

**DETEKSI WAJAH PADA CITRA DIGITAL  
UNTUK SISTEM PRESENSI KELAS**



Disusun Oleh:

N a m a : Dwitiya Pradipta Alwani  
NIM : 16523188

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA**

**2023**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

DETEKSI WAJAH PADA CITRA DIGITAL UNTUK SISTEM  
PRESENSI KELAS

TUGAS AKHIR



N a m a : Dwitiya Pradipta Alwani  
NIM : 16523188

الجامعة الإسلامية  
Yogyakarta, 15 Juni 2023  
الاندونيسية

Pembimbing,



( Arrie Kurniawardhani, S.Si., M.Kom )

**HALAMAN PENGESAHAN DOSEN PENGUJI**

**DETEKSI WAJAH PADA CITRA DIGITAL  
UNTUK SISTEM PRESENSI KELAS**

**TUGAS AKHIR**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 26 Juli 2023

Tim Penguji

Arrie Kurniawardhani, S.Si., M.Kom.

**Anggota 1**

Kurniawan Dwi Irianto, S.T., M.Sc.

**Anggota 2**

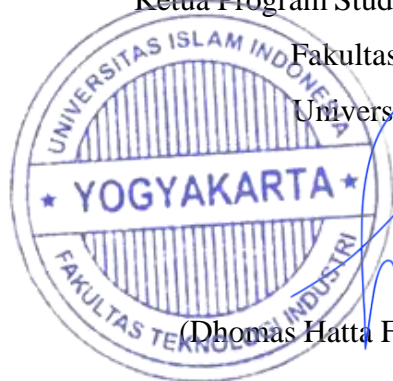
Rahadian Kurniawan, S.Kom., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

## HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Dwitiya Pradipta Alwani

NIM : 16523188

Tugas akhir dengan judul:

### **DETEKSI WAJAH PADA CITRA DIGITAL UNTUK SISTEM PRESENSI KELAS**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 15 Juni 2023



( Dwitiya Pradipta Alwani )

## HALAMAN PERSEMBAHAN

*Melalui halaman ini, penulis ingin mengucapkan bahwa tugas akhir ini dipersembahkan untuk:*

*Kedua orang tua tercinta, Ir. Henairi Harun dan Emy Asnidar, yang senantiasa memberikan dukungan dan doa yang tidak pernah putus untuk anak-anaknya agar selalu dipermudah dalam berbagai hal.*

*Ibu Arrie Kurniawardhani S.Si., M.Kom. yang banyak memberikan motivasi, bantuan serta bimbingan kepada penulis untuk menyelesaikan skripsi ini.*

*Saudara-saudaraku, Kak Dila, Izzan, Ilham, Mia, Cho, dan Choi yang memberikan motivasi kepada penulis.*

*Auwa, Davi, dan lainnya yang telah memilih untuk tidak menyerah pada keadaan.*

*Semoga Allah selalu meridhoi dalam setiap urusan serta membalas kebaikan kalian semua dengan berlipat ganda.*

## HALAMAN MOTO

لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا

(Q.S. Al Baqarah: 286)

*Courage is not the absence of despair;  
it's rather the capacity to move ahead in spite of despair.*

**Rollo May**

وَمَنْ يَتَّقِ اللَّهَ يَجْعَلْ لَهُ مَخْرَجًا وَيَرْزُقْهُ مِنْ حَيْثُ لَا يَحْتَسِبُ وَمَنْ يَتَوَكَّلْ عَلَى اللَّهِ فَهُوَ حَسْبُهُ إِنَّ اللَّهَ

بَلِغُ أَمْرِهِ قَدْ جَعَلَ اللَّهُ لِكُلِّ شَيْءٍ قَدْرًا

(Q.S At Talaq: 2-3)

*“Tidak ada kesuksesan tanpa kerja keras. Tidak ada keberhasilan tanpa kebersamaan. Tidak ada kemudahan tanpa doa.”*

**Ridwan Kamil**

## KATA PENGANTAR

Bismillahirrahmannirahim

Segala puji syukur kepada Allah SWT Tuhan Semesta Alam yang telah melimpahkan segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan Tugas Akhir ini yang berjudul “Deteksi Wajah Pada Citra Digital untuk Sistem Presensi Kelas”. Laporan ini dimaksudkan sebagai salah satu syarat untuk memperoleh gelar sarjana di jurusan Teknik Informatika Universitas Islam Indonesia. Selama proses pengembangan aplikasi ini, banyak kesulitan dan hambatan yang dialami penulis. Namun, berkat bantuan serta arahan dari pembimbing, penulis berhasil menyelesaikan Tugas Akhir ini. Oleh sebab itu, penulis ingin mengucapkan terima kasih kepada:

1. Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Sheila Nurul Huda, S.Kom., M.Cs. selaku Sekretaris Jurusan Informatika.
3. DThomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Program Studi Informatika Program Sarjana.
4. Dosen pembimbing, ibu Arrie Kurniawardhani, S.Si., M.Kom. yang telah meluangkan tenaga dan waktunya untuk membimbing serta memberi dukungan dalam menyelesaikan tugas akhir ini.
5. Bapak Fietyata Yudha, S. Kom., M.Kom. selaku dosen pembimbing akademik yang telah banyak membantu selama masa perkuliahan, serta memberikan berbagai tips dalam pengelolaan waktu.
6. Dosen-Dosen penguji yang telah memberikan arahan serta masukan, bapak Rahadian Kurniawan, S.Kom., M.Kom. dan bapak Kurniawan Dwi Irianto, S.T., M.Sc.
7. Kedua orang tua tercinta, Bapak Ir. Henairi Harun dan Ibu Emy Asnidar dengan segala pengorbanannya yang luar biasa disertai doa-doa yang tak pernah putus untuk penulis yang menjadi motivasi terbesar bagi penulis untuk dapat menyelesaikan tugas akhir.
8. Saudara-saudariku, Fadila Aulia Pritami S.Kom., Muhammad Izzan Harun, Muhammad Ilham Harun, Damia Azzyatul Salsa, Cho, dan Choi yang telah menjadi motivasi penulis dalam mencapai kesuksesan.
9. Sahabat-sahabatku, Erbial dan Noval, yang telah menasihati, mendukung, serta banyak membantu dalam berbagai hal.
10. Yayang dan Maulina, yang selalu berkenan menemani meski sudah lama tidak bertemu.

11. Kawan-kawan Hexadecima, Radivan, Galih Aji, dan Lantang, yang telah banyak membantu penulis dalam menyelesaikan Tugas Akhir ini.
12. Semua dosen dan staff di Jurusan Informatika UII yang telah berbagi pengetahuan, wawasan, serta pengalaman mereka selama masa pendidikan saya di UII.

Semoga semua dukungan, bantuan, pengetahuan, dan arahan yang telah diberikan kepada penulis mendapatkan berkah dari Allah SWT. Penulis sadar bahwa Laporan Tugas Akhir ini masih jauh dari kesempurnaan, dan dengan setulus hati, penulis siap menerima segala bentuk kritik dan saran yang membangun untuk meningkatkan kualitasnya. Harapannya, karya ini akan memberikan manfaat kepada semua pihak yang terlibat dan yang membutuhkan baik sekarang maupun di masa mendatang.

Yogyakarta, 15 Juni 2023



Dwitiya Pradipta Alwani



## SARI

Pengolahan citra telah menjadi bagian penting dari proses pendidikan, terutama dalam bidang administrasi presensi. Perlunya data terkait kehadiran siswa dalam setiap sesi pembelajaran menjadi sangat penting dalam kegiatan pembelajaran di sekolah atau perguruan tinggi. Namun saat ini masih banyak institusi atau sekolah yang masih menggunakan sistem presensi manual sehingga cukup menghabiskan waktu pembelajaran. Kekurangan lainnya sistem presensi manual juga yaitu masih menggunakan kertas yang mudah rusak, hingga adanya indikasi pemalsuan data kehadiran. Oleh karena itu, perlu adanya pengembangan sistem yang dapat secara otomatis dalam melakukan presensi yang tidak dapat dimanipulasi dan juga bisa melakukan penghitungan data kehadiran secara manual.

Aplikasi Deteksi Wajah Pada Citra Digital untuk Sistem Presensi Kelas dikembangkan sebagai salah satu alternatif sistem presensi. Perancangan sistem ini memanfaatkan penerapan teknologi *face detection* berbasis web agar dapat meningkatkan produktifitas kegiatan belajar dan mengajar, keamanan data, serta dapat mengurangi human error. Tekonologi multimedia dengan sistem pengolahan wajah (*face processing*) khususnya metode pendeteksian wajah (*face detection*) telah menjadi tren pada beberapa dekade terakhir. Teknologi ini banyak dikembangkan karena dianggap memiliki utilitas yang tinggi dalam berbagai bidang, terutama dalam visi komputer dan sistem kontrol akses secara otomatis (*automatic access control systems*).

Hasil yang didapat dari perancangan aplikasi ini adalah sistem memiliki 2 jenis pengguna yaitu *user* dan admin. Admin memiliki tugas dalam mengatur jadwal, ruangan, data guru, mata pelajaran, hingga data *user*. Sedangkan untuk *user* hanya memiliki fungsi untuk melakukan presensi dan *upload* foto. Sistem mampu menganalisa wajah menggunakan *face recognition* dengan baik. Penerapan algoritma deteksi wajah dapat diterapkan pada modul presensi pada PC menggunakan kamera *built-in* perangkat yang tersedia.

Kata kunci: *face recognition*, *face processing*, presensi kelas, absensi.

## GLOSARIUM

Back-End	Bagian di balik layar aplikasi atau sistem yang bertanggung jawab atas penyimpanan dan pemrosesan data, serta interaksi dengan database yang tidak terlihat oleh pengguna secara langsung.
Database	Basis data atau kumpulan informasi yang disimpan di dalam komputer secara sistematis.
Embed	Proses menggabungkan informasi ke dalam struktur yang lebih besar untuk mempermudah pemrosesan dan pemahaman data oleh komputer.
Face recognition	Teknologi Artificial Intelligence (AI) yang digunakan untuk mengekstraksi dan mengidentifikasi wajah dari gambar digital.
Front-End	Bagian dari sebuah aplikasi atau sistem yang berinteraksi langsung dengan pengguna untuk memfasilitasi input, output, dan interaksi visual dengan aplikasi tersebut.
Query	Permintaan yang diajukan ke database untuk mengambil informasi tertentu.
Software	Perangkat/piranti lunak.
Embedding	Representasi numerik dari objek atau data dalam bentuk vector.
Encoding	Proses mengubah data ke dalam format yang dapat dipahami atau digunakan oleh sistem komputer.

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING .....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI .....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR .....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR .....	vii
SARI .....	ix
GLOSARIUM .....	x
DAFTAR ISI .....	xi
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR .....	xiv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	3
1.6 Metodologi Penelitian .....	3
1.7 Sistematika Penulisan .....	4
BAB II TINJAUAN PUSTAKA .....	6
2.1 Sistem Presensi .....	6
2.2 Deteksi Wajah .....	6
2.3 Pengolahan Citra Digital .....	7
2.4 Machine Learning .....	8
2.5 Convolutional Neural Network (CNN) .....	9
2.6 Face-API.js .....	9
2.7 Penelitian Sebelumnya .....	10
BAB III METODOLOGI .....	13
3.1 Desain Arsitektur Sistem .....	13
3.2 Gambaran Umum Sistem .....	13
3.3 Diagram Use Case .....	15
3.4 Database .....	16
3.5 Pengumpulan Data Citra Siswa .....	18
3.6 Pengembangan Algoritma Sistem .....	18
BAB IV HASIL DAN PEMBAHASAN .....	20
4.1 Implementasi .....	20
4.1.1 Batasan Implementasi .....	20
4.1.2 Perangkat yang Digunakan .....	20
4.2 Implementasi Face-API.js .....	20
4.3 Kode dan Tampilan Sistem .....	26
4.3.1 Log In .....	26
4.3.2 Register Wajah .....	28
4.3.3 Halaman Kelola Jadwal .....	29
4.3.4 Halaman Kelola Data Guru .....	31
4.3.5 Halaman Kelola Kelas .....	33
4.3.6 Halaman Kelola Mapel .....	35

		xii
4.3.7	Halaman Kelola Ruangan .....	37
4.3.8	Halaman Tambah Presensi.....	38
4.4	Proses Presensi .....	41
4.5	Evaluasi Pengujian Sistem .....	43
	<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>45</b>
5.1	Kesimpulan.....	45
5.2	Saran.....	45
	<b>DAFTAR PUSTAKA.....</b>	<b>46</b>

**DAFTAR TABEL**

Tabel 2.1 Penelitian terkait deteksi wajah untuk sistem presensi dengan berbagai metode ...	10
Tabel 4.1 Test case sistem presensi.....	43

## DAFTAR GAMBAR

Gambar 2.1 Contoh hasil deteksi wajah menggunakan <i>bounding box</i> .....	7
Gambar 2.2 Arsitektur pada CNN.....	9
Gambar 3.1 Desain arsitektur sistem .....	13
Gambar 3.2 Gambaran umum sistem.....	14
Gambar 3.3 Diagram use case sistem.....	16
Gambar 3.4 Relasi tabel database sistem .....	17
Gambar 3.5 Ilustrasi 68 titik <i>landmark</i> pada citra wajah .....	19
Gambar 3.6 <i>FaceMatcher</i> pada Face-API.js untuk mencocokkan wajah .....	19
Gambar 4.1 Kode untuk melakukan deteksi wajah.....	21
Gambar 4.2 Kode untuk melakukan pengenalan wajah (1) .....	23
Gambar 4.3 Kode untuk melakukan pengenalan wajah (2) .....	24
Gambar 4.4 Kode untuk melakukan pengenalan wajah (3) .....	25
Gambar 4.5 Kode untuk halaman Log In.....	26
Gambar 4.6 Tampilan Halaman Log In .....	27
Gambar 4.7 Kode untuk Register Wajah .....	28
Gambar 4.8 Tampilan Register Wajah .....	29
Gambar 4.9 Kode fungsi Tambah Jadwal .....	30
Gambar 4.10 Kode fungsi Edit pada halaman Kelola Jadwal .....	30
Gambar 4.11 Kode fungsi Hapus pada halaman Kelola Jadwal .....	31
Gambar 4.12 Tampilan Kelola Jadwal .....	31
Gambar 4.13 Kode Tambah Guru .....	32
Gambar 4.14 Kode Edit Guru .....	32
Gambar 4.15 Kode Hapus Guru.....	33
Gambar 4.16 Tampilan Halaman Kelola Data Guru.....	33
Gambar 4.17 Kode Tambah Kelas .....	34
Gambar 4.18 Kode Edit Kelas.....	34
Gambar 4.19 Kode Hapus Kelas .....	34
Gambar 4.20 Tampilan Kelola Data Kelas .....	35
Gambar 4.21 Kode fungsi Tambah Mapel pada halaman Kelola Mata Pelajaran.....	35
Gambar 4.22 Kode fungsi Edit pada halaman Kelola Mata Pelajaran .....	36
Gambar 4.23 Kode fungsi Hapus pada halaman Kelola Mata Pelajaran.....	36
Gambar 4.24 Tampilan fungsi Edit pada halaman Kelola Mapel .....	37

Gambar 4.25 Kode fungsi Tambah Ruangan .....	37
Gambar 4.26 Kode fungsi Edit pada halaman Kelola Ruangan .....	37
Gambar 4.27 Kode fungsi Hapus pada halaman Kelola Ruangan .....	38
Gambar 4.28 Tampilan Kelola Data Ruangan .....	38
Gambar 4.29 Kode fungsi untuk melakukan presensi.....	39
Gambar 4.30 Kode ketika presensi berhasil dilakukan .....	40
Gambar 4.31 Tampilan sistem saat siswa melakukan presensi .....	40
Gambar 4.32 Tampilan saat siswa telah melakukan presensi .....	41
Gambar 4.33 Siswa melakukan presensi bukan dengan akun miliknya.....	42
Gambar 4.34 Presensi siswa berhasil ditambahkan ke database .....	42

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Perkembangan pesat teknologi multimedia beberapa dekade terakhir telah menarik banyak perhatian para peneliti, terutama yang berhubungan dengan sistem pengolahan wajah (*face processing*), terutama metode pendeteksian wajah (*face detection*). Tidak hanya insinyur saja, ahli saraf dan ahli psikofisik pun turut serta dalam meneliti dan mengembangkan teknologi ini karena dianggap memiliki utilitas yang tinggi dalam berbagai bidang, terutama dalam visi komputer dan sistem kontrol akses secara otomatis (*automatic access control systems*) (Chellappa et al., n.d.).

Penggunaan teknologi komputer dan pengolahan citra juga telah menjadi bagian penting dari proses pendidikan, terutama dalam bidang administrasi presensi. Kehadiran atau presensi siswa dalam setiap sesi pembelajaran merupakan aspek yang sangat penting dalam kegiatan pembelajaran di lingkungan sekolah. Pencatatan kehadiran siswa dalam kelas dapat membantu guru untuk memantau dan memastikan bahwa siswa hadir secara teratur dan tidak melewatkan materi yang diajarkan. Proses presensi kelas yang konsisten juga membantu mengajarkan siswa tentang kedisiplinan dan tanggung jawab. Informasi presensi kelas yang tercatat dapat digunakan untuk pengelolaan data akademik, administrasi, dan penilaian siswa. Selain itu, presensi kelas juga penting untuk alasan keamanan dan pengawasan. Dengan mencatat kehadiran siswa, pihak sekolah atau institusi dapat mengetahui siapa yang berada di dalam kelas pada saat tertentu.

Sistem presensi yang masih menggunakan cara manual dengan memakai kertas masih terbilang membuang waktu sehingga dapat mempengaruhi produktivitas belajar (Prathivi & Kurniawati, 2020). Selain itu, sistem presensi manual juga memiliki kekurangan seperti bahan kertas yang masih bisa rusak, hingga pemalsuan data kehadiran sehingga hal tersebut masih butuh pengembangan lebih lanjut demi menunjang efektivitas belajar di dalam kelas (Nugraha Ramdhon & Febriya, 2021). Oleh karena itu, diperlukan pengembangan sistem yang dapat secara otomatis dalam melakukan presensi yang tidak dapat dimanipulasi dan juga bisa melakukan penghitungan data kehadiran secara manual.



Dengan mempertimbangkan latar belakang tersebut, peneliti bertujuan untuk merancang sebuah sistem yang menggunakan teknologi deteksi wajah berbasis web untuk meningkatkan efisiensi kegiatan belajar mengajar, menjaga keamanan data, dan mengurangi *human-error*. Pada bagian *front-end*, sistem akan memungkinkan pengguna untuk mengambil gambar menggunakan kamera webcam yang terhubung dengan server untuk disimpan dalam database. Di sisi *back-end*, sistem akan melakukan pemantauan dan pengelolaan data presensi mahasiswa melalui analisis citra.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang sebelumnya, telah dirumuskan beberapa pertanyaan terkait dengan penelitian ini. Berikut ini pertanyaan penelitian yang telah disusun, antara lain :

- a. Bagaimana sistem dapat mendeteksi citra wajah manusia dalam sebuah gambar?
- b. Bagaimana cara mengintegrasikan sistem deteksi wajah dengan sistem manajemen presensi yang ada?

## 1.3 Batasan Masalah

Untuk memastikan fokus dan konsistensi dalam penyusunan tugas akhir ini, perlu ditetapkan batasan-batasan dalam mengatasi masalah yang akan dibahas. Berikut adalah beberapa batasan masalah yang akan diterapkan untuk menjaga kelangsungan penelitian tugas akhir ini.

- a. Penelitian ini membahas tentang perancangan sistem presensi dengan mengimplementasikan teknologi deteksi wajah.
- b. Pemrograman menggunakan bahasa pemrograman PHP.
- c. Pemrograman dilakukan dengan mengimplementasikan *library* face-API.js.
- d. Tingkat keakurasian sistem belum menjadi prioritas utama.
- e. *Dataset* menggunakan citra berukuran minimal 320x320 piksel.
- f. Penelitian ini dapat mengelola data presensi sesuai dengan jadwal kegiatan belajar.
- g. Penelitian ini berfokus pada kondisi kelas dan lingkungan pendidikan tertentu, dan mungkin diperlukan penyesuaian untuk implementasi di luar konteks tersebut.

#### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk merancang dan mengimplementasikan teknologi deteksi wajah yang dapat mengenali wajah-wajah siswa secara efisien, akurat, dan dapat diandalkan ke dalam sistem presensi.

#### 1.5 Manfaat Penelitian

Dalam pelaksanaan tugas akhir ini diharapkan dapat memberi manfaat antara lain:

a. Bagi Peneliti

Bermanfaat dalam pendalaman pembelajaran tentang pendeteksian wajah beserta implementasinya.

b. Bagi Akademisi

Bermanfaat sebagai acuan dan referensi untuk penelitian ataupun pengembangan sistem selanjutnya pada bidang pendeteksian wajah.

c. Bagi Masyarakat Umum

Bermanfaat untuk mengurangi kesalahan administrasi dalam pencatatan kehadiran siswa, serta memberikan pengetahuan pada bidang pendeteksian wajah. Penelitian ini juga dapat membantu pengembangan teknologi dalam bidang sistem manajemen kehadiran otomatis di setiap sekolah maupun tempat-tempat dengan kegiatan yang mewajibkan presensi.

#### 1.6 Metodologi Penelitian

Metode yang digunakan dalam penyusunan tugas akhir ini meliputi beberapa tahapan sebagai berikut:

a. Identifikasi Masalah

Identifikasi masalah merupakan tahap awal dalam penelitian ini. Tahap ini bertujuan mengidentifikasi keseluruhan masalah yang akan dipecahkan menggunakan sistem yang akan dibuat.

b. Analisis Kebutuhan

Pada tahap ini dilakukan analisis terkait segala kebutuhan pengembangan system seperti kebutuhan terkait *software* dan metode yang akan digunakan.

c. Perancangan

Perancangan adalah tahap untuk mendesain rancangan yang akan diaplikasikan pada sistem.

Pada tahap ini akan dirancang arsitektur sistem, diagram *use case*, gambaran umum sistem, dan penerapan metode yang digunakan dalam sistem.

d. Implementasi

Implementasi adalah proses transfer seluruh rancangan yang telah dibuat sebelumnya menjadi barisan kode atau perangkat lunak.

e. Pengujian

Pengujian yang dilakukan adalah pengujian performansi dari sistem yang sudah dikembangkan, apakah sudah sesuai dengan target atau tidak.

## 1.7 Sistematika Penulisan

Untuk menyajikan laporan penelitian ini secara terstruktur dan komprehensif, sistematika penulisan akan dibagi menjadi beberapa bab yang mengikuti urutan berikut. Dengan demikian, laporan penelitian ini akan disusun dengan baik dan menyajikan keseluruhan penelitian secara terperinci.

### **BAB I PENDAHULUAN**

Bagian ini membahas latar belakang mengenai permasalahan yang menjadi dasar penelitian, pentingnya deteksi wajah pada citra digital untuk sistem presensi, serta solusi yang diajukan untuk mengatasi masalah tersebut. Dengan mengacu pada latar belakang yang telah dijelaskan, selanjutnya akan dirumuskan permasalahan sebagai landasan untuk merencanakan penyelesaian, menetapkan batasan masalah, merumuskan tujuan penelitian, mengidentifikasi manfaat penelitian, menjelaskan metodologi penelitian yang digunakan, dan merangkum sistematika penulisan.

### **BAB II TINJAUAN PUSTAKA**

Bab ini menggambarkan konsep pengetahuan yang relevan dengan topik penelitian sebagai dasar untuk melaksanakan penelitian yang dilakukan, serta beberapa penelitian serupa yang telah dilakukan dengan berbagai metode. Teori-teori yang dijelaskan dalam bab ini mencakup sistem presensi, deteksi wajah, pengolahan citra digital, *machine learning*, *convolutional neural network* (CNN), dan Face-API.js.

### **BAB III METODOLOGI**

Bab ini berisi tentang tahapan dan metodologi penelitian yang menjadi pedoman untuk mencapai solusi terhadap permasalahan yang ada dalam penelitian ini. Bagian ini terdiri dari perancangan arsitektur sistem, diagram *use case*, diagram blok, dan pengembangan sistem.

### **BAB IV HASIL DAN PEMBAHASAN**

Bab ini membahas hasil dan pembahasan sistem yang telah dikembangkan, beserta implementasinya.

## **BAB V PENUTUP**

Bab ini memuat kesimpulan yang diperoleh dari sistem presensi kelas dengan teknologi *face detection* dan dapat digunakan sebagai acuan ataupun dapat dikembangkan lebih lanjut.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Sistem Presensi**

Sistem presensi merupakan sebuah sistem yang digunakan untuk memonitor dan mencatat kehadiran individu dalam suatu organisasi atau institusi. Sistem ini telah menjadi elemen kritis dalam berbagai sektor, termasuk bisnis, pendidikan, dan administrasi publik. Dalam konteks lembaga pendidikan, presensi mengacu pada kehadiran siswa secara teratur dan tepat waktu di kelas, perkuliahan, atau kegiatan akademik yang telah dijadwalkan. Hal ini menandakan partisipasi aktif siswa dalam proses pembelajaran dan kepatuhan mereka terhadap kebijakan kehadiran yang ditetapkan oleh institusi.

Catatan kehadiran dapat disimpan dan di analisis untuk memantau keterlibatan siswa, melacak perkembangan akademik (Hikmah et al., 2021), menghitung nilai, serta mengidentifikasi pola ketidakhadiran yang mungkin memerlukan intervensi (Widodo, 2016). Dengan menganalisis data kehadiran dalam hubungannya dengan kinerja akademik, institusi dapat mengevaluasi dampak kehadiran terhadap keberhasilan siswa. Selain itu, pencatatan kehadiran siswa secara konsisten juga membantu mengajarkan siswa tentang kedisiplinan dan tanggung jawab. Presensi kelas juga penting untuk alasan keamanan dan pengawasan. Dengan mencatat kehadiran siswa, pihak sekolah atau institusi dapat mengetahui siapa yang berada di dalam kelas pada saat tertentu.

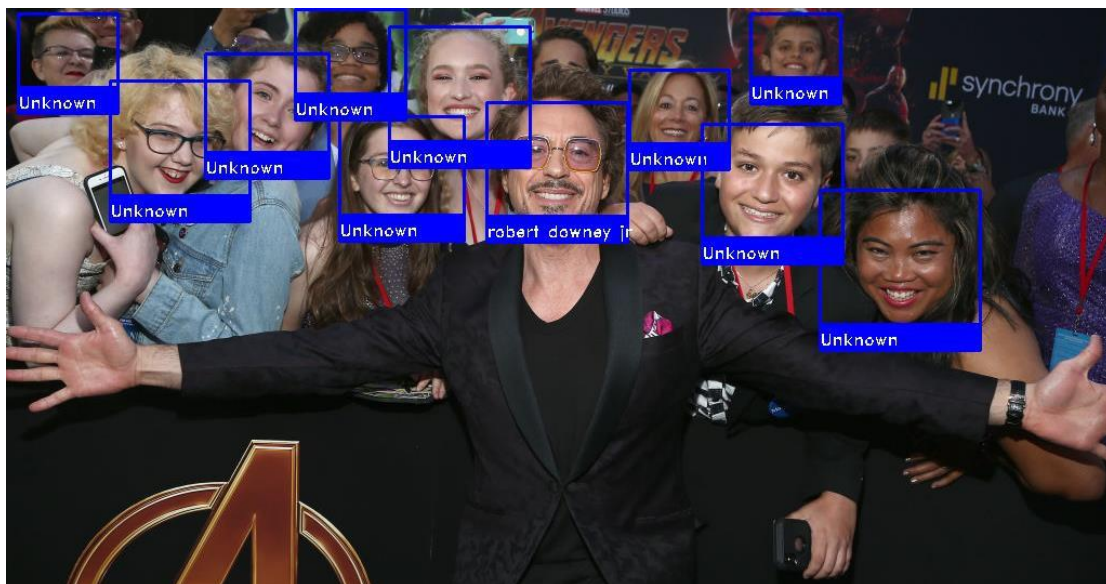
Tujuan utama sistem presensi adalah untuk memastikan keakuratan dan validitas data kehadiran, sekaligus memudahkan pengelolaan administrasi yang berkaitan dengan kehadiran siswa, karyawan atau anggota. Untuk mengidentifikasi seseorang secara unik, sistem presensi dapat menggunakan teknologi seperti sidik jari, pengenalan wajah, kartu akses, kode pin, atau cara lainnya.

#### **2.2 Deteksi Wajah**

Deteksi wajah merupakan sebuah algoritma untuk mengidentifikasi dan mengekstraksi ciri khusus pada area wajah pada sebuah citra atau video digital, dengan tujuan pengenalan atau pendeteksian wajah. Teknologi ini memungkinkan pengenalan wajah melalui ciri khusus atau karakteristik pada wajah seperti mata, hidung, mulut, dan kontur wajah tanpa memperhatikan elemen-elemen lain seperti tubuh manusia secara keseluruhan.

Deteksi wajah juga dapat diartikan sebagai identifikasi objek spesifik bagian-bagian tertentu pada wajah manusia, seperti mulut, hidung, pipi, mata, dagu, dan dahi, yang memiliki karakteristik yang unik. Ciri-ciri khusus pada wajah tersebut disebut dengan istilah fitur (Inrawansyah, 2017). Fitur-fitur wajah tersebut digunakan untuk mengukur jarak dan proporsi antara fitur satu dengan lainnya, sehingga dapat memberikan karakteristik unik yang membedakan satu wajah dari yang lain.

Deteksi wajah memegang peranan penting sebagai tahap awal atau pre-processing dalam berbagai jenis pemrosesan wajah. Beberapa bidang penelitian yang berhubungan dengan pemrosesan wajah meliputi pengenalan wajah, autentikasi wajah, lokalisasi wajah, pelacakan wajah, serta identifikasi dan interpretasi ekspresi wajah (Dwisnanto Putro et al., n.d.).



Gambar 2.1 Contoh hasil deteksi wajah menggunakan *bounding box*

### 2.3 Pengolahan Citra Digital

Citra digital adalah representasi dari citra visual dalam bentuk piksel-piksel diskret yang membentuk matriks atau array dua dimensi. Setiap piksel dalam citra digital memiliki nilai numerik yang mewakili intensitas cahaya atau warna pada titik tersebut. Dalam hal ini, citra digital merupakan representasi numerik dari informasi visual yang dapat diproses, disimpan, dan ditransfer menggunakan teknologi komputer (House, 2002). Pengolahan citra digital mengacu pada penerapan algoritma dan model matematika untuk memproses dan menganalisis citra digital. Tujuan utama dari pengolahan citra digital adalah untuk meningkatkan kualitas gambar, mengekstrak data terkait dari gambar tersebut, dan mengotomatiskan operasi yang bergantung pada analisis gambar.

Salah satu tahap penting dalam pengolahan citra digital adalah *pre-processing*, di mana gambar yang diperoleh dari sumber tertentu dipersiapkan sebelum tahap analisis. *Pre-processing* melibatkan operasi seperti pengurangan noise, peningkatan kecerahan dan kontras, serta penajaman dan pemadaman warna. Teknik-teknik ini dapat membantu memperbaiki kualitas citra dan memungkinkan analisis lebih lanjut.

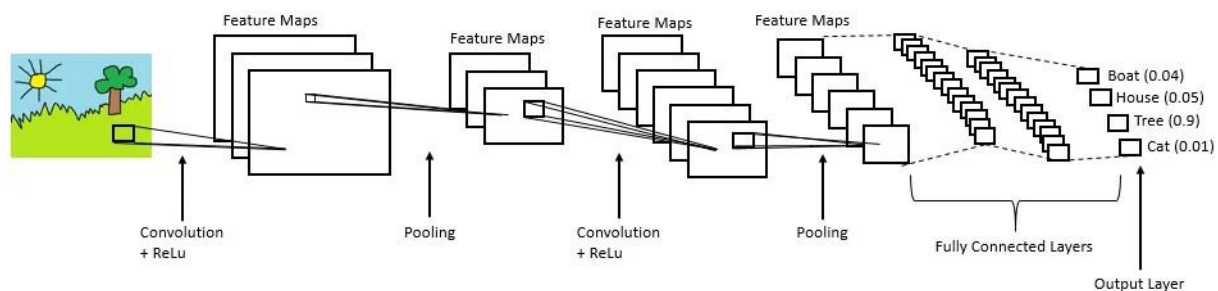
## 2.4 Machine Learning

Machine Learning atau pembelajaran mesin adalah cabang dari bidang kecerdasan buatan yang memungkinkan komputer untuk belajar dari data dan membuat prediksi atau keputusan secara mandiri sehingga sebuah mesin dapat meniru perilaku manusia yang cerdas (Brown, 2021). Konsep inti dari pembelajaran mesin berfokus pada pengembangan algoritme dan model yang mengenali pola, dan mengekstrak wawasan dari data untuk memfasilitasi pengambilan keputusan yang tepat (Sylaj, 2019). Dengan memanfaatkan pendekatan berbasis data, pembelajaran mesin memungkinkan komputer untuk secara otomatis belajar, beradaptasi, dan meningkatkan kinerja dalam berbagai tugas. Proses pembelajaran mesin umumnya melibatkan langkah-langkah berikut:

- **Persiapan Data:** Langkah ini melibatkan *pre-processing* data yang mencakup tugas-tugas seperti pengumpulan data, pembersihan data, normalisasi, dan rekayasa fitur untuk memastikan data dalam format yang sesuai untuk proses pembelajaran.
- **Pelatihan Model:** Algoritme ini belajar dari data yang telah disiapkan dengan menyesuaikan parameter internalnya berdasarkan pola dan hubungan yang ditemukannya. Algoritma atau model pembelajaran spesifik yang digunakan tergantung pada masalah yang sedang ditangani dan sifat data.
- **Evaluasi Model:** Setelah model dilatih, perlu dilakukan evaluasi untuk menilai kinerjanya. Evaluasi biasanya dilakukan dengan menggunakan dataset terpisah yang disebut set uji, yang tidak digunakan selama pelatihan.
- **Penerapan Model:** Jika model memiliki kinerja baik pada tahap evaluasi, model dapat digunakan untuk membuat prediksi atau mengambil tindakan pada data baru yang belum pernah ada sebelumnya. Model yang digunakan dapat diintegrasikan ke dalam aplikasi atau sistem untuk menyediakan fungsionalitas yang cerdas.

## 2.5 Convolutional Neural Network (CNN)

CNN adalah jenis jaringan saraf tiruan khusus yang telah digunakan secara luas untuk berbagai tugas visi komputer seperti klasifikasi gambar, deteksi objek, dan segmentasi gambar. CNN telah terbukti sangat efektif dalam memproses dan menganalisis data visual (Zufar & Setiono, 2016). Karakteristik utama CNN adalah kemampuannya untuk secara otomatis mempelajari representasi lapisan atau susunan data visual, sehingga CNN sangat cocok untuk mengolah data terstruktur seperti gambar dengan memanfaatkan konsep konvolusi, penyatuan, dan pelapisan hierarkis (Wang & Li, 2018). Komponen utama, proses, dan operasi yang terdapat pada CNN antara lain *Convolutional Layers* (Lapisan Konvolusi), *Activating Function* (Fungsi Aktivasi), *Pooling Layers* (Lapisan Penyatuan), *Fully Connected Layers* (Lapisan yang Terhubung Sepenuhnya), dan *Backpropagation* (Perambatan Balik), yang digambarkan secara sederhana pada Gambar 2.2.



Gambar 2.2 Arsitektur pada CNN

Sumber: (Raghav, 2018)

## 2.6 Face-API.js

Face-API.js adalah sebuah *library* JavaScript yang menggunakan TensorFlow.js sebagai *back-end* untuk menjalankan model-model deep learning yang diperlukan, sehingga memungkinkan untuk melakukan deteksi wajah, ekstraksi fitur wajah, dan pengenalan wajah yang akurat dan efisien.

Keunggulan dalam penggunaan Face-API.js adalah memungkinkan seluruh proses pengenalan wajah dapat di eksekusi secara langsung di browser web tanpa memerlukan ketergantungan eksternal, karena *library* ini telah menyediakan beberapa model *pre-trained* yang dioptimalkan untuk web yang dirancang secara khusus untuk dapat dijalankan secara efisien pada perangkat seluler dengan sumber daya terbatas. Selain itu, dengan memanfaatkan backend WebGL, Face-API.js dapat memanfaatkan kemampuan pemrosesan paralel GPU sehingga dapat mengeksekusi tugas analisis wajah yang lebih cepat dan efisien. Akselerasi



GPU ini meningkatkan kinerja keseluruhan Face-API.js dan memungkinkan analisis wajah secara real-time atau hampir real-time di dalam web browser, sehingga memberikan *user experience* yang lancar dan responsive (Mühler, 2018).

Beberapa fitur utama dari *library* Face-API.js meliputi:

- Deteksi Wajah: Face-API.js menggunakan algoritma deteksi wajah yang andal, seperti Tiny Face Detector, SSD Mobilenet V1, dan MTCNN. Algoritma-algoritma ini mampu mendeteksi wajah dengan akurasi tinggi, bahkan dalam berbagai kondisi pencahayaan dan sudut pandang.
- Ekstraksi Fitur Wajah: Face-API.js dapat mengekstraksi berbagai fitur penting dari wajah seperti mata, hidung, dan mulut. Ini memungkinkan untuk melakukan analisis lanjutan terhadap wajah yang terdeteksi.
- Pengenalan Wajah: Face-API.js menyediakan model pengenalan wajah yang kuat dan dilatih berdasarkan arsitektur yang mirip dengan FaceNet. Model ini menghasilkan vektor fitur wajah berdimensi 128D yang memungkinkan untuk mengidentifikasi wajah dengan akurasi yang tinggi.
- Deteksi Landmark Wajah: Face-API.js dapat mengidentifikasi *landmark* wajah yang penting, seperti posisi mata, hidung, dan mulut, yang dapat digunakan untuk penyalarsan wajah dan analisis ekspresi.
- Estimasi Usia dan Jenis Kelamin: *library* ini juga dapat memperkirakan usia dan jenis kelamin wajah yang terdeteksi, yang membuka kemungkinan untuk aplikasi khusus usia atau analisis demografis.
- Deteksi Emosi: Face-API.js dapat mengidentifikasi emosi dasar seperti senang, sedih, marah, dll., dalam ekspresi wajah secara *real-time*.

## 2.7 Penelitian Sebelumnya

Berikut adalah beberapa penelitian terdahulu terkait sistem presensi yang menerapkan teknologi deteksi wajah:

Tabel 2.1 Penelitian terkait deteksi wajah untuk sistem presensi dengan berbagai metode

No.	Judul	Author	Metode & Hasil
1.	Implementasi penggunaan OpenCV pada face recognition untuk sistem	(Santoso & Kristianto, 2020)	<i>Haar Cascade Classifier</i> dan <i>Local Binary Patterns Histograms</i> (LBPH).

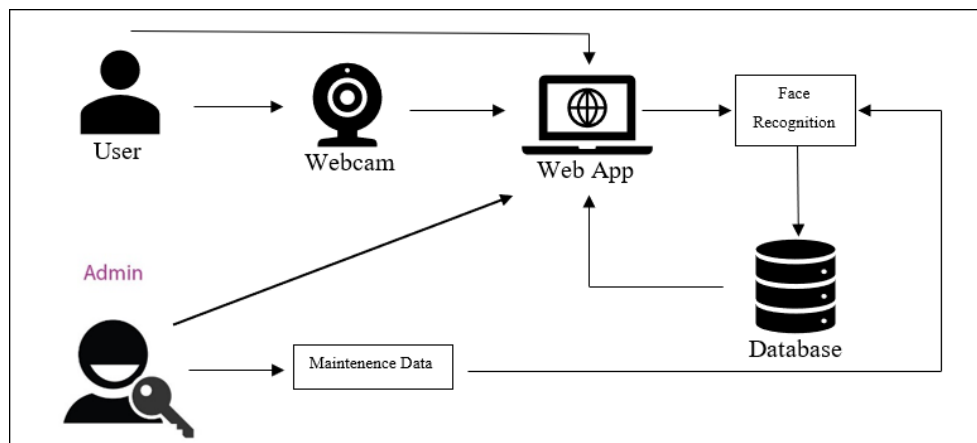
	presensi perkuliahan mahasiswa.		<ul style="list-style-type: none"> <li>- Jarak maksimal pendeteksian adalah 150cm.</li> <li>- Wajah yang tidak jelas tidak terdeteksi.</li> </ul>
2.	Sistem Monitoring Kehadiran Perkuliahan Berbasis Face Detection Menggunakan Algoritma Viola Jones	(Zulfachmi, Sudarma, & Jasa, 2019)	<i>Viola-Jones, Sobel, dan KNN</i> <ul style="list-style-type: none"> <li>- Jarak minimal pendeteksian adalah 20cm</li> <li>- Wajah harus tegak lurus.</li> <li>- Akurasi 94.79%.</li> </ul>
3.	Identifikasi Wajah Manusia untuk Sistem Monitoring Kehadiran Perkuliahan menggunakan Ekstraksi Fitur Principal Component Analysis (PCA)	(Suhery & Ruslianto, 2017)	<i>Haar-Cascade, dan Principal Component Analysis (PCA)</i> <ul style="list-style-type: none"> <li>- 1 dari 10 wajah gagal teridentifikasi.</li> <li>- Waktu rata-rata deteksi 0.63 detik.</li> <li>- Waktu rata-rata indentifikasi 0.89 detik.</li> </ul>
4.	Sistem Pencatatan Kehadiran Otomatis di Ruang Kelas Berbasis Pengenalan Wajah Menggunakan Metode Convolutional Neural Network (CNN)	(Endrianti et al., 2018)	<i>Convolutional Neural Network (CNN)</i> <ul style="list-style-type: none"> <li>- Akurasi 93.33%.</li> <li>- Hasil akurasi tergantung kondisi citra masukan.</li> </ul>
5.	Perancangan Sistem Presensi Pengenalan Wajah Pada Sistem Tata Kelola Perkuliahan Universitas Atma Jaya Makassar	(Syarif, 2021)	<i>Local Binary Patterns Histogram (LBPH)</i> <ul style="list-style-type: none"> <li>- Dapat diterapkan pada tablet PC</li> </ul>

6.	Student Attendance System using Face Recognition	(Sylaj, 2019)	<p><i>Haar-Cascade, Local Binary Patterns Histogram (LBPH)</i></p> <ul style="list-style-type: none"> <li>- Akurasi 78.40%</li> <li>- Lama proses tergantung jumlah input</li> <li>- Akurasi dapat ditingkatkan dengan menggunakan <i>adaptive threshold</i></li> </ul>
7.	Deep Learning untuk Deteksi Wajah Berhijab dengan CNN dan Tensorflow	(Anggraini, 2020)	<p><i>Convolutional Neural Network (CNN) dan Tensorflow</i></p> <ul style="list-style-type: none"> <li>- Akurasi 92%</li> <li>- Lama proses tergantung jumlah input</li> </ul>
8.	A Web-Based Mobile Attendance System with Facial Recognition Feature	(Nordin & Fauzi, 2020)	<p>JavaScript API, TensorFlow, Tiny Face Detector</p> <ul style="list-style-type: none"> <li>- Jenis evaluasi yang digunakan hanya dalam cangkupan kegunaan dan antarmuka</li> </ul>

## BAB III METODOLOGI

### 3.1 Desain Arsitektur Sistem

Desain arsitektur pada Gambar 3.1 berikut menggambarkan bagaimana sistem presensi pada penelitian ini dibuat.



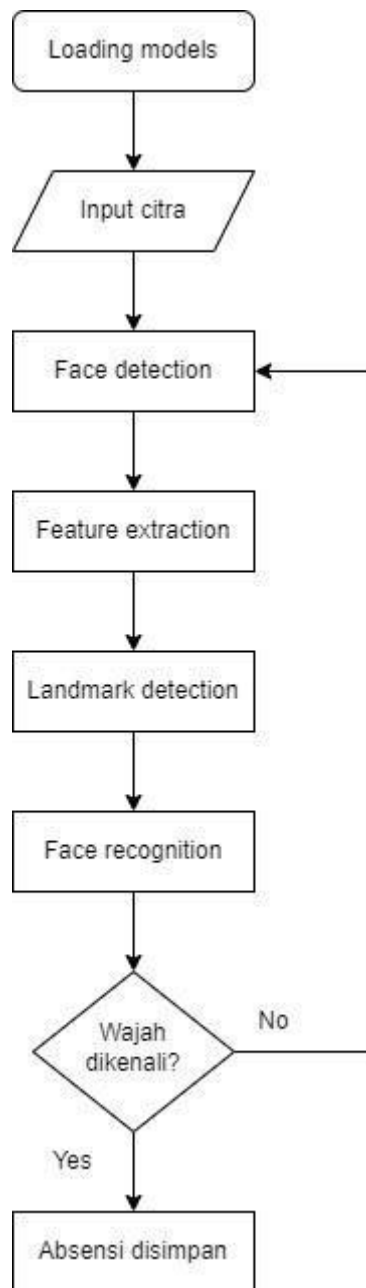
Gambar 3.1 Desain arsitektur sistem

Pada gambar di atas, user melakukan presensi melalui webcam dan data akan dikirimkan ke aplikasi web. Kemudian data masukan tersebut akan diproses oleh *face recognition* dengan cara mencocokkan data gambar yang ada di database. User sendiri dapat mengakses halaman aplikasi website untuk melihat data presensinya. Admin dapat melakukan *maintenance* data siswa agar nantinya sistem dapat mencocokkan citra dan wajah siswa tersebut, mengatur jadwal mata pelajaran setiap kelas dan mata pelajaran, serta juga dapat memberikan output laporan presensi kepada guru untuk dinilai.

### 3.2 Gambaran Umum Sistem

Deskripsi umum adalah penggambaran proses kerja yang umum dilakukan oleh perangkat lunak. Pada tahap ini, digunakan suatu diagram atau skema untuk menggambarkan bagaimana perangkat lunak bekerja mulai dari awal hingga selesai. Skema tersebut akan menunjukkan beberapa proses yang terjadi, termasuk memuat model, mengambil input citra, mendeteksi wajah, mengekstraksi fitur wajah, mendeteksi landmark wajah, pengenalan wajah,

pencocokkan wajah, dan lain sebagainya. Skema gambaran umum ini dapat dilihat dalam Gambar 3.2.



Gambar 3.2 Gambaran umum sistem

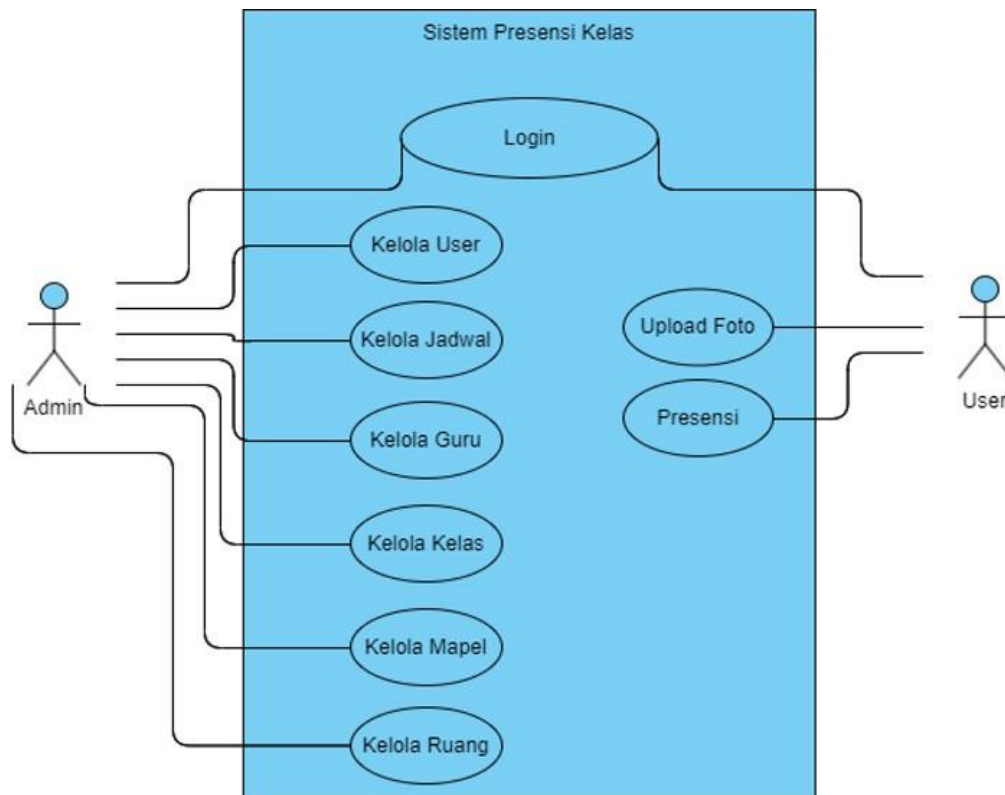
Berikut adalah ringkasan yang lebih rinci mengenai gambaran umum sistem pada penelitian ini:

1. Sistem dimulai dengan memuat model-model yang diperlukan untuk deteksi dan pengenalan wajah menggunakan Face-API.js.

2. *Input* untuk sistem ini diperoleh dari *webcam*, yang menyediakan sumber data yang kontinu.
3. *Frame* yang ditangkap diproses untuk mendeteksi wajah yang ada di dalam citra.
4. Jika terdeteksi wajah dalam *frame* yang ditangkap, sistem melanjutkan ke langkah berikutnya. Jika tidak ada wajah yang terdeteksi, sistem kembali ke tahap menangkap *frame* berikutnya dari *webcam*.
5. Untuk setiap wajah yang terdeteksi dalam *frame*:
  - Sistem mengekstraksi wilayah wajah, memisahkan wajah dari bagian lain gambar.
  - *Landmark* wajah, seperti mata, hidung, dan mulut, dideteksi untuk memberikan informasi lebih detail mengenai struktur dan fitur wajah.
  - *Landmark* wajah yang terdeteksi diproses sesuai kebutuhan sistem absensi.
  - Proses pengenalan wajah dilakukan dengan membandingkan wajah yang terdeteksi dengan *embedding* wajah yang telah diketahui atau tersimpan dalam sistem.
  - Jika terdapat kecocokan, menandakan bahwa wajah yang terdeteksi sesuai dengan individu yang dikenal, sistem melanjutkan untuk mencatat kehadiran dan menyimpan catatan absensi. Jika tidak ada kecocokan, sistem melanjutkan dengan wajah berikutnya dalam *frame*.
6. Sistem mengulangi langkah-langkah di atas untuk semua wajah yang terdeteksi dalam *frame* yang ditangkap.
7. Setelah semua wajah dalam *frame* diproses, catatan kehadiran dihasilkan.
8. Catatan kehadiran yang ditampilkan disimpan dalam *database*.
9. Proses dilanjutkan dengan menangkap *frame* berikutnya dari *webcam* atau dihentikan jika *webcam* tidak lagi tersedia atau sistem berhenti.

### 3.3 Diagram Use Case

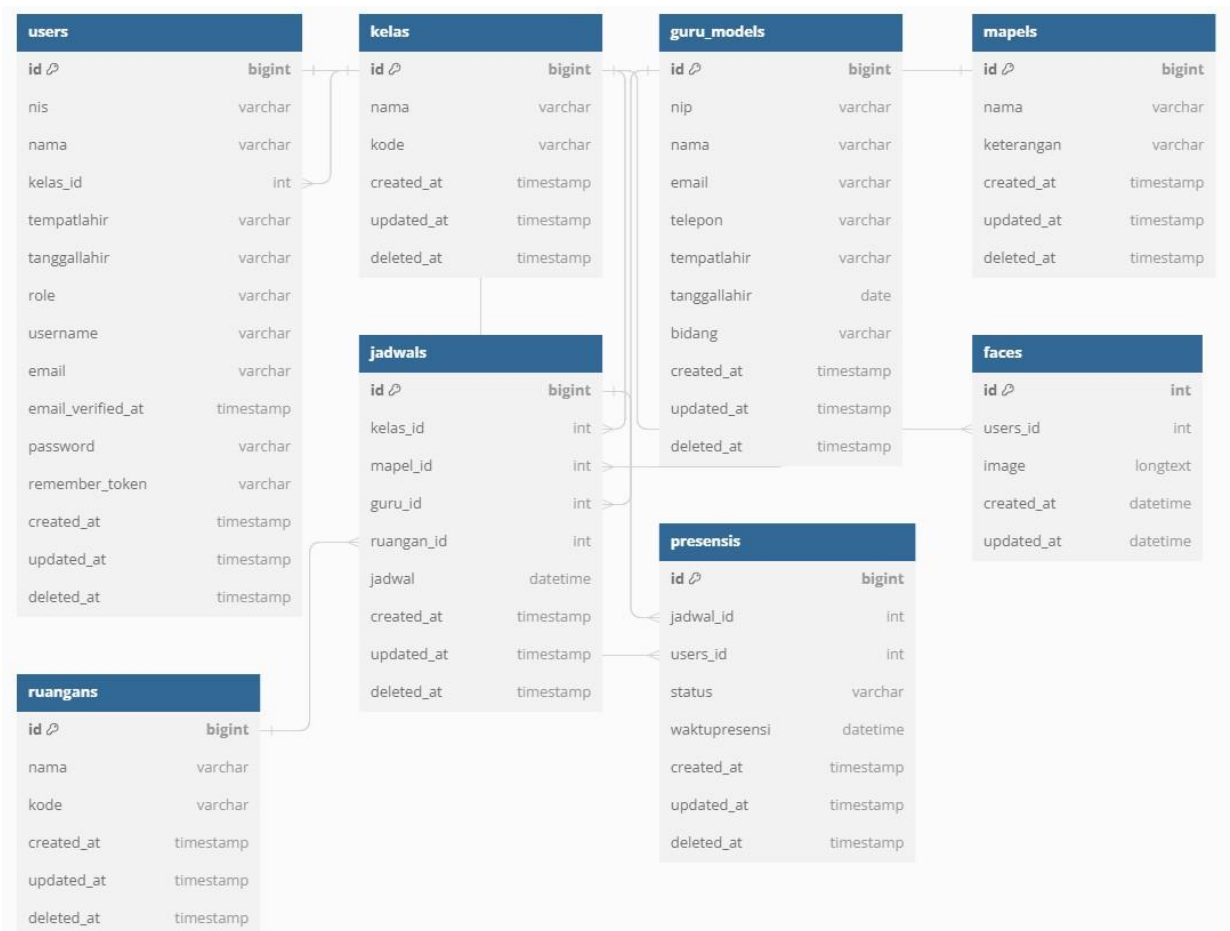
Diagram Use Case menggambarkan interaksi antara pengguna dan sistem dalam skenario penggunaan deteksi wajah pada citra digital untuk sistem presensi. Dengan menggunakan diagram Use Case ini, dapat dipahami secara visual bagaimana sistem berinteraksi dengan pengguna dalam konteks penggunaan deteksi wajah untuk sistem presensi. Aktor yang terlibat dalam sistem ini adalah User dan Admin. Pada Gambar 3.3, admin mendapatkan fungsi seperti Login, Kelola User, Kelola Jadwal, Kelola Guru, Kelola Kelas, Kelola Mapel, dan Kelola Ruangan. Sedangkan User mendapatkan fungsi berupa Login, Upload Foto, dan Presensi.



Gambar 3.3 Diagram use case sistem

### 3.4 Database

Database yang dibuat memiliki beberapa tabel yaitu: `faces`, `guru\_models`, `jadwals`, `kelas`, `mapels`, `presensis`, `ruangans`, dan `users`. Setiap tabel memiliki kolom-kolom yang berbeda yang menyimpan data terkait, dan beberapa tabel memiliki kolom yang menunjukkan hubungan atau relasi antar tabel.



Gambar 3.4 Relasi tabel database sistem

- Tabel `users` dan `kelas` memiliki relasi One-to-Many, di mana satu kelas dapat memiliki banyak siswa (users), tetapi satu siswa hanya terdaftar di satu kelas.
- Tabel `jadwals`, `mapels`, `guru\_models`, dan `ruangans` memiliki relasi One-to-Many dengan `kelas`.
- Tabel `jadwals`, `mapels`, dan `guru\_models` memiliki relasi Many-to-Many, karena satu jadwal dapat memiliki banyak mapel, dan satu mapel dapat diajar oleh banyak guru.
- Tabel `presensis`, `jadwals`, dan `users` memiliki relasi Many-to-Many, karena satu jadwal bisa memiliki banyak presensi dari banyak siswa, dan satu siswa dapat memiliki banyak presensi di berbagai jadwal.
- Tabel `faces` dan `users` memiliki relasi One-to-Many, di mana satu siswa dapat memiliki banyak data wajah (faces).



### 3.5 Pengumpulan Data Citra Siswa

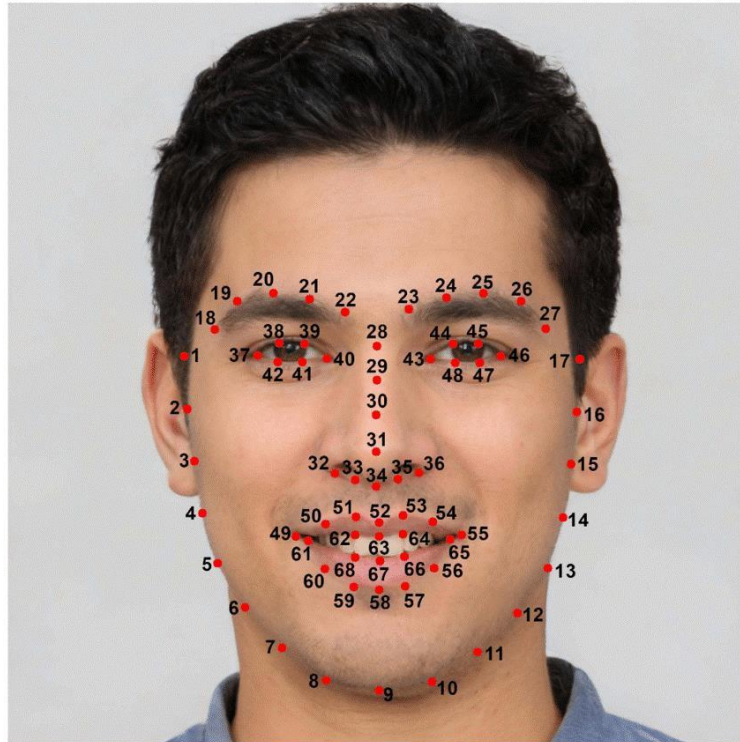
Dataset yang digunakan dalam penelitian ini adalah gambar-gambar wajah yang dapat dikumpulkan melalui proses menghimpun foto-foto siswa dengan mengajukan permintaan secara langsung ketika siswa mengakses sistem pertama kali.

### 3.6 Pengembangan Algoritma Sistem

Dalam penelitian ini, metode yang digunakan untuk pengembangan sistem presensi dengan cara mendeteksi wajah pada citra digital adalah melalui pemanfaatan Face-API.js. *Library* ini menyediakan kemampuan deteksi wajah yang kuat, memungkinkan sistem untuk menemukan citra wajah dalam gambar atau video. Proses pengembangan sistem dimulai dengan memuat model-model Face-API.js yang diperlukan. Model-model ini berperan penting dalam mendeteksi dan mengenali wajah secara akurat.

Untuk mempersiapkan dataset, sistem perlu mengumpulkan citra individu dan memberikan identitas unik pada masing-masing citra. Dataset ini diperoleh ketika siswa mengupload citra wajah mereka saat melakukan register wajah.

Dalam pendeteksian wajah, digunakan *pre-trained* model TinyFaceDetector. Model ini dipilih karena memiliki ukuran yang paling ringan (80KB) sehingga pendeteksian wajah dapat dilakukan dengan cepat. Hal ini karena TinyFaceDetector menggunakan *Depthwise Separable Convolution*, yaitu operasi konvolusi yang membagi konvolusi normal menjadi dua operasi independen: konvolusi menurut kedalaman dan menurut arah. Strategi ini mengurangi jumlah parameter dan komputasi yang harus dilakukan, sehingga menghasilkan model yang lebih efisien (Abdurrohman, Dini, & P. Muharram, 2018). Selanjutnya, dilakukan pemrosesan berupa *face alignment* dengan mengimplementasikan 68 titik *landmark* pada citra wajah yang telah terdeteksi.



Gambar 3.5 Ilustrasi 68 titik *landmark* pada citra wajah

Sumber: (Inkeaw, et al., 2022).

Pada wilayah wajah yang telah diproses sebelumnya perlu dilakukan ekstraksi fitur. Hasil dari ekstraksi fitur ini adalah daftar 128 representasi numerik yang mendeskripsikan fitur-fitur wajah (*face embeddings*). Pencocokan wajah dilakukan dengan membandingkan *face embedding* dari setiap citra wajah masukan yang telah diekstraksi dengan *face embedding* dari dataset yang telah diberi label. Tahap pencocokan wajah diimplementasikan dengan memanggil fungsi pada Gambar 3.6 berikut.

```
const faceMatcher = new faceapi.FaceMatcher(labeledFaceDescriptors,
maxDescriptorDistance)
```

Gambar 3.6 *FaceMatcher* pada Face-API.js untuk mencocokkan wajah

Sumber: (Mühler, 2018)

Pencocokan wajah pada Face-API.js menghitung skor kemiripan menggunakan *Euclidean Distance* dan menentukan kecocokan berdasarkan ambang batas atau *threshold* kemiripan yang ditetapkan. Semakin tinggi nilai *threshold* yang dipakai akan semakin akurat, namun semakin besar juga kemungkinan sistem mengalami *False-Positive*.

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Implementasi

#### 4.1.1 Batasan Implementasi

Batasan-batasan implementasi dalam program ini adalah sebagai berikut :

1. Aplikasi masih dibangun menggunakan *localhost*
2. Aplikasi dapat berjalan lancar pada RAM lebih dari 4 GB

#### 4.1.2 Perangkat yang Digunakan

Perangkat yang digunakan dalam pembuatan aplikasi ini adalah :

1. Perangkat Keras
  - **Notebook Acer Aspire A314-22**
  - **Processor AMD Athlon Silver 3050U**
  - **VGA AMD Radeon Graphics 2.30 GHz**
  - **RAM 4 GB**
  - **HDD 500 GB**
  - **Sistem operasi Widows 10**
2. Perangkat Lunak
  - **Visual Studio Code**
  - **XAMPP**
  - **Git Bash**
  - **PHP**

### 4.2 Implementasi Face-API.js

*Library* Face-API.js diimplementasikan dalam sistem presensi untuk mendeteksi dan mengenali wajah. Skrip-skrip pada fungsi ini diletakkan pada folder *public>js>script.js*. Berikut kode-kode yang digunakan untuk melakukan pengolahan citra wajah berupa *face detection* dan *face recognition* dengan memanfaatkan *library* Face-API.js.

```

// FD = Face Detection
const containerFD = document.getElementById('containerFD');
const videoFD = document.getElementById('videoFD');
const canvas = document.getElementById('canvas');
const buttonFD = document.getElementById('snapFD');
const submitFD = document.getElementById('submitFD');

if (videoFD) {
  Promise.all([
    faceapi.nets.tinyFaceDetector.loadFromUri('/js/models'),
    faceapi.nets.faceLandmark68Net.loadFromUri('/js/models'),
  ]).then(startWebcam)

  function startWebcam() {
    navigator.getUserMedia(
      { video: {} },
      stream => videoFD.srcObject = stream,
      err => console.error(err);
    );

    videoFD.addEventListener('play', () => {
      const canvas = faceapi.createCanvasFromMedia(videoFD)
      containerFD.appendChild(canvas)
      const displaySize = { width: videoFD.width, height: videoFD.height}
      faceapi.matchDimensions(canvas, displaySize)
      setInterval(async () => {
        const detections = await faceapi.detectAllFaces(videoFD, new
faceapi.TinyFaceDetectorOptions()).withFaceLandmarks()
        const resizedDetections = faceapi.resizeResults(detections,
displaySize)
        canvas.getContext('2d').clearRect(0, 0, canvas.width,
canvas.height)
        faceapi.draw.drawDetections(canvas, resizedDetections)
        faceapi.draw.drawFaceLandmarks(canvas, resizedDetections)
        resizedDetections.forEach( detection => {
          if (detection.detection.score >= 0.7) {
            buttonFD.classList.remove('d-none')
            Webcam.set({
              width : videoFD.width,
              height : videoFD.height,
              image_format : 'jpeg',
              jpeg_quality : 90,
            });
            Webcam.attach(videoFD);
          }
        })
      }, 100)
    })

    if (buttonFD) {
      buttonFD.addEventListener('click', () => {
        Webcam.snap(function (data_uri) {
          $(".imageFD").val(data_uri);
          document.getElementById('resultFD').innerHTML = '<img src =
'+data_uri+'>';
        });
        submitFD.classList.remove('d-none')
      })
    }
  }
}

```

Gambar 4.1 Kode untuk melakukan deteksi wajah

Gambar 4.1 menunjukkan kode sistem untuk melakukan deteksi wajah. Kode dimulai dengan memeriksa apakah elemen `videoFD` ada. Jika ada, maka kode melanjutkan dengan memuat beberapa model yang telah dilatih sebelumnya untuk deteksi wajah, penandaan wajah, dan pengenalan wajah menggunakan beberapa *library* dari Face-API.js. Setelah berhasil dimuat, fungsi `startWebcam` dipanggil untuk mengakses webcam pengguna.

Setelah menyiapkan webcam, sebuah *event listener* ditambahkan ke *event* `play` pada elemen `videoFD` yang dipicu ketika video dimulai. Di dalamnya, sebuah kanvas dibuat menggunakan metode `faceapi.createCanvasFromMedia`, dan kanvas ini ditambahkan ke elemen `containerFD`. Dimensi kanvas disesuaikan dengan ukuran video. Kemudian sebuah timer periodik diatur menggunakan `setInterval` untuk secara berulang melakukan deteksi wajah dan menampilkannya pada kanvas. Di dalam fungsi ini, deteksi wajah dilakukan menggunakan metode `faceapi.detectAllFaces` dengan `TinyFaceDetectorOptions`. Wajah yang terdeteksi kemudian diubah ukurannya agar sesuai dengan kanvas menggunakan `faceapi.resizeResults`. Kanvas dibersihkan, dan wajah yang terdeteksi serta penanda wajah digambar pada kanvas menggunakan fungsi-fungsi `faceapi.draw`.

Selain itu, `detection.score` berfungsi untuk memeriksa *confidence score* setiap wajah yang terdeteksi dengan menentukan apakah memenuhi *threshold* 0,7. Jika kondisi wajah memenuhi syarat, kode akan menampilkan tombol *snap* (`buttonFD`) dan mengkonfigurasi fitur tangkapan webcam untuk mengambil *snapshot* dari wajah yang terdeteksi.

```

// FR = Face Recognition
const containerFR = document.getElementById('containerFR');
const videoFR = document.getElementById('videoFR');
const buttonFR = document.getElementById('snapFR');
const submitFR = document.getElementById('submitFR');
const container = document.getElementById('container');

if (videoFR) {
  Promise.all([
    faceapi.nets.tinyFaceDetector.loadFromUri('/js/models'),
    faceapi.nets.faceRecognitionNet.loadFromUri('/js/models'),
    faceapi.nets.faceLandmark68Net.loadFromUri('/js/models'),
    faceapi.nets.ssdMobilenetv1.loadFromUri('/js/models'),
  ]).then(startWebcam)

  function startWebcam() {
    navigator.getUserMedia({
      video: true,
      audio: false,
    })
      .then ((stream => { videoFR.srcObject = stream;
      })
      .err => console.error(err)
    )
  }

  videoFR.addEventListener('play', () => {
    const canvas = faceapi.createCanvasFromMedia(videoFR)
    containerFR.appendChild(canvas)
    const displaySize = { width: videoFR.width, height: videoFR.height }
    faceapi.matchDimensions(canvas, displaySize)
    setInterval(async () => {
      const detections = await faceapi.detectAllFaces(videoFR, new
faceapi.TinyFaceDetectorOptions()).withFaceLandmarks()
      const resizedDetections = faceapi.resizeResults(detections,
displaySize)
      canvas.getContext('2d').clearRect(0, 0, canvas.width,
canvas.height)
      faceapi.draw.drawDetections(canvas, resizedDetections)
      faceapi.draw.drawFaceLandmarks(canvas, resizedDetections)
      resizedDetections.forEach( detection => {
        if (detection.detection.score >= 0.7) {
          buttonFR.classList.remove('d-none')
          Webcam.set({
            width : videoFR.width,
            height : videoFR.height,
            image_format : 'jpeg',
            jpeg_quality : 90,
          });
          Webcam.attach(videoFR);
        }
      })
    }, 100)
  })
}

```

Gambar 4.2 Kode untuk melakukan pengenalan wajah (1)

```

if (buttonFR) {
  buttonFR.addEventListener('click', () => {
    const imageUpload = document.getElementById('imageUpload');
    const resultFR = document.getElementById('resultFR');
    const contentType = 'image/png';

    const b64toBlob = (b64Data, contentType='', sliceSize=512) => {
      const byteCharacters = atob(b64Data);
      const byteArrays = [];

      for (let offset = 0; offset < byteCharacters.length; offset
+= sliceSize) {
        const slice = byteCharacters.slice(offset, offset +
sliceSize);

        const byteNumbers = new Array(slice.length);
        for (let i = 0; i < slice.length; i++) {
          byteNumbers[i] = slice.charCodeAt(i);
        }

        const byteArray = new Uint8Array(byteNumbers);
        byteArrays.push(byteArray);
      }

      const blob = new Blob(byteArrays, {type: contentType});
      return blob;
    }

    Webcam.snap(function (data_uri) {
      const data = data_uri;
      const dataImage = data.split(";base64,");
      const b64Data = dataImage[1];
      const blob = b64toBlob(b64Data, contentType);
      const blobUrl = URL.createObjectURL(blob);
      $.ajax({
        type: 'GET',
        url: '../allface',
        datatype: 'json',
        success: async function (response){
          if ($('#container div').length > 0) {
            $('#container').empty()
          }
          facedetection(blob, response)
        }
      });
    });
  });
}

```

Gambar 4.3 Kode untuk melakukan pengenalan wajah (2)

```

        async function facedetection(blob, response) {
            const rect = document.createElement('div')
            rect.style.position = 'relative'
            container.append(rect)
            const LabeledFaceDescriptors = await
loadLabeledImages(response[1])
            const faceMatcher = new
faceapi.FaceMatcher(LabeledFaceDescriptors, 0.6)
            let image
            let canvas
            if (image) image.remove()
            if (canvas) canvas.remove()
            image = await faceapi.bufferToImage(blob)
            rect.append(image)
            canvas = faceapi.createCanvasFromMedia(image)
            rect.append(canvas)
            const displaySize = { width: image.width, height:
image.height }
            faceapi.matchDimensions(canvas, displaySize)
            const detections = await
faceapi.detectAllFaces(image).withFaceLandmarks().withFaceDescriptors()
            const resizedDetections = faceapi.resizeResults(detections,
displaySize)
            const results = resizedDetections.map(d =>
faceMatcher.findBestMatch(d.descriptor))
            results.forEach((result, i) => {
                if (result.label == response[0]) {
                    const box = resizedDetections[i].detection.box
                    const drawBox = new faceapi.draw.DrawBox(box, {
label: result.toString() })
                    drawBox.draw(canvas)
                    $('#jadwalFR').removeClass('d-none')
                    $('#submitFR').removeClass('d-none')
                } else {
                    const box = resizedDetections[i].detection.box
                    const drawBox = new faceapi.draw.DrawBox(box, {
label: 'Anda Bukan ' + response[0] })
                    drawBox.draw(canvas)
                }
            })
        }

        function loadLabeledImages(response) {
            const labels = response;
            return Promise.all(
                labels.map(async label => {
                    const descriptions = []
                    for (let i = 1; i <= 5; i++) {
                        const img = await
faceapi.fetchImage(`../storage/face/${label}/${i}.png`)
                        const detections = await
faceapi.detectSingleFace(img).withFaceLandmarks().withFaceDescriptor()
                        descriptions.push(detections.descriptor)
                    }
                    return new faceapi.LabeledFaceDescriptors(label,
descriptions)
                })
            )
        }
    })
}
}

```

Gambar 4.4 Kode untuk melakukan pengenalan wajah (3)



Gambar 4.2 hingga Gambar 4.4 menunjukkan kode sistem dalam melakukan pengenalan wajah. Model-model yang digunakan dari kode ini sama seperti kode untuk pendeteksian wajah pada Gambar 4.1, namun memiliki penambahan model `faceRecognitionNet` yang bertugas untuk melakukan pengenalan wajah. Fungsi `loadLabeledImages` adalah mengambil deskriptor yang pada dasarnya merupakan representasi dari wajah yang dikenal, yang telah dilatih dan diberi label sebelumnya. Pada fungsi `faceMatcher`, pencocokan wajah dibuat dengan menggunakan deskriptor wajah berlabel dan dengan *threshold* 0,6. Nilai yang lebih rendah akan membuat pengenalan lebih ketat.

Pengenalan wajah dilakukan dengan mengekstrak landmark wajah dan deskriptor wajah untuk setiap wajah yang telah terdeteksi sebelumnya. Jika hasilnya menunjukkan wajah yang tidak dikenali, sistem akan menggambar *bounding box* dengan label yang menunjukkan bahwa wajah tersebut tidak dikenali.

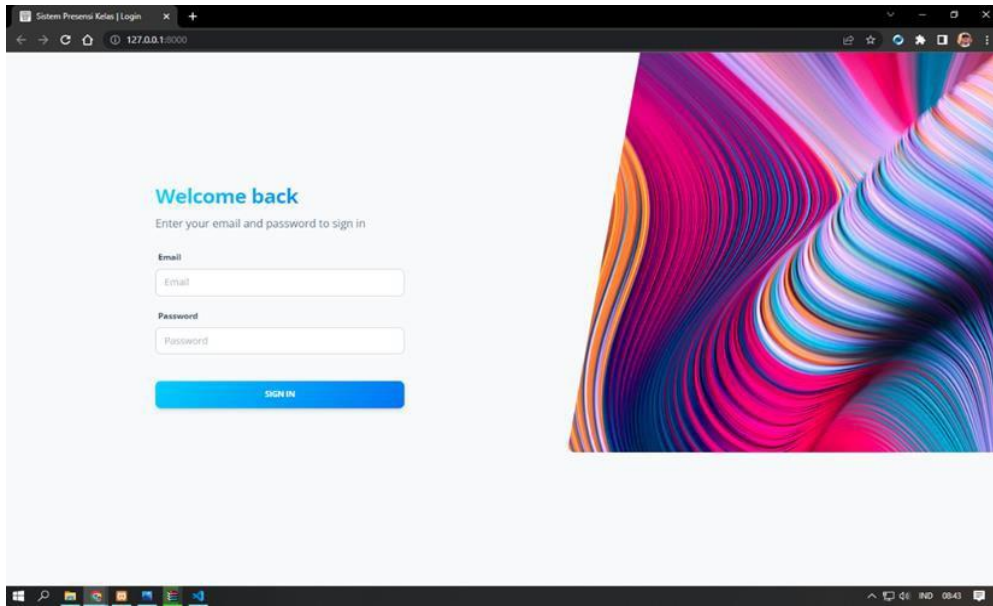
## 4.3 Kode dan Tampilan Sistem

### 4.3.1 Log In

```
public function index()
{
    return view('login.index', [
        'title' => 'Login',
    ]);
}
public function authenticate(Request $request)
{
    $credentials = $request->validate([
        'email' => ['required', 'email:dns'],
        'password' => ['required'],
    ]);
    if (Auth::attempt($credentials)) {
        $request->session()->regenerate();
        return redirect('/dashboard');
    }
    return back()->with('loginError', 'Login failed!');
}
public function logOut(Request $request)
{
    Auth::logout();
    $request->session()->invalidate();
    $request->session()->regenerateToken();
    return redirect('/');
}
```

Gambar 4.5 Kode untuk halaman Log In

Kode diatas berfungsi untuk menampilkan halaman login. Pada halaman ini, terdapat menu email dan password yang harus diisi oleh admin dan *user* untuk dapat mengakses aplikasi. Tampilan halaman *log in* dapat dilihat pada Gambar 4.6 berikut :



Gambar 4.6 Tampilan Halaman Log In

### 4.3.2 Register Wajah

```

public function store(Request $request)
{
    $img = $request->imageFD;
    $folderPath = "public/face/" . auth()->user()->username;

    Storage::makeDirectory($folderPath);

    $image_parts = explode(";base64,", $img);
    $image_type_aux = explode("image/", $image_parts[0]);
    $image_type = $image_type_aux[1];

    $faces = Faces::where('users_id', Auth()->id()->get());
    if (count($faces) > 0) {
        $name = count($faces[0]->image) + 1;
    } else {
        $name = 1;
    }

    $image_base64 = base64_decode($image_parts[1]);
    $fileName = $name . '.png';

    $file = $folderPath . '/' . $fileName;
    Storage::put($file, $image_base64);

    $image = [];
    if (count($faces) != 0) {
        $image = $faces[0]->image;
        $image[] = $fileName;
        $face = [
            'users_id' => Auth()->user()->id,
            'image' => $image,
        ];

        Faces::where('id', $faces[0]->id)->update($face);
    } else {
        $image[] = $fileName;
        $face = [
            'users_id' => Auth()->user()->id,
            'image' => $image,
        ];

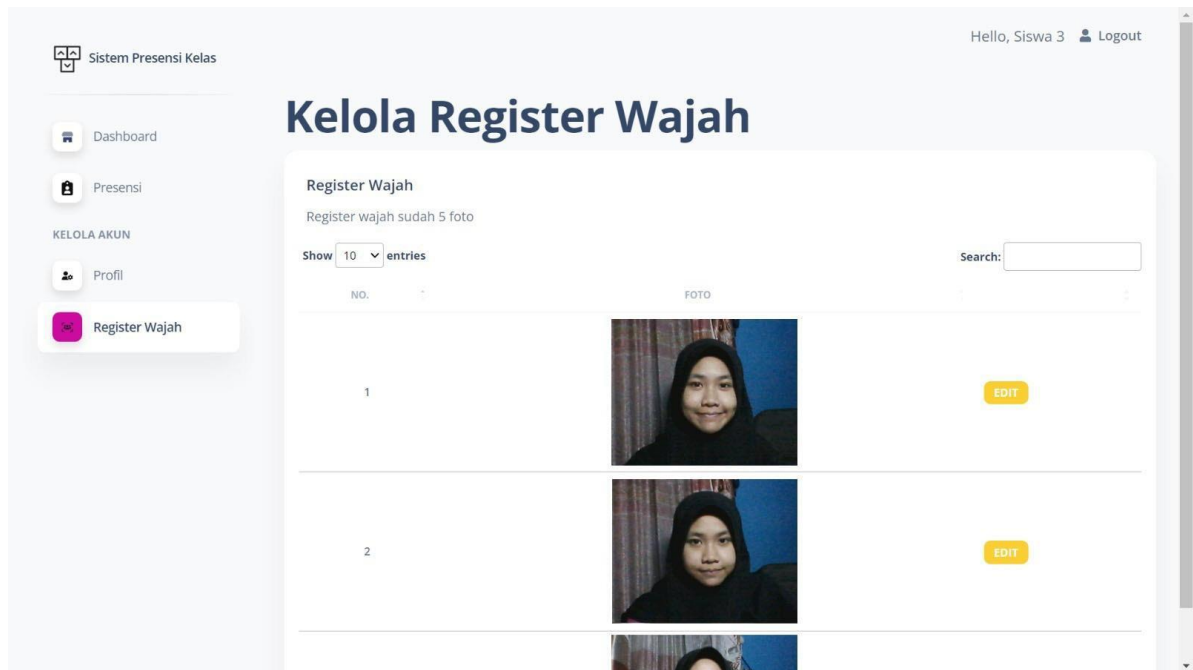
        Faces::create($face);
    }

    return redirect('/faces')->with('success', 'Register wajah berhasil
    disimpan!');
}

```

Gambar 4.7 Kode untuk Register Wajah

Gambar 4.7 diatas merupakan kode yang mengatur fungsi dari pengelolaan register wajah. Untuk tampilan halaman register wajah ditunjukkan pada Gambar 4.8 berikut:



Gambar 4.8 Tampilan Register Wajah

### 4.3.3 Halaman Kelola Jadwal

Pada halaman Kelola Jadwal, terdapat fungsi tambah jadwal, edit, dan hapus. Berikut adalah kode dari tiap-tiap fungsi tersebut.

```

public function create()
{
    $data = [
        'kelas' => Kelas::select('id', 'nama')->get(),
        'mapel' => Mapel::select('id', 'nama')->get(),
        'guru' => GuruModel::select('id', 'nama')->get(),
        'ruangan' => Ruangan::select('id', 'nama')->get(),
        'title' => 'Tambah Jadwal',
    ];
    return view('jadwal.create', $data);
}

public function store(Request $request)
{
    $data = [
        'kelas_id' => $request->kelas_id,
        'mapel_id' => $request->mapel_id,
        'guru_id' => $request->guru_id,
        'ruangan_id' => $request->ruangan_id,
        'jadwal' => $request->jadwal,
    ];

    Jadwal::create($data);
    return redirect('/jadwal')->with('success', 'Jadwal berhasil
ditambahkan!');
}

```

Gambar 4.9 Kode fungsi Tambah Jadwal

```

public function edit(Jadwal $jadwal)
{
    $data = [
        'kelas' => Kelas::select('id', 'nama')->get(),
        'mapel' => Mapel::select('id', 'nama')->get(),
        'guru' => GuruModel::select('id', 'nama')->get(),
        'ruangan' => Ruangan::select('id', 'nama')->get(),
        'title' => 'Edit Data Ruangan',
        'jadwal' => $jadwal,
    ];
    return view('jadwal.edit', $data);
}

public function update(Request $request, Jadwal $jadwal)
{
    $data = [
        'kelas_id' => $request->kelas_id,
        'mapel_id' => $request->mapel_id,
        'guru_id' => $request->guru_id,
        'ruangan_id' => $request->ruangan_id,
        'jadwal' => $request->jadwal,
    ];

    Jadwal::where('id', $jadwal->id)->update($data);
    return redirect('/jadwal')->with('success', 'Jadwal berhasil
diubah!');
}

```

Gambar 4.10 Kode fungsi Edit pada halaman Kelola Jadwal

```

public function destroy(Jadwal $jadwal)
{
    Jadwal::destroy($jadwal->id);
    return redirect('/jadwal')->with('success', 'Jadwal berhasil
dihapus!');
}

```

Gambar 4.11 Kode fungsi Hapus pada halaman Kelola Jadwal

Gambar 4.9 hingga Gambar 4.11 diatas merupakan kode untuk mengelola jadwal presensi. Berikut adalah tampilan halaman kelola jadwal.

The screenshot displays the 'Kelola Jadwal' page. At the top, there's a header with 'Sistem Presensi Kelas | Kelola Ja...' and a user profile 'Hello, Administrator' with a 'Logout' link. The main content area is titled 'Kelola Jadwal' and contains a 'Daftar jadwal' section with a '+ TAMBAH JADWAL' button. Below this is a table with 7 entries. The table has columns: JADWAL, KELAS, MATA PELAJARAN, GURU, TEMPAT, and action buttons (EDIT, HAPUS). The data rows are as follows:

JADWAL	KELAS	MATA PELAJARAN	GURU	TEMPAT	ACTION
2022-09-08 11:30:00	10 IPA 2	Biologi	Budi Gunawan	Ruang 102	[EDIT] [HAPUS]
2022-09-09 10:25:00	10 IPA 2	Fisika	Doni Firmansyah	Ruang 103	[EDIT] [HAPUS]
2022-09-11 00:00:00	10 IPA 1	Matematika	Budi Gunawan	Ruang 101	[EDIT] [HAPUS]
2022-09-11 17:00:00	11 IPA 1	Ekonomi	Budi Gunawan	Ruang 103	[EDIT] [HAPUS]
2022-09-11 17:50:00	10 IPA 2	Fisika	Budi Gunawan	Ruang 103	[EDIT] [HAPUS]
2022-09-12 09:00:00	11 IPA 1	Kimia	Doni Firmansyah	Ruang 103	[EDIT] [HAPUS]
2022-10-19 13:40:00	10 IPA 1	Matematika	Budi Gunawan	Ruang 101	[EDIT] [HAPUS]

At the bottom of the table, it says 'Showing 1 to 7 of 7 entries' and has 'Previous', '1', and 'Next' pagination controls.

Gambar 4.12 Tampilan Kelola Jadwal

#### 4.3.4 Halaman Kelola Data Guru

Pada halaman Kelola Data Guru terdapat beberapa fungsi yang dimplementasikan, yaitu fungsi Tambah Guru, Edit, dan Hapus. Berikut disertakan kode dari masing-masing fungsi.

```

public function create()
{
    $data = [
        'title' => 'Tambah Data Guru',
    ];
    return view('guru.create', $data);
}
public function store(Request $request)
{
    $data = [
        'nip' => $request->nip,
        'nama' => $request->nama,
        'email' => $request->email,
        'telepon' => $request->telepon,
        'tempatlahir' => $request->tempatlahir,
        'tanggallahir' => $request->tanggallahir,
        'bidang' => $request->bidang,
    ];

    GuruModel::create($data);
    return redirect('/guru')->with('success', 'Data guru berhasil
ditambahkan!');
}

```

Gambar 4.13 Kode Tambah Guru

```

public function edit(GuruModel $guru)
{
    $data = [
        'title' => 'Edit Data Guru',
        'guru' => $guru,
    ];
    return view('guru.edit', $data);
}
public function update(Request $request, GuruModel $guru)
{
    $data = [
        'nip' => $request->nip,
        'nama' => $request->nama,
        'email' => $request->email,
        'telepon' => $request->telepon,
        'tempatlahir' => $request->tempatlahir,
        'tanggallahir' => $request->tanggallahir,
        'bidang' => $request->bidang,
    ];

    GuruModel::where('id', $guru->id)->update($data);
    return redirect('/guru')->with('success', 'Data guru berhasil
diubah!');
}

```

Gambar 4.14 Kode Edit Guru

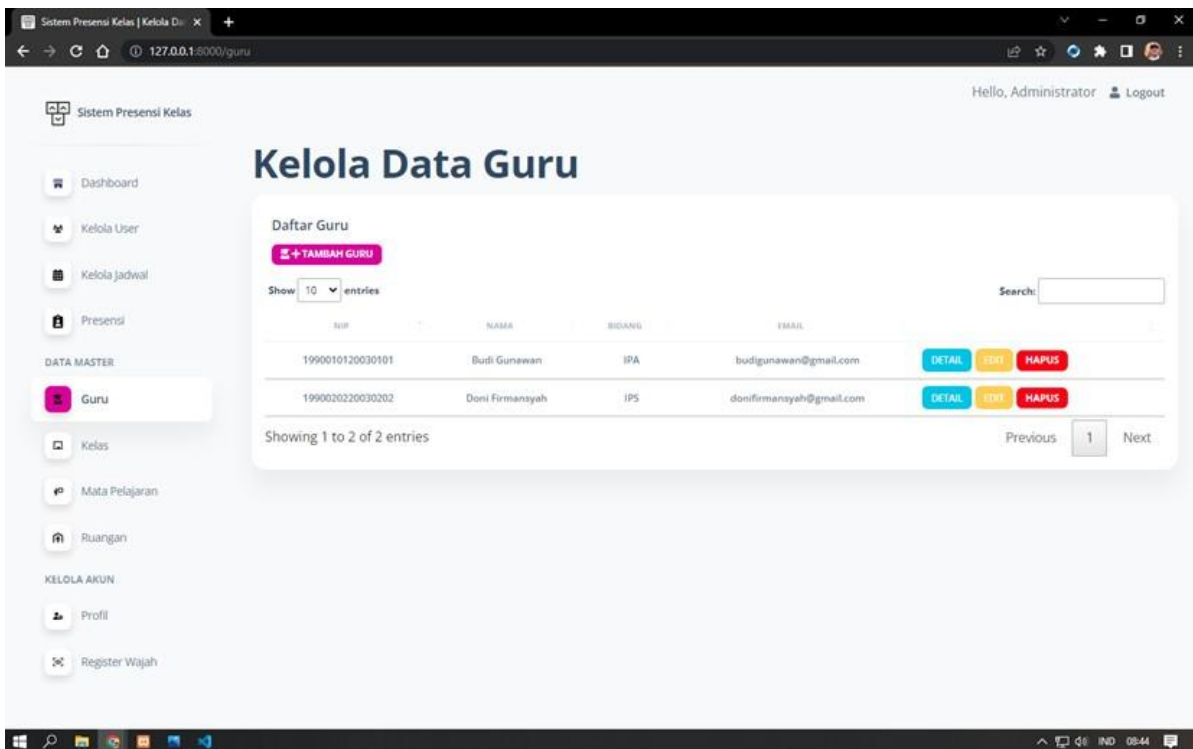
```

public function destroy(GuruModel $guru)
{
    GuruModel::destroy($guru->id);
    return redirect('/guru')->with('success', 'Data guru berhasil
dihapus!');
}

```

Gambar 4.15 Kode Hapus Guru

Gambar 4.13 hingga Gambar 4.15 di atas merupakan kode yang berfungsi untuk mengelola data guru. Halaman Kelola Data guru dapat dilihat pada Gambar 4.16.



Gambar 4.16 Tampilan Halaman Kelola Data Guru

### 4.3.5 Halaman Kelola Kelas

Pada halaman Kelola Data Kelas terdapat beberapa fungsi yang diimplementasikan, yaitu fungsi Tambah Kelas, Edit, dan Hapus. Berikut disertakan kode dari masing-masing fungsi.



```

public function create(){
    $data = [
        'title' => 'Tambah Data Kelas',
    ];
    return view('kelas.create', $data);
}
public function store(Request $request)    {
    $data = [
        'nama' => $request->nama,
        'kode' => $request->kode,
    ];
    Kelas::create($data);
    return redirect('/kelas')->with('success', 'Data kelas berhasil
ditambahkan!');
}

```

Gambar 4.17 Kode Tambah Kelas

```

public function edit(Kelas $kela){
    $data = [
        'title' => 'Edit Data Kelas',
        'kelas' => $kela,
    ];
    return view('kelas.edit', $data);
}
public function update(Request $request, Kelas $kela){
    $data = [
        'nama' => $request->nama,
        'kode' => $request->kode,
    ];
    Kelas::where('id', $kela->id)->update($data);
    return redirect('/kelas')->with('success', 'Data kelas berhasil
diubah!');
}

```

Gambar 4.18 Kode Edit Kelas

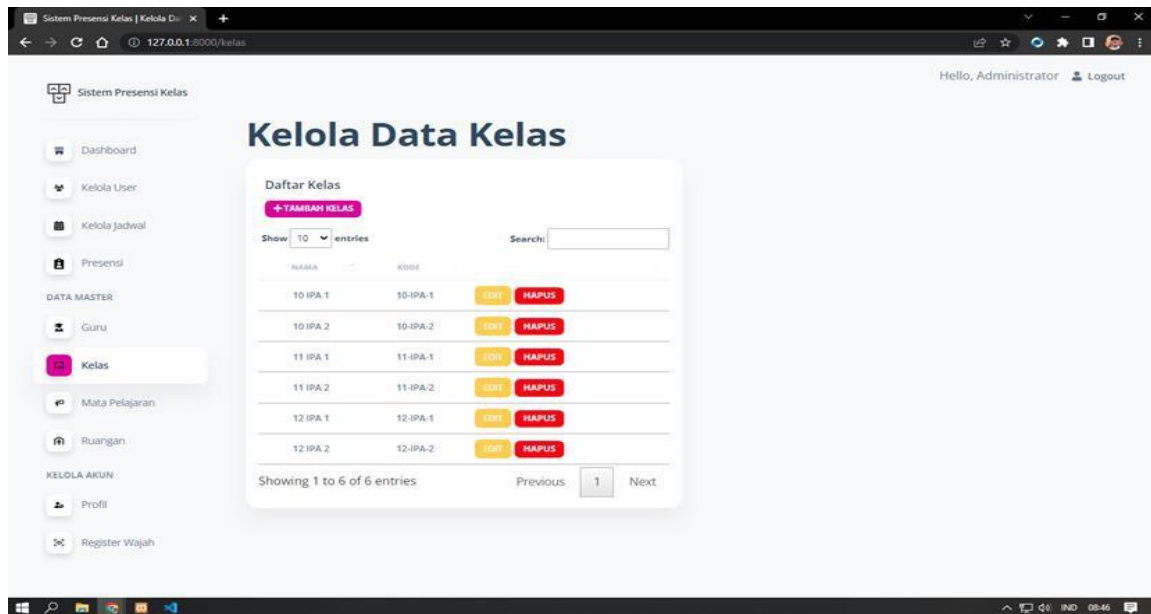
```

public function destroy(Kelas $kela){
    Kelas::destroy($kela->id);
    return redirect('/kelas')->with('success', 'Data kelas berhasil
dihapus!');
}

```

Gambar 4.19 Kode Hapus Kelas

Gambar 4.17 hingga Gambar 4.19 diatas merupakan kode yang berfungsi untuk mengelola data guru. Halaman Kelola Data guru dapat dilihat pada Gambar 4.20.



Gambar 4.20 Tampilan Kelola Data Kelas

#### 4.3.6 Halaman Kelola Mapel

Kode yang diimplementasikan untuk membuat halaman Kelola Mata Pelajaran adalah sebagai berikut.

```

public function create(){
    $data = [
        'title' => 'Tambah Data Mata Pelajaran',
    ];
    return view('mapel.create', $data);
}
public function store(Request $request)
{
    $data = [
        'nama' => $request->nama,
        'keterangan' => $request->keterangan,
    ];

    Mapel::create($data);
    return redirect('/mapel')->with('success', 'Data mata pelajaran
berhasil ditambahkan!');
}

```

Gambar 4.21 Kode fungsi Tambah Mapel pada halaman Kelola Mata Pelajaran

```

public function edit(Mapel $mapel){
    $data = [
        'title' => 'Edit Data Mata Pelajaran',
        'mapel' => $mapel,
    ];
    return view('mapel.edit', $data);
}
public function update(Request $request, Mapel $mapel)
{
    $data = [
        'nama' => $request->nama,
        'keterangan' => $request->keterangan,
    ];

    Mapel::where('id', $mapel->id)->update($data);
    return redirect('/mapel')->with('success', 'Data mata pelajaran
berhasil diubah!');
}

```

Gambar 4.22 Kode fungsi Edit pada halaman Kelola Mata Pelajaran

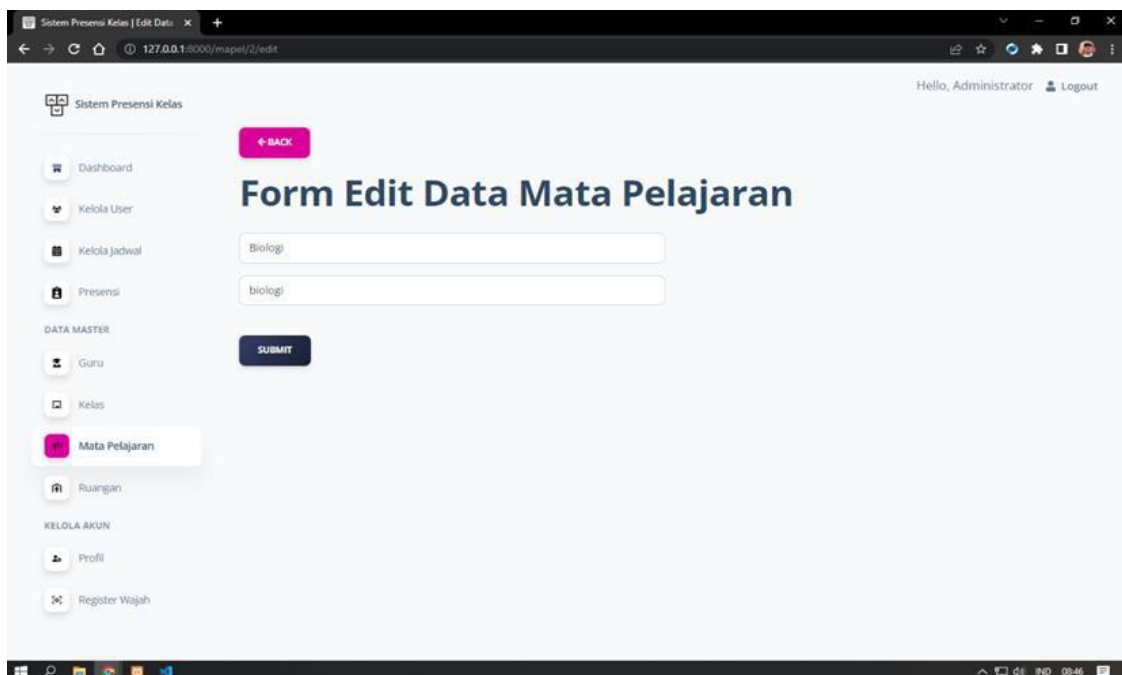
```

public function destroy(Mapel $mapel){
    Mapel::destroy($mapel->id);
    return redirect('/mapel')->with('success', 'Data mata pelajaran
berhasil dihapus!');
}

```

Gambar 4.23 Kode fungsi Hapus pada halaman Kelola Mata Pelajaran

Gambar 4.21 hingga Gambar 4.23 menunjukkan kode yang ada pada halaman Kelola Mata Pelajaran. Adapun Gambar 4.24 menunjukkan halaman Edit pada menu Mata Pelajaran.



Gambar 4.24 Tampilan fungsi Edit pada halaman Kelola Mapel

### 4.3.7 Halaman Kelola Ruangan

Pada halaman Kelola Data Ruangan terdapat beberapa fungsi yang dimplementasikan, yaitu fungsi Tambah Ruangan, Edit, dan Hapus. Berikut disertakan kode dari masing-masing fungsi.

```
public function index(){
    $data = [
        'ruangan' => Ruangan::select('id', 'nama', 'kode')->get(),
        'title' => 'Kelola Data Ruangan',
    ];
    return view('ruangan.index', $data);
}
public function create(){
    $data = [
        'title' => 'Tambah Data Ruangan',
    ];
    return view('ruangan.create', $data);
}
{
    $data = [
        'nama' => $request->nama,
        'kode' => $request->kode,
    ];
    Ruangan::create($data);
    return redirect('/ruangan')->with('success', 'Data ruangan berhasil
ditambahkan!');
}
```

Gambar 4.25 Kode fungsi Tambah Ruangan

```
public function edit(Ruangan $ruangan){
    $data = [
        'title' => 'Edit Data Ruangan',
        'ruangan' => $ruangan,
    ];
    return view('ruangan.edit', $data);
}
public function update(Request $request, Ruangan $ruangan){
    $data = [
        'nama' => $request->nama,
        'kode' => $request->kode,
    ];
    Ruangan::where('id', $ruangan->id)->update($data);
    return redirect('/ruangan')->with('success', 'Data ruangan berhasil
diubah!');
}
```

Gambar 4.26 Kode fungsi Edit pada halaman Kelola Ruangan

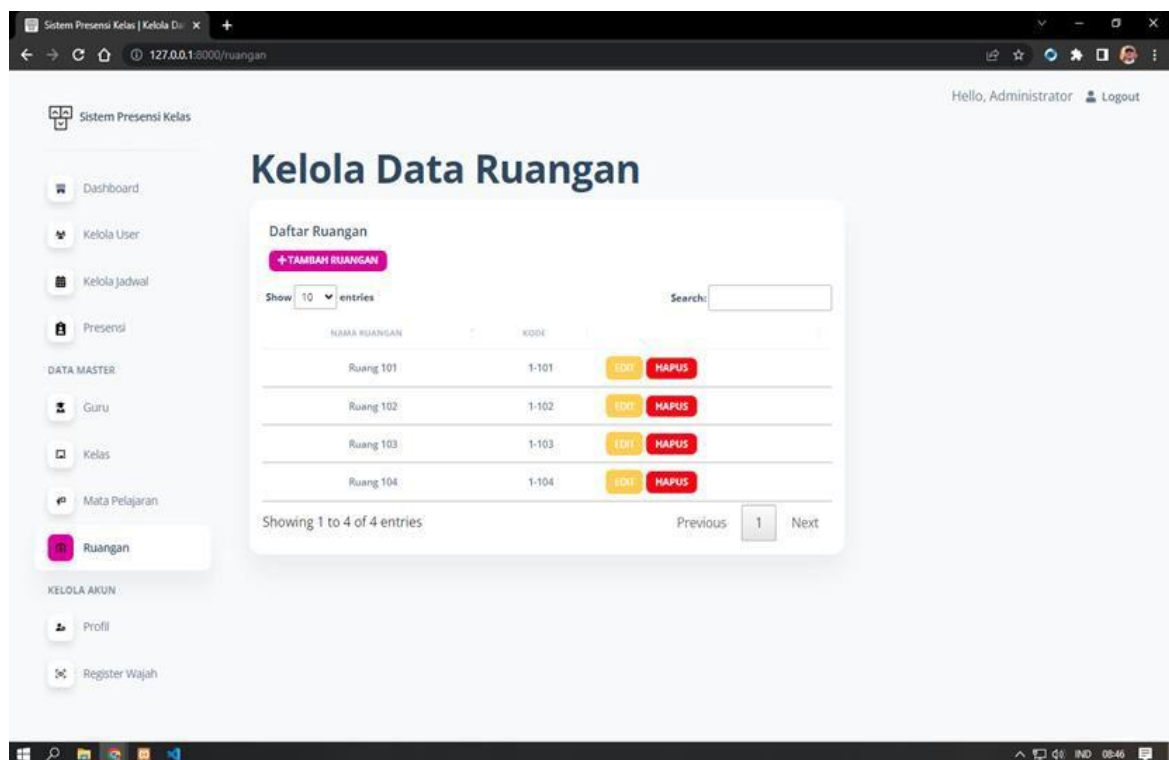
```

public function destroy(Ruangan $ruangan) {
    Ruangan::destroy($ruangan->id);
    return redirect('/ruangan')->with('success', 'Data ruangan berhasil
dihapus!');
}

```

Gambar 4.27 Kode fungsi Hapus pada halaman Kelola Ruangan

Gambar 4.25 hingga Gambar 4.27 menunjukkan kode dari berbagai fungsi yang ada pada halaman Kelola Data Ruangan. Adapun tampilan Kelola Ruangan ditunjukkan pada Gambar 4.28.



Gambar 4.28 Tampilan Kelola Data Ruangan

#### 4.3.8 Halaman Tambah Presensi

Pada halaman Presensi terdapat tombol untuk melakukan presensi. Apabila tidak ada kelas yang perlu dihadiri oleh siswa, maka sistem akan menampilkan *warning* “Belum ada jadwal untuk hari ini!”. Jika siswa sudah melakukan presensi maka akan muncul *warning* ”Anda sudah presensi untuk semua jadwal!”. Namun jika siswa yang ingin presensi belum melakukan registrasi wajah, maka sistem akan menampilkan *error* dengan perintah “Anda belum melakukan registrasi wajah. Registrasi terlebih dahulu di menu Registrasi Wajah!”.

```

public function create(){
    if (Auth()->user()->role == 'User') {
        $dataJadwal = Jadwal::where('kelas_id', Auth()->user()-
>kelas_id)->whereDate('jadwal', date('Y-m-d', time()))->get();
    } else {
        $dataJadwal = Jadwal::whereDate('jadwal', date('Y-m-d', time()))-
>get();
    }
    $jadwal = [];
    foreach ($dataJadwal as $k => $v) {
        $presensi = Presensi::where([
            ['jadwal_id', '=', $v->id],
            ['users_id', '=', Auth()->id()],
        ])->get();
        if (count($presensi) == 0) {
            $jadwal[] = $v;
        }
    }

    $faces = Faces::where('users_id', Auth()->id()->get());

    if (count($faces) != 0) {
        if (count($dataJadwal) != 0) {
            if ($jadwal != null) {
                $data = [
                    'jadwal' => $jadwal,
                    'title' => 'Tambah Presensi',
                ];

                return view('presensi.create', $data);
            } else {
                return redirect('/presensi')->with('success', 'Anda sudah
presensi untuk semua jadwal!');
            }
        } else {
            return redirect('/presensi')->with('success', 'Belum ada
jadwal untuk hari ini!');
        }
    } else {
        return redirect('/presensi')->with('error', 'Anda belum melakukan
registrasi wajah. Registrasi terlebih dahulu di menu Registrasi Wajah!');
    }
}

```

Gambar 4.29 Kode fungsi untuk melakukan presensi

Setelah siswa berhasil melakukan presensi, pada halaman Presensi akan ditampilkan status presensi siswa tersebut.

```

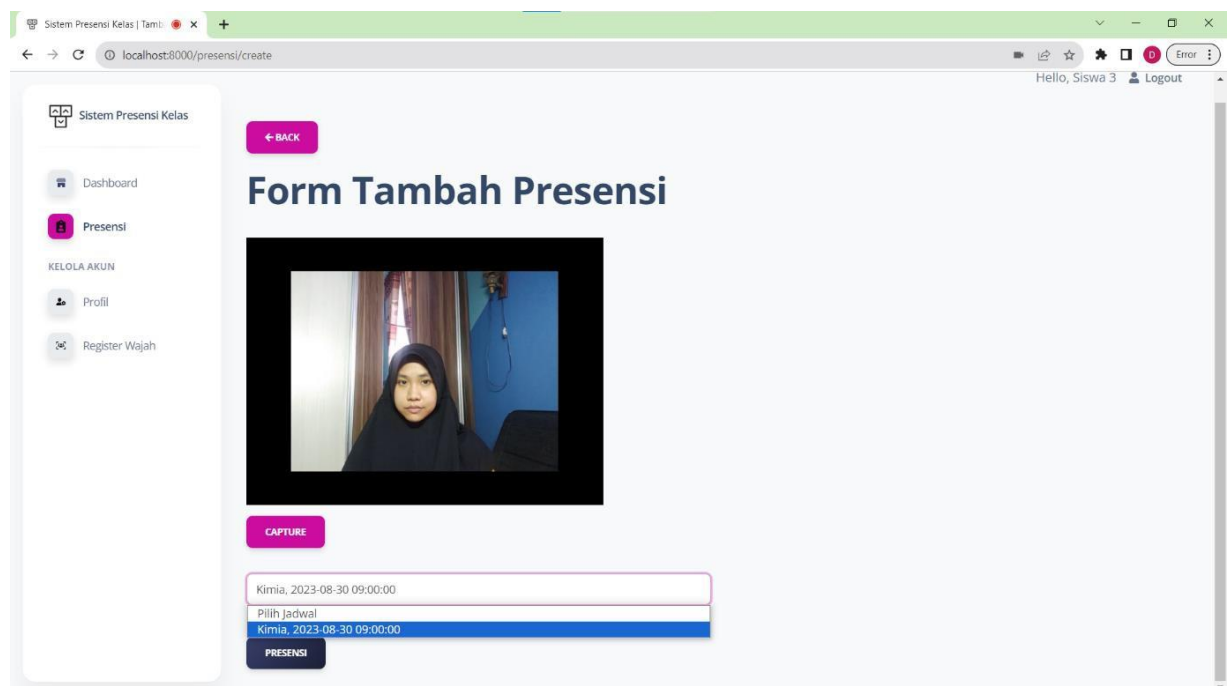
public function store(Request $request)
{
    $users_id = Auth()->id();
    $jadwal_id = $request->jadwal_id;
    $jadwal = Jadwal::select('jadwal')->where('id', $jadwal_id)->get();
    $waktupresensi = date('Y-m-d H:i:s', time());
    if ($jadwal[0]->jadwal >= $waktupresensi) {
        $status = 'Masuk, Tidak Terlambat';
    } else {
        $status = 'Masuk, Terlambat';
    }

    $data = [
        'jadwal_id' => $jadwal_id,
        'users_id' => $users_id,
        'waktupresensi' => $waktupresensi,
        'status' => $status,
    ];

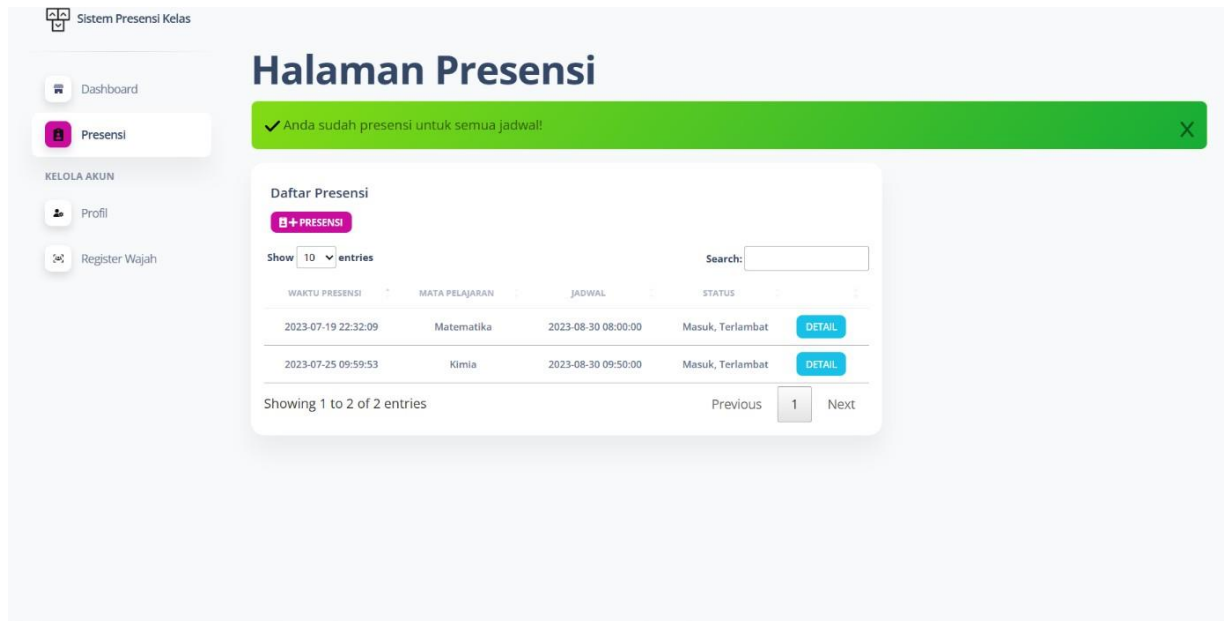
    Presensi::create($data);
    return redirect('/presensi')->with('success', 'Data presensi berhasil
ditambahkan!');
}

```

Gambar 4.30 Kode ketika presensi berhasil dilakukan



Gambar 4.31 Tampilan sistem saat siswa melakukan presensi



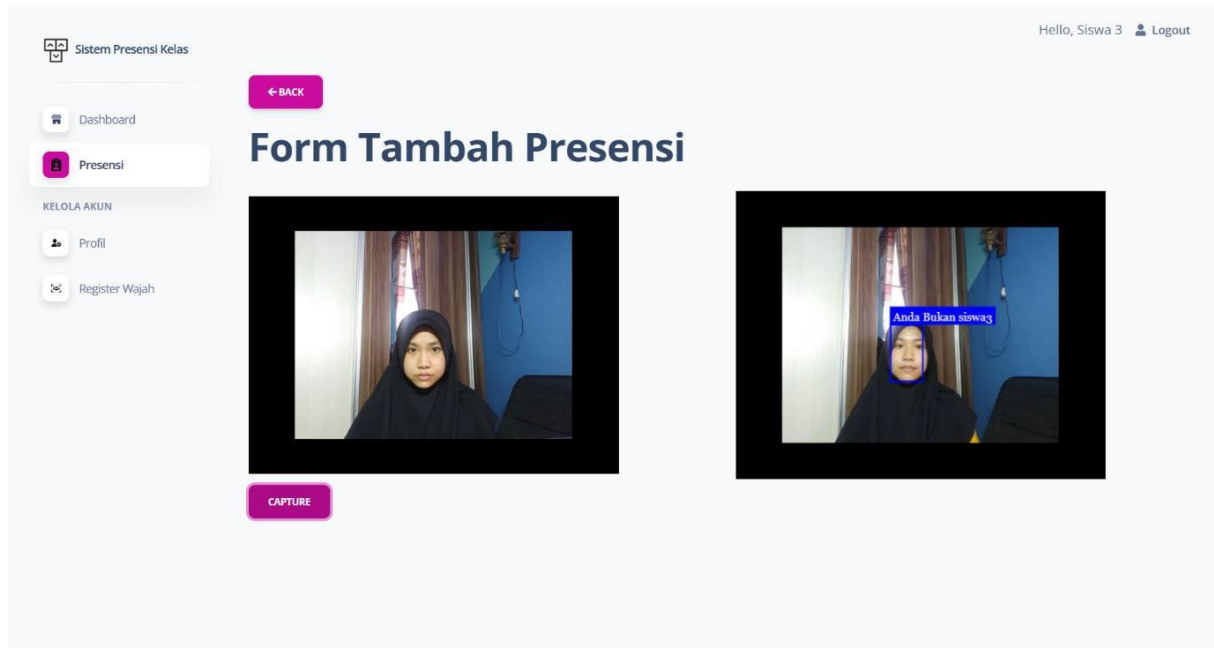
Gambar 4.32 Tampilan saat siswa telah melakukan presensi

#### 4.4 Proses Presensi

Pelaksanaan proses pembelajaran dilakukan di dalam ruang kelas sesuai dengan jadwal yang telah ditetapkan oleh admin dan diketahui dengan baik oleh siswa. Sebelum memulai pembelajaran, siswa dapat mengakses aplikasi web presensi dan login melalui perangkat masing-masing. Setiap siswa memiliki akun yang telah didaftarkan sebelumnya oleh admin.

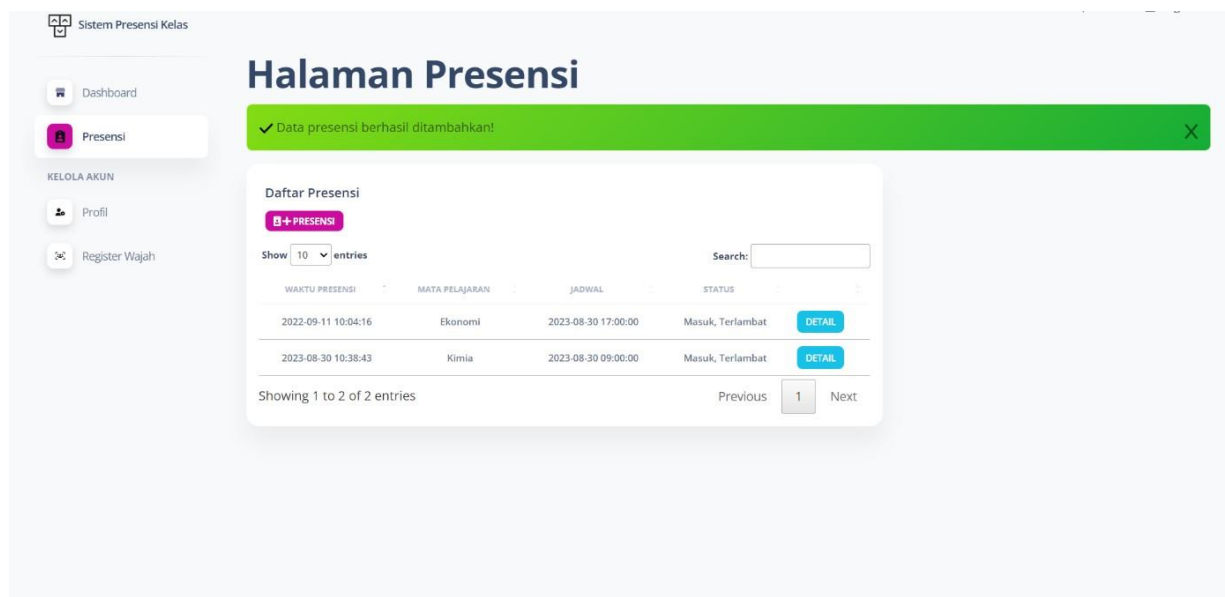
Presensi hanya dapat dilakukan oleh pemilik masing-masing akun dan tidak dapat diwakilkan siswa lain. Apabila siswa mencoba presensi dengan akun selain miliknya, akan muncul *bounding box* dengan tulisan “Anda bukan (nama siswa)” seperti pada Gambar 4.33 dan presensi siswa tersebut tidak akan dicatat oleh sistem.





Gambar 4.33 Siswa melakukan presensi bukan dengan akun miliknya

Namun apabila siswa melakukan presensi dengan akun miliknya sendiri, siswa akan diarahkan ke Halaman Presensi dan pada halaman tersebut akan muncul pemberitahuan bahwa presensi siswa berhasil dicatat oleh sistem seperti pada Gambar 4.34. Siswa hanya dapat melakukan presensi sesuai dengan jadwal yang telah ditentukan. Jika siswa terlambat melakukan presensi, maka sistem akan menampilkan status keterlambatannya.



Gambar 4.34 Presensi siswa berhasil ditambahkan ke database

#### 4.5 Evaluasi Pengujian Sistem

Tabel 4.1 menunjukkan serangkaian uji coba untuk memastikan bahwa alur sistem sesuai dengan rancangan awal. Pengujian ini melibatkan langkah dari inti alur sistem presensi, dimulai dari saat admin mengelola data-data yang diperlukan sistem, kemudian dilanjutkan oleh tindakan siswa untuk melakukan presensi dengan wajah mereka sendiri pada aplikasi web.

Tabel 4.1 Test case sistem presensi

No.	Skenario	Tindakan	Hasil	Keterangan
1.	Log In	Admin dan siswa log in ke aplikasi web dengan menggunakan username dan password	Berhasil	Akun siswa harus telah disediakan oleh admin
2.	Registrasi wajah	Siswa yang baru memiliki akun mendaftarkan wajahnya ke dalam aplikasi web	Berhasil	
3.	Kelola jadwal	Admin mengelola jadwal yang telah ditentukan ke dalam sistem	Berhasil	
4.	Kelola User	Admin menambahkan dan mengedit data siswa yang akan melakukan presensi	Berhasil	
5.	Deteksi wajah	Sistem mendeteksi wajah pada gambar siswa dengan Face-API.js	Berhasil	Tombol "capture" akan muncul ketika sistem mendeteksi wajah siswa
6.	Pengenalan wajah	Gambar wajah dari citra yang ditangkap dan di deteksi akan dikenali dengan Face-API.js	Berhasil	Dipengaruhi oleh perangkat masing-masing, memungkinkan terjadinya <i>lag</i>

7.	Presensi tidak sesuai data diri	Siswa melakukan presensi dengan menggunakan akun siswa lain	Gagal	Presensi tidak akan tercatat dalam sistem
8.	Presensi sesuai data diri	Siswa melakukan presensi dengan menggunakan akunnya sendiri	Berhasil	Siswa akan langsung dialihkan ke halaman presensi
9.	Siswa melihat status presensi	Siswa melihat status presensinya di halaman presensi	Berhasil	Hasil presensi juga akan menunjukkan status ketepatan waktu presensi

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan perancangan sistem presensi diatas, maka dapat disimpulkan bahwa:

1. Sistem telah dirancang dan memiliki 2 jenis pengguna yaitu *user* dan admin. Untuk admin memiliki tugas dalam mengatur jadwal, ruangan, data guru, mata pelajaran, hingga data *user*. Sedangkan untuk user hanya memiliki fungsi untuk melakukan presensi dan *upload* foto.
2. Sistem dapat menganalisa wajah dengan menggunakan *face recognition*, hal tersebut menandakan bahwa sistem dapat digunakan dengan baik.
3. Penerapan algoritma deteksi wajah dapat diterapkan pada modul presensi pada PC menggunakan kamera built-in perangkat yang tersedia meskipun dengan beberapa keterbatasan.

#### 5.2 Saran

Berdasarkan keterbatasan dan kekurangan pada aplikasi, maka saran yang dapat diberikan adalah sebagai berikut :

1. Pada pengembangan berikutnya, diharapkan desain UI aplikasi dapat dibuat lebih menarik.
2. Aplikasi sempat mengalami *lag* saat dijalankan. Sehingga perlu perbaikan logika dan sintaks aplikasi pada penelitian selanjutnya.
3. Aplikasi ini masih menggunakan *localhost*, oleh karena itu harapan kedepannya aplikasi sudah dapat dijalankan secara *online*.

## DAFTAR PUSTAKA

- Anggraini, W. (2020). *DEEP LEARNING UNTUK DETEKSI WAJAH BERHIJAB DGN CNN DAN TENSORFLOW*.
- Chellappa, R., Roy-Chowdhury, A. K., & Zhou, S. K. (n.d.). *Recognition of Humans and Their Activities Using Video*.
- Dwisnanto Putro, M., Magister, M., Elektro, I., Ugm, F. T., Grafika, J., Teguh Bharata, Y., Staf, A., Jurusan, P., Elektro, T., Bondhan, Y., Staf Pengajar, W., & Teknik, J. (n.d.). *Sistem Deteksi Wajah dengan Menggunakan Metode Viola-Jones*.
- Endrianti, F., Setiawan, W., & Wihardi, Y. (2018). *Sistem Pencatatan Kehadiran Otomatis di Ruang Kelas Berbasis Pengenalan Wajah Menggunakan Metode Convolutional Neural Network (CNN)* (Vol. 1, Issue 1). <https://ejournal.upi.edu/index.php/JATIKOM>
- Hikmah, A. B., Mulyani, Y. S., Alawiyah, T., Wiguna, W., Riksa, R., & Ridwan, A. (2021). IJCIT (Indonesian Journal on Computer and Information Technology) Rancang Bangun Sistem Informasi Absensi Siswa Berbasis Web Pada SMAN 1 Singaparna. In *IJCIT (Indonesian Journal on Computer and Information Technology)* (Vol. 6, Issue 2).
- House, D. H. (2002). *VIZA 654 / CPSC 646-The Digital Image Course Notes*.
- Inrawansyah, M. N. (2017). IMPLEMENTASI FACE DETECTION MENGGUNAKAN METODE VIOLA JONES UNTUK MEMBANTU MEMPERMUDAH PROSES COUNTER PENGUNJUNG GEDUNG. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 1, Issue 1).
- Nordin, N., & Fauzi, N. H. M. (2020). A web-based mobile attendance system with facial recognition feature. *International Journal of Interactive Mobile Technologies*, 14(5), 193–202. <https://doi.org/10.3991/IJIM.V14I05.13311>
- Nugraha Ramdhon, A., & Febriya, F. (2021). Penerapan Face Recognition Pada Sistem Presensi. *Journal of Applied Computer Science and Technology*, 2(1), 12–17. <https://doi.org/10.52158/jacost.v2i1.121>
- Prathivi, R., & Kurniawati, Y. (2020). SISTEM PRESENSI KELAS MENGGUNAKAN PENGENALAN WAJAH DENGAN METODE HAAR CASCADE CLASSIFIER. *SIMETRIS*, 11(1).
- Raghav, P. (2018, March 4). *Understanding of Convolutional Neural Network (CNN) — Deep Learning*. <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>

- Santoso, B., & Kristianto, R. P. (n.d.). *IMPLEMENTASI PENGGUNAAN OPENCV PADA FACE RECOGNITION UNTUK SISTEM PRESENSI PERKULIAHAN MAHASISWA*.
- Sylaj, A. (2019). *STUDENT ATTENDANCE SYSTEM USING FACE RECOGNITION*.  
<https://knowledgecenter.ubt-uni.net/etd/1621>
- Widodo, D. W. (2016). *Sistem Pendataan Presensi Mahasiswa Di Teknik Informatika Universitas Nusantara PGRI Kediri*.
- Abdurrohman, H., Dini, R., & P. Muharram, A. (2018). Evaluasi Performa metode Deep Learning untuk Klasifikasi Citra Lesi Kulit The HAM10000. *Seminar Nasional Instrumentasi, Kontrol dan Otomasi (SNIKO)*.
- Brown, S. (2021, 4 21). *Machine Learning, Explained*. Retrieved from mitsloan.mit.edu:  
<https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained>
- Hendrik, Anjomshooa, A., & Tjoa, A. M. (2014). Towards Semantic Mashup Tools For Big Data Analysis. *Proceeding of the Information & Communication Technology-EurAsia Conference 2014*, (pp. 100-145). Bali.
- Inkeaw, P., Srikummoon, P., Chaijaruwanich, J., Traisathit, P., Awiphan, S., Inchai, J., . . . Theerakittikul, T. (2022). Automatic Driver Drowsiness Detection Using Artificial Neural Network Based on Visual Facial Descriptors: Pilot Study. *Nature and Science of Sleep*.
- Mühler, V. (2018, 6 25). *face-api.js — JavaScript API for Face Recognition in the Browser with tensorflow.js*. Retrieved from Medium: <https://itnext.io/face-api-js-javascript-api-for-face-recognition-in-the-browser-with-tensorflow-js-bcc2a6c4cf07>
- Setiawan, A. M. (2013). *Integrated Framework For Business Process Complexity Analysis*. Retrieved from ECIS 2013 Completed Research: [http://aisel.aisnet.org/ecis2013\\_cr/49](http://aisel.aisnet.org/ecis2013_cr/49)
- Suhery, C., & Ruslianto, I. (2017). Identifikasi Wajah Manusia untuk Sistem Monitoring Kehadiran Perkuliahan menggunakan Ekstraksi Fitur Principal Component Analysis (PCA). *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*.
- Syarif, A. C. (2021). Perancangan Sistem Presensi Pengenalan Wajah Pada Sistem Tata Kelola Perkuliahan Universitas Atma Jaya Makassar. *Jurnal Tematika*.
- Taufiq, H. (2015). *Argumentasi dan Validitas*. Yogyakarta: Darqin.
- Wahid, F. (2014). The Antecedents And Impacts of a Green Eprocurement Infrastructure: Evidence From The Indonesian Public Sector. *International Journal of internet Protocol Technology*, 7(4), 210-218.

- Wang, J., & Li, Z. (2018). Research on Face Recognition Based on CNN. *IOP Conference Series: Earth and Environmental Science*.
- Zufar, M., & Setiono, B. (2016). Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time. *JURNAL SAINS DAN SENI ITS*.
- Zukhri, Z. (2014). *Algoritma Genetika: Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi*. Yogyakarta: Andi Publisher.
- Zulfachmi, Sudarma, M., & Jasa, L. (2019). Sistem Monitoring Kehadiran Perkuliahan Berbasis Face . *Majalah Ilmiah Teknologi Elektro*.