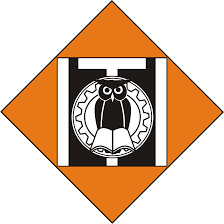
****

**INSTITUT TEKNOLOGI INDONESIA**

**IMPLEMENTASI SISTEM FACE RECOGNITION DALAM APLIKASI KEHADIRAN OTOMATIS DENGAN MENGGUNAKAN ALGORITMA LBPH (*LOCAL BINARY PATTERN HISTOGRAM*)**

**TUGAS AKHIR**

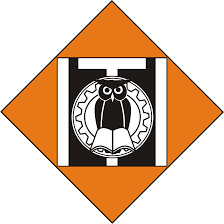
**MUHAMMAD NAUFAL MAHENDRA**

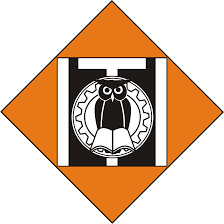
**1151700054**

**TEKNIK INFORMATIKA**

**TANGERANG SELATAN**

**AGUSTUS 2022**

****

****

**INSTITUT TEKNOLOGI INDONESIA**

**IMPLEMENTASI SISTEM FACE RECOGNITION DALAM APLIKASI KEHADIRAN OTOMATIS DENGAN MENGGUNAKAN ALGORITMA LBPH (*LOCAL BINARY PATTERN HISTOGRAM*)**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana**

**MUHAMMAD NAUFAL MAHENDRA**

**1151700054**

**TEKNIK INFORMATIKA**

**TANGERANG SELATAN**

**AGUSTUS 2022**

**HALAMAN PERNYATAAN ORISINIL**

Tugas Akhir ini adalah hasil karya saya sendiri,

dan semua sumber baik yang dikutip maupun dirujuk

telah saya nyatakan dengan benar.

|  |  |
| --- | --- |
| **Nama** | **: MUHAMMAD NAUFAL MAHENDRA** |
| **NPM** | **: 1151700054** |
| **Tanda Tangan** | **:** |
| **Tanggal** | **:** |

**HALAMAN PENGESAHAN**

Tugas Akhir ini diajukan oleh :

Nama : Muhammad Naufal Mahendra

NPM : 1151700054

Program Studi : Teknik Informatika

Judul Skripsi : Implementasi Sistem Face Recognition Dalam Aplikasi Kehadiran Otomatis Dengan Menggunakan Algoritma LBPH (Local Binary Pattern Histogram)

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer Pada Program Studi Teknik Informatika Institut Teknologi Indonesia.

**DEWAN PENGUJI**

Pembimbing : Husni, M.Kom (…….. tanda tangan ……..)

Penguji 1 : (…….. tanda tangan ……..)

Penguji 2 : (…….. tanda tangan ……..)

Penguji 3 : (…….. tanda tangan ……..)

Ditetapkan di : Kampus Institut Teknologi Indonesia, Tangerang Selatan

Tanggal :

**KETUA PROGRAM STUDI TEKNIK INFORMATIKA**

Dra. Sulistyowati, M.Kom

**KATA PENGANTAR**

Segala puji syukur penulis panjatkan kepada Allah Tuhan Yang Maha Esa atas segala nikmat-Nya sehinggan penulis dapat menyelesaikan tugas akhir yang berjudul “ Rancang Bangun Infrastructure As A Service Azure Cloud Untuk *Learning Management System* Moodle. Penulisan tugas akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar “Sarjana Komputer” pada Program Studi Teknik Informatika Institut Teknologi Indonesia.

Penulis mengucapkan terimakasih yang amat besar kepada semua pihak yang telah memberi dukungan, semangat, bimbingan, wadah, dan saran dari masa perkuliahan sampai penyusunan tugas akhir ini, maka dari itu penulis mengucapakan terimakasih dan pengharagaan setinggi-tingginya kepada:

1. Kedua orang tua dan keluarga yang mendukung penuh dan memfasilitasi kebutuhan penulis.
2. Ibu. Dra Sulistyowati M.Kom. selaku Ketua Program Studi Informatika.
3. Bapak Husni, M.Kom, MSc, selaku Penasehat Akademik.
4. Bpk.Dino Hariatma P, M.Kom. selaku Pembimbing Merdeka Belajar dan Tugas Akhir yang telah bersedia membimbing dan memberikan arahan selama penulisan Tugas Akhir ini.
5. Bpk. M. Ramli, S.T. selaku Dosen pembimbing Kerja Praktek.
6. Microsoft Indonesia dan MariBelajar yang telah melaksanakan program Studi Independen Bersertifikat.
7. Bpk. Wahyu Suprayogi, S.Stat, M.Si. selaku pembimbing kami di Badan Pusat Statistik Kabupaten Kudus.
8. Teman-teman MSIB Cloud Fundamental terkhususnya kelas Cloud Fundamental-005 yang telah banyak memberi ilmu tentang Cloud Fundamental.
9. Bapak Ir. Insap Paulus Santosa, M.Sc, Ph.D., IPU dan Mas Muhammad Revin selaku mentor dan asisten mentor kelas CF 005 Cloud Fundamental Track.
10. Keluarga Besar Himpunan Mahasiswa Teknin Informatika Institut Teknologi Indonesia yang telah memfasilitasi penulis selama proses belajar.
11. Teman-teman seperjuangan Teknik Informatika yang telah membantu, menyemangati, memberi saran, dan masih banyak lagi manfaat yang diberikan kepada penulis.

Dengan keterbatasan pengetahuan yang dimiliki oleh penulis, maka apabila ada kesalahan dalam pembuatan Tugas Akhir ini, maka penulis memohon maaf sebesar-besarnya. Demi kesempurnaan dalam pembuatan Tugas Akhir maka penulis menerima saran dan masukan dari pembaca. Atas perhatiannya penulis haturkan terimakas.

Tangerang Selatan, Juni 2022

Muhammad Naufal Mahendra

**HALAMAN PERSETUJUAN PERSETUJUAN PUBLIKASI**

**TUGAS AKHIR/SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Institut Teknologi Indonesia, saya yang bertanda tangan dibawah ini:

Nama : Bayu Khrisna Adj

NPM : 1151700026

Program Studi : Teknik Informatika

Jenis karya : Tugas Akhir

Demi pengembangan ilmu pengetahuan menyetujui untuk memberikan kepada Institut Teknologi Indonesia Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul :

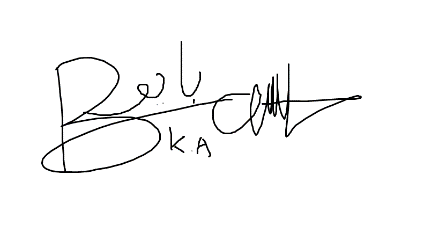
Rancang Bangun Infrastructure As A Service Azure Cloud Untuk *Learning Management System* Moodle

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Nonekslusif ini Institut Teknologi Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Tangerang Selatan

Pada Tanggal Februari 2022

Yang menyatakan,

Bayu Khrisna Adji

**ABSTRAK**

***ABSTRACK***

**DAFTAR ISI**

**DAFTAR GAMBAR**

**DAFTAR TABEL**



**INSTITUT TEKNOLOGI INDONESIA**

**IMPLEMENTASI SISTEM FACE RECOGNITION DALAM APLIKASI KEHADIRAN OTOMATIS DENGAN MENGGUNAKAN ALGORITMA LBPH (*LOCAL BINARY PATTERN HISTOGRAM*)**

**SKRIPSI**

**MUHAMMAD NAUFAL MAHENDRA**

**1151700054**

**TEKNIK INFORMATIKA**

**TANGERANG SELATAN**

**2022**

# BAB 1 PENDAHULUAN

## Latar Belakang

Pandemi Covid-19 yang saat ini sedang berlangsung telah menjadi perhatian utama masyarakat dunia dan membawa beragam implikasi, baik dalam bidang kesehatan, kebijakan publik, kesejahteraan, pendidikan, sosial, dan lainnya. Khususnya dalam bidang pendidikan membawa dampak terhadap proses pembelajaran. N*ovel Coronavirus* atau *Coronavirus Disease* (Covid-19) pertama kali muncul di daerah Wuhan pada negara Cina. Sejak meluasnya pandemi Covid-19 di dunia, termasuk telah melanda Negara Indonesia, maka sebagai usaha yang dilakukan untuk memutus rantai Covid-19 yaitu tidak boleh adanya kerumunan massa dan melakukan kontak fisik secara langsung baik dibidang apapun.

Dengan perkembangan zaman pada saat ini, pada ilmu Kecerdasan buatan (*Artificial Intelligenc*e) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik dari pada yang dilakukan manusia.

Salah satu contohnya yaitu sistem presensi. Yaitu merupakan tanda bukti kehadiran seseorang dalam menghadiri suatu pekerjaan atau aktivitas tertentu. Presensi dilakukan dengan cara menginput data seseorang dalam media tertentu sehinga dapat di dapatkan laporan presensi. Data yang diinput dalam hal ini bisa bermacam – macam bentuknya. Sebelum era digital datang, absen dilakukan dengan cara menuliskan nama pada selembar kertas atau buku tamu. Penulisan secara manual saat ini masih belum efektif mengingat keakuratan, konsistensi dan kejadian – kejadian yang tidak diinginkan seperti kertas hilang, basah, rusak masih belum maksimal. Demikian juga pada waktu yang digunakan belum maksimal lalu terdapat juga perusahaan yang masih melibatkan kontak fisik untuk melakukan absensi dengan menggunakan sistem *finger print.*

Oleh karena itu dibutuhkan suatu sistem untuk melakukan absensi dengan menggunakan pengenalan wajah seseorang. Sistem pengenalan wajah dapat dibangun dengan menggunakan berbagai macam metode dan algoritma. Salah satu nya dengan algoritma LBPH (*Local Binary Pattern Histogram*) yang digunakan dalam tugas akhir ini.

Algoritma LBPH adalah *Local Binary Pattern Histogram* merupakan salah satu jenis algoritma *machine learning*. LBPH merupakan hasil kombinasi dari dua metode yaitu metode *Local Binary Patterns* (LBP) yang dikombinasikan dengan *Histograms of Oriented Gradients* (HOG) . LBP merupakan operator tekstur yang mempunyai tugas memberi label pixel suatu gambar dengan cara membedakan lingkungan dari setiap *pixel* dan menganggap hasilnya sebagai angka biner sedangkan *Histograms of Oriented Gradient* diperoleh dengan mencari *histogram* dari citra hasil LBP sehingga metode LBPH merupakan metode yang digunakan untuk mendapatkan fitur yang ada pada wajah seseorang berdasarkan histogram yang diperoleh dari hasil LBP (Kosasih, R., & Daomara, C. 2021).

Maka dari itu dibutuhkan suatu sistem untuk melakukan presensi dengan menggunakan pengenalan wajah manusia. Dengan menggunakan salah satu metode algoritma LBPH (*Local Binary Pattern Histogram*).

## Perumusan Masalah

Berdasarkan latar belakang yang disebutkan diatas, masalah utama yang akan dibahas adalah sebagai berikut: Bagaimana cara melakukan implementasi algoritma LBPH (*Local Binary Pattern Histogram)* Untuk mendeteksi wajah manusia agar dapat melakukan pengenalan wajah secara *real time* dan melakukan presensi*.*

## Tujuan Penelitian

Tujuan dari tugas akhir ini adalah untuk dapat mendeteksi wajah manusia agar dapat melakukan pengenalan wajah secara *real time.* Dan juga diharapkan sistem ini dapat menghasilkan hasil yang lebih akurat, sehingga dapat membantu pengguna untuk melakukan presensi lebih cepat dan praktis.

## Batasan Masalah

Agar pengerjaan tugas akhir ini menjadi lebih terarah dan mendapatkan hasil yang lebih spesifik, maka sistem yang dirancang akan dibatasi dalam batasan masalah sebagai berikut:

* + 1. Dataset yang digunakan dalam membuat model *machine learning* ini adalah dataset yang telah diambil dari user yang telah didaftarkan.
    2. Sistem yang dirancang menggunakan bahasa python.
    3. Algoritma yang digunakan untuk dapat melakukan pengenalan wajah adalah *Local Binary Pattern Histogram* (LBPH)
    4. Bagian tubuh yang dapat dikenali hanya pada wajah yang sudah didaftarkan saja.
    5. Sistem tidak dapat mendeteksi suhu tubuh dan penggunaan masker.

## State of The Art

|  |  |
| --- | --- |
| **Judul Jurnal** | **Pembahasan** |
| **Implementasi *Face Recognition* Untuk Mengakses Ruangan**  **Peneliti**  Alwan Suryansah , Roni Habibi , Rolly Maulana Awangga , Rd. Nuraini Siti Fatonah.  **Lokasi**  Politeknik Pos Indonesia  **Tahun**  2020  **Nama Jurnal**  Jurnal Media Pendidikan Teknik Informatika dan Komputer. | **Hasil Penelitian :**  Hasil yang didapat pada jurnal ini adalah dapat mengimplementasikan algoritma LBPH (Local Binary Pattern Histogram) untuk pengenalan karakter wajah. memaksimalkan penggunaan komponen-komponen elektronik agar dapat digunakan sebagai alat yang dapat mengenal karakter wajah agar dapat mengakses ruangan.  **Alasan Menjadi Tinjauan Penelitian :**  Dengan pengimplementasian *Face Recognition* pada jurnal ini dapat dijadikan acuan dalam membangun sistem *Face Recognition* untuk melakukan presensi dalam mencari metode yang tepat untuk digunakan. |
| **Sistem Pengenalan Wajah dengan *Algoritma Haar Cascade* dan *Local Binary Pattern Histogram***  **Peneliti**  Sayeed Al-Aidid , Daniel S. Pamungkas  **Lokasi**  Politeknik Negeri Batam  **Tahun**  2018  **Nama Jurnal :**  Jurnal Rekayasa Elektrika | **Hasil Penelitian :**  Pada jurnal ini membahas tentang pembuatan Aplikasi pengenalan wajah dengan memanfaatkan 2 algoritma dan menggambungkan 2 metode yaitu *Haar Cascade* dengan algoritma *Local Binary Pattern Histogram* dapat digunakan sebagai pendeteksi dan pengenalan wajah manusia.  **Alasan Menjadi Tinjauan Penelitian :**  Dengan melihat penggunaan kedua algoritma tersebut yaitu algoritma *Haar Cascade* dengan algoritma *Local Binary Pattern Histogram* dapat melakukan pendeteksian wajah dan pengenalan wajah yang cukup baik dan mudah dimengerti , sehingga kedua algoritma tersebut dapat digunakan sebagai acuan untuk membuat sistem *Face Recognition* untuk melakukan presensi. |
| **Face Recognition Sebagai Sistem Pendataan dan Akses Masuk Perpustakaan Daerah**  **Peneliti**  Risyaf Fawwaz Pradipta , Denny Darlis , Syahban Rangkuti  **Lokasi**  Universitas Telkom  **Tahun**  2020  **Nama Jurnal**  Jurnal Teknologi Telekomunikasi | **Hasil Penelitian :**  Jurnal ini membahas tentang Sistem aplikasi *Face recognition* yang digunakan untuk pendataan dan akses masuk perpustakaan daerah dapat berjalan secara baik. Lalu memanfaatkan *face detection* dengan metode *Viola-Jones.* dan juga memanfaatkan salah satu *domain* kecerdasan buatan yaitu CV (*Computer Visual*).  **Alasan Menjadi Tinjauan Penelitian :**  Karena jurnal ini dapat memanfaatkan salah satu *domain* kecerdasan buatan yaitu CV (*Computer Visual*) dengan memanfaatkan *face detection* dengan menggunakan metode *Viola-Jones.* Jurnal ini dapat menjadi acuan untuk pembuatan Sistem *Face Recognition* untuk melakukan presensi. |
| **Pengenalan Wajah dengan Menggunakan Metode *Local Binary Patterns Histograms* (LBPH)**  **Peneliti**  Rifki Kosasih , Christian Daomara  **Lokasi**  Universitas Gunadarma, Depok, Indonesia  **Tahun**  2021  **Nama Jurnal**  Jurnal Media Informatika Budidarma | **Hasil Penelitian :**  Pada jurnal Ini membahas penelitian tentang absensi otomatis untuk membedakan wajah orang ke-1 dengan orang lainnya. Metode yang digunakan untuk melakukan pengenalan wajah secara otomatis. Pada penelitian ini digunakan metode *Local Binary Pattern*s Histograms (LBPH) yang merupakan kombinasi dari metode Local Binary Patterns (LBP).  **Alasan Menjadi Tinjauan Penelitian :**  Karena langkah-langkah yang digunakan pada metode *Local Binary Pattern Histogram* (LBPH) dijelaskan secara rinci dan mudah dimengerti sehingga dapat dijadikan acuan untuk membuat sistem *Face Recognition* untuk melakukan presensi. |
| **Implementasi Face Recognition pada Absensi Kehadiran Mahasiswa Menggunakan Metode *Haar Cascade***  ***Classifier***  **Peneliti**  Munawir, Liza Fitria, Muhammad Hermansyah  **Lokasi**  Universitas Samudra  **Tahun**  2020  **Nama Jurnal**  Jurnal Nasional Informatika dan Teknologi Jaringan | **Hasil Penelitian :**  Pada jurnal ini meneliti dan melakukan pengujian Sistem pengenalan dengan wajah dapat diterapkan pada absensi kehadiran mahasiswa. Dengan menggunakan metode *Haar Cascade Classifier.* Tetapi Sistem pengenalan wajah dengan banyak wajah (*multiple face recognition*) kurang cocok untuk diterapkan pada absensi kehadiran mahasiswa, karena terdapat banyak kesalahan dalam mengenali wajah sehingga proses absensi tidak sesuai dengan data yang sebenarnya. Dengan hasil pengujian satu wajah adalah 76% dan pengujian banyak wajah adalah 33.33%.  **Alasan Menjadi Tinjauan Penelitian :**  Dengan melihat penggunaan algoritma *Haar Cascade Classifier* yang dapat melakukan absensi kehadiran mahasiswa dengan cukup baik maka algoritma *Haar Cascade Classifier* dapat diimpelementasikan untuk pembuatan sistem Face Recognition untuk melakukan presensi. |

## Sistematika Penulisan

Sistematika penulisan disusun untuk memberikan gambaran secara umum mengenai permasalahan dan pemecahannya. Penyusunan ini diuraikan dalam beberapa pokok permasalahan yang terbagi dalam beberapa bab. Sistematika penulisan Tugas Akhir ini adalah sebagai berikut:

## Bab 1 PENDAHULUAN

Bab ini memuat pendahuluan penelitian yang terdiri dari latar belakang, rumusan masalah, tujuan dan manfaat penelitian, ruang lingkup penelitian, *state of the art*, serta sistematika penulisan penelitian.

## Bab 2 LANDASAN TEORI

Bab ini memuat landasan teori penelitian yang terdiri dari teori dasar mengenai sistem *Computer Vision* (CV) , *Artificial intellegence* (AI) , *Machine Learning,* Algoritma *Haar Cascade Classifier*, Algoritma *Local Binary Pattern Histogram (LBPH) , OpenCV.*

## Bab 3 ANALISIS

Bab ini berisi tentang analisis dan perancangan sebuah sistem yang dapat melakukan pengenalan wajah dan melakukan presensi. Dengan menggunakanalgoritma LBPH(*Local Binary Pattern Histogram*)*.*

## Bab 4 IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas mengenai implementasi dan analisa sistem yang dilakukan secara terstruktur menggunakan algoritma LBPH(*Local Binary Pattern Histogram*). Untuk hasil akhirnya akan berupa pengujian dan analisa terhadap pengenalan wajah.

## Bab 5 KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran dari pembahasan sistem Pengenalan Wajah yang telah dibangun dengan menggunakan algoritma LBPH(*Local Binary Pattern Histogram*)*.*

**LAMPIRAN**



**IMPLEMENTASI SISTEM FACE RECOGNITION DALAM APLIKASI KEHADIRAN OTOMATIS DENGAN MENGGUNAKAN ALGORITMA LBPH (*LOCAL BINARY PATTERN HISTOGRAM*)**

**SKRIPSI**

**MUHAMMAD NAUFAL MAHENDRA**

**1151700054**

**TEKNIK INFORMATIKA**

**TANGERANG SELATAN**

**2022**

# BAB II

# TINJAUAN PUSTAKA

## Face Recognition

*Face recognition* adalah teknik biometrik yang memungkinkan komputer atau mesin asli untuk mengenali wajah manusia. Teknologi pengenalan wajah dapat diterapkan dalam kehidupan sehari-hari untuk mempermudah aktivitas manusia. Dalam implementasi pengenalan wajah, banyak metode yang dapat digunakan, salah satunya adalah *machine learning,* yang mana hasil tangkapan dari kamera akan dilakukan pencocokan deengan menggunakan data yang telah dilatih sebelumnya menggunakan teknologi *machine learning*. Untuk dapat membaca dan memproses data inputan berupa citra, dapat menggunakan *library* OpenCV (Prapdipta, Darlis dan Rangkuti, 2020).

## Computer Vision

*Computer vision* merupakan cabang dari *artificial intellegence* (AI) atau kecerdasan buatan yang mempelajari ilmu mengenai bagaimana personal komputer bisa mengenali objek yang diamati. Implementasi dari teknologi AI sudah banyak sekali digunakan, baik pada teknologi *smartphone* juga pada dunia robotika. *computer vision* juga memungkinkan komputer bisa melihat objek atau benda yang terdapat pada lingkungan sekitar. Maka komputer bisa menganalisis benda atau gambar yang terdapat pada depannya sehingga informasi yang sudah terdekteksi bisa diterima & mampu membentuk perintah tertentu (Santoso, B. 2021).

## OpenCv

*Library* OpenCV (*Open Capture Vision*) merupakan suatu pemrograman aplikasi yang memungkinkan komputer untuk menampilkan objek seperti orang dan objek sehingga digunakan untuk pemrosesan citra atau yang disebut dengan *image processing* secara *real-time*. *Library* ini biasa ditujukan untuk *computer vision,* lalu komputer dapat mengambil keputusan, mengambil tindakan, dan mengenali objek yang objek sasarannya dalam penelitian ini adalah wajah. *Library* OpenCV saat ini bersifat open source atau gratis untuk digunakan oleh siapa saja dengan berbagai algoritma *computer vision* (Warnilah, A. I., Jaya-Mulyana, A., Siti-Nuraeni, F., & Aninditya-Widianto, T. 2022).

## Kecerdasan Buatan (Artificial Intelligence)

Kecerdasan buatan merupakan salah satu cabang ilmu komputer yang memiliki kapabilitas antara lain mesin pintar (*smart machine*) untuk menyelesaikan suatu persoalan yang rumit dengan cara yang lebih cepat dan tetap diarahkan oleh manusia. Hal itu dapat dilakukan oleh *artificial intelligence* (AI) dengan mencontoh karakteristik dan analogi proses berpikir dari kecerdasan manusia, dengan menerapkan algoritma (prosedur yang teratur) yang dapat dikenali oleh komputer. AI terkait erat dengan bidang ilmu lain, seperti statistika matematika, psikologi, pengamatan, biologi, filosofi, dan lainnya. (Pasaribu, M., & Widjaja, A. 2022). Kecerdasan diciptakan dan dimasukkan kedalam suatu mesin/komputer agar dapat melakukan pekerjaan seperti yang dapat dilakukan oleh manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer (*games*), logika fuzzy, jaringan saraf tiruan dan robotika. pengalaman dalam proses kerja teknologi *machine learning* ini biasanya diimplementasikan untuk menggunakan berbagai algoritma yang akan membuat prediksi dan klasifikasi data di masa depan. Hasil tersebut dicapai dalam dua fase, fase pelatihan dan fase uji coba (Roihan, Sunarya, Rafika, 2020).

## Machine Learning

*Machine Learning* merupakan bidang ilmu yang fokus kepada desain dan analisis algoritma sehingga dapat memungkinkan komputer untuk dapat belajar. Machine learning juga dapat diartikan sebuah komputer yang memilki kemampuan belajar tanpa diprogram secara eksplisit. Program tersebut memanfaatkan data untuk membangun sebuah model dan dapat mengambil keputusan berdasarkan model yang telah dibangun (Retnoningsih, E., & Pramudita, R. 2020). *Machine learning* sebagai salah satu bagian dari ilmu *soft computing* beririsan dengan berbagai ilmu lainya, seperti *mathematics* , *statistic*, *programming*, *data science*, *big data*, *data mining*, *database*, *information retrieval*, *computer vision*, *robotic*, *game programming*, *IoT* dan sebagainya. Salah satu ilmu yang populer di era *big data* pada saat ini, adalah *data* *science.* Secara umum *data* *science* adalah sebuah penggalian atau analisis data untuk menemukan data yang benar sehingga dapat menghasilkan suatu informasi atau pengetahuan yang akurat/tepat (Santoso, B., & Azis, A. I. 2020).

Agar dapat mencapai hasil yang diharapkan, dalam melakukan proses kerja *machine learning* dapat membantu dalam mengelola berbagai macam data, terutama data yang besar, kompleks dan memerlukan respon yang cepat. Berdasarkan input data pelatihan *machine learning* untuk menentukan jalannya proses kerja yang akan dilakukan. Dalam teknologi *machine learning*, dapat dibagi mejadi dua jenis yaitu:

* + 1. ***Supervised Learning***

*Supervised learning* merupakan pendekatan *machine learning* yang didasarkan pada penggunaan kumpulan data berlabel. Dataset ini memiliki “label”, yaitu satu kolom yang menjadi target keluaran model. Dalam Supervised learning, model dapat dilatih dalam dataset *training* dan dipantau untuk membuat klasifikasi atau prediksi sesuai dengan output berupa data berlabel yang ditentukan sebelumnya berdasarkan pola data *training* yang ada. *Supervised learning* dapat dibagi menjadi dua jenis yaitu:

* Klasifikasi

Klasifikasi adalah proses pembelajaran mesin yang menerapkan algoritme tertentu untuk secara akurat menetapkan data uji ke kategori tertentu. Sebagai contoh Supervised Learning, mesin menampilkan daftar gambar yang diberi label apel dan pisang. Kemudian menggunakan model hasil belajar dalam proses mengklasifikasikan gambar buah sebagai apel atau pisang.

* Regresi

Regresi adalah jenis lain dari metode *Supervised learning* yang menggunakan algoritma untuk memahami hubungan antara variabel dependen dan variabel independen memprediksi nilai numerik dari variabel dependen berdasarkan variabel independen. Contoh teknik ini adalah memprediksi harga tanah berdasarkan lokasi dan luas tanah (Hussein Saddam, 2022 ).

* + 1. ***Unsupervised Learning***

*Unsupervised Learning* adalah metode pembelajaran yang menggunakan algoritma pembelajaran mesin untuk menganalisis dan mengklasifikasikan kumpulan data yang tidak berlabel. Algoritma ini disebut tanpa pengawasan karena menemukan pola tersembunyi dalam data tanpa perlu campur tangan manusia. *Unsupervised Learning* digunakan untuk melakukan:

* *Clustering*

*Clustering* merupakan teknik pembelajaran mesin untuk mengelompokkan data yang tidak berlabel berdasarkan persamaan atau perbedaan. Contoh paling umum dari *machine learning* untuk analisis klaster adalah pengelompokan K-Means, yang banyak digunakan dalam proses analisis kelompok konsumen.

* Asosiasi

Asosiasi adalah jenis *Unsupervised Learning* yang menggunakan berbagai aturan untuk menemukan hubungan antara variabel dalam kumpulan data. Analisis asosiasi sering digunakan untuk membangun mesin rekomendasi. Contohnya kolom yang direkomendasikan untuk item berdasarkan pembelian atau pencarian sebelumnya.

* Dimensionality reduction

Dimensionality reduction adalah teknik untuk mengurangi jumlah variabel atau fitur dalam input atau data pelatihan. Proses ini dilakukan karena terdapat data dengan variabel yang sangat besar atau karena sering disebut dengan data berdimensi tinggi. Jenis data ini lebih sulit untuk dimodelkan (Hussein Saddam, 2022 ).

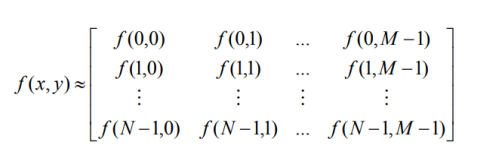
* 1. **Pengolahan Citra Digital**

Citra digital itu sendiri merupakan representasi yang digunakan untuk mengkorelasikan fungsi intensitas cahaya yang terdapat pada bidang 2D. Seringkali, gambar digital ini digunakan dalam bentuk foto atau video yang unik. Ada tiga jenis pengelompokan citra digital berdasarkan jenis warnanya: RGB, grayscale, dan binary, yang sebenarnya berbeda. Gambar RGB yang mendasari memiliki tiga saluran warna: merah, hijau, dan biru. Pada citra *grayscale*, warna yang digunakan adalah hitam dan putih, namun memiliki intensitas yang berbeda, seperti abu-abu. Untuk citra biner itu sendiri hanya menggunakan hitam dan putih, dan tidak ada perbedaan warna seperti yang dilakukan dengan skala abu-abu ( [Konsultan Data Penelitian & ArcGIS](https://patrastatistika.com/author/admin/), 2020).

Dari sini dapat disimpulkan bahwa citra digital merupakan citra 2D yang diubah menjadi citra dari proses analog 2D yang berkesinambungan dengan melewati proses *sampling*. Maka citra digital merupakan sebuah citra yang bisa diolah oleh komputer.

Ini terjadi karena data yang tersimpan di komputer hanyalah angka yang menunjukkan besarnya itensitas pada setiap piksel (Munantri et al, 2019).

Matriks dari citra digital dapat ditulis dalam bentuk matriks sebagai berikut ini :



**Gambar 2.1** Matriks Citra Digital f(x,y)

Keterangan :

M = Jumlah *pixel* baris (*row*)

N = Jumlah *pixel* kolom (*column*)

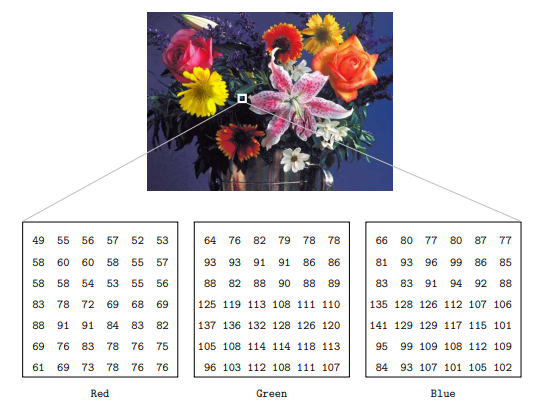
Nilai pada suatu irisan antara baris dan kolom ( pada posisi x,y) disebut juga dengan *picture elements, image elements, pels,* atau *pixel* (Hardiyanto, dan Sartika, 2018).

* + 1. **Jenis Citra Digital**

Secara umum, citra digital dibagi menjadi tiga jenis: citra berwarna, *Grayscale Image* dan citra biner.

1. Citra Berwarna

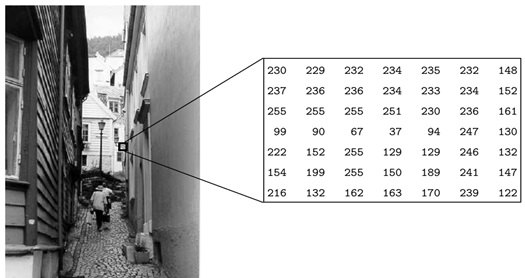
Setiap piksel dalam citra berwarna mewakili warna yang merupakan kombinasi dari tiga warna primer (RGB = *Red, Green, Blue*). Setiap warna dasar menggunakan 8 bit = 1 byte memori. Artinya, setiap warna memiliki gradien 255 warna. Setiap piksel memiliki 28 x 28 x 28 = 224 = 16 juta kombinasi warna. Untuk itu format ini disebut *true color* karena memiliki warna yang cukup untuk menutupi hampir semua warna yang ada di alam (Aqidah, 2019).



**Gambar 2.2**. RGB *Image*

1. *Grayscale Image*

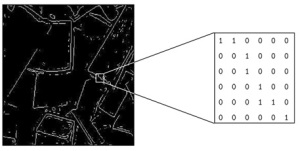
*Grayscale* adalah gambar yang hanya memiliki satu saluran warna. Akibatnya, nilai yang ditampilkan dalam jenis gambar ini disebut nilai intensitas, atau lebih umum skala abu-abu. Sebenarnya foto jenis ini sering dilihat namun istilah yang biasa digunakan adalah "hitam putih". Untuk citra hitam putih sebaiknya mengacu pada citra biner (Aqidah, 2019).



**Gambar 2.3** *Grayscale Image*

1. Binary Image

Citra biner (*binary image*) adalah citra yang hanya memiliki dua warna yaitu hitam dan putih, nilai *pixel* objek adalah 1 dan nilai piksel latar belakang adalah 0, Artinya latar belakangnya berwarna putih dan objeknya berwarna hitam. Dalam hal ini, nilai *pixel* hanya membutuhkan 1 bit (0 dan 1) atau 8 bit (0 dan 255) per *pixel*, yang sangat efisien dalam hal penyimpanan (Akbar, R. A., & Putra, R. E. 2019).



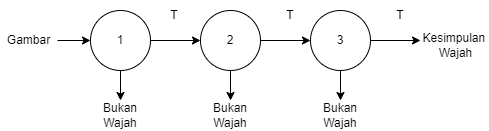
Gambar 2.10 *Binary Image*

* 1. **Haar Cascade Classifier**

Algoritma *Haar Cascade Classifier* merupakan salah satu *library* yang tersedia dalam OpenCV, dibangun dengan menggunakan bahasa C/C++ dengan *API* *(Application Programming Interface)* python yang dapat digunakan untuk mendeteksi wajah. Algoritma ini dapat dengan cepat mengenali objek, termasuk wajah manusia, secara *real time*. Algoritma klasifikasi *Haar Cascade* memiliki keunggulan lebih cepat karena hanya bergantung pada jumlah piksel kuadrat pada citra (Abidin, S, 2018).

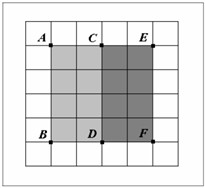
Untuk dapat memproses pendeteksian wajah digunakan algoritma *haar cascade classifier* digunakan untuk mengidentifikasi objek dalam *image digital*. Istilah *haar* berarti menunjukkan suatu fungsi matematika (*Haar Wavelet*) yang berbentuk kotak. Awalnya, pemrosesan gambar hanya terdiri dari pengecekan nilai RGB setiap piksel, tetapi metode ini terbukti tidak efektif. Kemudian Viola dan Jones mengembangkannya menjadi fitur *haar-like feature*. Fungsi *haar-like feature* dapat memproses gambar dalam kotak-kotak yang memiliki banyak piksel di dalam kotak tersebut*.* kemudian kotak tersebut diproses dan dapat menghasilkan nilai yang berbeda yang menunjukkan area gelap dan terang. Nilai-nilai ini digunakan sebagai dasar untuk pemrosesan gambar.

Nilai fungsi ini dihitung dengan mengurangkan nilai piksel pada daerah putih dari piksel di daerah hitam. Untuk dapat mempermudah proses penghitungan nilai fitur, algoritma *Haar* menggunakan media berupa *integral image*. *integral image* adalah citra yang nilai setiap pikselnya merupakan penjumlahan nilai piksel dari kiri atas ke kanan bawah. Misalnya, piksel (a, b) memiliki nilai kumulatif untuk semua piksel (x, y). Dimana x ≤ a dan y ≤ b. untuk dapat menggunakan metode *haar cascade* untuk memproses beberapa jenis gambar yang dapat diolah Salah satunya adalah *grayscale* (Al-Aidid, S., & Pamungkas, D. 2018).



**Gambar 2.4** Alur metode *haar cascade classifier*

Algoritma Haar menggunakan metode *statistical* dalam melakukan pendeteksian wajah. Metode ini menggunakan sample *haarlike fetures*. *Classifier* ini menggunakan gambar berukuran tetap (umumnya berukuran 24x24).



**Gambar 2.5** Fitur Persegi *Haar-like*

Nilai Fitur (ABFE) = Jumlah Nilai *Pixel*

(ABDC) – Jumlah Nilai *Pixel*

(CDFE)

* 1. **Local Binary Pattern Histogram (LBPH)**

*Local Binary Pattern* (LBP) adalah salah satu dari metode yang terkenal dalam mengenali sebuah objek. Sederhana tetapi sangat efisien yang dapat melabeli *pixel* berdasarkan suatu gambar dengan menggunakan ambang batas lingkungkan setiap *pixel* dan hasilnya sebagai bilangan biner. Dengan menggabungkan LBP dan *Histogram* bisa didapatkan deskripsi fitur yang bisa dipakai untuk merepsentasikan gambar wajah. Salah satu metode pendeteksian objek yang paling umum, Dalam hal ini, metode yang digunakan adalah membedakan objek dengan background. *Local Binary Pattern Histogram* (LBPH) merupakan algoritma kombinasi dari LBP dan *Histogram of Oriented Gradients* (HOG). Pengenalan wajah merupakan pengenalan wajah tingkat tinggi, dalam melakukan pengenalan wajah dapat menggunakan pencocokan dengan LBPH. Gambar wajah yang diambil secara *real time* oleh kamera dibandingkan dan dicocokkan dengan histogram yang diambil dari gambar wajah di database (Wibowo, A. W., Karima, A., Wiktasari, A. Y., & Fahriah, S., 2020).

Dikutip dari *website* medium sejarah & cara kerja *face recognition* pada *computer vision* adalah :

Dapat diketahui bahwa LBP memakai 4 paramater untuk perhitungannya. itu merupakan radius yang digunakan untuk mengatur jarak/radius seberapa besar lingkaran untuk merepresentasikan central *pixel-*nya, Kemudian *Neighbors* untuk seberapa banyak sample yang akan diambil untuk membuat lingkaran LBPnya, kemudian terdapat Grid X & Grid Y untuk membagi gambaran sebagai grid kotak-kotak kecil.

Langkah-langkah metode *Local Binary Pattern Histogram* :

* + - 1. Radius adalah jarak antar *threshold* dengan *neighbors* yang mengelilingi *threshold.*
      2. *Neighbor* adalah titik sampel yang mengelilingi *threshold* untuk membangun pola biner dan menghasilkan pola 8-bit.

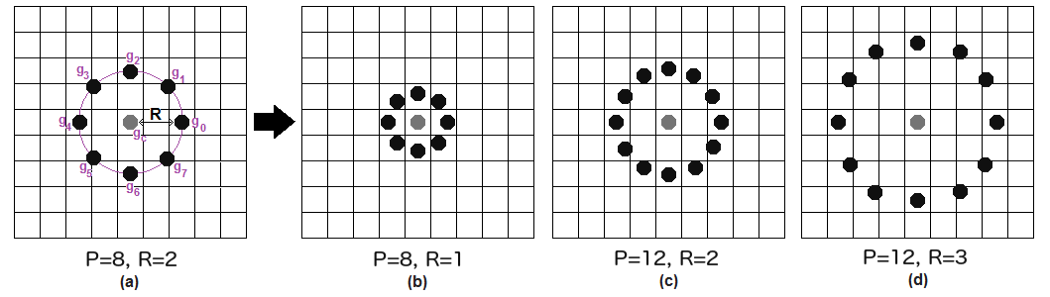


**Gambar 2.6** Konversi biner ke desimal.

Dari gambar 2.6, terdapat citra yang telah dikonversi dalam bentuk *grayscale*. Citra tersebut diambil sebagian pikselnya sebesar 3 x 3 piksel yang setiap pikselnya memiliki nilai masing-masing. Perbandingan yang dilakukan menggunakan rumus di bawah ini :

LBP P,R = s(gp - gc )2p , s(z) =

Operasi LBP dapat juga diexpand dengan menggunakan parameter yang telah diubah dari radius dan neighbors sehingga disebut sebagai *Circular* LBP. Karena *circular* maka terkadang akan terdapat beberapa titik yang berada pada antara *pixel,* maka dari itu bisa dipakai *bilinear interpolation* untuk mencari nilai tersebut. Berikut ini adalah contoh gambar yang ada dibawah ini :



**Gambar 2.7** *Bilinear interpolation*

* 1. **Tensorflow**

Tensorflow merupakan suatu *framework* yang sudah dikembangkan oleh Google Brain Team pada saat tahun 2015. Saat awal framework tensorflow ini dikembangkan digunakan untuk perhitungan numerik. Dengan semakin berkembang nya teknologi, dalam waktu ini framework tensorflow biasa dipakai untuk pengembangan aplikasi *Artificial Intelligence* (AI) oleh perusahaan-perusahaan besar. Aplikasi yang telah dikembangkan misalnya seperti pengklasifikasi gambar, penyematan kata, & pengembangan *chatbot* (Wiranda, Purba, Sukmawati, 2020).



**Gambar 2.8** Logo TensorFlow

*Framework* tensorflow kini telah menyediakan interface yang dapat digunakan untuk mengimplementasikan algoritma *machine learning* dan suatu aplikasi yang digunakan untuk menjalankan algoritma. Algoritma pemodelan yang didukung oleh *framework* ada begitu banyak tensorflow, contohnya adalah *Recurrent Neural Network* (RNN), *Local Binary Pattern Histogram* (LBPH), dan eksekusi paralel ( Wiranda , Purba, Sukmawati, 2020 ).

* 1. **Python**

Python adalah bahasa pemrograman yang dipakai buat menuntaskan problem numerik & analisis data. Dengan *library* yang majemuk Python sangat terkenal dikalangan programmer lantaran Python bisa dipakai menciptakan visualisasi data menggunakan aneka macam tampilan yang bagus. Lalu Python merupakan salah satu bahasa pemrograman terpopuler di dunia termasuk pada Indonesia. Selain itu, banyak pada bidang akademis telah menggunakan Python untuk menyelesaikan penelitian dalam ilmu komputer sains, robotika, *data science*, ekonomi, luar angkasa, dan banyak berbagai macam ilmu lainnya. Python diinstal secara *default* pada beberapa sistem operasi berbasis Linux seperti Ubuntu, Linux Mint, dan Fedora. Untuk sistem operasi lain, penginstal untuk sistem operasi ini disediakan.

Lalu Python juga mempunyai *package* yang populer bernama PIP. Anda dapat menggunakan PIP untuk mulai menginstal atau menghapus *library* Python yang digunakan atau tidak lagi digunakan. Ada banyak *website* yang menggunakan Python di Indonesia. *Website* yang menggunakan Python antara lain KelasKita, CodeSaya, dan Kargo.Co.Id. Berikut adalah gambar 2.9 adalah logo untuk bahasa pemrograman Python



**Gambar 2.9** Logo Python

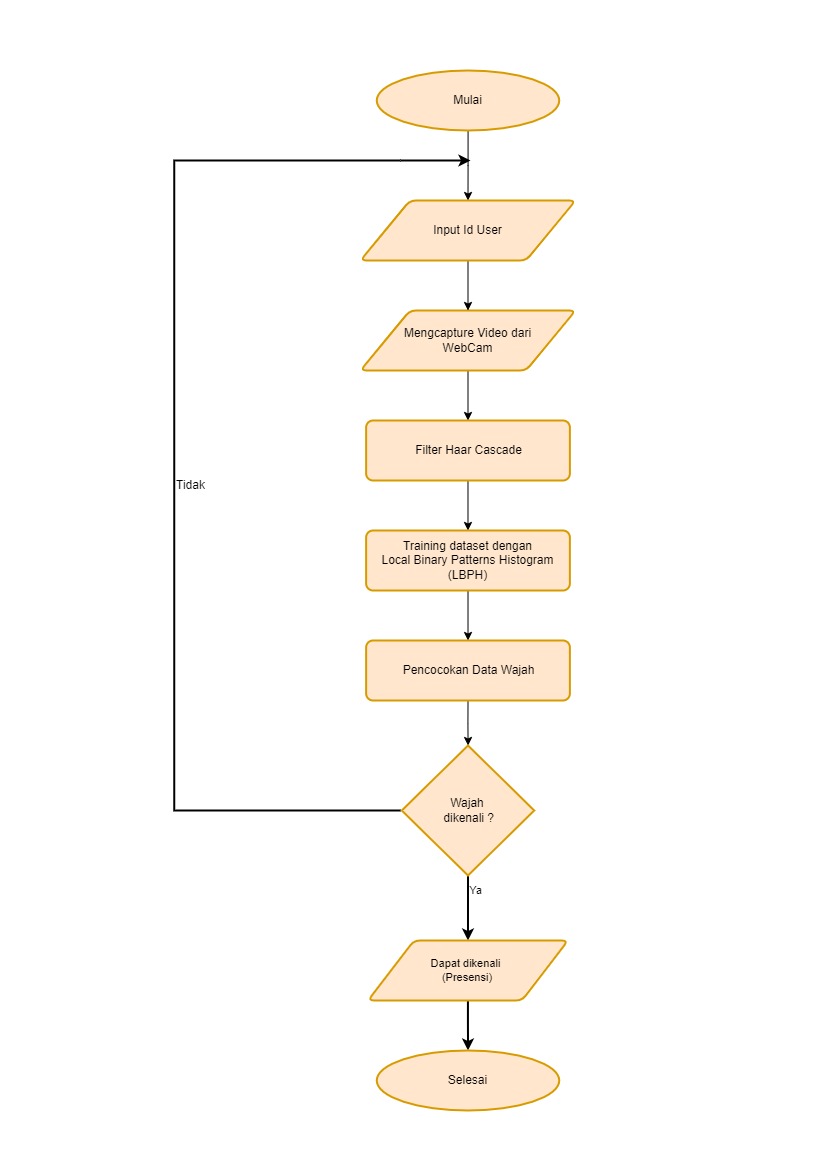
# BAB 3

# ANALISA DAN PERANCANGAN

Pada Bagian ini Bab ini akan membahas tentang analisa dan perancangan dalam sistem pengenalan wajah untuk melakukan kehadiran otomatis dalam aplikasi “*Smart Attendance Recognition*”. dalam Bab ini terdapat beberapa tahapan yang akan dianalisa, lalu tahap-tahap selanjutnya adalah sebagai berikut. Tahapan yang pertama adalah analisa data yang akan digunakan. Lalu tahapan kedua adalah analisa proses dalam mengolah citra yang akan digunakan. Selanjutnya yang terakhir adalah tahapan ketiga adalah analisa implementasi metode *Haar Cascade Classifier* dan dan *Local Binary Pattern Histogram* (LBPH)dalam melakukan pengenalan wajah untuk kehadiran secara *real time*.

## Gambaran Umum Alur Kerja Sistem

Metode yang akan digunakan dalam melakukan pengenalan wajah untuk kehadiran otomatis secara real timeadalah dengan menggunakan metode *Haar Cascade Classifier* dan *Local Binary Pattern Histogram* (LBPH)dengan melewati bebeberapa tahapan. Tahapan-tahapan tersebut akan dimulai dengan membuat data citra. Tahapan selanjutnya adalah tahapan *preprocessing* yang terdiri dari Proses *convert to grayscale* dan *resize,* proses ini akan mengubah citra agar proses *training* dapat lebih mudah dilakukan. Setelah tahap *preprocessing* sudah dilakukan, maka akan masuk ke dalam tahapan melakukan pengenalan wajah untuk kehadiran dengan mengimplementasi Algoritma *Haar Cascade Classifier* dan *Local Binary Pattern Histogram* (LBPH) . Setelah tahapan-tahapan tersebut dilakukan maka akan mendapatkan hasil pengenalan wajah user yang sudah terdaftar sebelumnya di *dataset*. Gambaran umum alur kerja dari sistem dapat dilihat melalui *flowchart* pada Gambar 3.1. berikut.



**Gambar 3.1 *Flowchart* alur kerja sistem**

Berikut uraian setiap proses yang terdapat pada *flowcharat,* yaitu :

1. Input Id User

Merupakan sebuah *input* atau masukan pada sistem yang dilakukan secara *real time.* Untuk melakukan pengimputan ID yang diperlukan untuk inisialisasi setiap wajah yang akan direkam.

1. Mengcapture Video dari *Webcam*

Merupakan sebuah *input* atau masukan pada sistem yang dilakukan secara *real time.* Untuk melakukan pendaftaran wajah diperlukan kamera atau *webcam* yang digunakan untuk merekam wajah.

1. Data Preproses dengan *Haar cascade*

Langkah ini dilakukan untuk melakukan *transformasi* pada gambar yang telah berhasil direkam melalui kamera atau *webcam* menjadi bentuk *grayscale* dan sudah dilakukan *resize* sesuai dengan wajah yang terdeteksi lalu akan ditunjukkan berupa garis bujur sangkar ROI (*Region of Interest*) pada wajah yang berhasil dideteksi.

1. Training Data dengan *Local Binary Pattern Histogram* (LBPH)

Langkah dalam mengenali wajah manusia adalah dengan mendeteksi wajah terlebih dahulu. Pada training wajah akan dilakukan proses pencocokan berdasarkan data yang sudah dilakukan *transformasi* sebelumnya dengan *haar cascade.*

1. Pencocokan Data Wajah

Pencocokan wajah dilakukan untuk mengenali wajah manusia berdasarkan dataset wajah yang sudah ada. Wajah akan dicocokan menggunakan metode *Haar Cascade Classifier.* Pada tahap ini terdapat perhitungan-perhitungan untuk pengenalan wajah.

1. Wajah dikenali

Wajah dikenali adalah sebuah kondisi apakah wajah manusia dikenali atau tidak. Berdasarkan *dataset* wajah yang sudah ada. Jika wajah dapat dikenali maka akan lanjut ke proses berikutnya , jika wajah tidak dikenali maka akan kembali ke proses sebelumnya yaitu Input Id *User*.

1. Dapat dikenali (Presensi)

Setelah melakukan pengujian dengan data wajah yang ada dalam *dataset* dan berhasil , sistem akan memberikan informasi nama wajah yang dikenali secara dan juga secara otomatis akan masuk kedalam file .csv yang berisi informasi berupa nama dan tanggal waktu melakukan presensi.

## Data yang digunakan

Data yang akan digunakan dalam penelitian ini terdiri dari sampel gambar yang diambil dari hasil *capture* sebuah kemera *webcam* sebanyak 100 *capture* gambar per-wajah yang didaftarkan, dengan beberapa batasan parameter yaitu : variasi posisi citra wajah dan jarak wajah terhadap kamera *webcam*.

Untuk variasi posisi wajah dilakukan beberapa posisi sebagai berikut:

* Menghadap tegak lurus ke depan
* Rotasi 10° ke kanan
* Rotasi 10° ke kiri
* Rotasi 10° ke ke atas
* Rotasi 10° ke bawah
* Mengangkat dagu 10° atas

Wajah yang di *capture webcam* tidak terhalangi sebagaian oleh objek lain, tidak banyak terpotong dan tidak bergerak. Untuk aspek jarak wajah terhadap kamera webcam akan dicari jarak ideal yaitu 40 cm sampai dengan 100 cm.

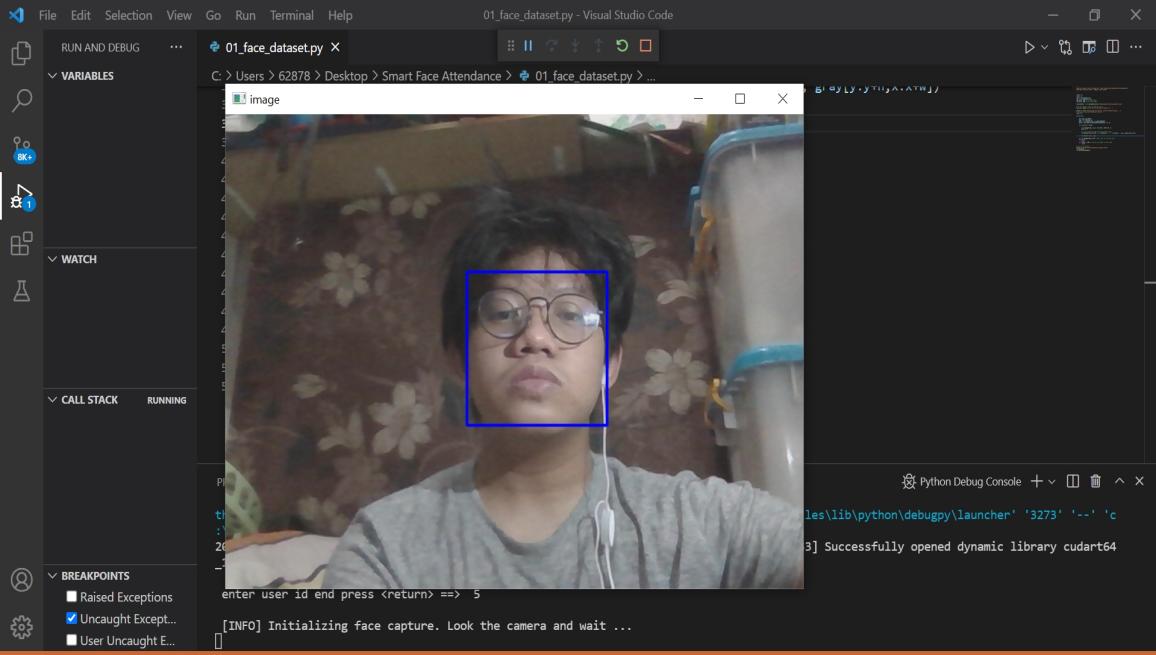
* 1. **Pembuatan *Dataset***

Sebelum sistem dapat mengenali wajah, wajah yang akan dikenali terlebih dahulu disimpan pada database agar sistem dapat mengetahui nilai *LBPH (Local Binary Patern Histogram)* atau nilai *histogram* dari sebuah *image*. Untuk dapat mengenali wajah dengan baik sistem memerlukan 100 input gambar dari kamera. Dari setiap subjek diambil datanya dengan berbagai pose dan sudut sebanyak 100 kali. Data tersebut akan disimpan sebagai *dataset* gambar. Pembuatan *dataset* diperlukan untuk mendapatkan data wajah yang akan dikenali. Sampel-sampel tersebut kemudian dimasukkan dalam sebuah folder dengan tiap orangnya memiliki nomor unik tersendiri. Nomor tersebut digunakan untuk mengidentifikasi sampel wajah dari orang tersebut. Sampel gambar kemudian diubah menjadi gambar yang bersifat *grayscale* sebelum disimpan pada *dataset* .Setelah itu, dilakukan proses training. Pada proses ini gambar pada *dataset* akan diekstraksi nilai histogramnya. Nilai tersebut disimpan dalam bentuk data array dan disimpan beserta nomor identitas masing-masing subjek.

* 1. ***Preprocessing***

Dalam proses preprocessing ini akan dilakukan tahap pengolahan citra agar data yang akan diolah mempunyai citra yang lebih baik untuk diproses ke tahapan selanjutnya. Pada tahap proses *preprocessing* ini terdiri dari 2 (dua) proses, yaitu proses *capture* gambar wajah *user* dan dikonversi dari citra RGB menjadi citra *grayscale* lalu dilakukan resize.

Hal ini berguna untuk membuat citra dapat diolah dengan menggunakan metode *Haar Cascade*, yang dimana metode ini akan digunakan untuk mendeteksi wajah. Setelah dapat terdeteksi wajah maka sistem akan menandai wajah dengan garis bujur sangkar ROI (*Region of Interest*) berwarna biru dan objek yang digunakan adalah muka. Hal ini dilakukan untuk membedakan dengan objek bukan wajah. Contoh pada tahapan ini akan digambarkan dalam Gambar berikut 3.2



**Gambar 3.2 Pengambilan gambar**

Lalu akan dijelaskan untuk tahapan-tahapan melakukan *preprocessing* seperti contoh pada gambar 3.3agar dataset yang akan digunakan dapat lebih mudah dan lebih cepat diproses sebagai berikut.

* + 1. ***Face Detection***

Perancangan face detection atau biasa disebut dengan pendeteksian wajah pada tugas akhir ini menggunakan metode *Haar Cascade Classifier.* Metode *Haar Cascade Classifier* digunakan karena metode ini sangat ringan dan memiliki kecepatan pendeteksian yang cepat. Penulis menggunakan *Library* yang disediakan oleh Open CV yakni *haarcascadefrontal\_face\_default.xml.* Library tersebut sudah dapat semua proses pendeteksian wajah dan berikut ini merupakan urutan cara kerja *library* Open CV haarcascadefrontal\_face\_default.xml yaitu.

Langkah pertama citra harus dikonversikan dari RGB menjadi *Grayscale.* Secara teori cara mengkonversikannya adalah dengan cara menggunakan rumus sebagai berikut.

*Grayscale =* 0.2989 R + 0.5870 G + 0.1140 B **(3.1)**

Pada persamaan 3.5 dapat melihat bahwa setiap nilai R nilai G dan nilai B pada setiap piksel akan dikalikan dengan nilai yang sesuai rumus agar sebuah citra dapat dikonversi menjadi *grayscale*. Maka contoh terdapat pada , gambar 3.3 merupakan citra RGB dengan dimensi 200 x 200.



**Gambar 3.1 Citra RGB**

Kemudian penulis akan menggunakan semua piksel secara acak untuk mengetahui nilai RGB dari salah satu piksel. Lalu pada piksel *x* = 100 dan *y* = 100 dapat diketahui RGB = 57 34 16. Dari data tersebut maka dapat dihitung *grayscale* dengan cara sebagai berikut ini :

Diketahui = *Red* (merah) = 57

*Green*(hijau) = 34

*Blue*(biru) = 16

Grayscale *=* 0.2989 R + 0.5870 G + 0.1140 B

*Grayscale* = *(* 0.2989 x 57) +( 0.5870 x 34) + (0.1140 x 16)

*Grayscale* =17.0373 + 19.968 + 1.824

Grayscale = 38.8193

Citra *grayscale* merupakan citra dengan ukuran 8 bit. Sehingga didapatkan

(28 – 1) yang mengubah warna mulai dari 0 hingga 255 , yaitu dimana 0 berarti hitam dan 255 berarti putih. Warna abu memiliki *range* dari 1 hingga 254 dimulai dari abu paling gelap sehingga abu terang dan akhirnya mendekati putih. Maka hasil *grayscale* = 38.8193 merupakan warna abu-abu yang hampir mendekati hitam. Lalu menggunakan rumus tersebut di semua poksel dan hasil dari konversi citra RGB pada gambar 3.1 akan menjadi seperti pada gambar 3.2 sebagai berikut :



**Gambar 3.2 Citra *grayscale***

Dari gambar 3.3 dapat diketahui bahwa setelah melakukan konversi menjadi citra *grayscale* sebagian besar daerah pada citra memiliki nilai yang akan lebih hitam. Proses selanjutnya adalah melakukan *Haar feature* yaitu adalah metode yang membutuhkan training terlebih dahulu untuk mendapatkan suatu keputusan apakah di *frame* tersebut dapat mendeteksi objek atau tidak. Selisih dari nilai fitur yang akan diimplementasikan akan dijadikan *threshold* klasifikasi terdeteksi objek atau tidak. Lalu dalam tugas akhir ini objek yang dideteksi adalah wajah. Berikut ini adalah contoh gambar 3.4 yang akan diletakan fitur *haar* dalam sebuah citra.



**Gambar 3.3 *Haar Feature***

Selanjutnya pada gambar 3.3 penulis meletakan sebuah feature haar pada gambar hasil konversi *grayscale. Feature* yang digunakan berupa *line feature* yang merupakan salah satu dari banyak *haar feature.* Untuk menentukan nilai feature tersebut dapat digunakan persamaan yaitu sebagai berikut ini :

*NILAI FITUR = |*(total piksel hitam) – ( total piksel putih)| **(3.2)**

* 1. ***Training Dataset***

Proses selanjutnya adalah mencari pola ciri wajah dari masing-masing individu untuk dapat membedakan antara individu satu dengan yang lainnya. Dibutuhkan operator untuk klasifikasi tekstur dan pada tugas akhir ini penulis menggunakan *Local Binary Patern*. Cara kerjanya adalah dengan mengkonversi seluruh citra *grayscale* yang telah dijadikan *dataset* sebelumnya menjadi *Local Binary Patern* Image dan dijadikan sebuah Histogram. Berikut ini adalah langkah-langkah membuat *Local Binary Patern*.

1. Langkah pertama menyiapkan citra grayscale yang sudah dijadikan *dataset* sebelumnya lalu akan diubah menjadi *local binary pattern image.* Maka Gambar 3.4 merupakan citra *grayscale*

**

**Gambar 3.4 Citra *grayscale***

1. Langkah kedua menggunakan rumus LBP 3 x 3 yang dihitung mulai dari piksel pojok kiri atas. Rumus yang akan digunakan diletakan pada persamaan 3.3

*LBPP.R =* (ɡ*p -* ɡ*c*)2p *S* = (*x*) = **(3.3)**

1. Setelah dilakukan zoom hingga terlihat jelas tiap piksel ambil matrik 3 x 3 pada piksel, maka gambar akan menjadi seperti pada gambar **3.5**



1. Lalu menghitung nilai matrik 3 x 3 yang telah ditandai tersebut dengan menggunakan rumus LBP. Nilai matriknya adalah sebagai berikut:

254 254 255

239 255 254

245 255 249

Titik pusatnya ialah 255 kemudian bandingkan dengan matrik disekitarnya dengan aturan jika pusat nilai piksel sekitar maka beri nilai 0. Jika titik pusat < nilai piksel sekitar maka beri nilai 1. Sesudah dilakukan seperti itu kemudian matriknya akan menjadi seperti berikut ini :

0 0 1

0 Pusat 0

0 1 0

Untuk mengetahui nilai pusat , hasil biner tersebut kemudian diubah menjadi desimal dengan cara sebagai berikut ini :

07 06 15

00  Pusat 04

01 12 03

07 06 15 03 12 01 00

25 22

0+0+32+0+0+4+0+0 = 36

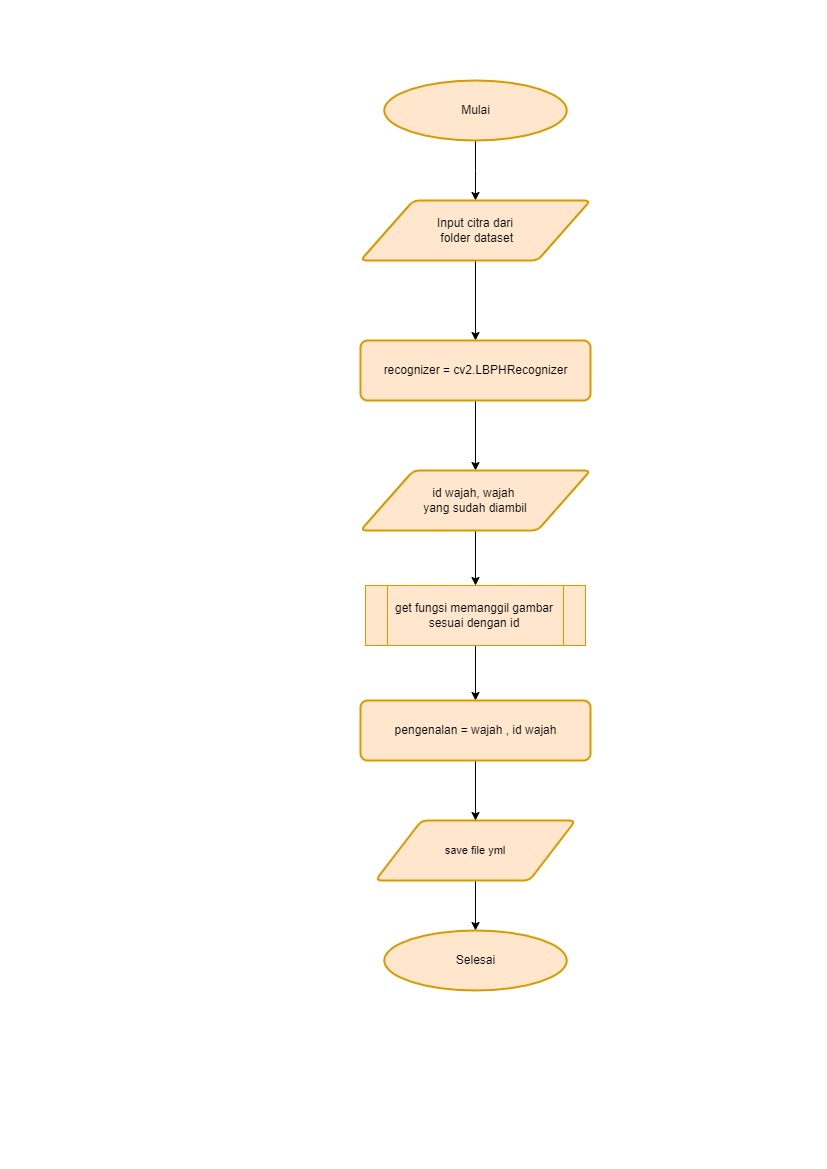
Dari hasil perhitungan tersebut diketahui nilai pusatnya adalah 36. Lalu akan diimplementasikan perhitungan pada semua bagian citra maka citra LBP akan menjadi seperti pada gambar 3.6



**Gambar 3.6 LBP *image***

Setelah menerapkan proses LBP maka histogram dari setiap gambar akan di ekstrak berdasarkan jumlah *grid* (X dan Y) yang dilewatkan parameter.setelah histogram dari setiap wilayah di ekstrak maka semua histogram yang ada akan digabungkan dan dibuatlah satu histogram baru yang ada digunakan untuk mempresentasikan gambar. Kemudian setiap hasil tersebut akan disimpan dalam sebuah file .yml.

Setelah melalui proses tersebut maka selanjutnya akan membuat algoritma yang dapat menyimpan hasil LBP dalam bentuk yml. Algoritma akan dirancang dalam bentuk *flowchart*. Untuk fungsi *Local Binary Pattern Histogram* menggunakan library buatan Open CV yaitu cv2.createRecognizer. pada library tersebut sudah mencakup keseluruhan cara kerja *Local Binary Pattern*. Maka pada gambar 3.7 akan dijelaskan *flowchart* yang akan dibuat.



**Gambar 3.7 Flowchart *training* data menggunakan LBPH**



**INSTITUT TEKNOLOGI INDONESIA**

**IMPLEMENTASI SISTEM FACE RECOGNITION DALAM APLIKASI KEHADIRAN OTOMATIS DENGAN MENGGUNAKAN ALGORITMA LBPH (*LOCAL BINARY PATTERN HISTOGRAM*)**

**SKRIPSI**

**MUHAMMAD NAUFAL MAHENDRA**

## 1151700054

**TEKNIK INFORMATIKA**

**TANGERANG SELATAN**

**2022**

**BAB 4**

**IMPLEMENTASI DAN PENGUJIAN**

Pada Bab ini akan menguji dan implementasi untuk menjelaskan mengenai perangkat yang akan digunakan dalam membangun sebuah model sistem pengenalan wajah untuk melakukan kehadiran otomatis dalam aplikasi “*Smart Attendance Recognition*”. dengan menggunakan metode *Haar Cascade Classifier* dan *Local Binary Patterns Histogram* (LBPH)*.* Perangkat keras *(hardware)* dan perangkat lunak *(software)* yang akan dijelaskan berikut ini.

1. **Perangkat Keras *(hardware)***

Pada kebutuhan perangkat keras (*hardware*) yang digunakan untuk membangun sebuah model sistem pengenalan wajah untuk dapat mengenali pengguna dan melakukan kehadiran*.* Akan ditunjukan pada Tabel 4.1 sebagai berikut.

**Tabel 4.1 Spesifikasi perangkat keras**

|  |  |  |
| --- | --- | --- |
| **Perangkat Keras** | **Spesifikasi Minimum** | **Spesifikasi yang Digunakan** |
| 1. Processor | Intel Celeron (1.4 GHz) -64bit | Intel® Core™ i7-8565U |
| 1. Graphic Card | Ethernet Card | NVIDIA® GeForce® MX150 |
| 1. Hardisk | 500GB | 512GB |
| 1. RAM | 2GB DDR3 | 8GB 2400MHz DDR4 Memory |
| 1. WebCam | 420p HD Camera | 720p HD Camera |

**4.1.2 Perangkat Lunak *(software)***

Perangkat lunak *(software)* yang digunakan pada masing-masing perangkat keras *(hardware)* yang sebelumnya telah dijelaskan, dalam membangun sebuah model kesesuaian metode *face recognition* untuk dapat mengidentifikasi seseorang.

Akan ditunjukan pada Tabel 4.2 sebagai berikut.

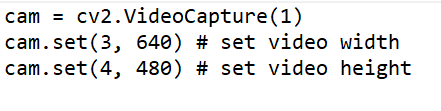
**Tabel 4.2 Spesifikasi perangkat lunak**

|  |  |
| --- | --- |
| **Perangkat Lunak** | **Spesifikasi** |
| 1. Operating System | Windows 10 64 bit Home |
| 1. Microsoft Visual Code | 1.63 Version |
| 1. Python | 3.9 Version |
| 1. Google Chrome | Version 98.0.4758.82 (Official Build) (64-bit) |

* 1. **Pengambilan *Dataset***

*Dataset* yang digunakan dalam implementasi ini berupa *dataset* yang dibuat sendiri dengan mengambil gambar wajah yang ingin diidentifikasi melalui kamera secara *real time*. Setelah itu citra tersebut akan diolah menggunakan metode pendeteksian wajah berupa *Haar Cascade*. Wajah yang terdeteksi ditunjukkan berupa garis bujur sangkar ROI (*Region of Interest*) yang digunakan adalah objek (muka)

Untuk dapat melakukan identifikasi sebuah objek (muka) , user harus melakukan pengambilan gambar terlebih dahulu secara *real time* sebanyak 100 gambar untuk dijadikan dataset. Maka implementasi *source code* untuk melakukan *open* kamera untuk pengambilan gambar secara *real time*. Akan ditampilkan pada *source code* Gambar 4.1 sebagai berikut.



**Gambar 4.1 *Source code* untuk menampilkan kamera *webcam***

Kemudian, python akan membaca video real time yang ada pada laptop. Setelah itu memberi program pada python agar dapat mendeteksi mana yang merupakan wajah dan mana yang bukan merupakan wajah dengan perintah *source code* pada gambar **4.2** sebagai berikut:



**Gambar 4.2 Source code mendeteksi wajah**

Kemudian menggambar ROI (*Region Of Interest*) dalam bentuk kotak berwarna biru sesuai dengan koordinat dan dimensinya yaitu (nilai x,y,w,h). maka contoh *source code* pada gambar 4.3 sebagai berikut :

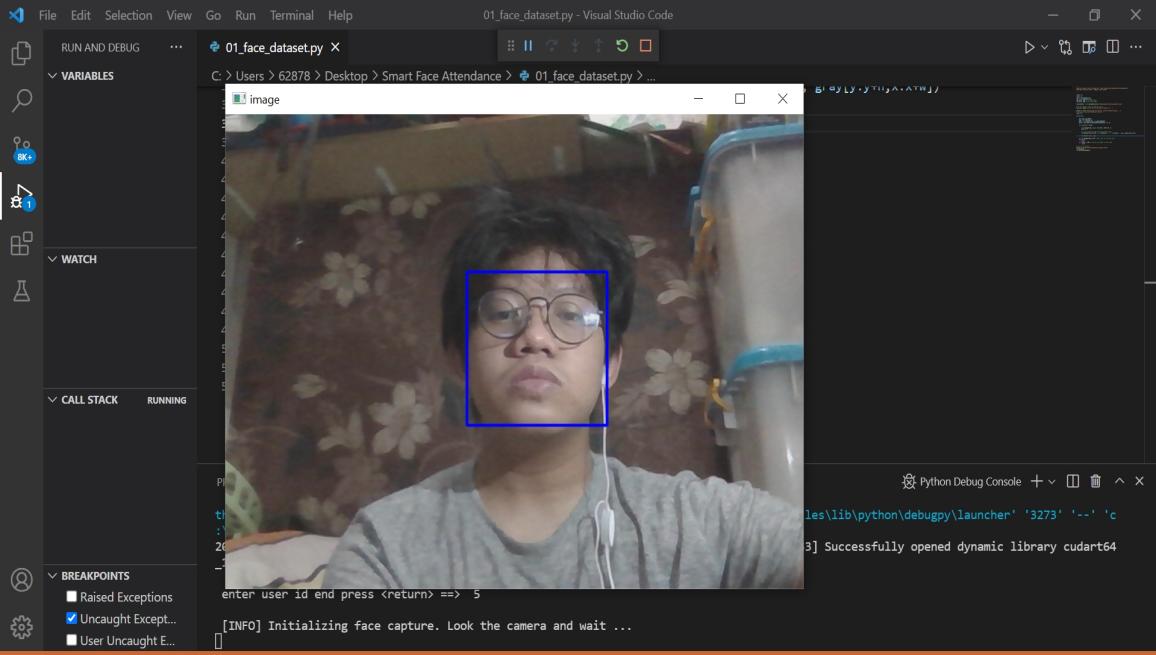


**Gambar 4.3 Source code menggambarkan ROI (*Region Of Interest*)**

Kemudian python akan meng-*capture* wajah *user* yang akan melakukan identifikasi wajah akan diminta untuk memasukan id untuk dapat menyimpan inisial dari nama *user* yang akan disimpan kedalam dataset, setiap wajah harus ditandai dengan no Id. Maka contoh *source code* pada gambar 4.4 sebagai berikut:

**Gambar 4.4 Source code untuk memasukan Id *User***

Lalu sistem akan mendeteksi wajah. Jika wajah terdeteksi maka akan ditunjukkan muncul tanda berupa garis bujur sangkar ROI (*Region of Interest*) pada wajah. Maka akan muncul sebuah tampilan video untuk meng-*capture* wajah user sampai 100 gambar. Akan ditampilkan pada gambar 4.4 sebagai berikut.



**Gambar 4.4 Pengambilangambar *user***

Data gambar yang telah melakukan identifikasi wajah akan memproses dan hasil dari pengolahan yang sudah dilakukan tersimpan pada folder “dataset” yang berada dalam *directory* yang sama dengan *source code*. Lalu data akan mentransformasi berbentuk *grayscale ,* oleh karena itu masing-masing data gambar yang ditangkap akan dikonversi menjadi *grayscale* akan ditampilkan source code pada gambar 4.3 dengan menggunakan fungsi sebagai berikut:



**Gambar 4.3 Fungsi mengubah citra RGB menjadi *grayscale***

* 1. **Pengolahan *Dataset***

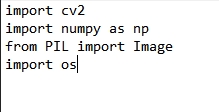
Proses pertama yang dilakukan oleh metode *Haar Cascade Classifier* untuk mendeteksi adanya fitur wajah pada sebuah gambar adalah dengan merubah gambartersebutmenjadi citra *grayscale* dan data yang akan diolah akan dilakukan proses *resize* (mengubah ukuran gambar) diubah menjadi dimensi yang berbeda-beda sesuai dengan rasio wajah yang tertangkap kamera. Proses ini dilakukan untuk dapat memudahkan proses pembelajaran sistem (*training*). Lalu hasil dari pengolahan yang sudah dilakukan tersimpan pada folder “dataset” yang berada dalam *directory* yang sama dengan *source code*. Maka hasil dari pengambilan gambar yang telah ditransformasi menjadi citra *grayscale* akan ditampilkan pada Tabel 4.3 sebagai berikut.

|  |  |  |  |
| --- | --- | --- | --- |
| Sampel 1 | | Sampel 2 | |
| C:\Users\62878\Desktop\Smart Face Attendance\dataset\User.1.27.jpg | C:\Users\62878\Desktop\Smart Face Attendance\dataset\User.1.8.jpg | User.2.5.jpg | User.2.57 |
| Sampel 3 | | Sampel 4 | |
| User.4.28.jpg | User.4.79.jpg | User.3.61.jpg | User.3.58.jpg |
| Sampel 5 | | Sampel 6 | |
| C:\Users\62878\AppData\Local\Microsoft\Windows\INetCache\Content.Word\User.5.83.jpg | C:\Users\62878\AppData\Local\Microsoft\Windows\INetCache\Content.Word\User.5.58.jpg | C:\Users\62878\AppData\Local\Microsoft\Windows\INetCache\Content.Word\User.6.77.jpg | C:\Users\62878\AppData\Local\Microsoft\Windows\INetCache\Content.Word\User.6.1.jpg |

**Tabel 4.3 Hasil pengolahan *dataset* yang sudah dikonversi menjadi *grayscale***

* 1. **Proses Pembelajaran Sistem (*Training*) **

Sebelum melakukan proses *training,* terdapat modul dari numpy , pillow yang akan digunakan. Modul-modul tersebut akan diimport terlebih dahulu seperti yang ditampilkan pada Gambar 4.4 berikut.



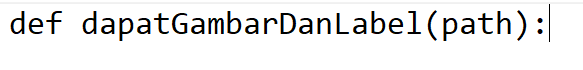
**Gambar 4.4 *Importing package* dan *modul training***

Pada proses *training* atau pembelajaran sistem ini akan menggunakan algoritma *Local Binary Patterns Histogram* (LBPH). Algoritma *Local Binary Pattern Histogram* (LBPH) akan dapat mengenali gambar dalam *folder dataset* secara terus-menerus untuk melatih sistem sehingga dapat mengenali gambar wajah tersebut dengan tepat. Maka ditampilkan source code pada Gambar 4.5 sebagai berikut:



**Gambar 4.5 Metode *Local Binary Pattern Histogram***

Kemudian membuat suatu fungsi perulangan untuk melatih sistem dalam mengenali gambar secara benar. Berikut adalah perintah *source code* pada gambar 4.6



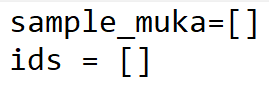
**Gambar 4.6 *Source code* perulangan untuk melatih**

Lalu dibutuhkan perintah untuk mengambil data dari *folder dataset* lalu akan melakukan *training* untuk keseluruhan dan menelusuri gambar yang ada pada *folder* *dataset.* Yang mana perintah source code pada gambar 4.7 berikut :



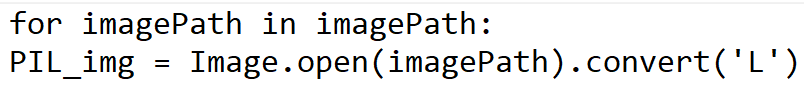
**Gambar 4.7 *Source code* untuk melakukan *training* pada seluruh *folder dataset***

Selanjutnya akan dibuat definisi untuk gambar wajah dan labelnya dengan perintah source code pada gambar 4.8 sebagai berikut :



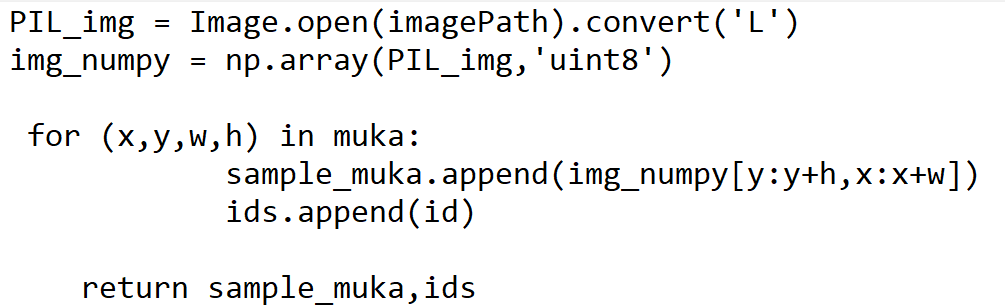
**Gambar 4.8 *Source code* untuk membuat definisi wajah dan label**

Lalu untuk mempelajari setiap gambar wajah, dapat menggunakan perulangan pada gambar 4.9 perintah *source code* berikut ini , yang mana perintah itu digunakan untuk mengkonversi gambar menjadi *grayscale* terlebih dahulu lalu dikonversikan lagi menjadi bentuk *array*.

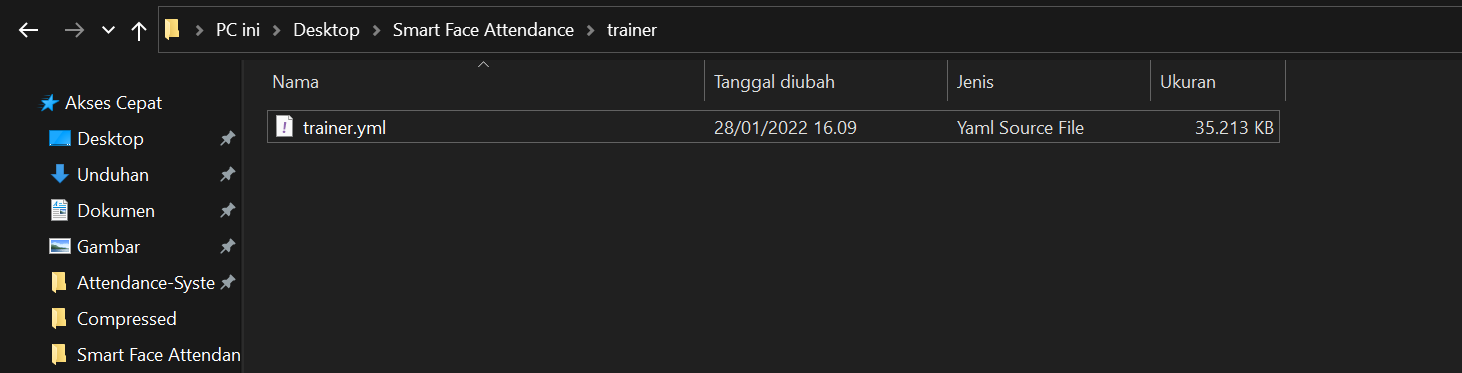


**Gambar 4.9 *Source code* untuk mengkonversi menjadi grayscale dan array**

Kemudian dibutuhkan juga modul untuk mengelola yaitu PIL(*Python Imaging Library*) yang akan digunakan untuk membuka, memanipulasi dan menyimpan diri dari berbagai format *file* gambar.



Jika semua pengulangan telah selesai, maka akan disimpan dalam sebuah file dengan ekstensi “trainer.yml”. Selanjutnya dapat mengambil semua data *user* dari *dataset* yang telah dibuat sebelumnya lalu membuat folder “trainer” dalam directory yang sama dengan *source code* selanjutnya di proses menggunakan **OpenCV Recognizer** dan dilakukan secara langsung dengan fungsi **OpenCV.** Lalu hasil dari proses training tersebut adalah dalam bentuk file “trainer.yml” yang akan disimpan pada folder “trainer”. Maka untuk hasil training dalam bentuk file “trainer.yml” tersebut akan ditampilkan pada Gambar 4.6



**Gambar 4.6 *Directory* hasil *training dataset***

* 1. **Pengujian Sistem Identifikasi Wajah**

Dalam proses pengujian (*testing*) ini dilakukan saat menguji akurasi dan kesesuaian metode *face recognition* untuk dapat mengidentifikasi seseorang. Tujuan dari pengujian dari proses ini adalah untuk memvalidasi bahwa sistem yang dibangun berjalan dengan baik.

Proses ini dilakukan dengan metode *Haar cascade classifier* dan *Local Binary Pattern Histogram* (LBPH) dengan menggunakan 6 Sampel wajah yang berbeda. Proses pengujian ini akan dilakukan dengan cara merekam atau meng-*capture* wajah *user* sebanyak 100 gambarlalu merubah gambar menjadi *grayscale.* Kemudian akan tersimpan kedalam folder “dataset” yang berada dalam *directory* yang sama dengan *source code* selanjutnya kumpulan *dataset* tersebut akan dilatih menggunakan metode *Local Binary Pattern Histogram* (LBPH) , setelah berhasil dilatih akan tersimpan pada folder trainer dan hasil pelatihan dataset tadi akan mengeluarkan *output* berupa file “trainer.yml”. Lalu selanjutnya menjalankan file “face\_recognition” untuk melakukan pengujian identifikasi wajah dapat dilakukan ketika proses training dataset telah berhasil.



**Gambar 4.7 *Source code* untuk inisialisasi id yang sudah terdaftar**

* + 1. **Hasil Pengujian dan Analisa**

Hasil pengujian akurasi *face recognition* akan disajikan dalam bentuk tabel agar lebih mudah dimengerti oleh pembaca. Tabel penyajian akan memiliki 4 kolom yaitu no, jarak, hasil ,mengenal wajah ya/tidak. Untuk dapat melakukan pengujian menggunakan parameter seperti berikut ini:

1. Pengaruh jarak terhadap tingkat keberhasilan dan akurasi pengenalan wajah yang sudah tersimpan di database. Tabel 4.3, 4.4, 4.5 , 4.6 merupakan hasil uji coba terhadap jarak untuk pengenalan wajah.

**Tabel 4.3 Hasil Pengujian Sampel 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Jarak | Hasil | Mengenal Wajah | |
| Ya | Tidak |
| 1.1 | 40 cm | Screenshot_81.jpg | ✓ |  |
| 1.2 | 70 cm | Screenshot_83.jpg | ✓ |  |
| 1.3 | 100 cm | Screenshot_85.jpg | ✓ |  |
| 1.4 | 180 cm | Screenshot_91.jpg |  | ✓ |

**Tabel 4.3 Hasil Pengujian Sampel 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Jarak | Hasil | Mengenal Wajah | |
| Ya | Tidak |
| 1.2.1 | 40 cm | Screenshot_32 | ✓ |  |
| 1.2.2 | 70 cm | Screenshot_33 | ✓ |  |
| 1.2.3 | 100 cm | Screenshot_34 | ✓ |  |
| 1.2.4 | 180 cm | Screenshot_35 |  | ✓ |

**Tabel 4.3 Hasil Pengujian Sampel 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Jarak | Hasil | Mengenal Wajah | |
| Ya | Tidak |
| 1.3.1 | 40 cm | Screenshot_27.jpg | ✓ |  |
| 1.3.2 | 70 cm | Screenshot_47 | ✓ |  |
| 1.3.3 | 100 cm | Screenshot_48 | ✓ |  |
| 1.3.4 | 180 cm | Screenshot_49.jpg |  | ✓ |

**Tabel 4.3 Hasil Pengujian Sampel 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Jarak | Hasil | Mengenal Wajah | |
| Ya | Tidak |
| 1.4.1 | 40 cm | Screenshot_72.jpg | ✓ |  |
| 1.4.2 | 70 cm | Screenshot_73.jpg | ✓ |  |
| 1.4.3 | 100 cm | Screenshot_76.jpg | ✓ |  |
| 1.4.4 | 180 cm | Screenshot_75.jpg |  | ✓ |

**Tabel 4.3 Hasil Pengujian Sampel 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Jarak | Hasil | Mengenal Wajah | |
| Ya | Tidak |
| 1.5.1 | 40 cm | Screenshot_31.jpg | ✓ |  |
| 1.5.2 | 70 cm | Screenshot_41.jpg | ✓ |  |
| 1.5.3 | 100 cm | Screenshot_42.jpg | ✓ |  |
| 1.5.4 | 180 cm | Screenshot_43.jpg |  | ✓ |

**Tabel 4.3 Hasil Pengujian Sampel 6**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Jarak | Hasil | Mengenal Wajah | |
| Ya | Tidak |
| 1.6.1 | 40 cm | Screenshot_63.jpg | ✓ |  |
| 1.6.2 | 70 cm | Screenshot_64.jpg | ✓ |  |
| 1.6.3 | 100 cm | Screenshot_65.jpg | ✓ |  |
| 1.6.4 | 180 cm | Screenshot_66.jpg |  | ✓ |

Pada Tabel 4.3 menunjukkan bahwa terdapat wajah yang dapat dideteksi dan dikenali pada jarak 40 cm hingga pada jarak 200 cm sudah tidak dapat mengenali wajah tetapi masih dapat mendeteksi wajah.

|  |  |
| --- | --- |
| **Total Pengujian Benar** | **20** |
| **Total Pengujian Salah** | **4** |
| **Total Data Pengujian** | **24** |

Lalu berdasarkan dari tabel Tabel 4.3, 4.4, 4.5, 4.6 di atas yaitu, dapat disimpulkan bahwa dari 5 sampel data wajah orang yang berbeda, terdapat 15 pengujian yang dapat dikenali oleh sistem dengan benar, dan terdapat 5 pengujian gambar yang tidak dapat dikenali tapi masih bisa mendeteksi wajah.

Cara untuk menghitung nilai akurasi dan kesalahan dari proses pengujian diatas dapat dihitung dengan menggunakan persamaan berikut ini.

x100%

x100%

*Akurasi =* 83.33%

Sedangkan proses perhitungan untuk data yang tidak berhasil dikenali atau dideteksi adalah sebagai berikut.

x100%

x100%

*Akurasi =* 16.67% atau 17%

Dari perhitungan akurasi dan kesalahan di atas, didapat bahwa persentase nilai akurasi sistem pengenalan wajah untuk melakukan kehadiran otomatis dalam aplikasi “*Smart Attendance Recognition*. sebesar 83.33% dengan kesalahan sebesar 16.67% atau jika dibulatkan menjadi 17%.

1. Pengaruh tingkat kemiringan wajah terhadap keberhasilan dan akurasi pengenalan wajah yang sudah tersimpan di *database*. Tabel 4.7, 4.8, 4.9, 4.10, 4.11 merupakan hasil dari pengujian terhadap kemiringan wajah untuk pengenalan wajah.

**Tabel 4.7 Hasil Pengujian *Sampel* 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Drajat kemiringan wajah | Hasil | Mengenal Wajah | |
| Ya | Tidak |
| 2.1.1 | Tegak lurus | Screenshot_82.jpg | ✓ |  |
| 2.1.2 | 10°  Ke kanan | Screenshot_93.jpg | ✓ |  |
| 2.1.3 | 10°  Ke kiri | Screenshot_95.jpg | ✓ |  |
| 2.1.4 | 10°  Ke atas | Screenshot_92.jpg | ✓ |  |
| 2.1.5 | 40°  Ke atas | Screenshot_96.jpg |  | ✓ |

**Tabel 4.8 Hasil Pengujian Sampel 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Drajat kemiringan wajah | Hasil | Mengenal Wajah | |
| Ya | Tidak |
| 2.2.1 | Tegak lurus | Screenshot_33.jpg | ✓ |  |
| 2.2.2 | 10°  Ke kanan | Screenshot_38.jpg | ✓ |  |
| 2.2.3 | 10°  Ke kiri | Screenshot_37.jpg | ✓ |  |
| 2.2.4 | 10°  Ke atas | Screenshot_39.jpg | ✓ |  |
| 2.2.5 | 40°  Ke atas | Screenshot_28.jpg |  | ✓ |

**Tabel 4.9 Hasil Pengujian Sampel 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Drajat kemiringan wajah | Hasil | Mengenal Wajah | |
| Ya | Tidak |
| 2.3.1 | Tegak lurus | Screenshot_47.jpg | ✓ |  |
| 2.3.2 | 10°  Ke kanan | Screenshot_52.jpg | ✓ |  |
| 2.3.3 | 10°  Ke kiri | Screenshot_51.jpg | ✓ |  |
| 2.3.4 | 10°  Ke atas | Screenshot_53.jpg | ✓ |  |
| 2.3.5 | 40°  Ke atas | Screenshot_29.jpg |  | ✓ |

**Tabel 4.10 Hasil Pengujian Sampel 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Drajat kemiringan wajah | Hasil | Mengenal Wajah | |
| Ya | Tidak |
| 2.4.1 | Tegak lurus | Screenshot_44.jpg | ✓ |  |
| 2.4.2 | 10°  Ke kanan | Screenshot_45.jpg | ✓ |  |
| 2.4.3 | 10°  Ke kiri | Screenshot_57.jpg | ✓ |  |
| 2.4.4 | 10°  Ke atas | Screenshot_58.jpg | ✓ |  |
| 2.4.5 | 40°  Ke atas | Screenshot_59.jpg |  | ✓ |

**Tabel 4.11 Hasil Pengujian Sampel 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Drajat kemiringan wajah | Hasil | Mengenal Wajah | |
| Ya | Tidak |
| 2.5.1 | Tegak lurus | Screenshot_72.jpg | ✓ |  |
| 2.5.2 | 10°  Ke kanan | Screenshot_77.jpg | ✓ |  |
| 2.5.3 | 10°  Ke kiri | Screenshot_78.jpg | ✓ |  |
| 2.5.4 | 10°  Ke atas | Screenshot_79.jpg | ✓ |  |
| 2.5.5 | 40°  Ke atas | Screenshot_80.jpg |  | ✓ |

**Tabel 4.12 Hasil Pengujian Sampel 6**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Drajat kemiringan wajah | Hasil | Mengenal Wajah | |
| Ya | Tidak |
| 2.6.1 | Tegak lurus | Screenshot_63 | ✓ |  |
| 2.6.2 | 10°  Ke kanan | Screenshot_67 | ✓ |  |
| 2.6.3 | 10°  Ke kiri | Screenshot_68 | ✓ |  |
| 2.6.4 | 10°  Ke atas | Screenshot_69 | ✓ |  |
| 2.6.5 | 40°  Ke atas | Screenshot_70 |  | ✓ |

Pada Tabel 4.7, 4.8, 4.9, 4.10, 4.11 menunjukakan bahwa tingkat kemiringan mempengaruhi pengenalan wajah. Dalam uji coba ini penulis melakukan kemiringan wajah ke kanan, ke kiri dan ke atas. Lalu ketika tingkat kemiringan kurang lebih 40 derajat wajah sudah tidak bisa dideteksi dan dikenali.

|  |  |
| --- | --- |
| **Total Pengujian Benar** | **24** |
| **Total Pengujian Salah** | **6** |
| **Total Data Pengujian** | **30** |

Lalu berdasarkan dari tabel 4.7, 4.8, 4.9, 4.10, 4.11 di atas yaitu, dapat disimpulkan bahwa dari 5 sampel data wajah orang yang berbeda terdapat 20 pengujian yang dapat dikenali oleh sistem dengan benar, dan terdapat 5 pengujian gambar yang tidak dapat terdeteksi.

Cara untuk menghitung nilai akurasi dan kesalahan dari proses pengujian diatas dapat dihitung dengan menggunakan persamaan berikut ini.

x100%

x100%

*Akurasi =* 80.00%

Sedangkan proses perhitungan untuk data yang tidak berhasil dikenali atau dideteksi adalah sebagai berikut.

x100%

x100%

*Akurasi =* 20.00%

Dari perhitungan akurasi dan kesalahan di atas, didapat bahwa persentase nilai akurasi sistem pengenalan wajah untuk melakukan kehadiran otomatis dalam aplikasi “*Smart Attendance Recognition*. sebesar 80.00% dengan kesalahan sebesar 20.00% atau jika dibulatkan menjadi 20%.

**4.5.1 *Database* Sistem Kehadiran**

Dari hasil pendeteksian dan pengenalan yang sudah dilakukan akan mengeluarkan *output* dalam bentuk file c.sv yang bernama absen.csv berfungsi sebagai penyimpanan data nama , tanggal dan waktu ketika melakukan presensi. Berikut tampilan *database* pada sistem kehadiran pengenalan wajah ini dapat dilihat pada Gambar 4.8

