

SKRIPSI

**RANCANGAN SOFTWARE SMART ATTENDENSE FACE RECOGNITION
MENGUNAKAN WEBCAME
(Program dan Database)**

Diajukan Sebagai Salah Satu Syarat

Untuk Memperoleh Gelar Sarjana Teknik Elektro



Oleh

TAUFIQ HIDAYATULLAH
191000220201020

PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH SUMATERA BARAT

2023

HALAMAN PENGESAHAN

RANCANGAN SOFTWARE SMART ATTENDENSE FACE RECOGNITION
MENGUNAKAN WEBCAME
(Program dan Database)

Oleh

Taufiq Hidayatullah
191000220201020

Dosen Pembimbing I,



Hariyadi, S.kom., M.Kom
NIDN. 1021068901

Dekan Fakultas Teknik
UM Sumatera Barat,



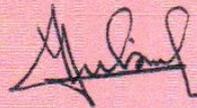
Musril, S.T., M.T.
NIDN. 1005057407

Dosen Pembimbing II,



Herris Yamashika, S.T., M.T.
NIDN. 1024038202

Ketua Program Studi
Teknik Elektro,



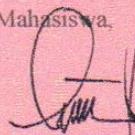
Ir. Yulisman, M.T.
NIDN. 8808220016

LEMBARAN PERSETUJUAN TIM PENGUJI

Skripsi ini telah dipertahankan dan disempurnakan berdasarkan masukan dan koreksi Tim Penguji pada ujian tertutup tanggal 28 februari 2023 di Fakultas Teknik Universitas Muhammadiyah Sumatera Barat

Bukittinggi, 28/februari 2023

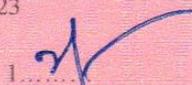
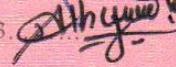
Mahasiswa,



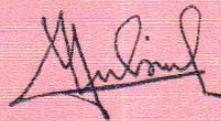
Taufiq Hidayatullah
191000220201020

Disetujui Tim Penguji Skripsi tanggal 28 februari 2023

1. Hariyadi, S.Kom.,M.Kom.
2. Herris yamashika,S.T.,M.T.
3. Mahyessi Kamil,S.T.,M.T.
4. Aggrivina Dwiharzandis,S.pd.,M.T.

1. 
2. 
3. 
4. 

Mengetahui,
Ketua Program Studi
Teknik Elektro



Ir. Yulisman, M.T
NIDN. 8808220016

LEMBARAN PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama Mahasiswa : TAUFIQ HIDAYATULLAH
Tempat dan Tanggal Lahir : PAKAN SINAYAN 11 DESEMBER 2000
NIM : 191000220201020
Judul Skripsi : RANCANGAN SOFTWARE SMART
ATTENDENSE FACE RECOGNITIONN
MENGUNAKAN WEBCAME

Menyatakan dengan sebenarnya bahwa penulisan Skripsi ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah laporan maupun kegiatan yang tercantum sebagai bagian dari Skripsi ini. Jika terdapat karya orang lain, saya akan mencantumkan sumber yang jelas.

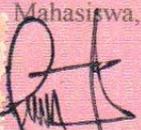
Demikian pernyataan ini saya buat dengan sesungguhnya dan apabila kemudian hari terdapat penyimpangan dan ketidak benaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai peraturan yang berlaku di UM Sumatera Barat.

Demikian pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun

Bukittinggi, 28 februari 2023

Mahasiswa,




Taufiq Hidayatullah
191000220201020

ABSTRAK

Penerapan *smart attendense* menggunakan *face recognition* banyak digunakan untuk tujuan mengurangi kecurangan pada pengambilan absensi. Penggunaan algoritma LBPH (*Local Binary Pattern Histogram*) dan algoritma *Haar Cascade Classifier* mampu mendeteksi dan mengenali wajah. Dalam penelitian ini penulis ingin merancang sebuah *soware* yang mampu mendeteksi wajah dan juga pengambilan absensi menggunakan pengenalan wajah. Dalam percobaan ini penulis menggunakan 210 dataset dari 3 orang mahasiswa masing masing menginputkan 70 wajah. Hasil dari pengujian software dengan menggunakan metode deteksi wajah didapat akurasi 90 – 100 %.

Kata Kunci : Smart Attendense, Face Recognition, Haar Cascade, Local Binary Pattern Histogram



ABSTRACT

Application smart attendense useface recognition widely used for the purpose of reducing fraud in taking attendance. Using the LBPH algorithm (Local Binary Pattern Histogram) and algorithmsHer Cascade Classifiercapable of detecting and recognizing faces. In this study the authors wanted to design a software capable of detecting faces and also taking attendance using facial recognition. In this experiment the authors used 210 datasets from 3 student to input 70 faces. The results of software testing using the face detection method obtained an accuracy of 90 – 100%.

keywords: Smart Attendense, Face Recognition, Haar Cascade, Local Binary Pattern Histogram



KATA PENGANTAR

Puji syukur kepada Allah SWT atas segala berkat yang telah diberikan-Nya, sehingga skripsi ini dapat diselesaikan. Skripsi ini merupakan salah satu kewajiban yang harus diselesaikan untuk memenuhi sebagian persyaratan akademik untuk memperoleh gelar Sarjana Teknik Elektro di Universitas Muhammadiyah Sumatera Barat (UM Sumatera Barat).

Penulis menyadari bahwa tanpa bimbingan, bantuan, dan doa dari berbagai pihak, skripsi ini tidak akan dapat diselesaikan tepat pada waktunya. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam proses pengujian skripsi ini, yaitu kepada:

1. Orang tua, kakak, dan adik serta seluruh keluarga yang telah memberikan dukungan moral, doa, dan kasih sayang;
2. Bapak Masril, S.T., M.T Selaku Dekan Fakultas Teknik UM Sumatera Barat;
3. Bapak Hariyadi, S.Kom., M.Kom Selaku Wakil Dekan Fakultas Teknik UM Sumatera Barat, Dosen pembimbing akademik, Pembimbing I skripsi yang telah memberikan bimbingan dan banyak memberikan masukan kepada penulis
4. Bapak Ir.Yulisman, M.T., Selaku Ketua Program Studi Teknik Elektro;
5. Bapak Herry Yamashika, S.T., M.T Selaku Dosen Pembimbing II skripsi yang telah memberikan bimbingan dan banyak memberikan masukan kepada penulis;
6. Bapak/Ibu Tenaga Kependidikan Fakultas Teknik UM Sumatera Barat;
7. Bapak/Ibu mentor *orbit future academy* yang telah memberi banyak ilmu tentang *artificial intelligence*;
8. Semua pihak yang namanya tidak dapat disebutkan satu per satu.

Akhir kata, penulis menyadari bahwa mungkin masih terdapat banyak kekurangan dalam skripsi ini. Oleh karena itu, saran dari pembaca semua pihak yang membacanya, khususnya mahasiswa teknik elektro.

Bukittinggi, 28 februari 2023

Taufiq Hidayatullah

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
LEMBARAN PERSETUJUAN TIM PENGUJI	iii
LEMBARAN PERNYATAAN KEASLIAN	iv
ABSTRAK.....	v
KATA PENGANTAR	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	1
1.3. Batasan Masalah	2
1.4. Tujuan dan Manfaat Penelitian	2
1.4.1. Bagi penulis	2
1.4.2. Bagi instansi	2
1.4.3. Bagi pembaca	2
1.5. Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	4
2.1 Landasan Teori.....	4
2.1.1 Wajah.....	4
2.1.2 Pengenalan Wajah	4
2.1.3 Absensi	4
2.1.4 Python.....	4
2.1.5 OpenCV.....	4
2.1.6 Haar Cascade.....	5
2.1.7 Local Binary Pattern Histogram.....	6

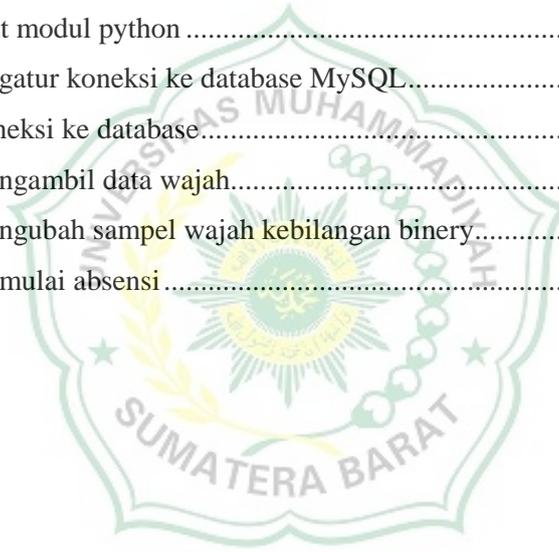
2.2	Kajian Penelitian Sebelumnya	6
BAB III METODOLOGI PENELITIAN		8
3.1	Lokasi dan Waktu Penelitian	8
3.1.1	Lokasi Penelitian	8
3.1.2	Waktu Penelitian	8
3.2	Data Penelitian	8
3.2.1	Jenis dan Sumber Data	8
3.2.2	Teknik Pengumpulan Data	9
3.3	Bagan alir Penelitian	10
3.3.1	Studi Pustaka	10
3.3.2	Pengumpulan Data	10
3.3.3	Perancangan Sistem.....	11
3.3.4	Implementasi	11
3.3.5	Analisa Hasil	11
3.3.6	Penyelesaian Laporan Penelitian	11
3.4	Rancangan sistem.....	12
3.4.1	Rancangan tampilan	12
3.4.2	Database	12
3.4.3	Flowchart.....	13
3.4.4	Metode Pengumpulan Data	14
3.4.5	Deteksi Wajah	14
3.4.6	Ekstrasi Fitur	15
3.4.7	Aplikasi yang digunakan	15
3.4.8	Libray python yang digunakan	16
BAB IV HASIL DAN PEMBAHASAN		17
4.1	Hasil rancangan sistem.....	17
4.1.1	Tampilan aplikasi absensi	17
4.1.2	Database.....	19

4.1.3 kode program	20
4.2 Pengujian sistem	33
4.2.1 deteksi wajah dan hasil akurasi.....	33
BAB V PENUTUP	35
5.1 Kesimpulan	35
5.2 Saran	35
Daftar Pustaka.....	36
Lampiran	



DAFTAR GAMBAR

Gambar 2. 1 struktur cascade classifier.....	5
Gambar 3. 1 Bagan akir penelitian.....	10
Gambar 3. 2 Rancangan tampilan aplikasi.....	12
Gambar 3. 3 Flowchart proses absensi.....	13
Gambar 4. 1 Tampilan halaman awal aplikasi.....	17
Gambar 4. 2 Halaman input data	18
Gambar 4. 3 Rekam data wajah	18
Gambar 4. 4 Pengambilan absensi	19
Gambar 4. 5 Hasil absensi	19
Gambar 4. 6 Data wajah.....	20
Gambar 4. 7 Data training.....	20
Gambar 4. 8 input modul python	20
Gambar 4. 9 mengatur koneksi ke database MySQL.....	23
Gambar 4. 10 koneksi ke database.....	25
Gambar 4. 11 mengambil data wajah.....	27
Gambar 4. 12 mengubah sampel wajah kebilangan binery.....	29
Gambar 4. 13 memulai absensi.....	30



DAFTAR TABEL

Tabel 2. 1 Penelitian sebelumnya	Error! Bookmark not defined.
Tabel 3. 1 Waktu penelitian	8
Tabel 3. 2 Presensi	12
Tabel 3. 3 Data wajah	13
Tabel 3. 4 Data user	13
Tabel 4. 1 Hasil pengujian sistem	33
No table of figures entries found. Tabel 4. 2 Hasil pengujian sistem	34



BAB I

PENDAHULUAN

1.1. Latar Belakang

Penerapan kecerdasan buatan atau *Artificial Intelligence* (AI) pada saat sekarang ini sudah banyak dikembangkan. Penggunaan AI pada berbagai bidang memberi banyak manfaat dan kemudahan, sehingga kita dituntut untuk mampu dalam mengikuti perkembangan teknologi *Artificial Intelligence* (AI). Salah satu kemampuan AI, mampu mendeteksi dan mengklasifikasikan berdasarkan data. Dengan data, AI mampu dilatih menggunakan menggunakan algoritma *machine learning* sehingga akurasi dari deteksi AI memiliki akurasi yang tinggi.

Kebutuhan akan data pada saat sekarang ini sangatlah penting karena dengan data, perusahaan mampu memberi penawaran yang terbaik bagi calon konsumen. Dengan data AI mampu memprediksi cuaca hari ini, besok ataupun selanjutnya dan dengan data, kepolisian mampu mengidentifikasi keberadaan oknum kejahatan sehingga dengan cepat ditemukan keberadaannya menggunakan data pengenalan wajah (*face detection*).

Penggunaan dan penerapan pengenalan wajah (*face detection*) pada sebuah aplikasi sudah banyak dikembangkan. Terdapat berbagai macam metode pengenalan wajah sudah tersedia. Salah satunya metode LBPH (*Local Binary Pattern Histogram*). Citra wajah yang diambil secara realtime menggunakan kamera akan dibandingkan dan dicocokkan menggunakan histogram yang sudah diekstraksi dengan citra wajah yang ada pada database. Metode ini banyak digunakan salah satunya pada absensi menggunakan wajah.

Pada saat ini masih banyak diterapkan absensi manual. Penggunaan absensi manual sangat berpotensi terjadinya kecurangan pada pengambilan absensi. Melihat kebutuhan ini maka perlu adanya absensi yang sulit dicurangi salah satunya dengan menggunakan absensi pengenalan wajah. Dari permasalahan tersebut maka penulis mengusulkan judul “**RANCANGAN SOFTWARE SMART ATTENDENSE FACE RECOGNITION MENGGUNAKAN WEBCAME**”

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah di uraikan di atas, maka penulis dapat mengabil rumusan masalah yaitu bagaimana merancang absensi

menggunakan wajah sehingga data absensi dapat diperkuat dengan sistem pengenalan wajah secara *real time*?

1.3. Batasan Masalah

Adapun beberapa poin batasan masalah agar skripsi ini lebih terfokus dan terperinci sehingga tercapai tujuan yang diantaranya sebagai berikut :

1. Untuk menganalisis pendeteksian pola wajah menggunakan metode haar cascade
2. Untuk membandingkan pola wajah yang terdeteksi dengan pola wajah yang ada di dataset menggunakan metode LBPH (*Local Binary Pattern Histogram*)
3. Aplikasi ini berupa rancangan *software* absensi menggunakan wajah yang terdiri dari : rekam data, training data, mendeteksi wajah dan menyimpan data absensi ke sql.
4. Aplikasi ini di bangun menggunakan bahasa pemrograman *python* dan dibuat menggunakan *software visual studio code*

1.4. Tujuan dan Manfaat Penelitian

1.4.1. Bagi penulis :

1. Membuka wawasan dan pengetahuan baru tentang teknologi *artificial intelligence*
2. Menerapkan ilmu baik secara teknis maupun teori yang telah diperoleh selama mengikuti pendidikan ke dalam aplikasi nyata.

1.4.2. Bagi instansi :

1. Membantu kampus um sumatera barat dalam pengambilan data absensi dengan menggunakan pendeteksian wajah.
2. Hasil data wajah yang didapat dari perekaman wajah bisa digunakan untuk pengembangan berbagai aplikasi pendeteksi wajah
3. Data hasil pelatihan *mechin* (data training) bisa di manfaatkan untuk algoritma deteksi wajah

1.4.3. Bagi pembaca :

1. Sebagai bahan referensi untuk penelitian.
2. Sebagai sumber bacaan.

1.5. Sistematika Penulisan

Sistematika penulisan yang digunakan untuk memudahkan pemahaman pembaca dalam mengambil inti yang terdapat pada skripsi ini, maka skripsi ini disusun dengan sistematika penulisan sebagai berikut:

Bab I Pendahuluan

Bab ini berisi latar belakang pengambilan tema, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian serta sistematika penulisan.

Bab II Tinjauan pustaka

Bab ini berisi tentang landasan teori dan hasil penelitian yang dilakukan sebelumnya.

Bab III. Metode Penelitian

Bab ini berisi tentang rencana dan prosedur penelitian yang digunakan oleh penulis untuk memperoleh jawaban yang sesuai dengan permasalahan dan tujuan penelitian. Diantaranya penjelasan lokasi penelitian, data penelitian, metode analisis data serta rancangan sistem.

Bab IV Hasil dan Pembahasan

Pada bab ini penulis membahas cara menggunakan aplikasi, tampilan aplikasi, hasil data absensi dan kesimpulan dari penggunaan metode (*Local Binary Pattern Histogram*).

Bab V Penutup

Pada bab ini penulis memberikan kesimpulan dari hasil penelitian, keterbatasan penelitian serta implikasi dan saran bagi penelitian pada topik yang sama di masa yang akan datang.

Daftar pustaka

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Wajah

Wajah adalah bagian depan dari kepala manusia meliputi wilayah dari dahi hingga dagu, termasuk rambut, dahi, alis, mata, hidung, mulut dan bibir. Wajah digunakan untuk menentukan ekspresi, penampilan, serta identitas dari seseorang. Tidak ada satupun wajah yang serupa, bahkan kembar identic sekalipun.[1]

2.1.2 Pengenalan Wajah

Suatu program komputer yang secara otomatis menggunakan gambar digital atau video untuk mengidentifikasi atau mengautentifikasi seseorang secara otomatis.[2]

2.1.3 Absensi

Absensi adalah suatu kegiatan atau rutinitas yang dilakukan untuk membuktikan seseorang hadir atau tidak dalam suatu kelas atau suatu pertemuan. [3]

2.1.4 Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.

Python merupakan salah satu bahasa pemrograman tingkat tinggi (*high level language*). *Python* dirancang untuk memberikan kemudahan bagi *programmer* melalui segi efisiensi waktu, kemudahan dalam pengembangan dan kompatibilitas dengan sistem. *Python* bisa digunakan untuk membuat aplikasi *standalone* (berdiri sendiri) dan pemrograman script (*scripting programming*). [4]

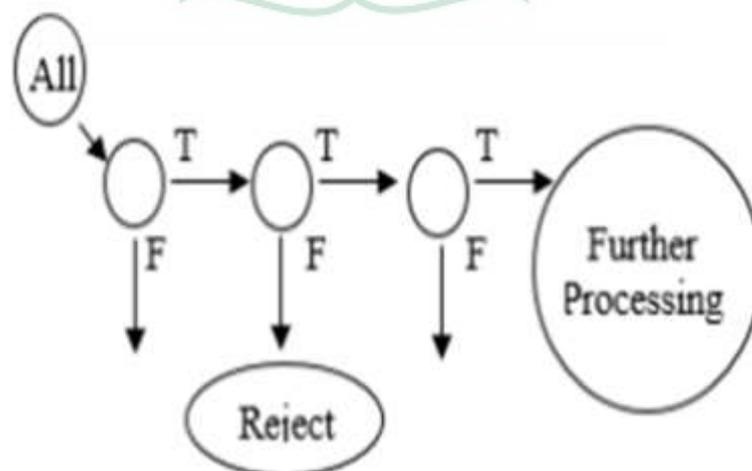
2.1.5 OpenCV

OpenCV adalah *open source C++ library* untuk *image processing* dan *computer vision*. Secara teori *OpenCV* digunakan seperti

meniru cara kerja sistem visual manusia yaitu dengan melihat objek melalui "penglihatan/mata" dan citra pada objek tersebut diteruskan ke otak untuk memproses sehingga mengerti objek apa yang tampak pada pandangan mata manusia. *OpenCV* merupakan salah satu cabang *Artificial intellegent* (kecerdasan buatan) yang digunakan untuk pengembangan atau analisis isi suatu gambar. mengidentifikasi objek, dan lain-lain. Di dalam *OpenCV*, untuk algoritma pengenalan wajah yang disediakan sampai pada saat penelitian ditulis, telah tersedia tiga algoritma pengenalan wajah, diantaranya *Eigenface*, *Fisherface*, dan *Local Binary Pattern Histogram* (LBPH).[5]

2.1.6 Haar Cascade

Haar Cascade Classifier yang merupakan sebuah *library* yang tersedia dalam *Open-CV*, dibangun di atas Bahasa *C/C++* dengan API (*Application Programming Interface*) *python*. *Haar Cascade* mengkombinasikan tiga hal yang menjadi dasar. Pertama adalah memiliki satu set fitur yang luas dapat dihitung secara tepat dan cepat, hal ini dapat mengurangi variabilitas dalam satu kelas dan meningkatkan variabilitas antar kelas. Kedua adalah menerapkan algoritma yang memungkinkan pemilihan dari sebuah fitur dan pelatihan. Ketiga adalah membentuk *cascade* secara bertahap dengan hasil klasifikasi dan skema deteksi yang lebih kompleks, cepat dan efisien. Struktur dari *Haar Cascade Classifier* bisa digambarkan pada gambar 1.



Gambar 2. 1 Struktur cascade classifier

Haar Cascade bisa dilatih untuk mendeteksi beberapa objek, yang harus kita lakukan adalah dengan menentukan area pada wajah yang memiliki kemungkinan tertinggi. Wajah tersebut memiliki kulit dan memiliki tingkat piksel warna pada kulit. Pemilihan teknik segmentasi dipilih untuk warna piksel pada wajah. Kemudian memvalidasinya dengan *haar cascade classifier*, Jika *piksel* yang divalidasinya sesuai dengan geometriknya maka sistem telah menemukan wajah yang dimaksud, jika tidak sesuai maka sistem mengabaikannya[6]

2.1.7 Local Binary Pattern Histogram

Local Binary Pattern Histogram (LBPH) adalah teknik baru dari metode *Local Binary Pattern* (LBP) untuk mengubah performa hasil pengenalan wajah. LBP adalah deskriptor tekstur yang dapat juga digunakan untuk mewakili wajah, karena gambar wajah dapat dilihat sebagai sebuah komposisi *micro-texture-pattern* yaitu suatu operator non parametrik yang menggambarkan tata ruang lokal citra.[7]

2.2 Kajian Penelitian Sebelumnya

Pada penelitian terdahulu, metode (*Face Recognition*) seperti Metode *Eigenface*, Metode *Principal Component Analysis* (PCA), dan Metode *Local Binary Pattern Histogram* (LBPH) telah banyak di kembangkan untuk mengembangkan aplikasi absensi pengenalan wajah. Pada penelitian ini penulis berfokus pada penggunaan metode *Local Binary Pattern Histogram* (LBPH).

Tabel 2. 1 Penelitian sebelumnya

No	Nama	Judul	Tahun	Hasil Penelitian
1	Fajri dkk	Sistem Absensi Berbasis Pengenalan Wajah Secara <i>Real Time</i> menggunakan Metode <i>Fisherface</i>	2020	hasil penelitian menunjukkan bahwa tingkat akurasi pengenalan wajah mencapai 80% dengan jumlah data yang di ambil sebanyak 10 sampel wajah, masing-masing wajah

				mencecapture sebanyak 10 gambar sehingga data training mencapai 100 gambar. Dalam pengambilan data untuk database menggunakan kamera <i>webcame</i> .
2	Angga wahyu wibowo dkk	Pendeteksian dan Pengenalan Wajah Pada Foto Secara <i>Real Time</i> Dengan <i>Haar Cascade</i> dan <i>Local Binary Pattern Histogram</i>	2020	hasil penelitian menunjukkan penggunaan metode <i>Haar Cascade</i> dan LBPH bisa mendekteksi wajah dari jarak 0 – 40 cm dan sistem ini tidak dapat mendeteksi wajah lebih dari 40 cm. penelitian ini menggunakan 240 sampel wajah dalam dataset dengan peengambilan data wajah menggunakan <i>webcame</i> .
3	Linda rahma yanti dkk	Sistem Absensi Mahasiswa Berdasarkan Citra Wajah Menggunakan Metode <i>Principal Component Alalysis (PCA)</i>	2017	Hasil penelitian menunjukkan tingkat akurasi wajah mencapai 85% dengan jumlah dataset sebanyak 25 gambar dari 5 orang mahasiswa masing – masing mahasiswa menginputkan 5 wajah.
4	Prince Richard setiono dkk	Aplikasi Pengenalan Wajah Untuk Sistem Absensi Kelas Berbasis <i>Raspberry Pi</i>	2020	pengunaan kamera <i>raspberry pi nightvision</i> untuk mendeteksi wajah kurang baik untuk digunakan karena terkendala pencahayaan. Penelitian ini menggunakan 5 dataset dari 5 mahasiswa yang berbeda dengan menggunakan algoritma <i>local binary pattern histogram (LBPH)</i>

BAB III

METODE PENELITIAN

3.1 Lokasi dan Waktu Penelitian

3.1.1 Lokasi Penelitian

Pada penelitian ini lokasi yang akan digunakan dalam pengambilan data yakni salah satu ruangan labor teknik elektro Kampus III Universitas Muhammadiyah Sumatera Barat, yang terletak di Jl. By Pass Aur Kuning, No. 1 km. 9, Kelurahan Tarok Dipo, Kecamatan Guguk Panjang, Kota Bukittinggi

3.1.2 Waktu Penelitian

Tabel 3.1 Waktu penelitian

No	Keterangan	Bulan		
		November	Desember	Januari
1	Menentukan judul skripsi	✓		
2	Mencari referensi skripsi	✓		
3	Membuat proposal pengajuan judul	✓		
4	Perancangan sistem kerja software		✓	
5	Pembuatan desain software		✓	
6	Pembuatan program software		✓	
7	Pengujian software		✓	
8	Evaluasi software			
9	Penelitian penggunaan algoritma LBPH			✓
10	Hasil penelitian			✓
11	Pembuatan skripsi			✓

3.2 Data Penelitian

3.2.1 Jenis dan Sumber Data

1. Sumber Data Primer

Metode penelitian ini dilakukan langsung pada objek penelitian, data serta keterangan yang dikumpulkan dengan cara observasi dan wawancara

2. Sumber Data Sekunder (Dokumentasi)

Sumber data sekunder dalam penelitian ini adalah dengan memperoleh dokumen bersangkutan dengan obyek yang diteliti, yang dimaksudkan sebagai bukti bahwa penelitian benar – benar dilakukan pada instansi, dan juga buku, ataupun pencarian dari internet

3.2.2 Teknik Pengumpulan Data

1. Observasi

Observasi adalah teknik pengumpulan data yang dilakukan melalui suatu pengamatan, dengan disertai pencatatan-pencatatan terhadap keadaan atau perilaku obyek sasaran. Dengan demikian Penulis mendatangi langsung lokasi penelitian dan melakukan serangkaian pengamatan untuk mendapatkan data.

2. Wawancara

Wawancara merupakan proses komunikasi yang sangat menentukan dalam proses penelitian. Dengan wawancara data yang diperoleh akan lebih mendalam, karena mampu menggali pemikiran atau pendapat secara detail. Wawancara yang digunakan peneliti adalah wawancara semi terstruktur, yaitu untuk menemukan permasalahan secara lebih terbuka dan pihak yang diajak wawancara diminta pendapatnya dan mencatat apa yang dikemukakan oleh informan.

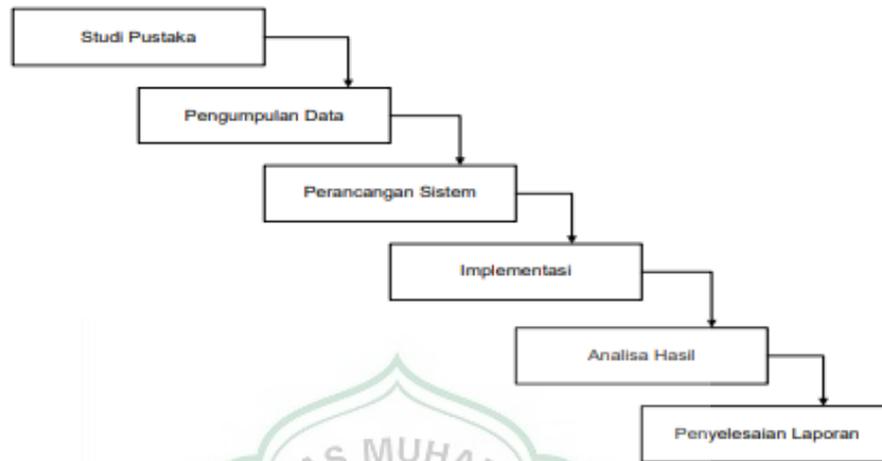
Proses wawancara ini, dilakukan peneliti dengan mewawancarai Bapak ir yulisman M.T selaku ketua program studi teknik elektro di Universitas Muhammadiyah Sumatera Barat, dan beberapa mahasiswa aktif yang berada di Universitas Muhammadiyah Sumatera Barat diantaranya Bapak July Mardi dan Afri Warman.

3. Dokumentasi

Metode dokumentasi Metode dokumentasi yaitu mencari data mengenai hal-hal atau variabel yang berupa catatan, buku, surat kabar, majalah, prasasti, notulen rapat, agenda, dan sebagainya. Adapun data yang diharapkan peneliti adalah sejarah berdirinya

Universitas Muahmmadiyah Sumatera Barat, letak geografis Universitas Muahmmadiyah Sumatera Barat, struktur organisasi, keadaan jumlah staf pendidikan dan jumlah mahasiswa, dan lain sebagainya.

3.3 Bagan alir Penelitian



Gambar 3. 1 Bagan alir penelitian

Perancangan metodologi penelitian pada tugas akhir ini terdiri dari Studi Pustaka, Pengumpulan Data, Perancangan Sistem, Implementasi, Analisa Hasil, dan Penyelesaian laporan. Berikut ini penjelasan dari metode yang digunakan:

3.3.1 Studi Pustaka

Penelitian yang dilakukan berdasarkan pada metode (*Local Binary Pattern Histogram*) LBPH dan dengan materi-materi tambahan sebagai bahan rujukan, diantaranya adalah beberapa jurnal yang berkaitan dengan kasus maupun metode yang peneliti lakukan, buku yang menunjang penelitian dan juga materimateri dari internet. Jurnal yang digunakan merupakan jurnal yang dipublikasi dalam kurung waktu lima tahun terakhir dan juga terdeteksi secara nasional maupun internasional. Buku yang digunakan merupakan buku dengan terbitan terbaru dengan kurung waktu lima tahun terakhir.

3.3.2 Pengumpulan Data

Dalam memperoleh informasi yang dibutuhkan dalam penelitian ini, dilakukan dengan cara observasi yang dilakukan selama bulan November 2022 di Fakultas Teknik Program Studi teknik elektro universitas

muhammadiyah sumatera barat. Informasi yang dibutuhkan dalam penelitian ini adalah bagaimana sistem absensi yang sedang berjalan saat ini dan juga data contoh wajah beberapa mahasiswa untuk uji coba deteksi wajah.

3.3.3 Perancangan Sistem

Penelitian yang akan dilakukan, peneliti membutuhkan beberapa data dan wajah mahasiswa dengan inputan wajah mahasiswa. Kemudian uji coba dilakukan. Pada proses deteksi wajah dan pengenalan wajah dilakukan pelatihan dari data training yang berjumlah 150 data citra. Data tersebut merupakan data dari 5 mahasiswa dengan masing-masing mahasiswa menginputkan 30 wajah.

3.3.4 Implementasi

Pada tahap ini meliputi pembuatan program yang terdiri dari pembuatan script dan interface aplikasi serta uji coba untuk mendapatkan nilai tingkat akurasi dari deteksi wajah, pengenalan wajah, dan aplikasi secara keseluruhan. Uji validitas dilakukan untuk menguji tingkat kehandalan aplikasi yang dibangun.

3.3.5 Analisa Hasil

Analisa hasil dari sistem absensi mahasiswa berdasarkan citra wajah adalah bagaimana aplikasi mampu mengenali wajah mahasiswa dan menampilkan akurasi pendeteksian.

3.3.6 Penyelesaian Laporan Penelitian

Penyelesaian laporan penelitian meliputi penulisan laporan dari Bab I Pendahuluan yang terdiri dari Latar belakang, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, Metodologi Penelitian, dan Sistematika Penulisan, Bab II Kajian Pustaka yang terdiri dari Penelitian Terdahulu dan Landasan Teori, Bab III Perancangan Sistem, Bab IV Implementasi, dan Bab V Penutup yang terdiri dari Simpulan dan Saran tentang sistem absensi yang dibangun.

3.4 Rancangan sistem

3.4.1 Rancangan tampilan

LOGO		Mulai presensi	Test Kamera	Remote	Train	Users	Pengaturan	About
LOGO		Papan Informasi		Input Data User				
		Jumlah User		Nim		Nama		
		Jam Masuk						
		Jam keluar						
Presensi Hari ini				Prodi		Kelas		
No	Nama	Jam masuk	Jam Keluar	Ubah Waktu Presensi				
				Jam Masuk		Jam Keluar		
				Dari		Dari		
				Simpan		Simpan		

Gambar 3. 2 Rancangan tampilan aplikasi

3.4.2 Database

Untuk penyimpanan data penulis menggunakan MySQL dengan nama database db_facial_recognition adapaun file data yang di perlukukan di antaranya

- 1) Presensi (tb_presensi)

Tabel 3. 2 Presensi

Id	integer	11
Nim	varchar	100
nama	varchar	300
tangal	date	
clockin	varchar	120
c_in	varchar	120
clocout	varchar	120
c_out	varchar	120
Img_in	varchar	300
Img_out	varchar	300

2) data wajah (tb_take_image)

Tabel 3. 3 Data wajah

Id	integer	11
nim	varchar	100
nama	varchar	100
prodi	Varchar	300
kelas	text	
status	integer	11

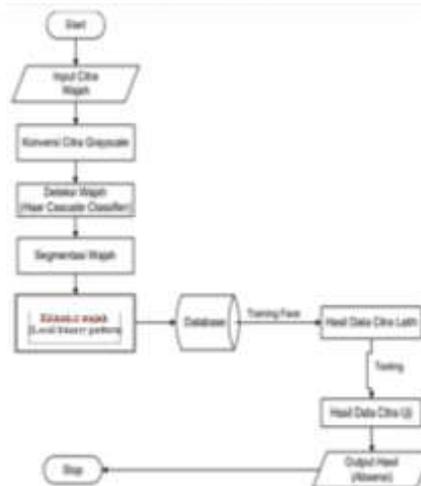
3) data user (tb_users)

Tabel 3. 4 Data user

Id	integer	11
Nim	varchar	100
nama	varchar	100
prodi	Varchar	300
kelas	text	
status	integer	11

3.4.3 Flowchart

Tahapan ini merupakan langkah-langkah pengenalan wajah untuk sistem absensi



Gambar 3. 3 Flowchart proses absensi

Berikut ini merupakan penjelasan dari proses pengenalan hingga identifikasi wajah:

1. Mahasiswa melakukan absensi dengan menghadap pada kamera
2. Aplikasi akan menangkap gambar pada layar
3. *Pre-processing* merupakan proses deteksi wajah, dimana pada proses ini peneliti menggunakan metode *Haar Cascade*, kemudian selanjutnya citra gambar akan dirubah dalam bentuk citra *grayscale*. Proses selanjutnya merupakan segmentasi wajah, dimana bagian wajah yang digunakan untuk perhitungan adalah bagian dahi hingga dagu.
4. Proses *ekstrasi* menggunakan metode LBPH dimana pada proses ini akan menyimpan nilai citra. Dalam hal ini, cara yang digunakan adalah membedakan objek dengan *background*.
5. Wajah yang dijadikan sebagai citra uji akan disimpan dalam database/folder *trained face*.
6. Saat mahasiswa melakukan proses absensi, wajah mahasiswa akan terdeteksi secara real time dan jika mahasiswa menekan tombol esc absensi selesai dan data akan masuk ke file sql.

3.4.4 Metode Pengumpulan Data

Data yang diambil pada penelitian ini berupa citra RGB dari webcam laptop, dimana pada citra tersebut terdapat objek wajah yang nantinya digunakan untuk proses pengenalan wajah. citra yang diambil pada saat mahasiswa menghadap webcam.

Data yang di ambil sebanyak 3 wajah dengan jumlah data sebanyak 210 gambar sehingga data training mencapai 210 gambar wajah yang sangat cukup untuk melakukan proses pengenalan wajah.

3.4.5 Deteksi Wajah

Mendeteksi wajah penulis menggunakan *algoritma Haar Cascade Classifier*. *Haar Cascade* mengkombinasikan tiga hal yang menjadi dasar. Pertama adalah memiliki satu set fitur yang luas dapat dihitung secara tepat dan cepat, hal ini dapat mengurangi variabilitas dalam satu kelas dan meningkatkan variabilitas antar kelas. Kedua adalah menerapkan algoritma yang memungkinkan pemilihan dari sebuah fitur

dan pelatihan. Ketiga adalah membentuk cascade secara bertahap dengan hasil klasifikasi dan skema deteksi yang lebih kompleks, cepat dan efisien.

3.4.6 Ekstraksi Fitur

Untuk dapat mengenali wajah pada dataset penulis menggunakan algoritma *Local binary pattern* (LBP) adalah salah satu dari metode yang terkenal dalam mengenali sebuah objek. Dalam hal ini, cara yang digunakan adalah membedakan objek dengan background. Citra wajah yang diambil secara realtime menggunakan kamera akan dibandingkan dan dicocokkan menggunakan histogram yang sudah diekstraksi dengan citra wajah yang ada pada database. Cara kerja LBP ditunjukkan pada gambar yang menjelaskan bahwa piksel yang berada di tengah diperoleh dengan cara membandingkan insensitasnya dengan insensitas piksel yang lain. Nilai dari piksel yang berada di tengah adalah ambang batas dari kedelapan piksel yang lainnya. Pada sebuah matriks tersebut nilai biner di tengah akan dibandingkan dengan nilai sekelilingnya. Jika nilai pada matriks tengah lebih tinggi dari nilai sekelilingnya, maka nilai matriks sekelilingnya akan bernilai '1' begitupun sebaliknya jika nilai pada matriks tengah lebih rendah dari nilai sekelilingnya, maka nilai matriks sekelilingnya akan bernilai '0'. Kemudian dihitung nilai histogram untuk membandingkan dan mencocokkan wajah yang ada pada kamera dengan yang ada di *database*.

3.4.7 Aplikasi yang digunakan

1. Visual Studio Code

Visual studio code adalah perangkat lunak penyunting kode sumber buatan Microsoft untuk *linux*, *macOS*, dan *windows*. *Visual studiocode* menyediakan berbagai fitur seperti penyorotan sintaksis, penyelesaian *code*, kutipan kode dan *merefaktor* kode.

2. Python

Python merupakan bahasa pemrograman yang dirancang untuk digunakan dalam berbagai aplikasi, pengembangan *software*, *website*, otomatisasi, hingga digunakan dalam berbagai ilmu data.

3. Xampp

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri, yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*.

3.4.8 Libray python yang digunakan

1. Opencv

OpenCV adalah sebuah pustaka perangkat lunak yang ditujukan untuk pengelolaan citra dinamis secara realtime, yang dibuat oleh *intel*, dan sekarang di dukung oleh *willow garage* dan *itseez*.

2. Tkinter

Tkinter adalah sebuah pengikat *Python* untuk peralatan *GUI Tk*. *Tkinter* merupakan antarmuka *Python* standar untuk peralatan *GUI Tk*, dan merupakan *GUI standar de facto* nya *Python*. *Tkinter* dimasukkan secara *default* kedalam instalasi *Python* untuk *Linux*, *Microsoft Windows* dan *Mac OS X*. Nama *Tkinter* berasal dari *Tk interface*.

3. Numpy

NumPy adalah sebuah pustaka untuk bahasa pemrograman *Python*, *NumPy* memberikan dukungan untuk himpunan dan matriks multidimensi yang besar, dan dilengkapi koleksi sejumlah besar fungsi matematika tingkat tinggi untuk beroperasi pada himpunan ini.

4. Pandas

pandas adalah perpustakaan perangkat lunak yang ditulis untuk bahasa pemrograman *Python* untuk manipulasi dan analisis data. Secara khusus, ia menawarkan struktur data dan operasi untuk memanipulasi tabel numerik dan deret waktu.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil rancangan sistem

4.1.1 Tampilan aplikasi absensi

1. Halaman awal

Pada tampilan awal aplikasi terdapat tampilan from input data,from ubah waktu presensi, hasil absensi, papan informasi, tombol mulai absensi, tombol tes kamera, tombol remote, tombol train, tombol user, tombol prngaturan, dan tombol about.



Gambar 4. 1 Tampilan halaman awal aplikasi

2. Input data

From input data user diwajibkan bagi user yang belum terdaftar datanya. Input data terdiri dari nim, nama, prodi, dan kelas. Setelah input data selesai user menekan tombol ambil gambar. Setelah itu akan tampil jendela untuk merekam wajah user. Data user di ambil sebanyak 70 sampel wajah. Terdapat tombol train yang digunakan untuk melatih mesin agar wajah user dapat di kenali untuk mengambil absensi.



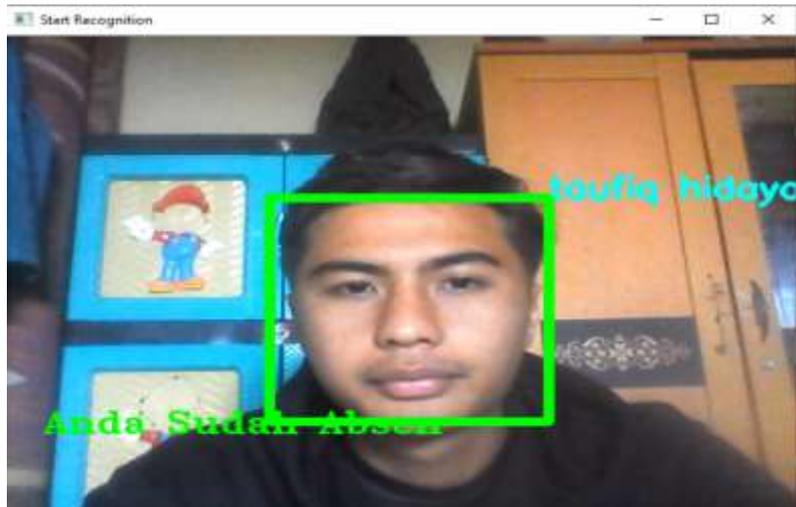
Gambar 4. 2 Halaman input data

Sebelum mengambil absensi admin diharuskan merubah waktu absensi sesuai dengan jadwal yang telah ditentukan. Setelah itu user dapat mengambil absensi dengan menekan tombol mulai absensi. Data absensi user dapat di lihat pada sebelah kiri aplikasi.



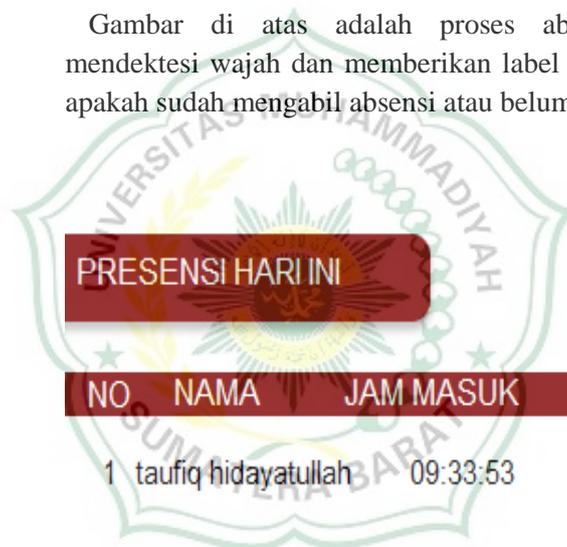
Gambar 4. 3 Rekam data wajah

Gambar di atas merupakan proses pengambilan data wajah. Hasil dari rekam data wajah digunakan untuk melatih sistem untuk bisa membedakan wajah satu dengan yang lainnya



Gambar 4. 4 Pengambilan absensi

Gambar di atas adalah proses absensi dimana aplikasi mendeteksi wajah dan memberikan label nama dan status absensi apakah sudah mengabil absensi atau belum.



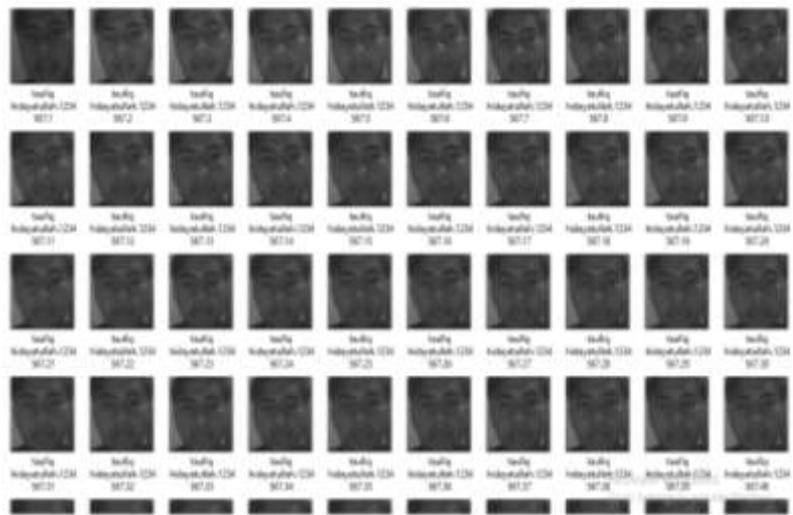
Gambar 4. 5 Hasil absensi

Gambar diatas merupakan hasil dari pengambilan absensi terdapat nama dan jam masuk.

4.1.2 Database

1. Data wajah

Data wajah yang sudah direkam sebanyak 70 sampel wajah disimpan di database



Gambar 4. 6 Data wajah

2. Data training

Hasil data wajah yang sudah di training disimpan didalam data base

Name	Date modified	Type	Size
trainer	21/01/2024 19:47	Yaml Source File	12.086 KB

Gambar 4. 7 Data training

4.1.3 kode program

1. Impor library

```
import tkinter as tk
from tkinter import *
import tkinter.font as tkFont
from tkinter import messagebox
import cv2
import csv
import os
import requests
import numpy as np
import hashlib
from PIL import Image, ImageTk
import pandas as pd
import datetime
import time
from pathlib import Path
from tkinter import ttk
import mysql.connector
```

Gambar 4. 8 input modul python

Gambar di atas adalah kode program yang diimport menggunakan *library Tkinter* dan beberapa *library* lainnya. Berikut adalah penjelasan tentang fungsi-fungsi yang digunakan dalam kode program tersebut:

1. `import tkinter as tk`
Mengimport *library Tkinter* dan memberikan alias `tk` untuk mempermudah pemanggilan fungsi.
2. `from tkinter import *`
Mengimport semua fungsi dari *library Tkinter* sehingga bisa diakses tanpa perlu menambahkan *prefix tk*.
3. `import tkinter.font as tkFont`
Mengimport *library tkFont* yang digunakan untuk mengatur font pada tampilan aplikasi.
4. `from tkinter import messagebox`
Kode program mengimport fungsi *messagebox* dari *library Tkinter* yang digunakan untuk menampilkan pesan pada jendela *pop-up*.
5. `import cv2`
Mengimport *library OpenCV* yang digunakan untuk memproses gambar.
6. `import csv`
Mengimport *library csv* yang digunakan untuk membaca dan menulis file `csv`.
7. `import os`
Mengimport *library os* yang digunakan untuk mengakses file dan direktori pada sistem operasi.
8. `import requests`
Mengimport *library requests* yang digunakan untuk melakukan *HTTP requests*.
9. `import numpy as np`
Mengimport *library NumPy* yang digunakan untuk melakukan operasi matematika pada array.
10. `import hashlib`
Mengimport *library hashlib* yang digunakan untuk menghasilkan *hash* dari *string*.
11. `from PIL import Image, ImageTk`

Mengimport *library* PIL (*Python Imaging Library*) yang digunakan untuk memproses gambar.

12. `import pandas as pd`

Mengimport *library pandas* yang digunakan untuk membaca dan menulis file csv dan excel.

13. `import datetime`

Mengimport *library datetime* yang digunakan untuk mengakses waktu dan tanggal.

14. `import time`

Mengimport *library time* yang digunakan untuk menunda eksekusi program dalam beberapa detik.

15. `from pathlib import Path`

Mengimport *library pathlib* yang digunakan untuk memanipulasi *path* dan *direktori* pada sistem operasi.

16. `from tkinter import tk`

Mengimport fungsi-fungsi yang diperlukan untuk mengatur tampilan *widget* pada aplikasi.

17. `import mysql.connector`

Mengimport *library mysql.connector* yang digunakan untuk mengakses dan memanipulasi database MySQL.

Dengan mengimport *library-library* tersebut, program dapat menggunakan berbagai fungsi dan metode yang disediakan oleh masing-masing *library*.

Adapun kode program digunakan untuk membuat jendela utama pada aplikasi. Berikut adalah penjelasan tentang fungsi-fungsi yang digunakan:

1. `window = tk.Tk()`

Membuat instance dari *class Tk* pada *library Tkinter* yang akan digunakan sebagai jendela utama pada aplikasi.

2. `icon_DA = PhotoImage(file='assets/img/logo-fk.png')`

Membuat *objek PhotoImage* pada *library Tkinter* yang digunakan untuk menampilkan logo pada jendela utama. `assets/img/logo-fk.png` adalah path dari file logo yang ingin ditampilkan.

3. `window.tk.call('wm', 'iconphoto', window._w, icon_DA)`

Mengatur logo yang telah dibuat sebelumnya untuk ditampilkan pada jendela utama.

4. width=1128

Mengatur lebar jendela utama menjadi 1128 piksel.

5. height=700

Mengatur tinggi jendela utama menjadi 700 piksel.

6. screenwidth = window.winfo_screenwidth()

Membaca lebar resolusi layar saat ini.

7. screenheight = window.winfo_screenheight()

Membaca tinggi resolusi layar saat ini.

8. alignstr = '%dx%d+%d+%d' % (width, height, (screenwidth - width) / 2, (screenheight - height) / 2)

Mengatur posisi jendela utama pada tengah layar dengan menggunakan string format.

9. window.geometry(alignstr)

Mengatur ukuran dan posisi jendela utama sesuai dengan string format yang telah dibuat sebelumnya.

10. window.resizable(width=False, height=False)

Membuat jendela utama tidak bisa diubah ukurannya.

11. cur_dir = Path.cwd()

Mengambil *current working directory* (cwd) dari sistem operasi pada saat program dijalankan. Digunakan untuk membaca file dari direktori tertentu

2. Koneksi ke database

```
global mydb, koneksi, username

dataDB = {}
def koneksiDB():
    data = []
    with open('{} /config/database/database.csv'.format(cur_dir)) as fileDb:
        reader = csv.DictReader(fileDb)
        for x in reader:
            data.append(x)
    for row in data:
        host = f"{row['HOST']}"
        user = f"{row['USER']}"
        pw = f"{row['PASSWORD']}"
        db = f"{row['DATABASE']}"
    return dataDB.update({'host':host,
                        'user':user,
                        'pw':pw,
                        'db':db})
```

Gambar 4.9 mengatur koneksi ke database MySQL

Gambar ini adalah kode program tersebut digunakan untuk membaca informasi koneksi ke database dari file csv dan melakukan koneksi ke database. Berikut adalah penjelasan tentang fungsi-fungsi yang digunakan:

1. `dataDB = {}`
Membuat *dictionary* kosong dengan nama `dataDB` yang digunakan untuk menyimpan informasi koneksi ke database.
2. `def koneksiDB():`
Membuat fungsi `koneksiDB` yang akan digunakan untuk membaca informasi koneksi ke database dari file csv dan melakukan koneksi ke database.
3. `data = []`
Membuat list kosong dengan nama `data` yang digunakan untuk menyimpan data dari file csv.
4. `with open('{}\config/database/database.csv'.format(cur_dir)) as fileDb:`
Membuka file csv yang berisi informasi koneksi ke database. `'{}\config/database/database.csv'.format(cur_dir)` adalah path dari file csv yang ingin dibaca.
5. `reader = csv.DictReader(fileDb)`
Membuat objek *DictReader* pada library csv untuk membaca file csv.
6. `for x in reader:`
Melakukan iterasi pada setiap baris pada file csv.
7. `data.append(x)`
Menambahkan setiap baris pada file csv ke dalam list `data`.
8. `for row in data:`
Melakukan iterasi pada setiap item dalam list `data`.
9. `host = f"{row['HOST']}"`
Menyimpan nilai `host` dari file csv ke dalam variable `host`.
10. `user = f"{row['USER']}"`
Menyimpan nilai `user` dari file csv ke dalam variable `user`.
11. `pw = f"{row['PASSWORD']}"`
Menyimpan nilai `password` dari file csv ke dalam variable `pw`.
12. `db = f"{row['DATABASE']}"`

Menyimpan nilai nama database dari file csv ke dalam variable db.

13. `return dataDB.update({'host':host,'user':user,'pw':pw,'db':db})`

Menyimpan nilai host, user, password, dan nama database ke dalam dictionary dataDB.

```
koneksiDB()
try:
    mydb = mysql.connector.connect( host=dataDB['host'],
                                   user=dataDB['user'],
                                   password=dataDB['pw'],
                                   database=dataDB['db'])

    if(dataDB["host"]==""):
        messagebox.showinfo('INFORMASI','Lakukan konfigurasi pada database terlebih dahulu')
        dbConnected = False
    else:
        koneksi = mydb.cursor()
        print("DATABASE CONNECTED")
        dbConnected = True
except Exception as tampil:
    messagebox.showinfo("DATABASE ERROR", tampil)
dbConnected = False
```

Gambar 4. 10 koneksi ke database

Gambar ini adalah kode Python untuk mencoba melakukan koneksi ke database MySQL menggunakan informasi yang disimpan dalam dictionary "dataDB". Kode ini menggunakan perintah "connect" dari modul `mysql.connector` untuk melakukan koneksi ke database. Jika koneksi berhasil, maka variabel "koneksi" akan ditempatkan pada objek cursor dan variabel "dbConnected" akan diatur sebagai True. Jika koneksi gagal, maka akan ditampilkan pesan "DATABASE ERROR" dan variabel "dbConnected" akan diatur sebagai False. Berikut adalah penjelasan kode program koneksi ke data base :

1. `global mydb, koneksi, username`: Mendefinisikan variabel global `mydb`, `koneksi`, dan `username`.
2. `dataDB = {}`: Membuat variabel dictionary kosong `dataDB`.
3. `def koneksiDB():` : Membuat sebuah fungsi bernama `koneksiDB`.
4. `data = []`: Membuat variabel list kosong `data`.
5. `with open('{ }/config/database/database.csv'.format(cur_dir)) as fileDb:` : membuka file `database.csv` yang berada dalam direktori `config/database` dan menyimpannya ke variabel `fileDb`.

6. `reader = csv.DictReader(fileDb)`: Membaca isi file `fileDb` menggunakan fungsi `csv.DictReader()` dan menyimpannya ke dalam variabel `reader`.
7. `for x in reader: data.append(x)`: *Looping* setiap baris data yang terdapat pada variabel `reader` dan menyimpannya ke dalam variabel `list data`.
8. `for row in data:` : *Looping* setiap baris data yang terdapat pada variabel `list data`.
9. `host = f"{row['HOST']}"`: Mengambil nilai kolom 'HOST' pada setiap baris data yang tersimpan di dalam variabel `row` dan menyimpannya ke dalam variabel `host`.
10. `user = f"{row['USER']}"`: Mengambil nilai kolom 'USER' pada setiap baris data yang tersimpan di dalam variabel `row` dan menyimpannya ke dalam variabel `user`.
11. `pw = f"{row['PASSWORD']}"`: Mengambil nilai kolom 'PASSWORD' pada setiap baris data yang tersimpan di dalam variabel `row` dan menyimpannya ke dalam variabel `pw`.
12. `db = f"{row['DATABASE']}"`: Mengambil nilai kolom 'DATABASE' pada setiap baris data yang tersimpan di dalam variabel `row` dan menyimpannya ke dalam variabel `db`.
13. `return dataDB.update({'host':host,'user':user,'pw':pw,'db':db})`: Mengupdate nilai dictionary `dataDB` dengan key-value baru yang terdiri dari 'host', 'user', 'pw', dan 'db' dengan value yang berasal dari variabel `host`, `user`, `pw`, dan `db`.
14. `koneksiDB()`: Memanggil fungsi `koneksiDB`.
15. `try:` : Menjalankan sebuah blok `try-except`.
16. `mydb=mysql.connector.connect(host=dataDB['host'],user=dataDB['user'], password=dataDB['pw'], database=dataDB['db'])`: Membuat koneksi ke database MySQL menggunakan `host`, `user`, `password`, dan `database` yang didapat dari variabel `dataDB`. Hasil koneksi disimpan ke dalam variabel `mydb`.
17. `if(dataDB['host']==")`:: Jika nilai pada key 'host' di dalam variabel `dataDB` kosong.

18. `messagebox.showinfo('INFORMASI','Lakukan konfigurasi pada database terlebih dahulu')`: Menampilkan pesan informasi pada popup window
 19. `dbConnected = False`: Menyimpan nilai False pada variabel `dbConnected`.
 20. `koneksi = mydb.cursor()`: Membuat koneksi ke database MySQL dan menyimpannya ke dalam variabel koneksi.
 21. `print('DATABASE CONNECTED')`: Mencetak pesan 'DATABASE
3. Pengambilan sampel data wajah

```

while (True):
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = detector.detectMultiScale(gray, 1.3, 5)
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
        # mengambil sample
        sampelnum = sampelnum + 1
        # simpan gambar
        cv2.imwrite(str(cur_dir + "/trainingImages/" + base + "." + nid + "." + str(sampelnum) + ".jpg"),
                    gray[y:y + h, x:x + w])
        cv2.imshow('Take Image', img)
        # wait for 100 milliseconds
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
        # break if the sample number is more than 100
        elif sampelnum > 70:
            break

```

Gambar 4. 11 mengambil data wajah

Gambar ini adalah bagian kode Python dari aplikasi yang menggunakan OpenCV untuk mengambil gambar dari kamera dan menyimpannya sebagai file gambar. Kode ini membaca gambar dari kamera dengan perintah "`cam.read()`" dan mengubah warna gambar menjadi grayscale dengan "`cv2.cvtColor`". Kemudian, fungsi "`detectMultiScale`" dari objek "`detector`" digunakan untuk mendeteksi wajah dalam gambar. Setiap wajah yang terdeteksi akan ditandai dengan kotak biru dan gambar wajah akan disimpan sebagai file gambar. Kode ini akan berulang selama user belum menekan tombol "q" atau jika jumlah sampel sudah lebih dari 70.

Berikut ini adalah penjelasan potongan kode Python untuk mengambil gambar dari kamera dan menyimpannya ke folder 'TrainingImage' dengan nama file yang diambil dari inputan nama dan nim. Selain itu, kode ini juga menambahkan entri baru ke file 'DataUser.csv' dan tabel 'tb_users' di database, yang berisi informasi pengguna seperti nim, nama, prodi, kelas, dan status.

Pertama, fungsi 'take_img' memanggil dua fungsi lain, yaitu 'dbPengaturan' dan 'database', tetapi tidak diberikan detail pada kode tersebut, jadi tidak dapat dijelaskan lebih lanjut.

Selanjutnya, fungsi 'take_img' mengambil input dari beberapa entri teks seperti 'en_nama', 'en_id', 'en_prodi', dan 'en_kelas'. Jika input untuk nama dan ID kosong, maka muncul pesan kesalahan dan program tidak akan berlanjut.

Setelah itu, fungsi membuka kamera menggunakan 'cv2.VideoCapture(0)' dan mendeteksi wajah menggunakan 'cv2.CascadeClassifier('haarcascade_frontalface_alt_tree.xml')'. Kemudian, gambar diambil saat wajah terdeteksi, dan gambar tersebut disimpan ke folder 'TrainingImage' dengan format nama file yang disesuaikan dengan nama, nim, dan nomor sampel gambar. Jumlah sampel gambar yang diambil dibatasi hingga 70 gambar. Setelah selesai, kamera dilepaskan dan jendela tampilan ditutup.

Terakhir, fungsi 'take_img' menambahkan informasi pengguna ke file 'DataUser.csv' dan tabel 'tb_users' di database. Setiap entri terdiri dari nim, nama, prodi, kelas, dan status 'registered'. Status ini menunjukkan bahwa pengguna telah terdaftar dan siap digunakan. Jika operasi berhasil, maka akan muncul pesan 'Berhasil' dengan informasi 'Input Data Berhasil!' dan catatan di file log. Namun, jika terjadi kesalahan saat menambahkan data, pesan kesalahan akan muncul dan catatan kesalahan akan ditulis di file log.

4. Training data

```
def training(self):
    database()
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    global detector
    detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
    try:
        global faces, Id
        faces, Id = getImagesAndLabels("TrainingImage")
        log("TrainingImage")
    except Exception:
        error("ID::(ImagesNotFound) :: suggested -- Take the training images first or create "TrainingImage" folder, place the Images")
        log("ERROR", "ID::(ImagesNotFound) :: suggested -- Take the training images first or create "TrainingImage" folder, place the Images")

    recognizer.train(faces, np.array(Id))
    try:
        recognizer.save("{}TrainingImageLabel/Trainer.yml".format(str(cur_dir)))
        log("Proses Train Berhasil")
        messagebox.showinfo('Berhasil', 'Proses Train Berhasil')
        time.sleep(2)
    except Exception:
        info("Peringatan", "Folder "TrainingImageLabel" tidak ditemukan")
        log("Peringatan", "Folder "TrainingImageLabel" tidak ditemukan")
    koneksi.execute("UPDATE tb_controls SET train = '0'")
    koneksi.execute("UPDATE tb_take_image SET status = '0'")
    with open('')
```

Gambar 4. 12 mengubah sampel wajah kebilangan binery

Gambar ini adalah fungsi training yang menggunakan library OpenCV. Fungsi ini melakukan koneksi ke database, menginisialisasi algoritma recognizer LBPHFaceRecognizer, dan mengambil semua gambar dan label wajah dari folder "TrainingImage". Setelah itu, algoritma recognizer dilatih menggunakan gambar dan label wajah yang diambil. Hasil pelatihan disimpan dalam file "Trainer.yml" di folder "TrainingImageLabel". Fungsi ini juga memperbarui status pada tabel database untuk menandakan bahwa proses training sudah selesai dilakukan.

Berikut ini adalah penjelasan kode program training data set :

1. database() adalah fungsi untuk membuat koneksi database dan mengambil nilai dari tabel yang digunakan di dalam fungsi ini.
2. recognizer adalah variabel untuk inisialisasi pengenalan wajah menggunakan metode Local Binary Patterns Histograms (LBPH).
3. detector adalah variabel untuk menginisialisasi klasifikasi pengenalan wajah menggunakan Cascade Classifier yang terdapat dalam file haarcascade_frontalface_alt_tree.xml.

4. faces dan Id adalah variabel yang digunakan untuk menyimpan wajah dan id pengguna yang terdapat pada direktori "TrainingImage".
 5. getImagesAndLabels("TrainingImage") adalah fungsi untuk mengambil gambar wajah dan id pengguna yang terdapat pada direktori "TrainingImage".
 6. recognizer.train(faces, np.array(Id)) adalah fungsi untuk melatih model pengenalan wajah menggunakan gambar wajah yang terdapat pada variabel faces dan id pengguna pada variabel Id.
 7. recognizer.save("{}TrainingImageLabel/Trainer.yml".format(str(cur_dir))) adalah fungsi untuk menyimpan model pengenalan wajah pada direktori "TrainingImageLabel" dengan nama file "Trainer.yml".
 8. koneksi.execute("UPDATE tb_controls SET train = '0'") dan koneksi.execute("UPDATE tb_take_image SET status = '0'") adalah fungsi untuk mengubah status pada tabel kontrol menjadi 0, yang menandakan bahwa model pengenalan wajah telah dilatih.
 9. mydb.commit() adalah fungsi untuk melakukan commit pada database.
5. Memulai absensi

```

if sub != 'log':
    log("Presensi dimulai")
    #load model
    recognizer = cv2.face.LBPHFaceRecognizer_create() # cv2.createLBPHFaceRecognizer()
    try:
        recognizer.read("TrainingImageLabel/Trainer.yml")
    except:
        info('Peringatan', 'Lakukan train terlebih dahulu')
        log("belum melakukan train image")

    harcasadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcasadePath)
    #membaca data user pada file DataUser.csv
    df = pd.read_csv("{}DataUser/DataUser.csv".format(str(cur_dir)))
    #object memulai camera video
    cam = cv2.VideoCapture(0)
    #font font...

```

Gambar 4. 13 memulai absensi

Kode di atas adalah skrip Python untuk pengenalan wajah. Skrip ini dimulai dengan melakukan log "Presensi dimulai" dan memuat model pengenalan wajah yang sudah dilatih sebelumnya (LBPHFaceRecognizer) dari "TrainingImageLabel/Trainer.yml". Jika model tidak ditemukan, skrip akan log pesan peringatan "belum melakukan train image". Skrip menggunakan klasifikasi Haar cascade untuk deteksi wajah dan membaca data pengguna dari file CSV di "DataUser/DataUser.csv". Kode ini mengatur objek pengambilan video menggunakan kamera default (indeks kamera 0).

Beriku adalah penjelasan isi kode program untuk memulai absensi menggunakan wajah

1. `def start(self,remote=False):`
Fungsi start akan memulai program absensi dan akan dijalankan setiap kali tombol "Start" pada program dijalankan.
`remote=False` adalah parameter yang diisi dengan nilai default False. Parameter ini akan digunakan jika program akan dijalankan dengan menggunakan server.
2. `dbPengaturan()`
Fungsi `dbPengaturan` akan melakukan koneksi ke database untuk mengambil pengaturan awal untuk program absensi.
3. `koneksi.execute("SELECT dari, sampai FROM tb_waktu_masuk")`
`hasil_wm = koneksi.fetchall()`
`for data in hasil_wm:`
`dari_wm = int(data[0])`
`sampai_wm = int(data[1])`
Baris kode tersebut akan mengambil data waktu presensi masuk dan keluar dari tabel `tb_waktu_masuk` dan `tb_waktu_keluar`.
Nilai dari `dari_wm` dan `sampai_wm` akan diambil dari hasil query yang sudah dijalankan.
4. `koneksi.execute("SELECT dari, sampai FROM tb_waktu_keluar")`
`hasil_wk = koneksi.fetchall()`
`for r in hasil_wk:`
`dari_wk = int(r[0])`
`sampai_wk = int(r[1])`

Baris kode tersebut akan mengambil data waktu presensi masuk dan keluar dari tabel tb_waktu_keluar.

Nilai dari dari_wk dan sampai_wk akan diambil dari hasil query yang sudah dijalankan.

```
5. future      = time.time() + 5
   now         = int(time.strftime("%H%M%S"))
   masuk       = int(dari_wm)
   keluar      = int(dari_wk)
```

Baris kode tersebut akan mengambil waktu sekarang dan waktu masuk serta waktu keluar dari variabel yang sudah diambil pada query sebelumnya.

```
6. global sub
   if now <= keluar and now >= masuk:
       sub = 'tb_presensi_masuk'
       jenis = 'clockin'
       if now >= dari_wm and now <= sampai_wm:
           ket = "hadir"
       elif now > sampai_wm:
           ket = "terlambat"
       else:
           ket = "error"
   elif now >= keluar and now >= masuk:
       sub = 'tb_presensi_keluar'
       jenis = 'clockout'
       if now >= dari_wk and now <= sampai_wk:
           ket = "hadir"
       elif now >= sampai_wk:
           ket = "terlambat"
       else:
           ket = "error"
   else:
       sub = 'log'
       jenis = 'log'
       ket = "waktu error"
   retak()
```

```
info("INFO","Absensi Belum Dimulai")
```

```
log("Absensi belum dimulai")
```

Baris kode tersebut akan menentukan apakah saat ini kategori presensi masuk atau presensi keluar dan menentukan terlambat atau tidak terlambat.

Variabel sub akan diisi dengan nama tabel yang akan diisi dengan data absensi.

Variabel jenis akan diisi dengan tipe presensi (masuk/keluar).

Variabel ket akan diisi dengan keterangan status presensi (had)

4.2 Pengujian sistem

4.2.1 Deteksi wajah

Tabel 4.1 pengujian absensi

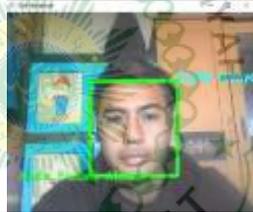
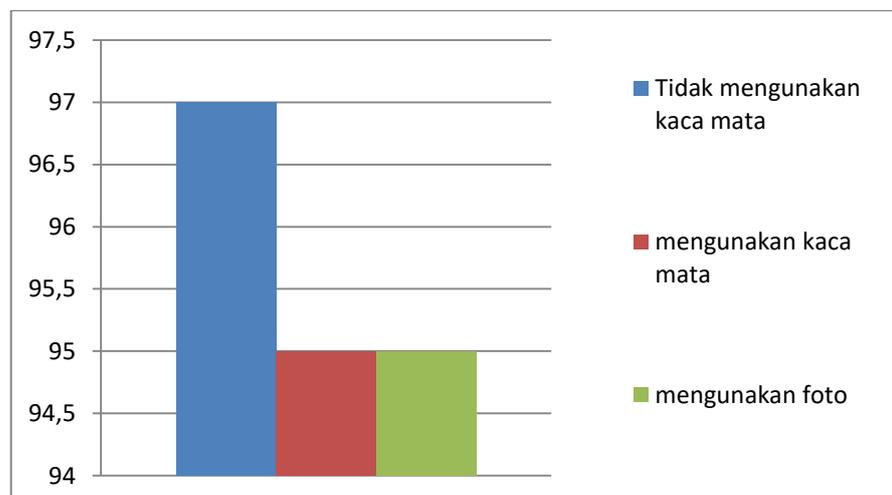
No	Nama	Absensi	Keterangan
1	Taufiq		Terdeteksi
2	azmi		Terdeteksi
3	afri		Terdeteksi

Table di atas merupakan hasil dari input data absensi mahasiswa dari table di atas terlihat aplikasi berjalan baik di lihat dari keterangan absensi.

Tabel 4. 2 hasil pengujian sistem

NO	Uji coba sistem	Output nilai akurasi	nilai Rata rata akurasi	Hasil deteksi	keterangan
1	Tidak menggunakan kaca mata	98.67	97.5		Terdeteksi
		97.99			
		97.29			
		96.26			
		97.33			
2	Menggunakan kaca mata	97.85	95.5		Terdeteksi
		98.88			
		96.21			
		93.33			
		91.23			
3	Menggunakan foto	91.86	95.4		Terdeteksi
		98.33			
		94.56			
		97.14			
		95.17			

Dari hasil pengujian sistem aplikasi absensi dengan menggunakan salah satu wajah mahasiswa dengan melakukan 3 pengujian di antaranya : tidak menggunakan kacamata, menggunakan kacamata, dan menggunakan foto terdapat hasil rata – rata dari nilai akurasi dengan hasil di atas 90 %. Namun pada saat pengujian menggunakan foto wajah wajah masiswa terdeteksi dan ini bisa menimbulkan kecurangan pada saat absensi.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari pengujian sistem absensi dengan pengenalan wajah maka dapat di tarik kesimpulan antara lain :

1. Akurasi pengenalan wajah sangat maksimal saat pengujian dikarenakan hasil akurasi yang cukup tinggi
2. Proses pengenalan wajah dengan metode LBPH dilakukan dengan cara mencocokkan inputan gambar dengan database dari gambar yang telah tersimpan pada saat pengambilan gambar melalui webcam.
3. Hasil akurasi sangat di pengaruhi pada posisi user melihat kamera terlihat pada hasil akurasi yang memiliki nilai output yang beragam
4. Sistem ini masih memiliki kelemahan karena tidak dapat membedakan foto dengan manusia
5. Pada saat menjalankan aplikasi proses absensi berjalan lambat pada saat uji coba di karenakan komputasi yang rendah.

5.2 Saran

Saran penulis bagi peneliti selanjutnya antara lain:

1. Dalam membangun sebuah aplikasi harap di perhatikan penulisan koding karena setiap kata dapat menyebabkan eror.
2. untuk peneliti selanjunya diharapkan bisa menambahkan algoritma yang dapat membedakan antara foto dengan manusia
3. untuk peneliti selanjunya diharapkan bisa menambahkan pengujian menggunakan data user kembar.
4. untuk peneliti selanjunya diharapkan bisa menggunakan computer /leptop yang memiliki komputasi tinggi agar proses menjalankan aplikasi tidak memakan waktu yang lama.

Daftar Pustaka

- [1] Fajri, R., Effendi, T. R., & Fadillah, N. (2020). Sistem Absensi Berbasis Pengenalan Wajah Secara Real Time Menggunakan Metode Fisherface. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, 4(2), 350-354.
- [2] Kumaran, I., Firmansyah, M. R., Fauziah, E., Hutahaeen, Y. B., Suryana, A., Sidik, A. D. W. M., ... & Kusumah, I. H. (2021). Pengenalan Wajah Menggunakan Pendekatan Berbasis Pengukuran dan Metode Segmentasi dalam Berbagai Posisi dan Pencahayaan. *FIDELITY: Jurnal Teknik Elektro*, 3(1), 5-8.]
- [3] Aulia, A. G. (2020). Sistem Informasi Absensi berbasis WEB di Politeknik Negeri Padang. *Jurnal Teknologi Informasi Indonesia (JTII)*, 5(1), 11-17.
- [4] aragih, R. R. (2016). Pemrograman dan bahasa Pemrograman. *STMIK-STIE Mikroskil*, 1-91.
- [5] Fajri, R., Effendi, T. R., & Fadillah, N. (2020). Sistem Absensi Berbasis Pengenalan Wajah Secara Real Time Menggunakan Metode Fisherface. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, 4(2), 350-354.
- [6] wibowo, A. W., Karima, A., Wiktasari, A. Y., & Fahriah, S. (2020). Pendeteksian dan Pengenalan Wajah Pada Foto Secara Real Time Dengan Haar Cascade dan Local Binary Pattern Histogram. *JTET (Jurnal Teknik Elektro Terapan)*, vol, 9(2020).
- [7] rfa, A., & Cokrowibowo, S. (2020). Deteksi Wajah dengan Metode Local Binary Pattern Histogram pada OpenCV menggunakan Pemrograman Python. *Journal of Computer and Information System (J-CIS)*, 3(2).