

**RANCANG BANGUN APLIKASI ABSENSI PENGENALAN
WAJAH MENGGUNAKAN ALGORITMA *LOCAL BINARY
PATTERN HISTOGRAM* DAN DETEKSI KEDIPAN MATA
BERBASIS WEBSITE**

Studi Kasus Kantor Desa Kertawinangun

SKRIPSI

Diajukan untuk Memenuhi Persyaratan
guna memperoleh gelar
Sarjana Komputer



DISUSUN OLEH:

ELISWATI KUSUMA AYU

NIM : 0402201032

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS NAHDLATUL ULAMA CIREBON
2024 M / 1446 H**



UNIVERSITAS NAHDLATUL ULAMA (UNU) CIREBON

LEMBAR PENGESAHAN STATUS SKRIPSI

JUDUL : RANCANG BANGUN APLIKASI ABSENSI PENGENALAN
WAJAH MENGGUNAKAN ALGORITMA *LOCAL BINARY
PATTERN HISTOGRAM* DAN DETEKSI KEDIPAN MATA
BERBASIS WEBSITE

SAYA : ELISWATI KUSUMA AYU

NIM : 0402201032

Mengijinkan Skripsi Sarjana Komputer ini disimpan di Perpustakaan
Universitas Nahdlatul Ulama dengan syarat-syarat kegunaan sebagai berikut :

1. Skripsi adalah hak milik Universitas Nahdlatul Ulama.
2. Perpustakaan Univeristas Nahdlatul Ulama dibenarkan membuat Salinan untuk referensi saja.
3. Perpustakaan juga dibenarkan membuat salinan Skripsi ini sebagai bahan pertukaran antar institusi Pendidikan tinggi.
4. Berikan tanda ☒ sesuai dengan kategori Skripsi.

☐ Sangat Rahasia (Mengandung isi tentang keselamatan atau kepentingan negara Indonesia)

☐ Rahasia (Mengandung isi tentang kerahasiaan dari suatu organisasi/badan di mana penelitian Skripsi ini dikerjakan)

☒ Biasa

Disahkan oleh :

ROSIDIN, S.Kom, M. Kom

Tanggal : 14 September 2024

H. SUKARSA, S.T, M. Kom

Tanggal : 14 September 2024



UNIVERSITAS NAHDLATUL ULAMA (UNU) CIREBON

LEMBAR PERNYATAAN PENULIS

JUDUL : RANCANG BANGUN APLIKASI ABSENSI PENGENALAN
WAJAH MENGGUNAKAN ALGORITMA *LOCAL BINARY
PATTERN HISTOGRAM* DAN DETEKSI KEDIPAN MATA
BERBASIS WEBSITE

SAYA : ELISWATI KUSUMA AYU

NIM : 0402201032

"Saya menyatakan dan bertanggungjawab dengan sebenarnya bahwa Skripsi ini adalah hasil karya saya sendiri kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya. Jika pada waktu selanjutnya ada pihak lain yang mengklaim bahwa Skripsi ini sebagai karyanya, yang disertai dengan bukti-bukti yang cukup, maka saya bersedia untuk membatalkan gelar Sarjana Komputer saya beserta segala hak dan kewajiban yang melekat pada gelar tersebut".

Cirebon, 4 September 2024



ELISWATI KUSUMA AYU

NIM : 0402201032



UNIVERSITAS NAHDLATUL ULAMA (UNU) CIREBON

LEMBAR PERSETUJUAN SKRIPSI

JUDUL : RANCANG BANGUN APLIKASI ABSENSI PENGENALAN
WAJAH MENGGUNAKAN ALGORITMA *LOCAL BINARY
PATTERN HISTOGRAM* DAN DETEKSI KEDIPAN MATA
BERBASIS WEBSITE

SAYA : ELISWATI KUSUMA AYU

NIM : 0402201032

Skrripsi ini telah diperiksa dan disetujui,
Cirebon, 4 September 2024

Pembimbing I

ROSIDIN, S. Kom, M. Kom
NIDN. 0415028303

Pembimbing II

H. SUKARSA, S.T, M. Kom
NIDN. 0418117605

Mengetahui,
Ketua Program Studi

DICKY ANDIKA SULAEMAN, S.T, M. Kom
NIDN. 0409069402



UNIVERSITAS NAHDLATUL ULAMA (UNU) CIREBON

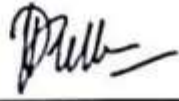
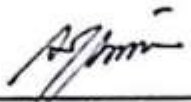
LEMBAR PENGESAHAN SKRIPSI

JUDUL : RANCANG BANGUN APLIKASI ABSENSI PENGENALAN
WAJAH MENGGUNAKAN ALGORITMA *LOCAL BINARY
PATTERN HISTOGRAM* DAN DETEKSI KEDIPAN MATA
BERBASIS WEBSITE

SAYA : ELISWATI KUSUMA AYU

NIM : 0402201032

Skripsi ini telah diujikan dan dipertahankan dihadapan Dewan Penguji pada Sidang Skripsi tanggal 7 September 2024 menurut pandangan kami, Skripsi ini memadai untuk penganugrahan gelar Sarjana Komputer (S.Kom).

	Tanggal	Tanda Tangan
Pembimbing 1, <u>ROSIDIN, S. Kom, M. Kom</u> NIDN. 0415028303	19/09/2024	
Pembimbing 2, <u>H. SUKARSA, S.T, M. Kom</u> NIDN. 0418117605	19/09/2024	
Penguji 1, <u>DICKY ANDIKA SULAEMAN, S.T, M. Kom</u> NIDN. 0409069402	19/09/2024	
Penguji 2, <u>ABDUL KOHAR, M. Kom</u> NIDN. 0412068704	19/09/2024	

Mengetahui,
Dekan Fakultas Ilmu Komputer
Universitas Nahdlatul Ulama Cirebon


ROSIDIN, S. Kom, M. Kom
NIDN. 0415028303

MOTO DAN PERSEMBAHAN

MOTO

"Dan janganlah kamu berputus asa dari rahmat Allah. Sesungguhnya tidak ada yang berputus asa dari rahmat Allah kecuali kaum yang kafir."

(QS. Yusuf: 87)

"Believe in your own efforts, dream big, be patient, stay focused, and surround yourself with good people."

PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah Subhanahu Wa Ta'ala dan Nabi Muhammad Shallallahu 'Alaihi Wasallam, saya memberikan karya ini untuk Mamah, Papah, dan Adik yang selalu setia mendukung saya. Terima kasih atas segala pengorbanan, kasih sayang, dan doa yang selalu mengiringi perjalanan hidup saya. Setiap langkah yang saya tempuh, setiap rintangan yang saya hadapi, semuanya terasa lebih ringan karena kehadiran kalian. Kalian adalah alasan utama saya memberikan harapan, dorongan untuk menghadapi setiap tantangan, dan menjadi penyemangat terbesar dalam setiap langkah perjuangan saya. Tanpa kalian, saya tidak akan bisa mencapai titik ini. Semoga karya ini bisa menjadi wujud dari penghargaan, rasa hormat, dan rasa terima kasih saya untuk kalian. Semoga Allah Subhanahu Wa Ta'ala senantiasa melimpahkan kesehatan, kebahagiaan, dan keberkahan kepada kalian.

ABSTRACT

Attendance at the Kertawinangun Village Office is still done manually and is vulnerable to various forms of fraud, such as attendance on another day independently or entrusting attendance with the help of other people. The attendance recapitulation process also takes time because it is done manually. Facing this challenge, this study aims to design and build a web-based attendance application that utilizes facial recognition technology with Local Binary Pattern Histogram (LBPH) and eye blink detection through facial landmarks from the dlib library. With the application of this technology, face-based attendance is expected to be able to effectively identify employee faces on workdays, and eye blink detection ensures that the physical presence of detected employees is actually present, and speeds up the recapitulation process. This application is also equipped with a feature to download the recapitulation results in CSV format, which simplifies the process of processing attendance data and increases overall time efficiency. With this application, the potential for fraud in attendance can be reduced and significantly increase the reliability and speed of employee attendance management.

Keywords: facial recognition, dlib, Local Binary Pattern Histogram.

ABSTRAK

Absensi di Kantor Desa Kertawinangun masih dilakukan secara manual dan rentan terhadap berbagai bentuk kecurangan, seperti absensi pada hari lain secara mandiri atau menitip absen dengan bantuan orang lain. Proses rekapitulasi absensi juga memakan waktu karena dilakukan secara manual. Menghadapi tantangan ini, penelitian ini bertujuan untuk merancang dan membangun aplikasi absensi berbasis web yang memanfaatkan teknologi pengenalan wajah dengan *Local Binary Pattern Histogram* (LBPH) serta deteksi kedipan mata melalui facial landmark dari library dlib. Dengan penerapan teknologi ini, absensi berbasis wajah diharapkan dapat secara efektif mengidentifikasi wajah pegawai pada hari kerja, dan deteksi kedipan mata memastikan kehadiran fisik pegawai yang terdeteksi benar-benar hadir, serta mempercepat proses rekapitulasi. Aplikasi ini juga dilengkapi fitur untuk mengunduh hasil rekapitulasi dalam format CSV, yang mempermudah proses pengolahan data absensi dan meningkatkan efisiensi waktu secara keseluruhan. Dengan adanya aplikasi ini, potensi kecurangan dalam absensi dapat berkurang dan secara signifikan meningkatkan keandalan serta kecepatan dalam pengelolaan kehadiran pegawai.

Kata Kunci: pengenalan wajah, dlib, *Local Binary Pattern Histogram*.

KATA PENGANTAR



Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan penelitian skripsi dengan judul “Rancang Bangun Aplikasi Absensi Pengenalan Wajah Menggunakan Algoritma *Local Binary Pattern Histogram* Dan Deteksi Kedipan Mata Berbasis Website”. Shalawat beriringan salam semoga selalu tercurah kepada junjungan Nabi Muhammad SAW beserta keluarga, sahabat, kerabat, serta muslimin dan muslimat, semoga kita semua mendapatkan syafa’at dari beliau di akhirat kelak. Aamiin.

Dalam penyusunan skripsi ini penulis telah mendapat banyak bantuan dan bimbingan serta semangat dari berbagai pihak, tentunya penyusunan laporan ini memiliki proses dalam penyelesaian. Oleh karena itu, penulis ingin menyampaikan terima kasih kepada :

1. Allah Subhanahu Wa Ta’la yang telah memberikan rahmat, hidayah serta kesehatan sehingga penulis dapat menyelesaikan skripsi ini.
2. Nabi Muhammad SAW, teladan utama bagi umat Islam, yang ajarannya menjadi pedoman dalam setiap langkah hidup ini.
3. Kedua Orang tua memberikan doa dan memberikan semangat serta kasih sayang sehingga penulis dapat menyelesaikan penelitian ini.
4. Untuk diri saya sendiri, terima kasih karena telah mau berjuang dan bertahan sampai saat ini dan mampu berada di titik ini.
5. Bapak Dr. Agus Sugiarto, SH., MH., MM selaku Rektor Universitas Nahdlatul Ulama Cirebon.
6. Bapak Rosidin, S. Kom, M. Kom selaku Dekan Fakultas Ilmu Komputer Universitas Nahdlatul Ulama Cirebon.
7. Bapak Dicky Andika Sulaeman, S.T, M.Kom, selaku Ketua Program Studi Teknik Informatika.

8. Bapak Rosidin, S. Kom, M. Kom selaku Dosen Pembimbing I yang selalu memberikan masukan dan arahan dalam penyusunan laporan skripsi ini.
9. Bapak H. Sukarsa, S.T, M. Kom selaku Dosen Pembimbing II yang telah meluangkan waktu, arahan, dan motivasi pada saat bimbingan sehingga penelitian ini dapat terselesaikan.
10. Dosen-dosen Fakultas Ilmu Komputer Universitas Nahdlatul Ulama Cirebon.
11. Ibu Putri selaku Staff Kantor Desa Kertawinangun Cirebon yang telah membantu penulis dengan memberikan waktu dan informasi.
12. Dan seluruh pihak yang tidak dapat disebutkan satu persatu yang telah membantu hingga terselesaikannya penelitian ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat banyak kekurangan yang disebabkan oleh keterbatasan pengetahuan penulis. Oleh karena itu, penulis memohon maaf dan berharap agar para pembaca dapat memaklumi kekurangan yang ada dalam penelitian ini.

Akhir kata penulis berharap semoga skripsi ini dapat bermanfaat bagi para pembaca pada umumnya. Terima Kasih.

Cirebon, 15 Agustus 2024

Eliswati Kusuma Ayu

DAFTAR ISI

LEMBAR PENGESAHAN STATUS SKRIPSI	ii
LEMBAR PERNYATAAN PENULIS	iii
LEMBAR PERSETUJUAN SKRIPSI.....	iv
LEMBAR PENGESAHAN SKRIPSI	v
MOTO DAN PERSEMBAHAN.....	vi
<i>ABSTRACT</i>	vii
ABSTRAK.....	viii
KATA PENGANTAR.....	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Identifikasi Masalah	3
1.3. Rumusan Masalah	3
1.4. Batasan Masalah.....	3
1.5. Tujuan Penelitian.....	4
1.6. Manfaat Penelitian	4
1.7. Metode Dan Teknik Penelitian.....	5
1.7.1. Metode.....	5
1.7.2. Teknik Penelitian.....	5
1.8. Literatur Review.....	6
1.9. Sistematika Penulisan	11
1.10. Kerangka Berfikir.....	12

BAB II LANDASAN TEORI	13
2.1. Keterkaitan Dengan Al-Qur'an Surat Al-Baqarah (42) Ayat 2.....	13
2.2. Citra Digital.....	13
2.3. <i>Computer vision</i>	14
2.4. <i>Face Recognition</i>	16
2.5. <i>Haar Cascade Clasifier</i>	17
2.6. <i>Local Binary Pattern Histogram (LBPH)</i>	18
2.7. Dlib.....	19
2.8. Python	20
2.9. Tools Aplikasi.....	21
2.9.1. Xampp	21
2.9.2. Mysql	22
2.9.3. Apache Websiteserver	23
2.9.4. Django Framework.....	24
2.9.5. OpenCV	25
2.9.6. PhpMyAdmin.....	26
2.9.7. Visual Studio Code.....	28
2.9.8. Figma Atau Adobe	29
2.9.9. Blackbox	32
2.9.10. Entity Relationship Diagram (ERD)	33
2.10. Metode Waterfall.....	34
2.11. <i>Use Case</i>	35
2.12. Activity Diagram.....	37
2.13. <i>Flowchart</i>	38
BAB III ANALISIS DAN PERANCANGAN SISTEM	41

3.1.	Tempat Dan Waktu Penelitian.....	41
3.1.1.	Tempat.....	41
3.1.2.	Waktu Penelitian	41
3.2.	Analisis Sistem.....	41
3.2.1.	Analisis Masalah	41
3.2.2.	Analisis Kebutuhan Perangkat Keras Dan Perangkat Lunak.....	42
3.2.3.	Analisis Proses	43
3.3.	Perancangan Sistem Database.....	52
3.4.	Tampilan Perancangan	55
3.4.1.	Tampilan Perancangan Home	55
3.4.2.	Tampilan Perancangan Signup.....	56
3.4.3.	Tampilan Perancangan Login.....	56
3.4.4.	Tampilan Perancangan Registrasi Absensi.....	57
3.4.5.	Tampilan Perancangan Dataset Wajah	57
3.4.6.	Tampilan Perancangan Absensi.....	58
3.4.7.	Tampilan Perancangan Rekap Absensi	58
3.4.8.	Tampilan Perancangan Admin	59
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM.....		60
4.1.	Implementasi Interface.....	60
4.1.1.	Halaman Signup	60
4.1.2.	Halaman Login.....	60
4.1.3.	Halaman Register Absensi	61
4.1.4.	Halaman Proses Penyimpanan Dataset Wajah	62
4.1.5.	Dataset Wajah.....	62
4.1.6.	File Model LBPH.....	63

4.1.7. Halaman Absensi.....	64
4.1.8. Halaman Rekap Absensi	64
4.1.9. Halaman Admin	65
4.2. Implementasi Program	66
4.2.1. Implementasi Program Dataset	66
4.2.2. Implementasi Program Pengenalan Wajah.....	68
4.2.3. Implemetasi Program Rekap Absensi	72
4.3. Pengujian Sistem.....	74
BAB V KESIMPULAN DAN SARAN.....	78
5.1. Kesimpulan	78
5.2. Saran.....	78
DAFTAR PUSTAKA	79

DAFTAR GAMBAR

Gambar 1. 1 Kerangka Berfikir.....	12
Gambar 2. 1 Ilustrasi <i>Computer vision</i>	15
Gambar 2. 2 <i>Haar Cascade</i> Classier.....	17
Gambar 2. 3 LBPH.....	18
Gambar 2. 4 Dlib.....	19
Gambar 2. 5 Analog Mata	20
Gambar 2. 6 Python	20
Gambar 2. 7 Xampp.....	21
Gambar 2. 8 Mysql.....	22
Gambar 2. 9 Django	24
Gambar 2. 10 OpenCv.....	25
Gambar 2. 11 PHP MyAdmin	26
Gambar 2. 12 Visual Studio Code.....	28
Gambar 2. 13 Figma Atau Adobe.....	29
Gambar 2. 14 Tahap Metode Waterfall	34
Gambar 3. 1 Flowchart Simpan Data Wajah.....	43
Gambar 3. 2 Foto Wajah RGB	44
Gambar 3. 3 Foto Wajah Grayscale	44
Gambar 3. 4 Flowchart Proses LBPH.....	45
Gambar 3. 5 Membagi 3x3 Pixel Wajah	46
Gambar 3. 6 Flowchart Deteksi Kedipan Mata.....	47
Gambar 3. 7 Titik Landmark Mata.....	48
Gambar 3. 8 <i>Use Case</i>	49
Gambar 3. 9 Activity Diagram	50
Gambar 3. 10 Entity relationship diagram (ERD)	52
Gambar 3. 11 Tampilan Perancangan Home.....	55
Gambar 3. 12 Tampilan Perancangan Signup	56
Gambar 3. 13 Tampilan Perancangan Login.....	56
Gambar 3. 14 Tampilan Perancangan Registrasi Absensi.....	57
Gambar 3. 15 Tampilan Perancangan Dataset Wajah	57

Gambar 3. 16 Tampilan Perancangan Absensi.....	58
Gambar 3. 17 Tampilan Perancangan Rekap Absensi.....	58
Gambar 3. 18 Tampilan Perancangan Admin.....	59
Gambar 4. 1 Halaman Signup	60
Gambar 4. 2 Halaman Login.....	60
Gambar 4. 3 Halaman Register Absensi	61
Gambar 4. 4 Proses Simpan Dataset Wajah	62
Gambar 4. 5 Dataset Wajah.....	62
Gambar 4. 6 File Model LBPH.....	63
Gambar 4. 7 Halaman Absensi.....	64
Gambar 4. 8 Halaman Rekap Absensi.....	64
Gambar 4. 9 Halaman Admin	65
Gambar 4. 10 File Rekap	66

DAFTAR TABEL

Tabel 1. 1 Literatur Review	10
Tabel 2. 1 <i>Use Case</i> Diagram.....	36
Tabel 2. 2 Activity Diagram	38
Tabel 2. 3 Flowchart.....	40
Tabel 3. 1 Waktu Penelitian.....	41
Tabel 3. 2 Tabel Kebutuhan Perangkat Lunak	42
Tabel 3. 3 Perancangan Database Pekerja.....	53
Tabel 3. 4 Perancangan Database Data Pekerja	53
Tabel 3. 5 Perancangan Database Image Dataset.....	53
Tabel 3. 6 Perancangan Database Absensi	54
Tabel 3. 7 Perancangan Database Rekap Absensi	54
Tabel 3. 8 Perancangan Database Auth User	55
Tabel 4. 1 Pengujian Blackbox.....	77

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi saat ini memungkinkan pemanfaatan data biologis manusia untuk berbagai tujuan. Ini disebabkan oleh perbedaan karakteristik biologis setiap orang yang dapat digunakan sebagai informasi identifikasi, seperti sidik jari, retina, suara, dan wajah. Dalam penerapan teknologi biometrik, perangkat sensor yang sesuai dengan karakteristik data yang digunakan sebagai input sangat diperlukan. *Face Recognition* merupakan salah satu teknologi biometrik yang telah dikembangkan oleh para ahli, menggunakan algoritma pengenalan wajah untuk membedakan individu berdasarkan data yang ada dalam database wajah.

Banyak aplikasi presensi di internet menghadapi masalah, seperti kesulitan dengan input sidik jari akibat mesin error atau pemindai yang tidak berfungsi. Mesin sering gagal membaca sidik jari ketika tangan pegawai basah karena air menghalangi pembacaan garis pada jari. Penggunaan mesin dalam kondisi ini dapat mempercepat kerusakan alat. Selain itu, pemadaman listrik juga dapat mengganggu proses absensi sidik jari. Di sisi lain, sensor wajah yang menggunakan kamera cenderung lebih mudah ditemukan dibandingkan dengan karakteristik lainnya dan lebih murah. Algoritma pengenalan wajah seperti *Eigenface*, Fisherface, dan *Local Binary Patterns Histograms* (LBPH) menunjukkan bahwa setiap metode memiliki karakteristik unik dalam menangani variabilitas, seperti tingkat perubahan pencahayaan dan ekspresi wajah. “Meskipun *Eigenface* dan Fisherface memberikan hasil yang baik, LBPH terbukti lebih akurat dan efisien dalam kondisi yang lebih kompleks, seperti perubahan pencahayaan dan ekspresi wajah” (Detila et al., 2019).

Proses absensi di Kelurahan Kertawinangun saat ini masih dilakukan secara manual, yaitu dengan menuliskan nama dan menandatangani buku kehadiran setiap hari kerja. Proses absensi yang menggunakan buku, seringkali terjadi kecurangan seperti pegawai yang menandatangani absensi secara mandiri pada hari lain meskipun sebenarnya tidak hadir, atau bahkan menitipkan absensi pada pegawai

lain dengan cara menuliskan nama dan meniru tanda tangan. Selain itu, proses rekapitulasi absensi memakan waktu yang lama karena melibatkan penghitungan manual dari buku absensi, seperti jumlah hari kerja dan kehadiran. Berdasarkan absensi di Kelurahan Kertawinangun yang diuraikan dapat diketahui bahwa masih memiliki kekurangan, seperti rentan terhadap kecurangan absensi yang bisa dilakukan secara mandiri ataupun bantuan orang lain dan membutuhkan waktu yang lama untuk rekapitulasi kehadiran dan hari kerja.

Dikelurahan kertawinangun terdapat kecurangan menitipkan absensi pada pegawai lain sedangkan teknologi pengenalan wajah banyak digunakan sebagai salah satu teknologi untuk melakukan identifikasi wajah seseorang. Namun pada beberapa perangkat belum memiliki komponen pengenalan wajah yang mampu mengenali orang asli yang sudah dikenali atau hanya foto saja karena hanya menyematkan Pengenalan Wajah 2D (dua dimensi). Untuk meningkatkan keandalan teknologi ini, diterapkan deteksi kedipan mata menggunakan facial landmark dari library dlib, yang berfungsi memastikan bahwa individu yang melakukan absensi adalah orang asli, bukan hanya foto atau gambar statis. Dengan penerapan deteksi kedipan mata ini, absensi pengenalan wajah dapat menjadi lebih efektif dan handal dalam mendeteksi kehadiran yang sebenarnya. Selain itu, dengan menggunakan absensi pengenalan wajah dapat mempercepat proses rekapitulasi berdasarkan hari kerja dan kehadiran.

Berdasarkan permasalahan di atas, maka penelitian ini merancang dan membangun aplikasi absensi berbasis web yang menggunakan teknologi pengenalan wajah dengan algoritma *Local Binary Patterns Histograms* (LBPH) serta deteksi kedipan mata menggunakan facial landmark dlib, dan juga absensi pengenalan wajah untuk mempercepat proses rekapitulasi kehadiran dan hari kerja. Dengan penerapan penelitian diatas diharapkan absensi pengenalan wajah dapat mengidentifikasi kehadiran seseorang untuk melakukan absensi, serta dengan penerapan deteksi kedipan mata dapat membantu mengetahui kehadiran palsu. Selain itu, rekapitulasi absensi berdasarkan hari kerja dan kehadiran dapat dilakukan dengan cepat, dan hasil rekapitulasi dapat diunduh dalam format CSV.

1.2. Identifikasi Masalah

Berdasarkan latar belakang di atas, maka identifikasi masalah pada penelitian ini adalah sebagai berikut:

1. Absensi manual menggunakan buku kehadiran memungkinkan pegawai untuk menandatangani absensi secara mandiri pada hari yang berbeda, meskipun mereka tidak hadir pada hari tersebut.
2. Absensi manual menggunakan buku kehadiran juga memungkinkan penitipan absensi, seperti meminta bantuan pegawai lain untuk menuliskan nama dan meniru tanda tangan.
3. Proses rekapitulasi absensi manual memerlukan waktu karena melibatkan pengumpulan data dari buku absensi, perhitungan manual jumlah hari kerja dan kehadiran.

1.3. Rumusan Masalah

Berdasarkan identifikasi masalah yang telah diuraikan, maka dapat dirumuskan beberapa permasalahan sebagai berikut:

1. Bagaimana merancang bangun absensi pengenalan wajah dan deteksi kedipan mata untuk mencegah kecurangan seperti absensi mandiri pada hari lain serta penitipan absensi oleh pegawai lain?
2. Bagaimana penerapan algoritma LBPH untuk absensi pengenalan wajah serta penggunaan library dlib dengan facial landmarks dalam mendeteksi kedipan mata?
3. Bagaimana rekapitulasi kehadiran berdasarkan absensi pengenalan wajah untuk hari kerja Senin-Jumat dalam sebulan secara efisien guna mempercepat proses rekapitulasi?

1.4. Batasan Masalah

Batasan masalah untuk memastikan penelitian tetap terfokus dan dapat diselesaikan dalam waktu yang ditentukan, maka batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Menggunakan framework full-stack Django dengan database MySQL.
2. Fitur-fitur yang disediakan meliputi registrasi absensi untuk menyimpan data wajah pegawai, absensi dengan pengenalan wajah dan deteksi kedipan mata, serta rekapitulasi daftar kehadiran.
3. Bahasa pemrograman yang digunakan dalam pengembangan sistem ini adalah Python.
4. Pengenalan wajah untuk absensi harus dilakukan dengan wajah menghadap langsung ke kamera dalam kondisi pencahayaan yang memadai.

1.5. Tujuan Penelitian

Berdasarkan identifikasi dan rumusan masalah yang telah diuraikan, tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang bangun absensi pengenalan wajah dengan deteksi kedipan mata untuk mengurangi kecurangan absensi pada hari lain dan mencegah kecurangan penitipan absensi.
2. Menggunakan algoritma LBPH pengenalan wajah untuk mengidentifikasi kehadiran pada hari kerja dan library dlib facial landmark untuk memverifikasi kehadiran individu secara fisik bukan penitipan absensi.
3. Mempercepat rekapitulasi absensi kehadiran pegawai dari hari kerja Senin hingga Jumat dalam sebulan, serta menyediakan hasil rekapitulasi yang dapat diunduh dalam format CSV.

1.6. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat bagi beberapa pihak sebagai berikut:

1. Bagi Penulis

Penelitian ini akan memberikan penulis kesempatan untuk mengasah keterampilan dalam bidang teknologi biometrik, pengenalan wajah dan deteksi kedipan mata untuk absensi. Penulis juga akan mendapatkan pengalaman berharga dalam pengembangan aplikasi berbasis web, yang

akan memperkuat keahlian teknis dan pemahaman tentang algoritma pengenalan wajah.

2. Bagi Universitas

Penelitian ini akan memperbanyak sumber daya akademik universitas dengan menyediakan studi kasus terkini mengenai teknologi biometrik untuk absensi. Selain itu, hasil penelitian ini dapat digunakan sebagai bahan ajar dan referensi bagi mahasiswa dan peneliti lainnya.

3. Bagi Masyarakat

Absensi berbasis pengenalan wajah dan deteksi kedipan mata dapat meningkatkan kemudahan dalam proses kehadiran di berbagai institusi, seperti kelurahan dan perusahaan. Teknologi ini dapat mengurangi kecurangan absensi dan mempermudah proses rekapitulasi data, sehingga memberikan manfaat dalam pengelolaan sumber daya manusia dan administrasi.

1.7. Metode Dan Teknik Penelitian

1.7.1. Metode

Metode merupakan cara atau langkah-langkah sistematis yang digunakan untuk mencapai suatu tujuan dalam penelitian atau pengembangan. Salah satu metode yang digunakan adalah metode pengembangan SDLC model Waterfall, yang meliputi beberapa tahapan berurutan: analisis kebutuhan, desain, implementasi, dan pengujian.

1.7.2. Teknik Penelitian

a. Wawancara

Teknik pengumpulan data dengan berbicara langsung kepada narasumber untuk mendapatkan informasi mendalam.

b. Studi Literatur

Teknik pengumpulan data dari sumber tertulis seperti buku dan jurnal untuk memahami topik dan menemukan informasi relevan.

1.8. Literatur Review

No	Jurnal	Permasalahan	Metode	Kesimpulan
1.	Judul: Perbandingan Metode <i>Eigenface</i>, Fisherface, dan LBPH pada Sistem Pengenalan Wajah Peneliti: Qadrisa Mutiara Detila dan Eri Prasetyo Wibowo. Tahun: 2019	Tantangan dalam sistem pengenalan wajah disebabkan oleh variasi ekspresi wajah, penampilan, serta kondisi cahaya, jarak, dan kualitas kamera yang mempengaruhi akurasi pengenalan wajah.	Penelitian ini membandingkan tiga metode pengenalan wajah yaitu <i>Eigenface</i>, Fisherface, dan <i>Local Binary Patterns</i> (LBPH)	Metode LBPH merupakan metode yang paling akurat dan efisien dibandingkan dengan metode <i>Eigenface</i> dan Fisherface, dengan rata-rata akurasi LBPH sebesar 83%, sedangkan <i>Eigenface</i> 46% dan Fisherface 54%.
2.	Judul: Implementasi Metode <i>Local Binary Pattern Histogram</i> Dan Facial Landmark Pada	Beberapa perangkat belum memiliki komponen pengenalan wajah yang mampu mengenali orang asli yang sudah	Penelitian ini menerapkan metode pengembangan System Development Life Cycle (SDLC) dengan pendekatan	Sistem login dengan menggunakan pengenalan wajah dan pendeteksian kedipan mata telah berhasil dibangun dan dapat dijalankan dengan

No	Jurnal	Permasalahan	Metode	Kesimpulan
	Keamanan Akses Login Peneliti: Frindi Mangimbulu de, Pinrolinvic D.K. Manembu, dan Feisy Diane Kambey. Tahun: 2023	dikenali atau hanya foto saja, terutama pada perangkat dengan spesifikasi rendah. Penelitian ini bertujuan untuk merancang sistem keamanan dengan pengenalan wajah dan pendeteksi kedipan mata pada keamanan akses login ke suatu sistem pada perangkat berspesifikasi rendah.	Model Waterfall, serta menggunakan metode <i>Local Binary Pattern Histogram</i> (LBPH) untuk pengenalan wajah dan Facial Landmark untuk pendeteksi kedipan mata.	baik pada perangkat lawas atau perangkat dengan spesifikasi rendah.
3.	Judul: Desain dan Implementasi Sistem Absensi Mahasiswa Berdasarkan	Sistem absensi konvensional yang dilakukan dengan menandatangani lembar kertas memungkinkan	Menggunakan sistem pengenalan wajah berbasis desktop dan website dengan metode Haar-	Sistem absensi mahasiswa berdasarkan pola wajah telah berhasil dibangun dan diimplementasikan

No	Jurnal	Permasalahan	Metode	Kesimpulan
	Fitur Pengenalan Wajah dengan Menggunakan Metode Haar-Like Feature Peneliti: Evta Indra, M Diarmansyah Batubara, Muhammad Yasir, dan Sugandi Chau. Tahun: 2019	terjadinya kecurangan dan membutuhkan waktu lama untuk proses rekapitulasi data absensi.	Like Feature, serta Raspberry Pi sebagai media akses kontrol.	, yang mengurangi waktu rekapitulasi data absensi dari 6-8 hari menjadi 3-4 hari.
4.	Judul: Prototipe Pendeteksi dan Pengenalan Wajah Berbasis Web Menggunakan Algoritma	Sistem absensi yang biasanya menggunakan sidik jari menjadi kurang efektif selama pandemi Covid-19, karena alat absensi yang	Penelitian ini menggunakan analisis pendeteksi dan pengenalan wajah dengan pendekatan fitur algoritma <i>Local Binary Pattern</i>	Sistem absensi wajah mampu mendeteksi wajah dengan akurasi 100% dan mengenali wajah yang telah terdeteksi dengan akurasi 60%.

No	Jurnal	Permasalahan	Metode	Kesimpulan
	<i>Local Binary Pattern Histogram</i> untuk Absensi Peneliti: Ita Augustina Tarigan, Asep Kurniawan Tahun: 2022	terkontaminasi dapat menjadi media penyebaran virus. Diperlukan sistem absensi yang tidak memerlukan kontak langsung dengan alat.	<i>Histogram</i> (LBPH) untuk absensi berbasis web, dengan pengembangan perangkat lunak menggunakan metode prototipe.	Pendeteksian dan pengenalan wajah sangat bergantung pada faktor seperti jarak wajah dengan webcam, intensitas cahaya, dan penggunaan aksesoris pada wajah.
5.	Judul: Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OpenCv dan Dlib Python Peneliti: Afrizal Zein Tahun: 2018	Kecelakaan di Indonesia yang dipicu oleh mengantuk meningkat drastis, dengan data menunjukkan enam kasus kecelakaan setiap hari akibat drowsiness.	Penelitian menggunakan sistem komputer vision untuk mendeteksi kantuk pengemudi secara otomatis melalui pemantauan kamera web real-time, yang kemudian membunyikan alarm dan menghubungi	Detektor rasa kantuk yang dikembangkan dapat diandalkan untuk mendeteksi kondisi mengantuk pengemudi dengan akurasi di atas 90%, menggunakan teknik penglihatan komputer seperti deteksi landmark wajah dan aspek rasio mata.

No	Jurnal	Permasalahan	Metode	Kesimpulan
			keluarga jika pengemudi terlihat mengantuk. Metode ini melibatkan deteksi landmark wajah dan penghitungan aspek rasio mata.	
Penelitian yang diusulkan		Hasil Penelitian		
Rancang Bangun Aplikasi Aplikasi Absensi Pengenalan Wajah Menggunakan Algoritma <i>Local Binary Pattern Histogram</i> Dan Deteksi Kedipan Mata Berbasis Website		Absensi pengenalan wajah dengan LBPH dapat mengidentifikasi kehadiran pegawai hanya pada hari saat mereka melakukan absensi, sehingga mencegah kecurangan seperti absensi mandiri pada hari-hari lain. Deteksi kedipan mata dengan menggunakan facial landmark dari library dlib dapat memastikan bahwa pegawai yang terdeteksi adalah individu asli, bukan perwakilan yang mencoba menitipkan absensi. Selain itu, rekapitulasi absensi dapat langsung diketahui secara real-time dan diunduh dalam format CSV tanpa perlu menunggu waktu.		

Tabel 1. 1 Literatur Review

1.9. Sistematika Penulisan

Dalam penulisan ilmiah ini, sistematika penulisan terdiri dari berbagai bab pembahasan sebagai berikut :

BAB I PENDAHULUAN

Dalam BAB I dibahas berbagai hal yang berkaitan dengan masalah yang akan dibahas dalam penulisan ilmiah ini, diantaranya latar belakang masalah, identifikasi masalah, rumusan masalah, tujuan penelitian, batasan masalah, review penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Dalam BAB II ini akan menjelaskan kajian jurnal yang terkait, teori utama penelitian, teori pendukung penelitian yang dapat diperoleh dari berbagai sumber terpercaya seperti buku, jurnal prosiding maupun laporan.

BAB III RANCANGAN SISTEM INTERFACE

Dalam BAB III akan membahas tentang lingkungan pengembangan perangkat keras, alasan pemilihan interface yang digunakan, rangkaian atau diagram yang menjadi dasar dalam perancangan interface, dan garis besar alur kerja dari interface yang dibangun.

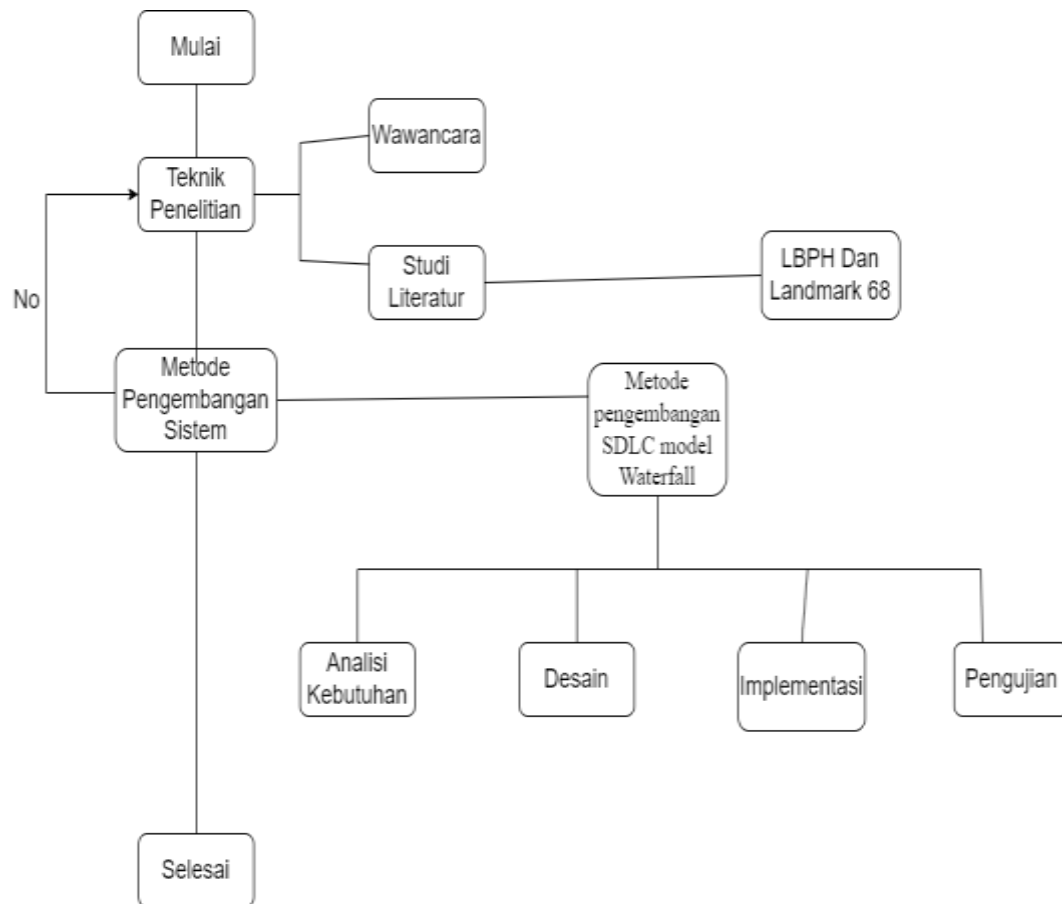
BAB IV IMPLEMENTASI SISTEM DAN PENGUJIAN

Dalam BAB IV akan dijelaskan mengenai hasil implementasi dari interface yang dikembangkan, pengujian secara simulasi, pengujian dalam situasi nyata, cara penggunaan interface yang telah dikembangkan, beserta hasil Analisa yang didapat dari interface yang telah dibuat.

BAB V PENUTUP

Dalam BAB V ini dijelaskan tentang kesimpulan dan saran-saran dari penulis tentang penulisan ilmiah ini.

1.10. Kerangka Berfikir

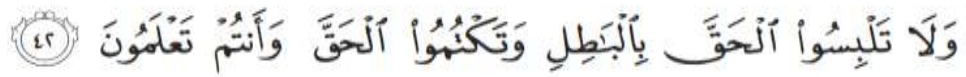


Gambar 1. 1 Kerangka Berfikir

BAB II

LANDASAN TEORI

2.1. Keterkaitan Dengan Al-Qur'an Surat Al-Baqarah (42) Ayat 2



Artinya : Janganlah kamu mencampuradukkan yang benar dengan yang salah, dan janganlah kamu menyembunyikan yang benar, padahal kamu mengetahuinya (Al – Baqarah : 42).

Surah Al-Baqarah (2:42) mengingatkan kita untuk tidak mencampur kebenaran dengan kebohongan. Dalam konteks absensi, berbohong mengenai kehadiran, seperti mengklaim hadir padahal tidak, merupakan contoh mencampurkan kebenaran (kehadiran) dengan kepalsuan (keterangan tidak hadir). Tindakan ini merugikan orang lain yang bekerja keras dan dapat menciptakan ketidakadilan dalam lingkungan kerja.

2.2. Citra Digital

Citra digital adalah representasi gambar yang dihasilkan oleh mesin melalui proses sampling dan kuantisasi. Sampling merujuk pada ukuran kotak-kotak yang tersusun dalam baris dan kolom, atau dengan kata lain, mencerminkan ukuran piksel (titik) pada gambar. Kuantisasi, di sisi lain, menggambarkan tingkat kecerahan yang dinyatakan dalam skala abu-abu (grayscale) sesuai dengan jumlah bit biner yang digunakan oleh mesin. Dengan kata lain, kuantisasi menunjukkan jumlah warna yang terdapat pada gambar. Ada berbagai metode untuk menyimpan citra digital di dalam memori, dan cara penyimpanan ini akan mempengaruhi jenis citra digital yang dihasilkan. Beberapa jenis citra digital yang umum digunakan meliputi citra biner, citra grayscale, dan citra warna (Yanu F et al., 2022).

1. Citra Biner (Monokrom): Hanya terdiri dari dua warna, yaitu hitam dan putih. Untuk menyimpan kedua warna ini, diperlukan 1 bit memori.
2. Citra Grayscale (Skala Keabuan): Jumlah warna yang ditampilkan bergantung pada jumlah bit yang dialokasikan dalam memori. Citra dengan

2 bit mampu menampilkan 4 warna, citra 3 bit menampilkan 8 warna, dan seterusnya. Semakin besar alokasi bit untuk warna, semakin halus gradasi warna yang bisa dihasilkan.

3. Citra Warna (True Color): Setiap piksel dalam citra warna terbentuk dari kombinasi tiga warna dasar, yaitu merah (Red), hijau (Green), dan biru (Blue) atau RGB. Masing-masing warna dasar disimpan menggunakan 8 bit atau 1 byte, yang artinya setiap warna memiliki gradasi hingga 255 tingkat. Kombinasi dari ketiga warna dasar ini menghasilkan lebih dari 16 juta variasi warna ($28 \times 28 \times 28 = 244$). Oleh sebab itu, format ini disebut sebagai true color karena jumlah warnanya yang sangat banyak sehingga hampir mencakup seluruh warna yang ada di alam.

2.3. *Computer vision*

Computer vision atau visi komputer adalah teknologi yang memadukan kamera, sistem komputasi, perangkat lunak, serta kecerdasan buatan (AI) untuk memungkinkan sistem mengamati dan mengenali objek. Teknologi ini diterapkan di berbagai lingkungan, misalnya untuk mengenali objek dan individu dengan cepat, menganalisis demografi audiens, memeriksa kualitas produksi, melacak gerakan, dan masih banyak aplikasi lain yang bisa dikembangkan (Dona Marleny, 2021).

Bidang visi komputer adalah cabang ilmu multidisiplin yang berfokus pada cara komputer memahami gambar atau video digital pada tingkat yang lebih tinggi. Dari sudut pandang rekayasa, visi komputer berusaha mengotomatiskan berbagai tugas yang biasanya dilakukan oleh penglihatan manusia. Ini mencakup metode pengambilan, pemrosesan, analisis, dan pemahaman citra digital, serta mengekstraksi data yang kompleks dari dunia nyata untuk menghasilkan informasi dalam bentuk angka atau simbol, seperti membuat keputusan. Dalam hal ini, proses transformasi visual gambar menjadi deskripsi yang mampu berinteraksi dengan pemikiran sistem menjadi penting untuk tindakan yang tepat. Pemahaman ini dicapai dengan menguraikan informasi simbolis dari data citra menggunakan model

yang dikembangkan melalui berbagai disiplin ilmu, seperti geometri, fisika, statistik, teori pembelajaran, serta algoritma komputasi (Dona Marleny, 2021).

Sub-bidang dalam visi komputer meliputi rekonstruksi adegan, deteksi peristiwa, pelacakan video, pengenalan objek, estimasi gambar 3D, pembelajaran, pengindeksan, estimasi gerakan, dan pemulihan citra. Salah satu aspek penting dari visi komputer adalah bagaimana komputer dapat melihat gambar atau citra. Proses ini melibatkan berbagai teknik pemrosesan citra yang memungkinkan komputer untuk mengidentifikasi dan memahami pola-pola visual yang terkandung dalam gambar, seperti bentuk, warna, tekstur, dan gerakan. Melalui teknik-teknik tersebut, sistem komputer dapat mengenali objek, memprediksi pergerakan, atau bahkan mengidentifikasi emosi atau ekspresi wajah dalam sebuah gambar atau video (Dona Marleny, 2021).



Yang dilihat Manusia

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

Yang dilihat computer

Gambar 2. 1 Ilustrasi Computer vision

(Sumber: Dona Marleny, 2021)

2.4. *Face Recognition*

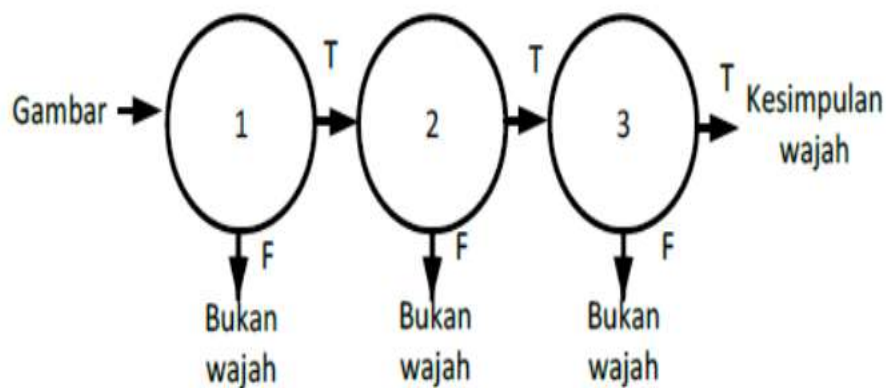
Face Recognition adalah teknologi yang memungkinkan identifikasi wajah seseorang melalui gambar digital. Teknologi ini berfungsi sebagai aplikasi biometrik yang dapat mengidentifikasi atau memverifikasi wajah dengan cara yang unik. Proses biometrik ini menganalisis dan membandingkan pola berdasarkan bentuk wajah. Meskipun umumnya digunakan untuk tujuan keamanan, teknologi ini juga banyak diterapkan di berbagai bidang lainnya. *Face Recognition* tidak memerlukan kontak fisik. Teknik ini sangat berguna dalam pengumpulan data waktu dan kehadiran. Namun, ada beberapa kelemahan, seperti ketidakmampuan mengenali wajah dalam kondisi pencahayaan yang buruk. Selain itu, variasi ekspresi wajah dapat mengurangi efektivitas identifikasi (Clara, 2022). Secara umum, proses *Face Recognition* terdiri dari:

1. Modul akuisisi: Merupakan tahap pengumpulan data untuk proses pengenalan wajah, yang bisa diperoleh melalui citra digital atau kamera.
2. Modul pre-processing: Ini adalah tahap yang melibatkan penyesuaian citra input, seperti normalisasi ukuran citra, pemfilteran median untuk mengurangi noise akibat pergeseran frame atau kamera, serta penyesuaian histogram untuk meningkatkan kualitas citra tanpa kehilangan informasi penting. Selain itu, dilakukan pemfilteran high pass untuk mendapatkan tepi citra, penghapusan latar belakang agar hanya wajah yang diproses, dan konversi ke skala abu-abu dari citra RGB. Proses preprocessing bertujuan untuk mengatasi masalah yang mungkin muncul selama proses pengenalan wajah.
3. Modul ekstraksi fitur: Bertujuan untuk mendapatkan bagian paling signifikan yang berfungsi sebagai vektor unik yang merepresentasikan wajah.
4. Modul klasifikasi: Dalam tahap ini, pola-pola yang dihasilkan dibandingkan dengan fitur-fitur yang tersimpan dalam database untuk menentukan apakah citra wajah dapat dikenali.

5. Training set: Modul ini digunakan selama fase pelatihan (training), proses identifikasi, dan semakin kompleks serta seringnya pengenalan wajah dilakukan, maka hasilnya akan semakin baik.

2.5. *Haar Cascade Clasifier*

Istilah Haar pada dasarnya merujuk pada sebuah fungsi matematika yang dikenal sebagai Haar Wavelet, berbentuk kotak. Pada awalnya, pengolahan gambar hanya berdasarkan nilai RGB setiap piksel, namun pendekatan ini terbukti kurang efisien. Oleh karena itu, peneliti Viola dan Jones mengembangkan sebuah metode baru untuk mengolah gambar yang disebut Haar-Like feature. Haar-Like feature bekerja dengan memproses gambar dalam bentuk kotak-kotak kecil, di mana masing-masing kotak berisi beberapa piksel. Setiap kotak diproses untuk menghasilkan perbedaan nilai yang menunjukkan area gelap dan terang. Nilai-nilai ini digunakan sebagai dasar dalam pemrosesan gambar. Cara menghitung nilai fitur ini adalah dengan mengurangi nilai piksel di area putih dengan nilai piksel di area hitam. Untuk mempermudah penghitungan, algoritma Haar memanfaatkan Integral Image, yaitu citra di mana setiap nilai piksel merupakan hasil penjumlahan dari piksel kiri atas hingga kanan bawah. Saat menggunakan metode *Haar Cascade*, salah satu jenis gambar yang dapat diproses adalah citra grayscale. *Haar Cascade classifier* merupakan langkah untuk mencapai hasil yang lebih akurat dengan cara menghitung banyak dan berulang kali nilai Haar Feature (Ludia Ramadini & Haryatmi, 2022).



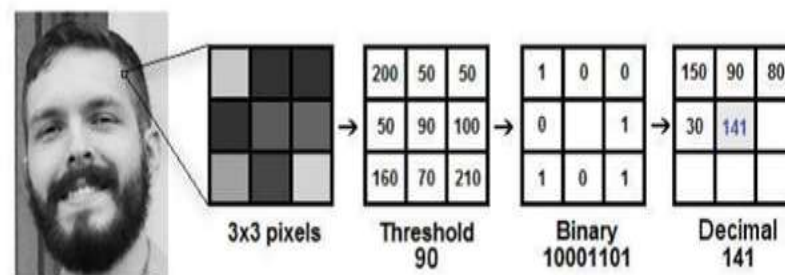
Gambar 2. 2 *Haar Cascade Classier*

(Sumber: Febrin Ludia Ramadini, Emy Haryatmi, 2022)

Haar Cascade dapat diprogram untuk mendeteksi berbagai objek dengan cara mengidentifikasi area wajah yang memiliki kemungkinan tertinggi. Wajah tersebut memiliki warna kulit yang khas, sehingga segmentasi dilakukan berdasarkan warna piksel kulit. Setelah itu, dilakukan validasi menggunakan *Haar Cascade classifier*. Jika piksel yang divalidasi cocok dengan struktur geometris wajah, maka perangkat akan menganggapnya sebagai wajah yang dicari, namun jika tidak cocok, perangkat akan mengabaikannya (Ludia Ramadini & Haryatmi, 2022).

2.6. Local Binary Pattern Histogram (LBPH)

Pengenalan wajah merupakan tahap lanjutan dari sistem pendeteksian wajah, di mana prosesnya dilakukan menggunakan metode LBPH. Citra wajah yang diambil secara real-time melalui kamera laptop akan dibandingkan serta dicocokkan dengan citra wajah yang sudah tersimpan di database melalui histogram yang telah diekstraksi. *Local Binary Pattern Histogram* (LBPH) berfungsi untuk mengklasifikasikan citra dengan menggabungkan histogram, dan merupakan salah satu teknik terbaru dari metode LBP yang mampu meningkatkan performa pengenalan wajah. LBP sendiri awalnya dirancang untuk mengenali tekstur. Ide utama dari Local Binary Pattern adalah merangkum struktur lokal dalam gambar dengan membandingkan setiap piksel dengan sekitarnya, lalu hasilnya diperlakukan sebagai angka biner. Angka biner ini kemudian diubah menjadi angka desimal yang akan menjadi nilai baru untuk piksel tengah. Angka desimal tersebut disebut nilai piksel LBPH, dengan rentang nilai 0-255 (Ludia Ramadini & Haryatmi, 2022).



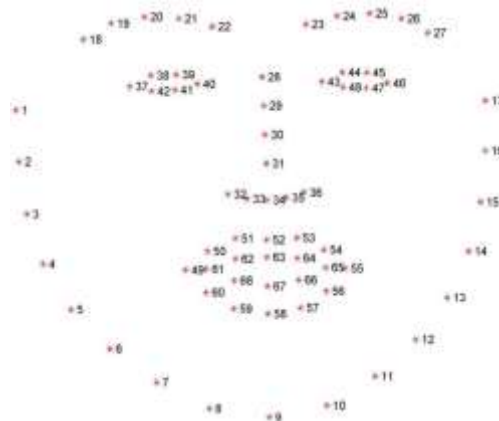
Gambar 2. 3 LBPH
(Sumber: Ahmad Jalaluddin, 2023)

Pada ilustrasi cara kerja LBPH dijelaskan bahwa nilai piksel di tengah didapat dengan membandingkan intensitas gambar yang diambil oleh kamera dengan intensitas piksel di sekitarnya. Nilai piksel di tengah ini merupakan ambang batas dari delapan piksel lainnya. Dalam sebuah matriks, nilai biner di tengah akan dibandingkan dengan nilai di sekelilingnya. Jika nilai di tengah lebih besar daripada nilai di sekitarnya, maka piksel di sekelilingnya diberi nilai 1, dan jika lebih kecil, nilainya menjadi 0. Setelah itu, nilai histogram dihitung untuk membandingkan dan mencocokkan wajah yang ditangkap kamera dengan yang ada di database. Berikut adalah persamaan untuk menghitung histogramnya (Ludia Ramadini & Haryatmi, 2022).

$$D = \sqrt{\sum_i^n = 1 (hist\ 1_i - hist\ 2_i)^2}$$

2.7. Dlib

Detektor landmark wajah yang telah dilatih sebelumnya dalam pustaka dlib digunakan untuk mengidentifikasi posisi 68 koordinat (x, y) yang mencerminkan struktur wajah. Indeks dari 68 koordinat ini dapat dilihat dalam gambar berikut:

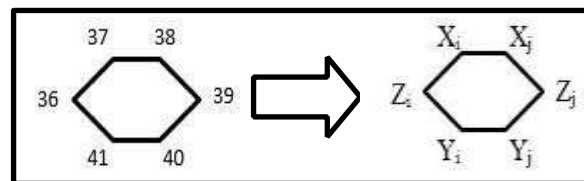


Gambar 2. 4 Dlib

(Sumber: Zein et al., 2018)

Notasi ini merupakan bagian dari dataset iBUG 300-W yang berisi 68 titik landmark wajah, di mana prediktor landmark dlib dilatih. Penting untuk dicatat bahwa ada pendeteksi wajah lainnya, termasuk model dengan 194 titik yang bisa

dilatih menggunakan dataset HELEN. Terlepas dari dataset yang dipakai, kerangka kerja dlib dapat digunakan untuk melatih prediktor bentuk pada data pelatihan yang Anda miliki berguna jika Anda ingin mengembangkan pendeteksi landmark wajah atau prediktor bentuk khusus sendiri. Detektor landmark wajah dari dlib akan menghasilkan objek berbentuk yang berisi 68 koordinat (x, y) yang menunjukkan lokasi landmark pada wajah. Dengan fungsi `shape_to_np`, objek ini dapat diubah menjadi array NumPy, sehingga lebih kompatibel dengan kode Python. Setelah area mata dikenali, kita bisa menghitung eye aspect ratio (rasio aspek mata) untuk menentukan apakah mata dalam keadaan tertutup atau terbuka (Zein et al., 2018).



Gambar 2. 5 Analog Mata

Menurut (Awasekar, Ravi, Doke dan Shaikh, 2018), pengukuran mata sebagai berikut:

$$EAR = \frac{|(X_i + X_j) - (Y_i + Y_j)|}{2 * |Z_i - Z_j|}$$

2.8. Python

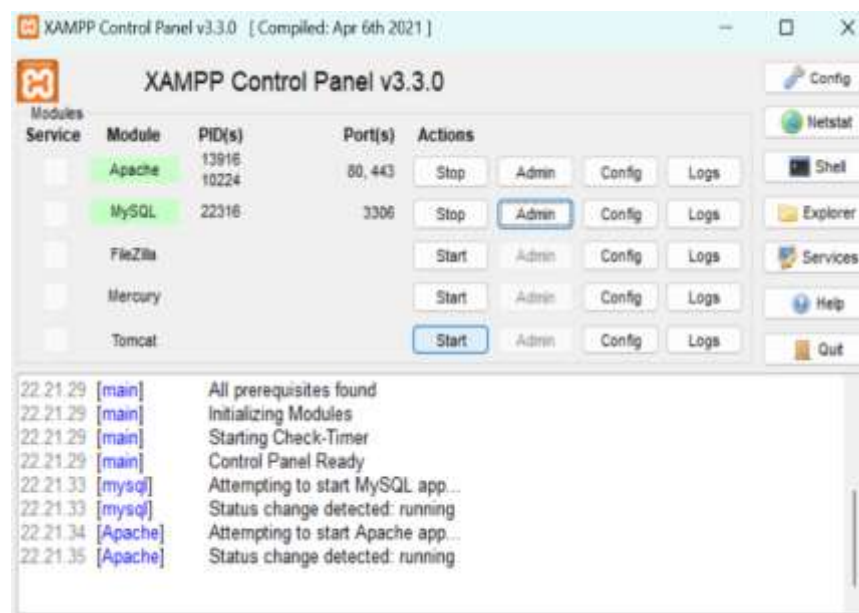


Gambar 2. 6 Python
(Sumber: <https://www.python.org>)

Python merupakan bahasa pemrograman tingkat tinggi yang berorientasi objek, yang dikembangkan oleh Guido Van Rossum. Desainnya fokus pada keterbacaan kode, dan sintaksnya memungkinkan para pengembang untuk menyampaikan ide dalam jumlah baris yang lebih sedikit dibandingkan dengan bahasa pemrograman lain seperti C. Struktur bahasa ini memudahkan pengguna untuk membuat program yang mudah dipahami, baik untuk proyek kecil maupun besar. Salah satu fitur utama Python adalah kemampuannya untuk mendukung berbagai paradigma pemrograman, termasuk pemrograman berorientasi objek dan prosedural imperatif. Selain itu, Python memiliki tipe data dinamis dan pengelolaan memori otomatis, serta dilengkapi dengan berbagai pustaka yang luas (Diana, 2021).

2.9. Tools Aplikasi

2.9.1. Xampp



Gambar 2. 7 Xampp

(Sumber: <https://www.researchgate.net/>)

XAMPP adalah perangkat lunak sumber terbuka yang berfungsi sebagai web server dan mencakup berbagai program. XAMPP kompatibel dengan berbagai sistem operasi, termasuk Linux, Windows, dan macOS. Software ini berfungsi sebagai server lokal atau localhost, yang di dalamnya terdapat program Apache,

MySQL, dan PHP. Menggunakan XAMPP sebagai database adalah pilihan karena XAMPP menyediakan MySQL dengan antarmuka yang lebih mudah digunakan. Tool-tool yang tersedia cukup lengkap untuk memenuhi kebutuhan perancangan database. Selain itu, XAMPP juga merupakan aplikasi gratis yang tidak memerlukan biaya (Dimasmewa Widyatmoko, 2022).

Sebagaimana dijelaskan oleh (Smith dan Johnson, 2020), XAMPP menyediakan lingkungan yang ideal untuk pengembangan dan pengujian aplikasi web, membuatnya menjadi pilihan populer di kalangan pengembang pemula maupun berpengalaman. Dengan berbagai fitur dan kemudahan penggunaannya, XAMPP tetap menjadi alat yang sangat berharga dalam dunia pengembangan web.

2.9.2. Mysql



Gambar 2. 8 Mysql

(Sumber: Dimasmewa Widyatmoko, 2022)

MySQL adalah sistem manajemen basis data (DBMS) yang memanfaatkan perintah dasar SQL (Structured Query Language). Ini adalah DBMS sumber terbuka yang memiliki dua jenis lisensi: perangkat lunak bebas dan shareware, di mana shareware memiliki batasan penggunaan. Server database MySQL tersedia

secara gratis di bawah lisensi GNU dan GPL (General Public License). Dalam penggunaannya, MySQL sering digunakan untuk berbagai aplikasi web, mulai dari sistem manajemen konten hingga aplikasi bisnis yang berbasis data. Kelebihan MySQL terletak pada kemampuannya untuk menangani volume data besar dengan performa yang cepat dan efisien. MySQL juga dilengkapi dengan berbagai fitur, termasuk dukungan untuk kunci asing dan kemampuan replikasi yang memudahkan pengelolaan basis data terdistribusi (Dimasmewa Widyatmoko, 2022).

2.9.3. Apache Websiteserver

Apache Website Server, yang sering dikenal dengan sebutan Apache, adalah salah satu server Website yang paling populer dan banyak digunakan di seluruh dunia. Fungsi utamanya adalah untuk mengirimkan halaman-halaman Website kepada pengguna melalui protokol HTTP dan HTTPS. Server ini dikembangkan dan dikelola oleh Apache Software Foundation, menjadikannya sebagai pilihan utama bagi penyedia layanan hosting situs Web. Keunggulan Apache terletak pada fleksibilitasnya, keandalannya, serta kemampuannya untuk beroperasi di berbagai sistem operasi seperti UNIX, Linux, dan Windows. Salah satu fitur utama Apache adalah kemampuannya untuk mengonfigurasi berbagai modul tambahan, yang mendukung fitur seperti URL rewriting, autentikasi pengguna, serta enkripsi SSL/TLS. Server ini juga mendukung berbagai bahasa pemrograman, termasuk PHP, Python, dan Perl, yang memberikan fleksibilitas luar biasa dalam pengembangan aplikasi Website yang kompleks dan dinamis. Apache dikenal luas karena kemampuannya dalam menangani lalu lintas Website yang tinggi dan stabil, sehingga menjadi pilihan ideal untuk situs-situs dengan jumlah pengunjung yang besar. Selain itu, Apache dilengkapi dengan berbagai fitur keamanan yang canggih, membantu melindungi situs Website dari serangan siber yang berbahaya, seperti DDoS dan brute force.

Seperti yang dinyatakan oleh Kar (2019, 30-35), Apache adalah server Website yang paling banyak digunakan karena kesederhanaannya, kehandalan, dan fleksibilitas yang ditawarkan. Modul-modul yang dapat diintegrasikan membuatnya sangat cocok untuk berbagai kebutuhan hosting dan pengembangan

aplikasi Website. Dengan dukungan dari komunitas yang besar dan pembaruan yang terus dilakukan, Apache tetap menjadi pilihan yang relevan dan terpercaya di dunia pengembangan Website modern, beradaptasi dengan perubahan dan kebutuhan teknologi yang terus berkembang.

2.9.4. Django Framework



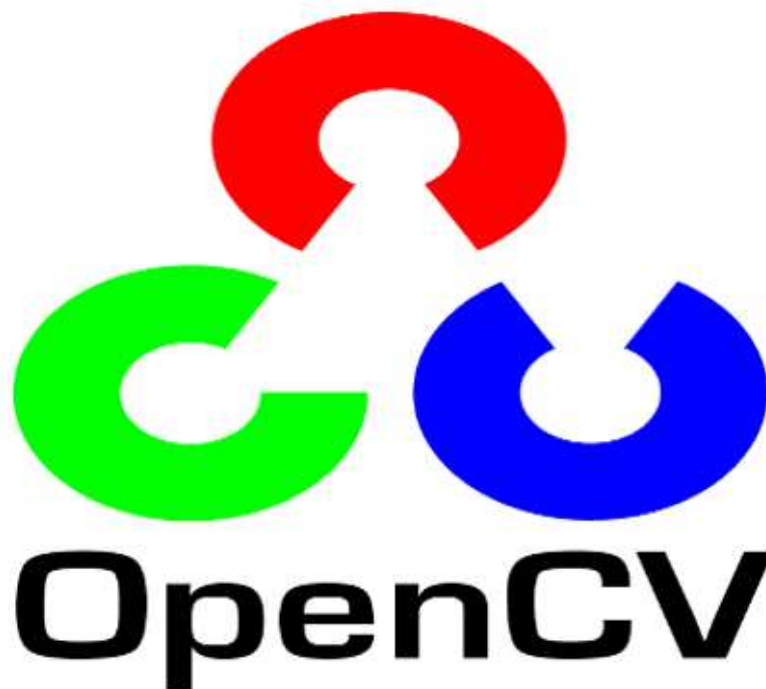
Gambar 2. 9 Django

(Sumber: <https://www.djangoproject.com/>)

Django adalah sebuah framework yang dirancang untuk membangun aplikasi web dinamis menggunakan bahasa pemrograman Python. Django menerapkan konsep *Model View Controller* (MVC). Salah satu keunggulannya adalah adanya panel admin otomatis yang memudahkan pengelolaan halaman admin, serta menyediakan pustaka yang mendukung otomatisasi pembuatan fungsi baru tanpa perlu menulis kode secara manual. Selain itu, Django menggunakan Object Relation Mapper (ORM) yang memudahkan pengembang untuk berinteraksi dengan database dan secara otomatis mengelola transfer data ke dalam database (Nofear & Pratama, 2019). Model dalam Django berfungsi sebagai lapisan akses

data, mencakup definisi tentang struktur data, cara akses, validasi, dan sebagainya. Template berperan dalam lapisan presentasi, menentukan bagaimana tampilan web disajikan kepada pengguna. View mengurus logika aplikasi, bertindak sebagai penghubung antara model dan template, serta mengelola proses logika untuk mengakses model dan template (Dwi Kristianto, 2019).

2.9.5. OpenCV



Gambar 2. 10 OpenCv

(Sumber: Ahmad Jalaludin, 2023)

OpenCV merupakan sebuah pustaka open source yang sangat populer di bidang *Computer vision* dan sering digunakan untuk pengolahan serta analisis gambar. Pustaka ini memiliki lebih dari 2500 algoritma yang dioptimalkan untuk berbagai keperluan, seperti manipulasi, pengolahan, dan pengeditan gambar. OpenCV mendukung sejumlah bahasa pemrograman, termasuk C, C++, Python, Java, dan PHP. Beberapa penerapan OpenCV meliputi Interaksi Manusia dan

Komputer (HCI), pengenalan serta identifikasi objek, deteksi dan pengenalan wajah, pengenalan gerakan, pelacakan objek bergerak, pemrosesan gambar, serta robotika mobile (Wiratna Eka & Sahputro Aghil, 2021).

Seperti yang dikemukakan oleh (Zhang et al., 2020), OpenCV memungkinkan pengembang untuk dengan mudah mengimplementasikan algoritma canggih dalam aplikasi real-time, serta menyediakan fungsionalitas yang luas untuk berbagai tugas pengolahan gambar. Dengan kemampuannya yang luas, OpenCV terus menjadi alat utama dalam bidang *Computer vision* dan pengembangan aplikasi berbasis gambar.

2.9.6. PhpMyAdmin



Gambar 2. 11 PHP MyAdmin

(Sumber: <https://commons.wikimedia.org/>)

PhpMyAdmin adalah perangkat lunak open-source yang banyak digunakan untuk manajemen basis data MySQL melalui antarmuka web yang intuitif. Dikembangkan menggunakan bahasa pemrograman PHP, phpMyAdmin dapat diakses baik melalui jaringan lokal maupun internet, serta memudahkan berbagai operasi administrasi database. Sebagai alat administrasi, phpMyAdmin memungkinkan pengguna untuk melaksanakan berbagai tugas tanpa perlu menulis perintah SQL secara langsung, sehingga sangat cocok untuk pengguna yang tidak memiliki banyak pengetahuan tentang pemrograman atau manajemen database tingkat lanjut.

MySQL berperan sebagai sistem manajemen basis data relasional (RDBMS) yang berfungsi untuk menyimpan data dan menyediakan alat-alat untuk mengelola basis data tersebut. Sebagai perangkat lunak server, MySQL memungkinkan aplikasi untuk menyimpan, mengambil, dan mengelola data secara efisien. Sebaliknya, phpMyAdmin berfungsi sebagai antarmuka berbasis web yang mempermudah penggunaan MySQL. Tanpa phpMyAdmin, administrator harus menggunakan command line untuk mengeksekusi perintah SQL dalam pengelolaan basis data. Dengan phpMyAdmin, proses ini menjadi lebih sederhana melalui antarmuka grafis yang ramah pengguna, yang membuat pengelolaan database menjadi lebih mudah, terutama bagi mereka yang tidak memiliki keahlian teknis yang mendalam dengan MySQL. Dengan demikian, perbedaan inti antara keduanya adalah bahwa MySQL adalah sistem manajemen basis data yang menyimpan data, sedangkan phpMyAdmin adalah alat yang memfasilitasi operasi MySQL dengan menyediakan cara yang mudah dan intuitif untuk mengelola basis data melalui browser (Dionisa Dero, 2019).

Menurut (Alvarez et al., 2021), phpMyAdmin merupakan alat yang sangat penting bagi pengembang dan administrator database, karena memberikan cara yang sederhana dan efisien untuk mengelola MySQL dan MariaDB tanpa harus bergantung pada baris perintah. Dengan komunitas yang besar dan dukungan yang aktif, phpMyAdmin terus berkembang dan menjadi salah satu alat favorit di kalangan pengembang web di seluruh dunia.

2.9.7. Visual Studio Code



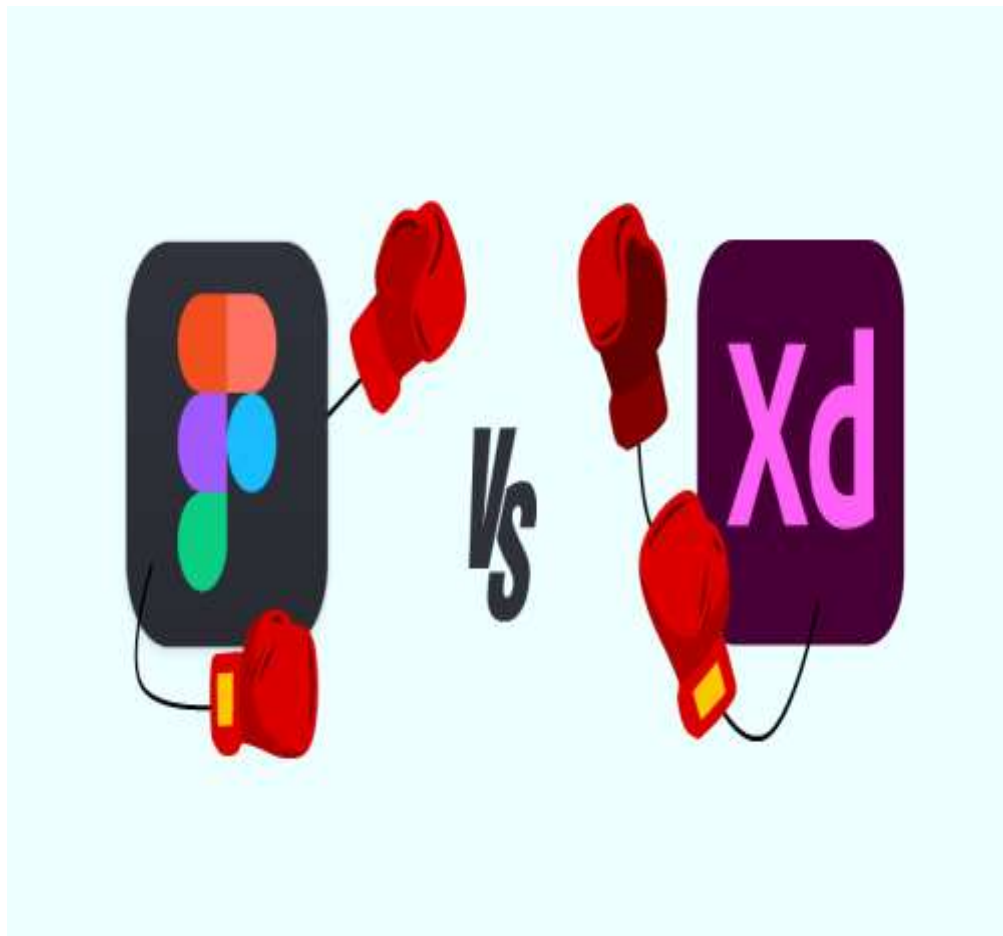
Gambar 2. 12 Visual Studio Code

(Sumber: <https://northsoft.co/>)

Visual Studio Code (VS Code) adalah editor teks yang ringan namun kuat, dikembangkan oleh Microsoft untuk berbagai platform, termasuk Linux, Mac, dan Windows. Editor ini mendukung secara langsung bahasa pemrograman seperti JavaScript, Typescript, Node.js, dan bahasa lainnya melalui plugin yang bisa diinstal melalui marketplace Visual Studio Code, seperti C++, C#, Python, Go, dan Java. VS Code memiliki berbagai fitur unggulan seperti Intellisense, integrasi Git, debugging, serta kemampuan untuk menambahkan ekstensi yang memperkaya fungsionalitas editor. Fitur-fitur tersebut terus berkembang seiring dengan pembaruan yang dirilis setiap bulannya, membedakan VS Code dari editor teks lainnya. Selain itu, VS Code bersifat open source, sehingga kode sumbernya dapat dilihat di Github, dan para pengguna pun dapat berkontribusi dalam pengembangannya. Hal ini menjadikan VS Code pilihan favorit bagi para

pengembang aplikasi. Meskipun VS Code dapat digunakan tanpa ekstensi tambahan jika alur kerja pemrograman yang dibutuhkan sudah didukung, beberapa fitur seperti snippets code dan peningkatan Intellisense memerlukan ekstensi untuk performa optimal. Banyak ekstensi VS Code dikembangkan oleh pihak ketiga yang juga berperan sebagai kontributor pengembangan editor ini (Wiratna Eka & Sahputro Aghil, 2021).

2.9.8. Figma Atau Adobe



Gambar 2. 13 Figma Atau Adobe

Figma dan Adobe XD adalah dua alat desain UI/UX yang banyak digunakan oleh desainer untuk membuat antarmuka pengguna (UI) dan pengalaman pengguna (UX) pada aplikasi dan situs web. Keduanya menawarkan berbagai fitur dan fungsionalitas untuk membuat desain prototipe interaktif, wireframes, dan mockup

yang realistis. Walaupun memiliki tujuan yang sama, kedua alat ini memiliki pendekatan dan kelebihan unik yang membedakan mereka satu sama lain.

Figma adalah alat desain berbasis cloud yang memungkinkan kolaborasi secara real-time. Karena berbasis web, Figma dapat diakses melalui browser tanpa memerlukan instalasi perangkat lunak khusus. Ini berarti desainer, pengembang, dan pemangku kepentingan lain dapat bekerja bersama-sama pada desain yang sama, dari mana saja dan kapan saja, selama mereka memiliki koneksi internet. Fitur kolaborasi langsung ini membuat Figma sangat berguna untuk tim yang bekerja dari jarak jauh atau yang memerlukan umpan balik dan pengeditan cepat dari berbagai pihak. Selain itu, Figma juga memiliki aplikasi desktop yang memungkinkan penggunaan offline dan sinkronisasi otomatis ketika kembali online.

Figma mendukung komponen dan gaya yang dapat digunakan kembali, memudahkan desainer untuk menjaga konsistensi dalam desain. Komponen seperti tombol, ikon, dan elemen UI lainnya dapat didefinisikan sekali dan digunakan berkali-kali di seluruh proyek. Jika komponen tersebut diubah, setiap instance-nya akan diperbarui secara otomatis, menghemat waktu dan usaha desainer. Selain itu, Figma memiliki fitur gaya yang memungkinkan pengaturan global untuk warna, teks, dan elemen lainnya. Hal ini sangat membantu ketika membuat dan memelihara desain sistem atau antarmuka yang konsisten di seluruh aplikasi.

Adobe XD, di sisi lain, adalah alat desain antarmuka pengguna dan prototipe yang dikembangkan oleh Adobe. Berbeda dengan Figma, Adobe XD adalah aplikasi desktop yang dapat diunduh dan diinstal di Windows dan macOS. Salah satu keunggulan utama Adobe XD adalah integrasinya yang mendalam dengan ekosistem Adobe, seperti Adobe Photoshop, Illustrator, dan Creative Cloud. Hal ini memudahkan desainer yang sudah menggunakan alat Adobe lainnya untuk mengimpor grafik, gambar, dan aset desain langsung ke dalam proyek Adobe XD. Selain itu, Adobe XD mendukung fitur "Repeat Grid" yang memungkinkan desainer untuk membuat dan mengatur elemen yang berulang seperti daftar dan galeri dengan lebih cepat dan mudah.

Adobe XD juga menawarkan fitur prototipe yang kuat, memungkinkan desainer untuk membuat interaksi dan transisi yang realistis antara layar. Dengan fitur prototipe ini, desainer dapat menghubungkan artboard, menambahkan animasi, dan menentukan aksi pengguna seperti klik, geser, atau hover untuk membuat simulasi pengalaman pengguna yang mendekati aplikasi nyata. Setelah prototipe selesai, desainer dapat membagikannya kepada klien atau pengembang melalui tautan yang dapat diakses secara online, memberikan cara yang mudah untuk mendapatkan umpan balik dan menguji desain.

Meskipun Adobe XD tidak mendukung kolaborasi real-time seperti Figma, Adobe telah menambahkan beberapa fitur kolaboratif, seperti Coediting dan Share for Review, yang memungkinkan beberapa pengguna bekerja pada file yang sama dan memberikan umpan balik. Selain itu, Adobe XD memiliki plugin dan integrasi dengan alat lain seperti Slack, Zeplin, dan Jira, yang memudahkan pengelolaan proyek dan komunikasi dalam tim. Plugin ini menambahkan berbagai fungsionalitas tambahan, seperti mengekspor desain ke format pengembangan, memeriksa tata letak responsif, atau menambahkan elemen desain pihak ketiga.

Secara keseluruhan, baik Figma maupun Adobe XD adalah alat desain yang sangat kuat dan memiliki kelebihan masing-masing. Figma unggul dalam kolaborasi real-time dan fleksibilitas akses berbasis cloud, membuatnya ideal untuk tim desainer yang tersebar secara geografis. Adobe XD menawarkan integrasi yang kuat dengan ekosistem Adobe dan fitur desain serta prototipe yang kuat, yang sangat berguna bagi desainer yang sudah terbiasa dengan produk Adobe. Pilihan antara keduanya biasanya bergantung pada preferensi pribadi, kebutuhan kolaborasi, dan ekosistem kerja yang digunakan oleh tim desain dan pengembangan.

Figma atau Adobe XD digunakan untuk membuat prototipe antarmuka pengguna (UI/UX) sebelum aplikasi diimplementasikan. Ini memastikan bahwa aplikasi memiliki desain yang ramah pengguna dan memenuhi kebutuhan pengguna.

2.9.9. Blackbox

Pengujian Black Box adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas aplikasi tanpa mengetahui struktur internal atau kode sumbernya. Dalam pengujian ini, penguji hanya memeriksa input dan output aplikasi untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditentukan.

1. Karakteristik pengujian blackbox
 - a. Berfokus pada Fungsionalitas: Pengujian ini dilakukan berdasarkan spesifikasi dan kebutuhan sistem. Penguji hanya memeriksa apakah sistem memberikan output yang benar untuk berbagai input yang diberikan, tanpa memeriksa bagaimana proses atau logika internal di dalam aplikasi bekerja.
 - b. Tidak Memerlukan Pengetahuan Kode Sumber: Penguji tidak perlu mengetahui atau memahami struktur kode sumber, algoritma, atau logika internal dari aplikasi. Fokus utama adalah pada interaksi pengguna dengan sistem dan bagaimana aplikasi merespons masukan.
 - c. Pengujian Fungsionalitas: Memastikan bahwa semua fitur berfungsi sesuai dengan persyaratan.
 - d. Pengujian Validasi Input: Memeriksa bagaimana aplikasi menangani input yang valid maupun tidak valid.
 - e. Pengujian Keandalan: Memeriksa seberapa baik sistem dapat menangani beban atau penggunaan berulang.
 - f. Pengujian Kompatibilitas: Memeriksa apakah aplikasi berfungsi di berbagai platform atau lingkungan yang berbeda.
2. Kelebihan Pengujian blackbox
 - a. Sederhana: Pengujian ini relatif mudah dilakukan karena hanya berfokus pada input dan output, tanpa memerlukan pemahaman tentang kode atau logika internal.
 - b. Perspektif Pengguna: Pengujian Black Box memberikan gambaran tentang bagaimana aplikasi akan berfungsi dari perspektif pengguna akhir.

- c. Komprehensif: Memungkinkan pengujian terhadap berbagai kondisi input, termasuk batasan dan skenario ekstrem.
- 3. Kekurangan Pengujian blackbox
 - a. Terbatas pada Fungsionalitas Eksternal: Tidak dapat digunakan untuk menemukan kesalahan yang terjadi di dalam kode atau logika internal sistem.
 - b. Keterbatasan Skenario Uji: Pengujian ini mungkin tidak mencakup semua kemungkinan skenario karena penguji hanya melihat aplikasi dari luar.
- 4. Contoh pengujian blackbox

Misalkan, sebuah aplikasi memiliki fitur login yang meminta pengguna untuk memasukkan nama pengguna dan kata sandi. Pengujian Black Box akan memeriksa berbagai skenario input, seperti:

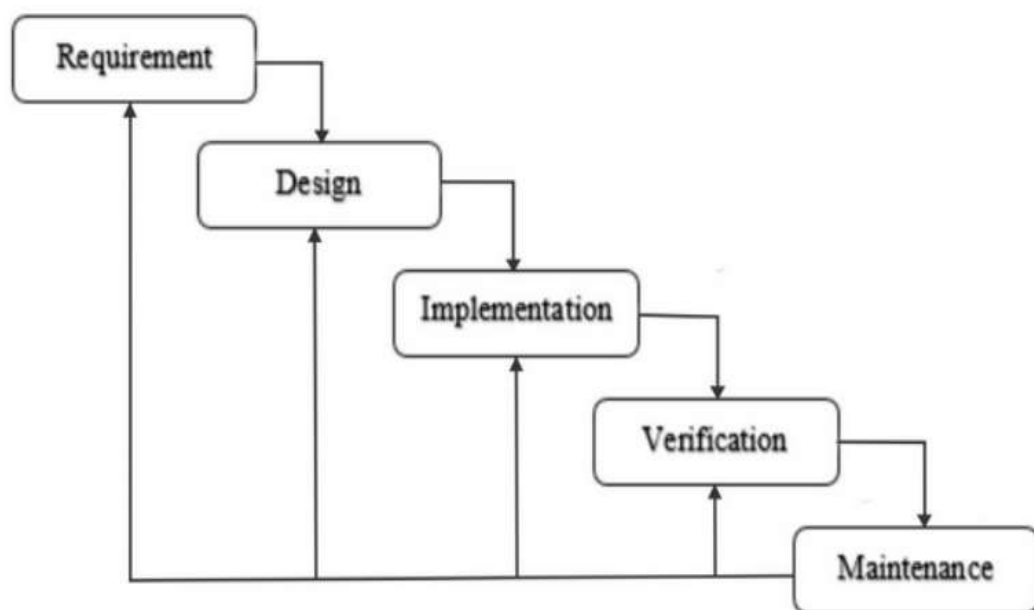
- a. Menggunakan nama pengguna dan kata sandi yang benar.
- b. Memasukkan kata sandi yang salah.
- c. Mengosongkan salah satu atau kedua kolom input.
- d. Memasukkan karakter khusus atau input yang tidak valid

2.9.10. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan representasi grafis yang digunakan untuk mendeskripsikan struktur sistem informasi secara konseptual. ERD menggambarkan hubungan antar entitas dalam sistem, data yang disimpan oleh setiap entitas, serta atribut yang dimilikinya. Dengan ERD, analisis sistem dapat dengan mudah memetakan keterkaitan data, memahami alur pengolahan data, dan mengidentifikasi peluang untuk mengoptimalkan pengelolaan data. ERD umumnya diterapkan dalam tahap desain sistem guna memastikan bahwa alur data dan struktur penyimpanan informasi sesuai dengan kebutuhan bisnis dan dapat mendukung pengambilan keputusan yang efektif. Selain itu, ERD juga membantu pengembang dalam mengintegrasikan data dari berbagai sumber secara konsisten. Hal ini menjadikan ERD sebagai alat penting dalam pengembangan sistem yang efisien dan terstruktur (Kadir, 2019).

2.10. Metode Waterfall

Metode Waterfall adalah salah satu pendekatan dalam pengembangan perangkat lunak yang juga dikenal sebagai *Software Development Life Cycle* (SDLC). Metode ini dianggap sebagai yang paling awal karena alurnya yang alami dengan pendekatan yang terstruktur, dimulai dari tahap pengumpulan kebutuhan sistem, dilanjutkan dengan analisis, desain atau perencanaan, implementasi atau pengkodean, pengujian/verifikasi, hingga tahap pemeliharaan (Santo, 2022).



Gambar 2. 14 Tahap Metode Waterfall
(Sumber: Santo, 2020)

1. Analisis Kebutuhan (Requirement)

Pada tahap ini, dilakukan proses pengumpulan dan pemahaman mengenai kebutuhan sistem dari pengguna untuk aplikasi yang akan dibuat. Informasi yang dikumpulkan kemudian diproses dan dianalisis, sehingga menghasilkan data lengkap terkait apa saja yang diperlukan oleh pengguna dalam pengembangan perangkat lunak tersebut.

2. Desain

Tahap ini melibatkan pengolahan informasi atau data yang diperoleh dari analisis kebutuhan, yang kemudian diterjemahkan ke dalam bentuk

rancangan pengembangan. Tujuan perancangan ini adalah untuk memberikan gambaran menyeluruh mengenai apa yang akan dilakukan pada tahap implementasi atau pengkodean. Selain itu, pada tahap ini pengembang juga mempersiapkan dan merencanakan kebutuhan perangkat keras untuk membangun arsitektur sistem perangkat lunak yang akan dikembangkan.

3. Implementasi

Di tahap ini, rancangan yang telah dibuat sebelumnya diwujudkan dalam bentuk modul-modul kode yang akan membentuk perangkat lunak sesuai dengan desain yang direncanakan. Setelah perangkat lunak selesai dibuat, dilakukan pengujian dan pengecekan untuk memastikan apakah perangkat lunak tersebut sudah memenuhi kriteria atau spesifikasi yang ditetapkan.

4. Verifikasi

Setelah perangkat lunak melewati tahap pengujian di fase implementasi, dilakukan proses integrasi dengan sistem. Selanjutnya, pemeriksaan dan pengujian perangkat lunak secara menyeluruh dilakukan untuk mendeteksi kemungkinan adanya kesalahan atau kegagalan dalam perangkat lunak tersebut.

5. Maintenance

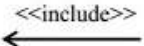
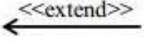
Pada tahap ini, perangkat lunak yang telah selesai akan melalui proses pemeliharaan. Pemeliharaan ini memungkinkan pengembang untuk memperbaiki kesalahan yang mungkin belum terdeteksi pada tahap sebelumnya. Proses ini mencakup perbaikan, peningkatan, serta penyesuaian sistem agar sesuai dengan kebutuhan pengguna.

2.11. *Use Case*

Use Case Diagram digunakan untuk menggambarkan interaksi antara pengguna (aktor) dan sistem yang sedang dalam proses perancangan. Aktor dalam diagram ini mewakili entitas eksternal yang berinteraksi dengan sistem, baik itu individu, perangkat keras, atau sistem lainnya. Di sisi lain, *Use Case* menjelaskan fungsi atau layanan yang ditawarkan oleh sistem kepada aktor. Garis yang menghubungkan aktor dan *Use Case* menggambarkan interaksi atau komunikasi di

antara keduanya. Diagram ini sering dipakai pada tahap awal pengembangan sistem untuk membantu dalam memahami kebutuhan fungsional sistem dari sudut pandang pengguna.

Seperti yang dijelaskan oleh (Irzan & Sutriyono, 2021), *Use Case Diagram* berfungsi untuk menentukan, menggambarkan, dan mendokumentasikan perilaku sistem secara menyeluruh. Diagram ini berguna bagi pengembang dalam menganalisis harapan pengguna dan bagaimana sistem harus berfungsi dalam berbagai kondisi. Oleh karena itu, diagram ini sangat penting dalam pengembangan sistem yang berorientasi pada kebutuhan pengguna, karena memberikan gambaran yang jelas tentang interaksi antara aktor dan sistem.

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Tabel 2. 1 *Use Case Diagram*

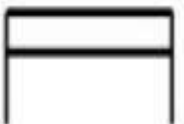
(Sumber: <https://www.dicoding.com>)

2.12. Activity Diagram

Activity Diagram digunakan untuk mendeskripsikan urutan aktivitas yang berlangsung dalam suatu proses atau operasi, serta hubungan antar aliran tersebut. Dalam pengembangan perangkat lunak, diagram ini sangat berharga dalam menggambarkan logika aliran proses dari awal hingga akhir, termasuk aktivitas, pengambilan keputusan, dan sinkronisasi. Menurut (Pratama & Febriani, 2021), diagram ini sangat membantu dalam memahami aktivitas yang muncul dalam suatu operasi, sehingga dapat diterapkan di berbagai sistem lainnya.

Seperti yang dijelaskan oleh (Siahaan et al., 2021), penilaian semantik otomatis pada diagram penting untuk meminimalkan risiko kesalahan dalam pemodelan sistem. Ini juga relevan untuk Activity Diagram, yang harus memastikan bahwa setiap aktivitas dan alur dalam diagram sesuai dengan logika bisnis yang diinginkan, tanpa ada langkah yang terlewat atau terulang. Penilaian semantik yang efektif memungkinkan pengembang mengidentifikasi ketidakjelasan dalam diagram. Dalam konteks sistem yang besar, otomatisasi penilaian semantik ini sangat penting untuk menjaga kualitas dan efisiensi dalam pemodelan.

Di samping itu, (Zulfa et al., 2020) menunjukkan pentingnya mengukur kesamaan struktur baik antar maupun dalam diagram. Pada Activity Diagram, pengukuran ini bisa dilakukan dengan menggunakan teknik Greedy Graph Edit Distance untuk memastikan setiap aktivitas dalam alur proses memiliki konsistensi struktural yang sesuai. Teknik ini memungkinkan para pengembang untuk melakukan perubahan pada diagram secara efisien tanpa kehilangan makna dari aktivitas atau proses yang ditampilkan. Penggunaan teknik ini juga mendukung kolaborasi antar anggota tim, di mana semua pihak dapat dengan mudah memahami dan menyetujui alur kerja yang telah dirancang. Activity Diagram berfungsi tidak hanya sebagai alat visualisasi tetapi juga sebagai sarana komunikasi yang efektif dalam pengembangan perangkat lunak. Dengan demikian, Activity Diagram berfungsi sebagai alat yang esensial dalam pengembangan perangkat lunak, karena tidak hanya memudahkan visualisasi alur proses, tetapi juga meningkatkan kolaborasi dan komunikasi antar anggota tim serta pemangku kepentingan.

Simbol	Keterangan
	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
	Asosiasi percabangan dimana jika ada pilihan
	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Tabel 2. 2 Activity Diagram
(Sumber: Elvan Adi Pratama, 2021)

2.13. Flowchart

Flowchart adalah alat yang sangat penting dalam pemrograman dan sistem informasi. Secara sederhana, *flowchart* berfungsi sebagai diagram yang menunjukkan alur proses atau langkah-langkah untuk menyelesaikan suatu masalah, sehingga sangat berguna dalam memvisualisasikan logika desain yang rumit. *Flowchart* bukan hanya berguna bagi para programmer, tetapi juga digunakan oleh berbagai profesional, termasuk analis sistem dan pengembang perangkat lunak, untuk merancang, menganalisis, dan menyelesaikan masalah.

Menurut (Santi, 2019), flowchart dapat dianggap sebagai representasi dari suatu algoritma. Algoritma merupakan rangkaian instruksi logis yang digunakan

untuk menyelesaikan masalah, dan flowchart menggambarannya dalam bentuk diagram dengan berbagai simbol yang mewakili keputusan, proses, input atau output, serta hubungan antar langkah. Ini menjadikan flowchart sebagai alat yang efektif untuk menyederhanakan konsep pemrograman dan membantu orang memahami cara kerja suatu sistem.

Dalam konteks flowchart, otomatisasi penilaian semantik sangat penting untuk memastikan bahwa logika yang digambarkan adalah akurat dan sesuai dengan tujuan sistem. Menurut (Fauzan et al., 2021), dalam “pendekatan otomatis dalam menilai diagram semantik diperlukan untuk meminimalkan risiko kesalahan dalam pemodelan. Hal ini juga relevan untuk flowchart, di mana alur proses yang keliru dapat menyebabkan kesalahan dalam implementasi system”. Penilaian semantik otomatis memungkinkan analisis yang lebih mendalam dan tepat mengenai logika alur proses yang ditunjukkan dalam flowchart. Sebagai contoh, jika terdapat redundansi atau inkonsistensi dalam alur logika, penilaian semantik otomatis dapat membantu mendeteksinya lebih awal dalam proses pengembangan.

Selain itu, seperti yang diungkapkan oleh (Zulfa et al., 2020) dalam penelitian mereka tentang Greedy Graph Edit Distance, pengukuran kesamaan struktur pada diagram sangat penting untuk menjaga konsistensi antara perencanaan dan pelaksanaan. Teknik ini dapat diterapkan pada flowchart untuk mengevaluasi kesamaan antara aliran proses yang digambarkan dan proses yang sebenarnya terjadi dalam sistem. Jika ada perubahan atau pembaruan pada flowchart, penting untuk memastikan bahwa perubahan tersebut tidak merusak alur logika yang telah direncanakan.

Pentingnya flowchart juga terletak pada kemampuannya untuk meningkatkan komunikasi di antara anggota tim. Dengan menggunakan flowchart, informasi dapat disampaikan dengan cara yang jelas dan ringkas, yang membantu semua pemangku kepentingan memahami langkah-langkah yang terlibat dalam suatu proses. Hal ini tidak hanya meningkatkan efisiensi, tetapi juga mengurangi kemungkinan terjadinya kesalahpahaman di antara anggota tim dan pengembang. Sebuah studi oleh (Goris et al., 2020) menunjukkan bahwa penggunaan flowchart

dalam perancangan sistem dapat mempercepat proses pengembangan serta mempermudah pelatihan bagi anggota tim baru, yang pada akhirnya berdampak positif terhadap kualitas dan ketepatan waktu penyelesaian proyek.

Simbol	Nama Simbol	Keterangan
	Garis Alir (Flow Line)	Simbol untuk menghubungkan antar proses atau antar simbol
	Connector	Simbol konektor untuk menyambung proses pada lembar kerja yang sama
	Offline Connector	Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda
	Process	Simbol pemrosesan yang terjadi pada sebuah alur kerja
	Manual	Simbol yang mendefinisikan proses yang dilakukan secara manual
	Decision	Simbol untuk memutuskan proses lanjutan dari kondisi tertentu
	Terminal	Simbol <i>start</i> atau <i>end</i> yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i>
	Input/Output	Simbol <i>input/output</i> yang mendefinisikan masukan dan keluaran proses
	Magnetic Tape	Simbol masukan atau keluaran dari atau ke sebuah pita magnetik
	Document	Simbol masukan atau keluaran dari atau ke sebuah dokumen
	Display	Simbol yang menyatakan piranti keluaran, seperti layar monitor, printer, dll
	Data Storage	Simbol database atau basis data
	Sub Program	Simbol yang menyatakan bagian dari program (sub program)

Tabel 2. 3 Flowchart
(Sumber: Santi, 2019)

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Tempat Dan Waktu Penelitian

3.1.1. Tempat

Penulis melakukan penelitian berdasarkan fokus penelitian fakta-fakta yang ada pada lapangan.

Tempat : Kelurahan Kertawinangun Kabupaten Cirebon

Alamat : Jl. Cideng Raya, Kertawinangun, Kec.Kedawung,
Kabupaten Cirebon, Jawa Barat, 45153

3.1.2. Waktu Penelitian

No.	Prosedur Penelitian	Bulan															
		Mei				Juni				Juli				Agustus			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Analisis Kebutuhan																
2.	Desain																
3.	Implementasi																
4.	Pengujian																

Tabel 3. 1 Waktu Penelitian

3.2. Analisis Sistem

Analisis ini bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada dalam sistem serta menentukan kebutuhan dari sistem yang dibangun.

3.2.1. Analisis Masalah

Proses absensi di Kelurahan Kertawinangun saat ini masih dilakukan secara manual, yaitu dengan menuliskan nama dan menandatangani buku kehadiran setiap hari kerja. Proses absensi yang menggunakan buku, seringkali terjadi kecurangan seperti pegawai yang menandatangani absensi secara mandiri pada hari lain meskipun sebenarnya tidak hadir, atau bahkan menitipkan absensi pada pegawai

lain dengan cara menuliskan nama dan meniru tanda tangan. Selain itu, proses rekapitulasi absensi memakan waktu yang lama karena melibatkan penghitungan manual dari buku absensi, seperti jumlah hari kerja dan kehadiran. Oleh karena itu penelitian ini merancang dan membangun aplikasi absensi berbasis web yang menggunakan teknologi pengenalan wajah dengan menggunakan *Local Binary Patterns Histograms* (LBPH) serta deteksi kedipan mata menggunakan facial landmark dlib, dan juga absensi pengenalan wajah untuk mempercepat proses rekapitulasi kehadiran dan hari kerja.

3.2.2. Analisis Kebutuhan Perangkat Keras Dan Perangkat Lunak

a) Analisis Kebutuhan Perangkat Keras

Spesifikasi standar yang dapat digunakan untuk membangun aplikasi ini dapat dilihat pada tabel dibawah ini:

Laptop

Processor AMD Ryzen 7 4700U 2.00 GHz

Memory 8 GB

SSD berkapasitas 512 GB

b) Analisis Kebutuhan Perangkat Lunak

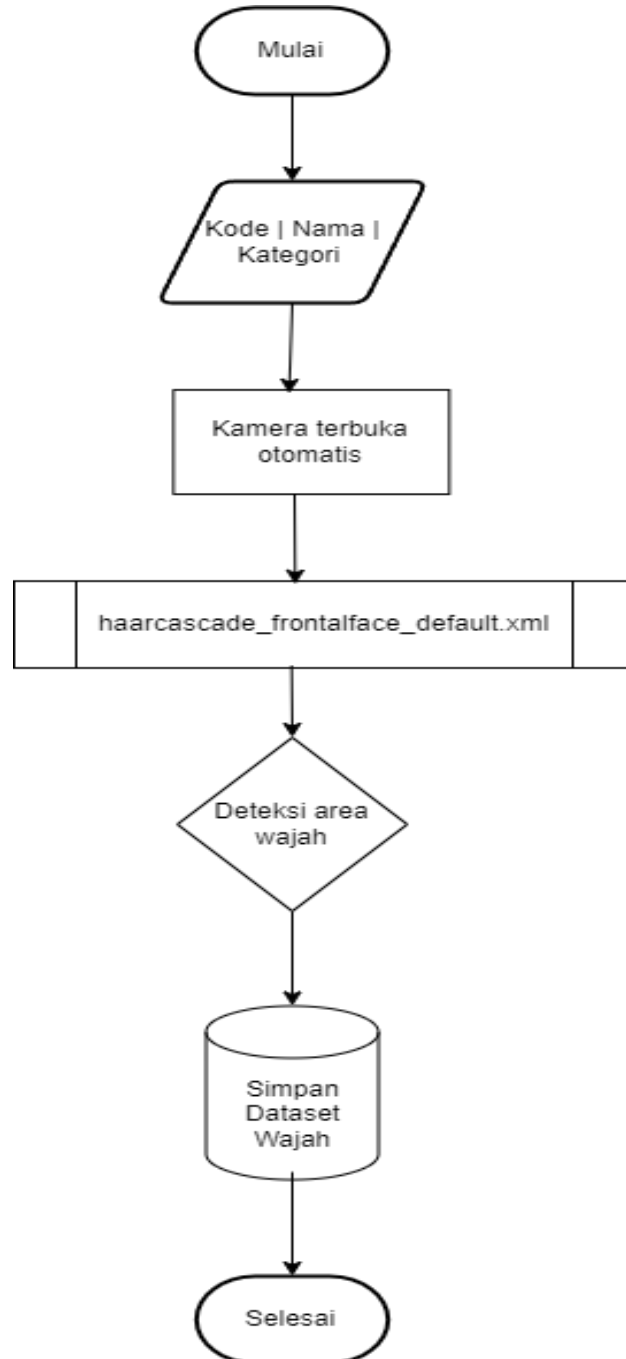
Spesifikasi standar yang dapat digunakan untuk membangun aplikasi ini dapat dilihat pada tabel dibawah ini:

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 11
2	Tool Pembangun	Visual Studio Code
3	Tool Design	Diagram.io & Figma
4	Bahasa Pemograman	Python
5	Database	MYSQL, dan XAMPP
6	Fullstack	Django
7	Library	Haarcascade, Dlib landmark 68

Tabel 3. 2 Tabel Kebutuhan Peragkat Lunak

3.2.3. Analisis Proses

a) Flowchart Penyimpanan Dataset Wajah



Gambar 3. 1 Simpan Data Wajah

- 1) Registrasi absensi menampilkan kode pendaftaran yang otomatis diperbarui berdasarkan pendaftaran sebelumnya. Pengguna diminta untuk memasukkan nama dan memilih kategori.

- 2) Kamera kemudian akan menampilkan gambar wajah yang berada di depan kamera.
- 3) `haarcascade_frontalface_default.xml` file XML yang berisi model deteksi classifier yang telah dilatih menggunakan algoritma *Haar Cascade* untuk mendeteksi wajah manusia dari tampak depan. File ini merupakan bagian dari pustaka OpenCV yang sering digunakan dalam deteksi wajah.
- 4) Wajah RGB yang terdeteksi diubah menjadi gambar grayscale dan dipotong menjadi ukuran 200x200 piksel.



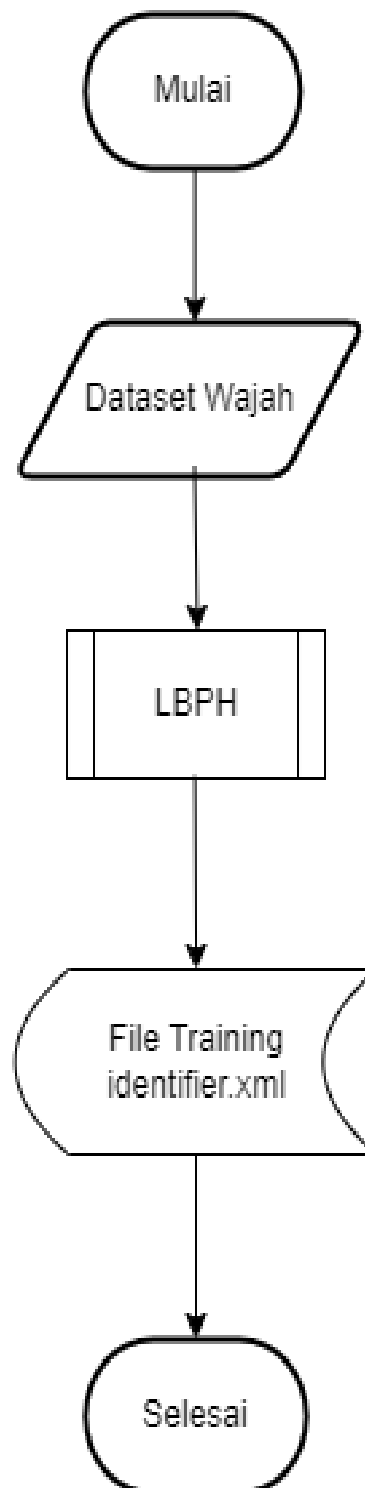
Gambar 3. 2 Foto Wajah RGB

- 5) Dataset wajah grayscale disimpan hingga mencapai 100 data wajah.



Gambar 3. 3 Foto Wajah Grayscale

b) Flowchart Proses *Local Binary Pattern Histogram*



Gambar 3. 4 Proses LBPH

- 1) Dataset grayscale wajah berukuran 200x200 piksel, yang disimpan sesuai dengan ID masing-masing, digunakan untuk melatih model pengenalan wajah. Dataset ini memungkinkan tengah mempelajari fitur unik dari setiap wajah, sehingga dapat mengenali individu.
- 2) Membagi citra wajah menjadi bagian-bagian kecil yang disebut dengan sel, dimana tiap sel berukuran 3x3 piksel.



Gambar 3. 5 Membagi 3x3 Pixel Wajah

- 3) Tentukan nilai pusat dari tiap sel yaitu piksel tengah untuk digunakan sebagai nilai ambang. Nilai tersebut digunakan untuk menentukan nilai-nilai baru dari ke 8 pixel disekitarnya piksel tetangga, nilai matrix pixel 3x3 sebagai berikut:

Kordinat (0, 0)

21 19 19

20 **19** 23

18 18 24

nilai pusat yang digunakan sebagai nilai ambang 19.

- 4) Untuk setiap piksel tetangga dari piksel tengah, akan ditetapkan nilai biner baru dengan aturan 1 untuk nilai yang sama atau lebih tinggi dari ambang dan 0 untuk nilai lebih rendah dari ambang. Berikut ini adalah nilai biner:

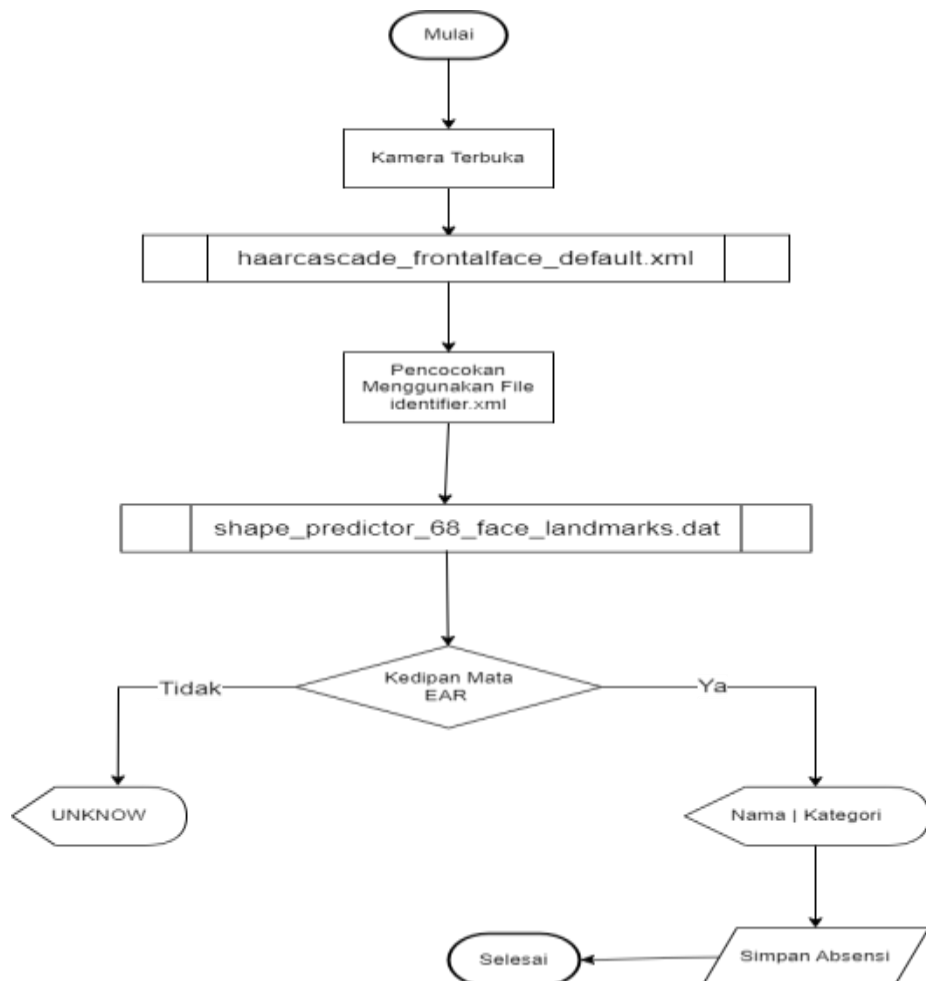
```

1 1 1
1  1
0 0 1

```

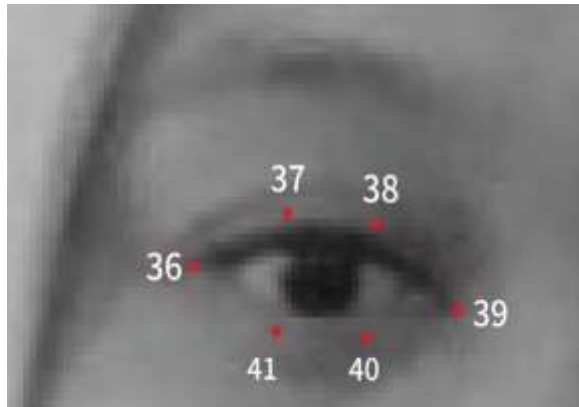
- 5) Nilai binery 11111001 dan desimalnya 249. Setelah diubah menjadi nilai biner maka akan menghasilkan deret biner yang kemudian jika diubah menjadi nilai decimal yang akan menghasilkan nilai baru untuk nilai tengah pixel selanjutnya. Hasil dari deret biner ini disimpan didalam file identifier.xml yang akan digunakan untuk proses pengenalan wajah.

- c) Flowchart Deteksi Kedipan Mata



Gambar 3. 6 Deteksi Kedipan Mata

- 1) Kamera terbuka kemudian akan menampilkan gambar wajah yang berada di depan kamera.
- 2) `haarcascade_frontalface_default.xml` file XML yang berisi model deteksi classifier yang telah dilatih menggunakan algoritma *Haar Cascade* untuk mendeteksi wajah manusia dari tampak depan. File ini merupakan bagian dari pustaka OpenCV yang sering digunakan dalam deteksi wajah.
- 3) Wajah yang terdeteksi kemudian dicocokkan dengan data model LBPH (*Local Binary Patterns Histograms*) yang telah dilatih sebelumnya. Model LBPH membandingkan ciri-ciri wajah yang disimpan dalam file seperti `identifier.xml` untuk menentukan apakah wajah tersebut cocok dengan data wajah yang dikenal. Jika sesuai, sistem akan mengenali identitasnya.
- 4) `shape_predictor_68_face_landmarks.dat` adalah file model yang digunakan oleh dlib untuk mendeteksi 68 titik landmark pada wajah manusia.
- 5) EAR dihitung dari jarak antara landmark mata. Rumusnya menggunakan jarak vertikal dan horizontal mata. Jika nilai EAR turun di bawah threshold (0.2), mata dianggap tertutup. Jika EAR kembali di atas threshold setelah sebelumnya di bawah, maka mata dianggap telah berkedip.



Gambar 3. 7 Titik Landmark Mata

$$EAR = \frac{|(37 + 38) - (41 + 40)|}{2 \times |36 - 39|}$$

Titik Kordinat Right Eye Landmarks (36-41):

36: (52, 79)

37: (61, 76)

38: (71, 76)

39: (81, 83)

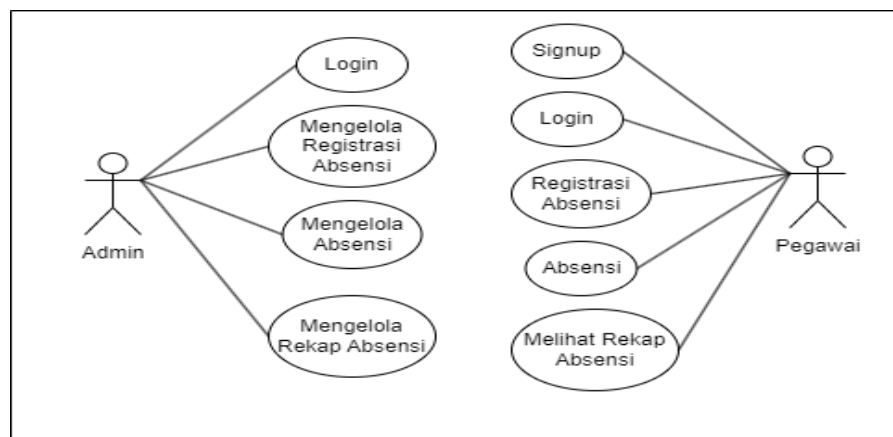
40: (71, 85)

41: (60, 85)

$$EAR = \frac{9.01}{29.28} = 0.31$$

- 6) EAR bernilai 0.31 dengan threshold 0.2. Semakin tinggi nilai EAR, maka mata semakin terbuka. Sebaliknya, semakin rendah nilai EAR, mata semakin tertutup.
- 7) Jika mata terdeteksi, sistem akan menampilkan nama dan kategori, kemudian melakukan absensi, hasilnya akan ditampilkan dan disimpan ke dalam database.
- 8) Jika mata tidak terdeteksi, akan menampilkan 'UNKNOWN'. Pengguna tidak dapat melakukan absensi, dan data absensi tidak akan disimpan ke dalam database.

d) *Use Case*

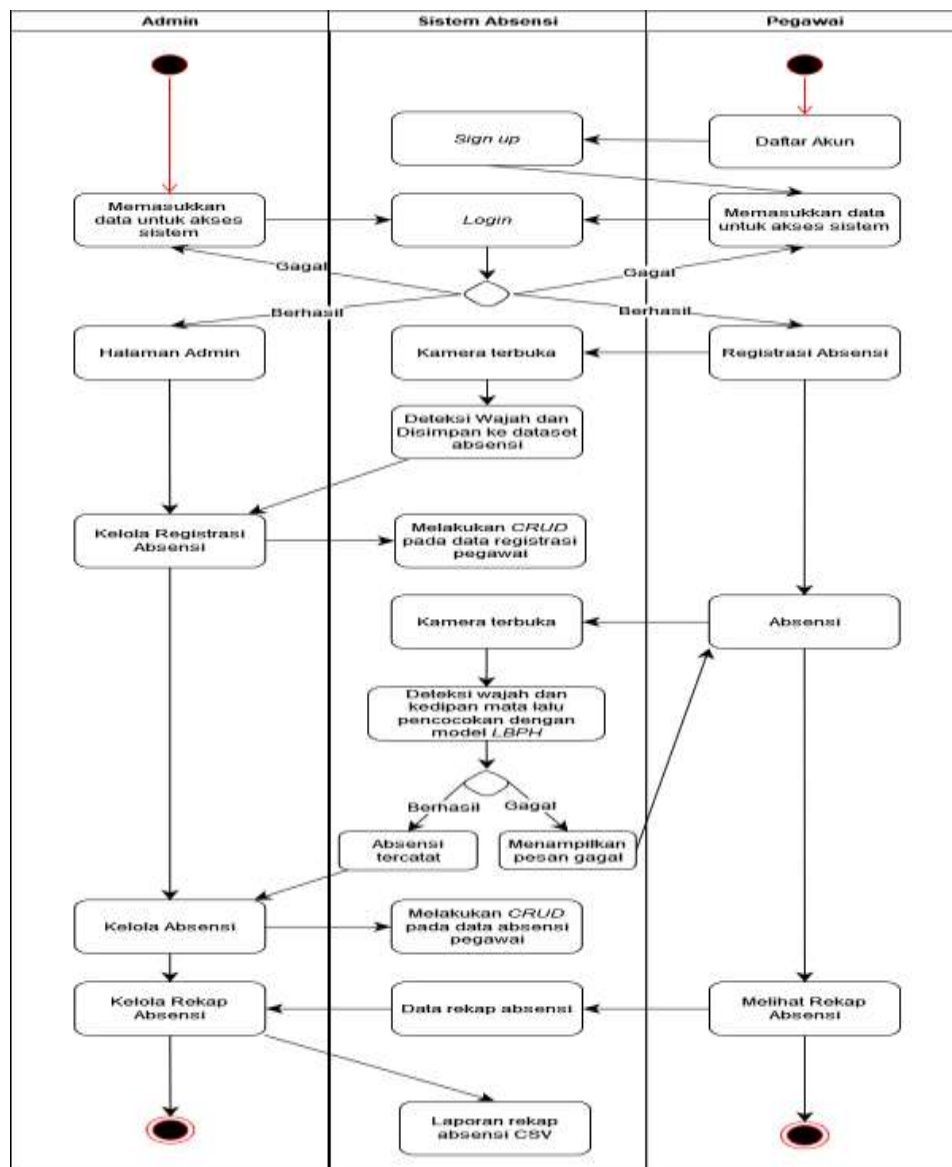


Gambar 3. 8 Use Case

Pegawai dapat melakukan registrasi kehadiran dengan menyimpan data wajah mereka, yang digunakan untuk identitas saat absensi. Selain itu, pegawai juga dapat melihat rekap absensi pribadi mereka. Di sisi lain, admin memiliki

kemampuan untuk mengelola data absensi secara menyeluruh seperti melihat rekap absensi, mengedit rekap absensi dan menghapus rekap absensi. Admin juga dapat mengunduh rekap absensi dalam format file CSV.

e) Activity Diagram



Gambar 3. 9 Activity Diagram

Alur pada pengguna atau user Admin yang ada pada *activity diagram*, dimulai dengan *menginputkan* data untuk mengakses sistem absensi yang biasa disebut dengan login. Apabila data tersebut benar maka admin dapat mengakses fitur yang ada pada halaman admin dan apabila data salah maka akan diarahkan

kembali untuk memasukkan data yang dibutuhkan. Fitur-fitur apabila *login* berhasil dilakukan dijabarkan seperti dibawah ini:

1) Kelola Registrasi Absensi

Fitur dimana admin dapat melakukan proses umum seperti *create*, *read*, *update*, dan *delete* pada data registrasi absensi yang didaftarkan oleh para pegawai yang mendaftar.

2) Kelola Absensi

Fitur dimana admin dapat melakukan proses umum yang sama seperti registrasi admin yaitu melihat data absensi, menghapus data absensi, mengedit data absensi para pegawai yang sudah melakukan absensi.

3) Kelola Rekap Absensi

Admin dapat melihat rekap absensi dari para pegawai dan dapat di ekspor menjadi file *csv* untuk kepentingan laporan yang lebih mudah dilihat.

Dan pada pengguna atau *user* Pegawai yang ada pada *activity diagram*, dimulai dengan mendaftar akun atau *sign up*. Para pegawai harus memiliki akun terlebih dahulu sebelum bisa menggunakan sistem absensi. Setelah para pegawai selesai *sign up*, maka mereka dapat melakukan *login* seperti sistem pada umumnya. Beberapa fitur yang dapat diketahui pada alur aktivitas pegawai antara lain:

1) Registrasi Absensi

Pegawai harus mendaftarkan wajah mereka ke *database* sistem absensi dengan fitur registrasi absensi. Wajah dan kedipan akan dicatat dan disimpan. Jika pendaftaran wajah dan kedipan gagal maka langkah tersebut harus diulang sampai pegawai benar-benar berhasil mendaftarkan wajah dan kedipan mereka.

2) Absensi

Apabila sudah terdaftar, maka fitur absensi sudah bisa digunakan untuk mencatat kehadiran pegawai. Apabila proses absensi mengalami kegagalan, maka

pegawai diharuskan untuk melakukan ulang deteksi wajah dan kedipan pada absensi agar hasil absensi bisa dibandingkan dengan hasil registrasi absensi yang sudah tercatat sebelumnya pada database.

3) Rekap Absensi

Fitur lainnya yaitu pegawai yang dapat melihat rekap absen dari mereka sendiri untuk melihat performa kehadiran pengguna itu sendiri sebagai pegawai.

3.3. Perancangan Sistem Database



Gambar 3. 10 Entity relationship diagram (ERD)

a) Pekerja

No	Nama Field	Tipe Data	Length field	Keterangan
1.	id	integer	20	primary key untuk pekerja
2.	nama_depan	varchar	50	menyimpan nama depan pekerja, dengan panjang maksimum 50 karakter
3.	nama_belakang	varchar	50	menyimpan nama belakang pekerja, dengan panjang maksimum 50 karakter
4.	email	varchar	254	menyimpan alamat email pekerja
5.	password	varchar	100	menyimpan kata sandi pekerja

Tabel 3. 3 Perancangan Database Pekerja

b) data_pekerja

No	Nama Field	Tipe Data	Length field	Keterangan
1.	kode_dp	varchar	3	merupakan primary key yang menyimpan kode unik data pekerja
2.	nama_dp	varchar	50	menyimpan nama data pekerja
3.	kategori_dp	varchar	30	menyimpan kategori atau jabatan yang dipilih dari pilihan yang tersedia Kuwu, Sekertaris Desa, dsb
4.	aktif_dp	date	1	secara default diisi dengan 'Y', menyimpan status aktif atau tidak
5.	tanggal_dp	time	-	secara otomatis menyimpan tanggal ketika data dibuat.

Tabel 3. 4 Perancangan Database Data Pekerja

c) img_dataset

No	Nama Field	Tipe Data	Length field	Keterangan
1.	id_img	integer	11	primary key untuk dataset gambar, secara otomatis bertambah.
2.	kode_img	varchar	3	foreignkey ke data pekerja, menyimpan referensi identitas.
3.	foto	varchar	100	menyimpan path file gambar, dengan panjang maksimum 100 karakter.

Tabel 3. 5 Perancangan Database Image Dataset

d) absensi

No	Nama Field	Tipe Data	Length field	Keterangan
1.	id_ab	integer	11	primary key otomatis, id unik untuk setiap entri absensi
2.	kode_ab	varchar	3	foreignkey ke data pekerja, menyimpan referensi identitas untuk absensi
3.	nama_ab	varchar	50	menyimpan nama pegawai
4.	tanggal_ab	date	-	menyimpan tanggal absensi
5.	waktu_ab	time	6	menyimpan waktu absensi (jam dan menit)
6.	keterangan_ab	varchar	20	menyimpan keterangan absensi seperti tepat waktu, terlambat/jam kerja
7.	ktdp_ab	varchar	20	menyimpan kategori atau jabatan
8.	bulan_tahun	varchar	7	menyimpan bulan dan tahun absensi dalam format YYYY-MM

Tabel 3. 6 Perancangan Database Absensi

e) rekap_absensi

No	Nama Field	Tipe Data	Length field	Keterangan
1.	id_ra	integer	11	primary key untuk rekap absensi
2.	kode_ra	varchar	10	foreignkey ke data pekerja, menyimpan referensi identitas untuk rekap absensi
3.	nama_ra	varchar	50	menyimpan nama pegawai
4.	bulan_tahun	varchar	7	menyimpan bulan dan tahun dalam format YYYY-MM
5.	kategori_ra	varchar	50	menyimpan kategori atau jabatan
6.	total_absensi	integer	11	menyimpan total kehadiran dalam satu bulan
7.	total_hari_kerja	integer	11	menyimpan total hari kerja senin-jumat dalam satu bulan

Tabel 3. 7 Perancangan Database Rekap Absensi

f) auth_user

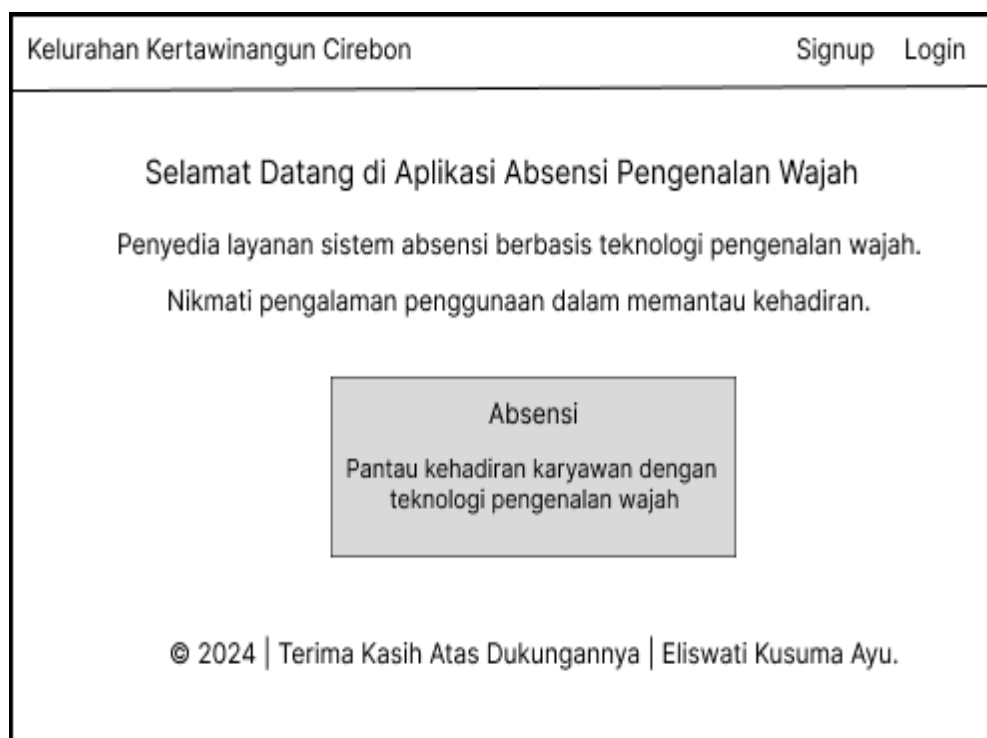
No	Nama Field	Tipe Data	Length field	Keterangan
1.	id	integer	11	primary key yang unik untuk setiap pengguna

2.	username	varchar	128	menyimpan nama pengguna (username) yang digunakan untuk login
3.	email	varchar	254	menyimpan alamat email pengguna
4.	password	varchar	128	menyimpan kata sandi pengguna

Tabel 3. 8 Perancangan Database Auth User

3.4. Tampilan Perancangan

3.4.1. Tampilan Perancangan Home

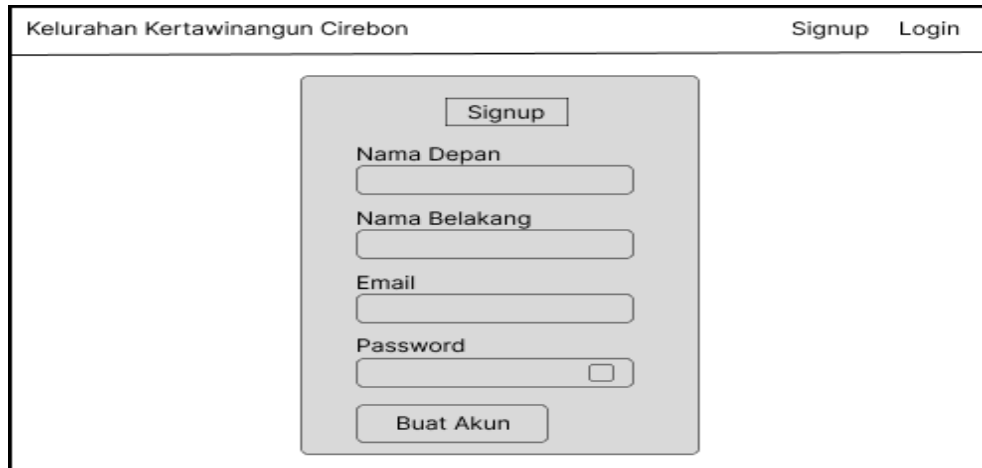


Gambar 3. 11 Tampilan Perancangan Home

Perancangan tersebut merupakan desain tampilan halaman home, pada bagian navbar terdapat class nav-link halaman signup dan login. Tampilannya sederhana dan minimalis, dengan judul Selamat Datang di Aplikasi Absensi Pengenalan Wajah dan deskripsi singkat tentang aplikasi. Terdapat juga kotak informasi dengan judul Absensi yang menjelaskan fungsi aplikasi dalam mengelola

kehadiran karyawan. Di bagian bawah terdapat informasi hak cipta, ucapan terima kasih, dan nama pengembang.

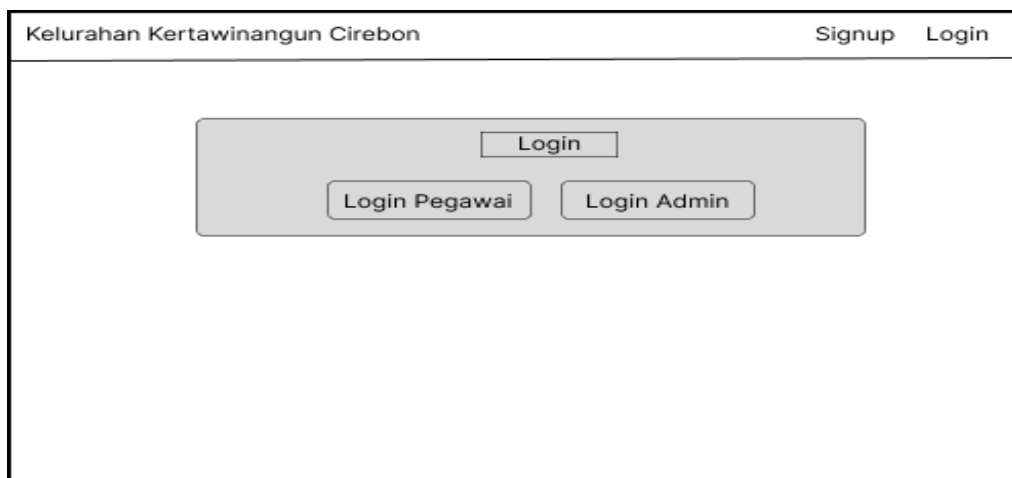
3.4.2. Tampilan Perancangan Signup



Gambar 3. 12 Tampilan Perancangan Signup

Perancangan formulir signup ini dirancang dengan tampilan minimalis dan sederhana, memiliki label untuk Nama Depan, Nama Belakang, Email, dan Password. Terdapat checkbox pada Password yang memungkinkan pengguna menampilkan atau menyembunyikan password. Di bagian bawah, terdapat button Buat Akun untuk mengirimkan dan menyimpan data formulir signup pegawai ke dalam database.

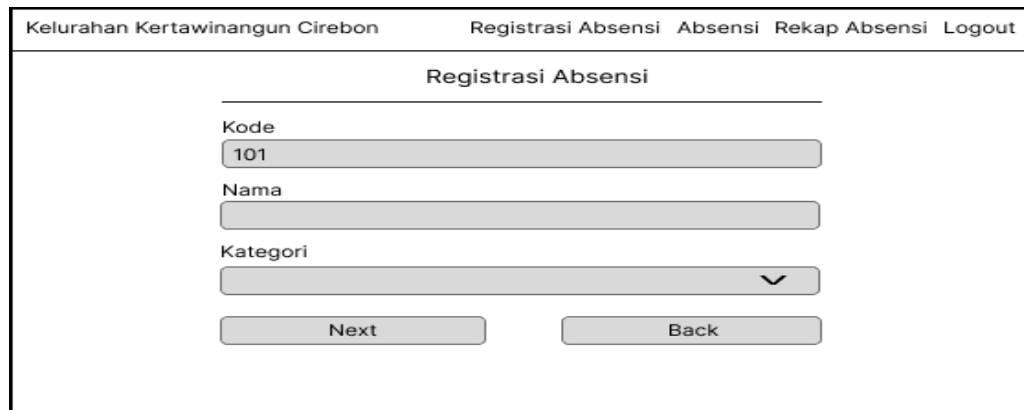
3.4.3. Tampilan Perancangan Login



Gambar 3. 13 Tampilan Perancangan Login

Tampilan perancangan login terdapat dua button yaitu login pegawai dan login admin, kedua button ini berfungsi untuk memisahkan akses login antara pegawai dan admin, sehingga pegawai ataupun admin dapat login sesuai dengan peran mereka.

3.4.4. Tampilan Perancangan Registrasi Absensi



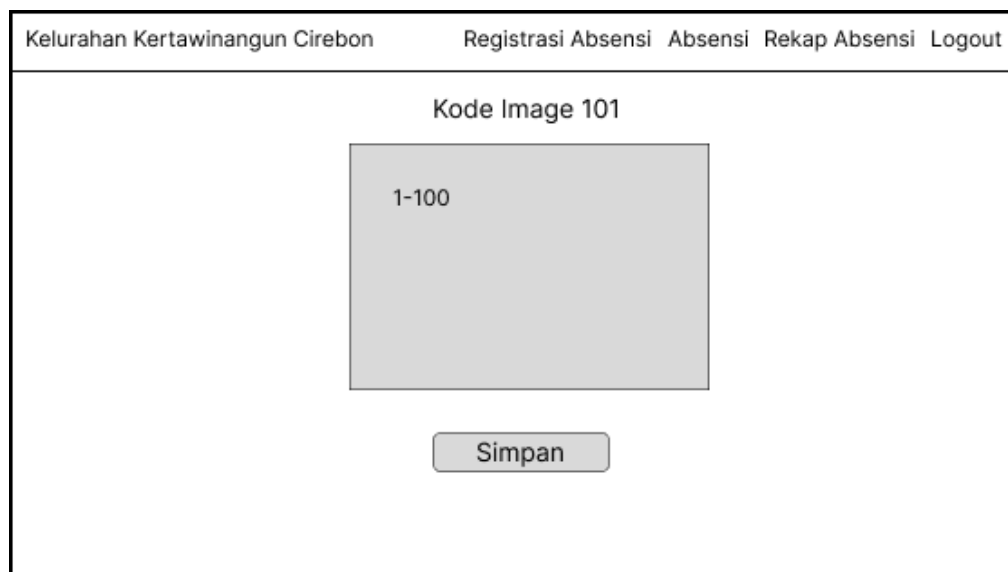
The screenshot shows a web application interface for 'Kelurahan Kertawinangun Cirebon'. The navigation bar includes 'Registrasi Absensi', 'Absensi', 'Rekap Absensi', and 'Logout'. The main content area is titled 'Registrasi Absensi' and contains the following elements:

- A 'Kode' field with the value '101'.
- A 'Nama' field.
- A 'Kategori' dropdown menu.
- 'Next' and 'Back' buttons.

Gambar 3. 14 Tampilan Perancangan Registrasi Absensi

Tampilan perancangan registrasi absensi terdapat kode unik yang otomatis berbeda setiap melakukan registrasi absensi, formulir label nama, lalu select pada kategori yang dapat memilih posisi atau jabatan pegawai dari pilihan yang tersedia.

3.4.5. Tampilan Perancangan Dataset Wajah



The screenshot shows the 'Dataset Wajah' (Face Dataset) form. The navigation bar is the same as in the previous image. The main content area is titled 'Kode Image 101' and contains:

- A large image placeholder labeled '1-100'.
- A 'Simpan' (Save) button.

Gambar 3. 15 Tampilan Perancangan Dataset Wajah

Tampilan perancangan dataset wajah akan menampilkan kode unik yang telah ditetapkan saat registrasi absensi. Kamera akan menyimpan gambar wajah dalam dataset hingga mencapai angka 100. Setelah itu, button simpan akan digunakan untuk menyimpan dataset.

3.4.6. Tampilan Perancangan Absensi

Kelurahan Kertawinangun Cirebon	Registrasi Absensi	Absensi	Rekap Absensi	Logout												
Hari Ini																
		<table border="1"> <thead> <tr> <th>Kode</th> <th>Nama</th> <th>Kategori</th> <th>Tanggal</th> <th>Waktu</th> <th>Keterangan</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>			Kode	Nama	Kategori	Tanggal	Waktu	Keterangan						
		Kode	Nama	Kategori	Tanggal	Waktu	Keterangan									

Gambar 3. 16 Tampilan Perancangan Absensi

Perancangan absensi akan menampilkan kamera saat proses absensi dilakukan, kemudian akan menampilkan informasi dalam bentuk tabel yang berisi nama, kategori, tanggal, waktu, dan keterangan.

3.4.7. Tampilan Perancangan Rekap Absensi

Kelurahan Kertawinangun Cirebon	Registrasi Absensi	Absensi	Rekap Absensi	Logout												
Rekap Absensi																
<table border="1"> <thead> <tr> <th>Kode</th> <th>Nama</th> <th>Kategori</th> <th>Bulan-Tahun</th> <th>Total Absensi</th> <th>Total Hari Kerja</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>					Kode	Nama	Kategori	Bulan-Tahun	Total Absensi	Total Hari Kerja						
Kode	Nama	Kategori	Bulan-Tahun	Total Absensi	Total Hari Kerja											

Gambar 3. 17 Tampilan Perancangan Rekap Absensi

Perancangan rekap absensi dirancang untuk memberikan tampilan informasi kepada pegawai setiap bulan dalam bentuk tabel yang mencakup kode, nama, kategori, bulan-tahun, total absensi, dan total hari kerja senin-jumat.

3.4.8. Tampilan Perancangan Admin

The screenshot shows a web interface for 'Administration Kelurahan Kertawinangun'. At the top right is a 'Logout' link. Below the header, there is an 'Action' section with a dropdown menu currently set to 'Eksport to CSV' and a 'Go' button. Below this is a table with the following headers: Kode, Nama, Kategori, Bulan-Tahun, Total Absensi, and Total Hari Kerja.

Kode	Nama	Kategori	Bulan-Tahun	Total Absensi	Total Hari Kerja
------	------	----------	-------------	---------------	------------------

Gambar 3. 18 Tampilan Perancangan Admin

Perancangan admin mengelola rekap absensi pegawai secara efektif, mencakup fitur untuk melihat seperti nama, bulan-tahun, kategori, total absensi, dan total hari kerja. Selain itu, tersedia opsi untuk mengekspor data rekap absensi ke dalam format CSV, yang kompatibel dengan Excel, guna keperluan laporan. Admin juga dapat menambah, mengedit, atau menghapus data rekap absensi secara langsung.

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Interface

4.1.1. Halaman Signup



Gambar 4. 1 Halaman Signup

Pada halaman signup untuk membuat akun absensi, pegawai akan diminta untuk mengisi formulir pendaftaran dengan beberapa informasi penting. Data yang dimasukkan mencakup nama depan, nama belakang, email, dan password. Setelah semua informasi diisi, pegawai dapat mengirimkan formulir untuk pendaftaran akun, yang nantinya akan digunakan untuk mengakses platform absensi.

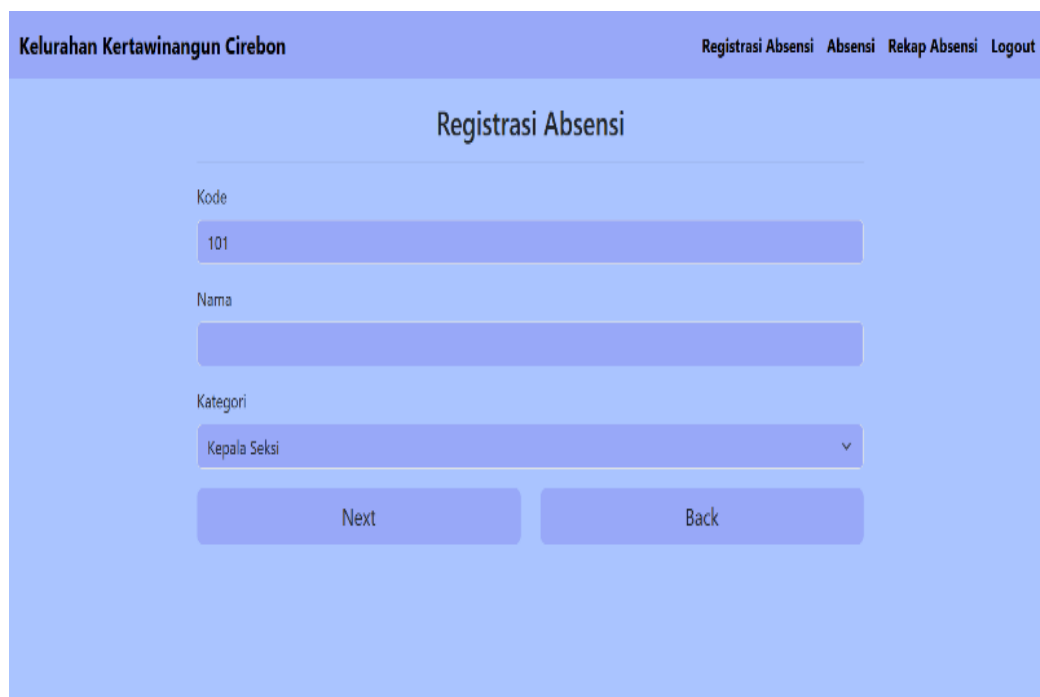
4.1.2. Halaman Login



Gambar 4. 2 Halaman Login

Pada halaman login, terdapat dua tombol login yang berbeda, yaitu Login Pekerja dan Login Admin. Tombol Login Pekerja memungkinkan pegawai untuk mengakses akun mereka dengan memasukkan email dan password yang terdaftar. Sementara itu, tombol Login Admin ditujukan untuk administrator, yang akan memasukkan username dan password untuk mengelola data dan fitur lebih lanjut dalam sistem. Kedua tombol ini berfungsi untuk memisahkan akses antara pekerja dan admin, sehingga pengguna dapat masuk sesuai dengan peran mereka.

4.1.3. Halaman Register Absensi



Gambar 4. 3 Halaman Register Absensi

Halaman registrasi absensi menampilkan kode pendaftaran yang otomatis diperbarui berdasarkan pendaftaran sebelumnya. Misalnya, jika kode pendaftaran sebelumnya adalah 104, maka kode berikutnya akan ditampilkan sebagai 105. Selain itu, halaman ini menyediakan kolom untuk mengisi nama lengkap serta pilihan kategori untuk menentukan posisi pegawai sesuai dengan posisi/jabatan masing-masing pegawai.

4.1.4. Halaman Proses Penyimpanan Dataset Wajah



Gambar 4. 4 Proses Simpan Dataset Wajah

Pada halaman dataset wajah, kamera akan mengubah gambar wajah menjadi grayscale, mendeteksi wajah, dan menyimpan hingga 100 data wajah ke dalam folder dataset.

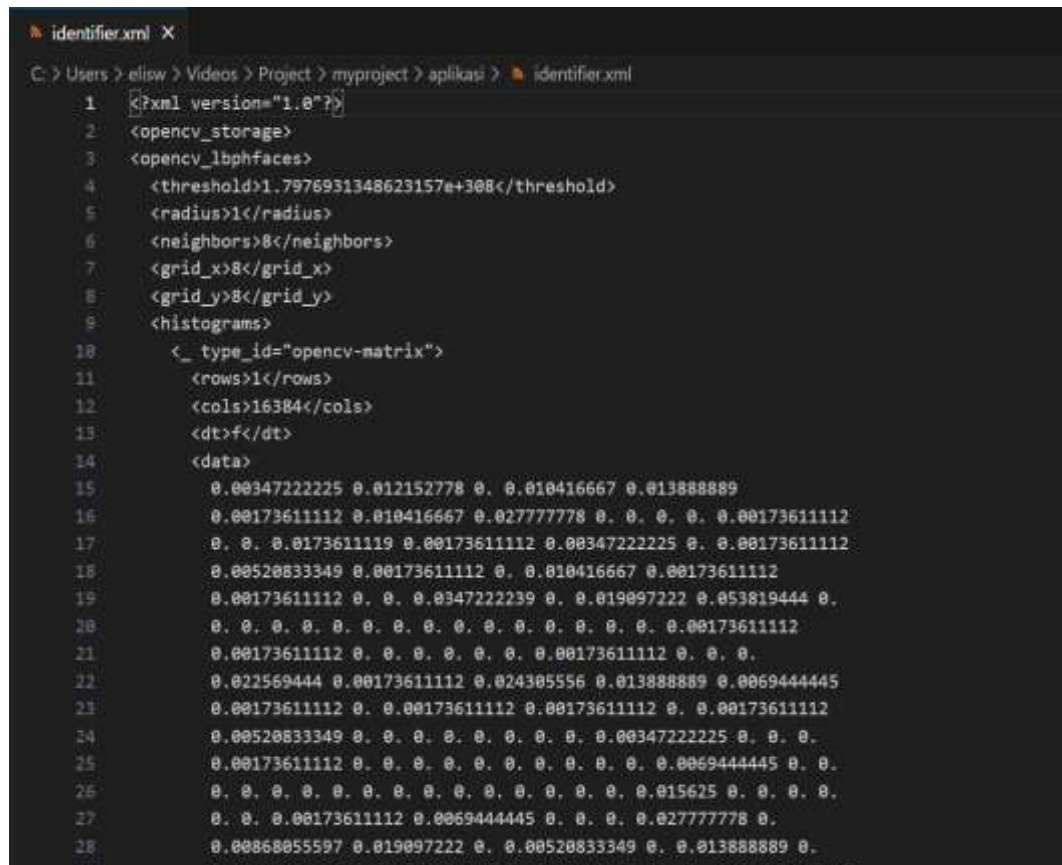
4.1.5. Dataset Wajah



Gambar 4. 5 Dataset Wajah

Dataset wajah disimpan setelah dikonversi ke grayscale, dideteksi dengan Haar Cascade, dipotong wajahnya, dan diubah ukurannya menjadi 200x200 piksel. Dataset ini digunakan untuk melatih model LBPH dalam pengenalan wajah. Setiap gambar diberi nama unik berdasarkan identitas subjek dan nomor urut gambar.

4.1.6. File Model LBPH



```

1  <?xml version="1.0"?>
2  <opencv_storage>
3  <opencv_lbphfaces>
4    <threshold>1.7976931348623157e+308</threshold>
5    <radius>1</radius>
6    <neighbors>8</neighbors>
7    <grid_x>8</grid_x>
8    <grid_y>8</grid_y>
9    <histograms>
10     <_ type_id="opencv-matrix">
11       <rows>1</rows>
12       <cols>16384</cols>
13       <dt>f</dt>
14       <data>
15         0.00347222225 0.012152778 0. 0.010416667 0.013888889
16         0.00173611112 0.010416667 0.027777778 0. 0. 0. 0.00173611112
17         0. 0. 0.0173611119 0.00173611112 0.00347222225 0. 0.00173611112
18         0.00520833349 0.00173611112 0. 0.010416667 0.00173611112
19         0.00173611112 0. 0. 0.0347222239 0. 0.019097222 0.053819444 0.
20         0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.00173611112
21         0.00173611112 0. 0. 0. 0. 0. 0. 0.00173611112 0. 0. 0.
22         0.022569444 0.00173611112 0.024305556 0.013888889 0.0069444445
23         0.00173611112 0. 0.00173611112 0.00173611112 0. 0.00173611112
24         0.00520833349 0. 0. 0. 0. 0. 0. 0. 0.00347222225 0. 0. 0.
25         0.00173611112 0. 0. 0. 0. 0. 0. 0. 0. 0.0069444445 0. 0.
26         0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.015625 0. 0. 0. 0.
27         0. 0. 0.00173611112 0.0069444445 0. 0. 0. 0.027777778 0.
28         0.00868055597 0.019097222 0. 0.00520833349 0. 0.013888889 0.
29         0.00173611112 0. 0.055555556 0. 0. 0. 0. 0. 0.015625 0. 0.

```

Gambar 4. 6 File Model LBPH

identifier.xml file yang menyimpan model LBPH (*Local Binary Patterns Histograms*) yang telah dilatih untuk mengenali wajah. File ini berisi data yang dihasilkan setelah proses pelatihan menggunakan dataset gambar wajah, mencakup informasi tentang fitur-fitur wajah yang dikenali. Hal ini memungkinkan pengenalan wajah berdasarkan pola tekstur wajah yang telah dipelajari. File ini digunakan untuk membandingkan wajah yang baru diambil dengan wajah-wajah yang sudah terlatih dalam dataset, sehingga identifikasi orang tersebut dapat dilakukan.

4.1.7. Halaman Absensi



Gambar 4. 7 Halaman Absensi

Absensi diidentifikasi menggunakan pengenalan wajah. Jika kedipan mata tidak terdeteksi hingga status bar penuh, sistem akan menampilkan status "Unknown" dan menghentikan aliran video. Jika kedipan mata terdeteksi, status bar akan menampilkan nama pegawai dan kategori. Data absensi akan disimpan dengan mencatat nama pegawai dan kategori yang ditampilkan, serta waktu absensi yang terdeteksi.

4.1.8. Halaman Rekap Absensi

Kelurahan Kertawinangun Cirebon					
Rekap Absensi					
Kode	Nama	Kategori	Bulan-Tahun	Total Absensi	Total Hari Kerja
101	Dian Putri Rahayu	Kepala Seksi	2024-08	1	22
104	Jodi Septian	Kepala Dusun	2024-08	1	22
103	Rita Anggraeni	Kepala Dusun	2024-08	1	22

Gambar 4. 8 Halaman Rekap Absensi

Halaman rekap absensi menampilkan informasi seperti bulan-tahun, nama pegawai, kategori, total absensi, dan total hari kerja. Halaman ini hanya untuk tampilan dan tidak dapat dikelola oleh pegawai.

4.1.9. Halaman Admin

The screenshot displays the Admin interface for 'Administration Kelurahan Kertawinangun'. The top header includes the site name and user information (WELCOME ADMIN, VIEW SITE / CHANGE PASSWORD / LOG OUT). The sidebar on the left contains navigation links: Home, Aplikasi, and Rekap absensi. The main content area features a search bar, a table of absence records, and an action menu. The table lists three records for August 2024, each with a unique ID RA, KODE RA, NAMA RA, and associated totals for absences, working days, and salary.

ID RA	KODE RA	NAMA RA	BULAN TAHUN	KATEGORI RA	TOTAL ABSENSI	TOTAL HARI KERJA	Gaji FINAL
4	103	Rita Anggraeni	2024-08	Kepala Dusun	1	22	780000
3	104	Jodi Septian	2024-08	Kepala Dusun	1	22	780000
1	101	Dian Putri Rahayu	2024-08	Kepala Seksi	1	22	780000

Gambar 4. 9 Halaman Admin

Pada halaman admin, admin dapat mengelola rekap absensi pegawai, termasuk melihat data seperti nama, bulan-tahun, kategori, total absensi, dan total hari kerja. Admin juga memiliki opsi untuk mengekspor data rekap absensi ke dalam format CSV yang dapat dibuka di Excel untuk keperluan laporan. Selain itu, admin bisa menambah, mengedit, atau menghapus data rekap absensi secara langsung dari panel admin, memberikan fleksibilitas dalam mengelola informasi absensi pegawai.

	A	B	C	D	E	F	G
1	ID RA	Kode RA	Nama RA	Bulan Tahun	Kategori RA	Total Absensi	Total Hari Kerja
2	4	103	Rita Anggraeni	2024-08	Kepala Dusun	1	22
3	3	104	Jodi Septian	2024-08	Kepala Dusun	1	22
4	1	101	Dian Putri Rahayu	2024-08	Kepala Seksi	1	22

Gambar 4. 10 File Rekap

4.2. Implementasi Program

4.2.1. Implementasi Program Dataset

```
def generate_dataset(nbr):
    # Inisialisasi Cascade Classifier
    face_cascade = cv2.CascadeClassifier('aplikasi/resources/haarcascade_frontalface_default.xml')

    # Fungsi internal untuk mendeteksi dan memotong wajah dari gambar
    def face_cropped(img):
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)

        if len(faces) == 0:
            return None

        x, y, w, h = faces[0]
        cropped_face = img[y:y+h, x:x+w]

        return cropped_face

    # Mulai tangkapan video dari kamera
    cap = cv2.VideoCapture(0)
    try:
        # Dapatkan ID terakhir dari dataset gambar
        lastid = ImgDataset.objects.aggregate(max_id=models.Max('id_img'))['max_id'] or 0

        id_img = lastid
        max_idimg = id_img + 100
        count_img = 0

        # Peroleh objek DataPekerja berdasarkan nomor yang diberikan
        dp_instance = DataPekerja.objects.get(kode_dp=nbr)

        base_dir = os.path.join("media", "dataset")
        if not os.path.exists(base_dir):
            os.makedirs(base_dir)

        # Loop utama untuk menghasilkan dataset
        while True:
            ret, img = cap.read()
            face = face_cropped(img)
            if face is not None:
                count_img += 1
                id_img += 1
                face_resized = cv2.resize(face, (280, 280))
                face_gray = cv2.cvtColor(face_resized, cv2.COLOR_BGR2GRAY)

                # Simpan gambar tanpa teks
                file_name = f'{nbr}_{id_img}.jpg'
                file_name_path = os.path.join(base_dir, file_name)
                cv2.imwrite(file_name_path, face_gray)

                with open(file_name_path, 'rb') as f:
                    content = ContentFile(f.read())
                    img_instance = ImgDataset.objects.create(id_img=id_img, kode_img=dp_instance)
                    img_instance.foto.save(file_name, content)
```

```

# Tambahkan teks pada gambar untuk tampilan streaming
face_with_text = cv2.putText(face_resized.copy(), str(count_img), (50, 50), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
frame = cv2.cvtColor(face_with_text, cv2.COLOR_BGR2GRAY)
encoded_frame = cv2.imencode('.jpg', frame)[1].tobytes()
yield (b'--frame\r\n\r\nContent-Type: image/jpeg\r\n\r\n' + encoded_frame + b'\r\n\r\n')

# Periksa apakah perlu menghentikan proses
if cv2.waitKey(1) == 13 or int(id_img) == int(max_idimg):
    break

except Exception as e:
    # Tangani pengecualian (cetak atau catat kesalahan, dll.)
    print(f'Terjadi kesalahan: {e}')

finally:
    # Stop tangkapan video dan bersihkan jendela
    cap.release()
    cv2.destroyAllWindows()

def vidfeed_dataset(request, nbr):
    return StreamingHttpResponse(generate_dataset(nbr), content_type='multipart/x-mixed-replace; boundary=frame')

def identifier(request, nbr):
    dataset_dir = os.path.join('media', 'dataset')

    for image_path in paths:
        file_name = os.path.basename(image_path)
        try:
            code, id_img, _ = file_name.split('.')
            id_img = int(id_img)
        except ValueError:
            # Skip file if it does not match the expected format
            continue

        img = Image.open(image_path).convert('L')
        image_np = np.array(img, 'uint8')

        faces.append(image_np)
        ids.append(id_img)

    ids = np.array(ids)

    # Train the classifier and save
    idfr = cv2.face_LBPHFaceRecognizer_create()
    idfr.train(faces, ids)

    # Lokasi penyimpanan file classifier.xml di dalam folder 'aplikasi'
    identifier_path = os.path.join('aplikasi', 'identifier.xml')
    idfr.write(identifier_path)

    return redirect('home') # Sesuaikan dengan nama fungsi URL yang merender halaman utama

```

- a) `generate_dataset(nbr)` berfungsi untuk mengambil gambar wajah dari kamera, menyimpannya sebagai dataset, dan menampilkan video secara langsung. Menggunakan *Haar Cascade* untuk mendeteksi wajah dari frame video. Jika wajah terdeteksi, gambar dipotong `face_cropped`, diubah menjadi grayscale, dan disimpan di folder `media/dataset`. Informasi terkait gambar seperti ID ataupun kode disimpan dalam model `ImgDataset`. Streaming video secara langsung jumlah gambar yang telah diambil. Proses berhenti jika 100 gambar sudah terkumpul atau pengguna menekan tombol Enter.
- b) `vidfeed_dataset(request, nbr)` mengirimkan video feed langsung dari kamera kepada pengguna. Menggunakan `StreamingHttpResponse` untuk mengirimkan video dari fungsi `generate_dataset` secara langsung.

- c) `identifier(request, nbr)` melatih model pengenalan wajah menggunakan dataset yang telah dibuat. Membaca gambar dari folder `media/dataset`. Mengekstrak ID ataupun kode dari nama file. Gambar diubah ke format grayscale untuk diproses. Data digunakan untuk melatih model LBPH, yang kemudian disimpan sebagai file `identifier.xml` di folder aplikasi. Setelah selesai, pengguna diarahkan kembali ke halaman utama.

4.2.2. Implementasi Program Pengenalan Wajah

```
# Initialize the face and eye detectors
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("aplikasi/resources/shape_predictor_68_face_landmarks.dat")

# Define constants and variables
threshold = 0.2
eye_closed = False
kedip = 'X'
color_kedip = (0, 0, 255)
justscanned = False
pause_cnt = 0
cnt = 0
stream_stop = False

def reset_variables():
    global eye_closed, kedip, color_kedip, justscanned, pause_cnt, cnt, stream_stop
    eye_closed = False
    kedip = 'X'
    color_kedip = (0, 0, 255)
    justscanned = False
    pause_cnt = 0
    cnt = 0
    stream_stop = False

def mid_line_distance(p1, p2, p3, p4):
    p5 = np.array([int((p1[0] + p2[0]) / 2), int((p1[1] + p2[1]) / 2)])
    p6 = np.array([int((p3[0] + p4[0]) / 2), int((p3[1] + p4[1]) / 2)])
    return norm(p5 - p6)
```

- a) `reset_variables()` mereset semua variabel global yang digunakan dalam proses deteksi wajah, verifikasi mata, dan pengelolaan alur streaming video. Fungsi ini dipanggil saat sesi video feed dimulai ulang atau ketika reset diperlukan.
- b) `mid_line_distance(p1, p2, p3, p4)` menghitung jarak rata-rata antara dua garis tengah, biasanya digunakan untuk mengukur jarak antara bagian atas dan bawah mata. `p1` dan `p2` titik-titik di bagian atas mata, `p3` dan `p4` titik-titik di bagian bawah mata. Jarak Euclidean antara dua garis tersebut.

```

def aspect_ratio(landmarks, eye_range):
    eye = np.array([np.array([landmarks.part(i).x, landmarks.part(i).y]) for i in eye_range])
    B = norm(eye[0] - eye[3])
    A = mid_line_distance(eye[1], eye[2], eye[5], eye[4])
    ear = A / B
    return ear

def detect_eye_blink(frame):
    global eye_closed, kedip, color_kedip
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = detector(gray)

    for rect in faces:
        landmarks = predictor(gray, rect)
        left_aspect_ratio = aspect_ratio(landmarks, range(42, 48))
        right_aspect_ratio = aspect_ratio(landmarks, range(36, 42))
        ear = (left_aspect_ratio + right_aspect_ratio) / 2.0

        # Display the EAR value on the frame
        cv2.putText(frame, "EAR: {:.2f}".format(ear), (10, 50), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 0, 0), 2)

        if ear < threshold:
            eye_closed = True
        elif ear >= threshold and eye_closed:
            kedip = 'Berhasil'
            color_kedip = (0, 255, 0)
            eye_closed = False

```

- c) `aspect_ratio(landmarks, eye_range)` menghitung rasio aspek mata Eye Aspect Ratio/EAR, yang digunakan untuk mendeteksi apakah mata sedang terbuka atau tertutup. Jika EAR berada di bawah ambang batas, mata dianggap tertutup. Landmarks titik-titik landmark wajah. `eye_range` Indeks landmark untuk mata kiri atau kanan.
- d) `detect_eye_blink(frame)` mendeteksi kedipan mata berdasarkan nilai EAR. Jika EAR turun di bawah threshold, status `eye_closed` diatur ke True. Jika EAR kembali naik, sistem mencatat kedipan berhasil dan memperbarui variabel global kedip menjadi Berhasil `color_kedip` menjadi hijau (0, 255, 0).

```

for n in range(36, 48):
    x = landmarks.part(n).x
    y = landmarks.part(n).y
    cv2.putText(frame, "Pejadian Mata :", (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 0, 0), 2)
    cv2.putText(frame, "{}".format(kedip), (200, 60), cv2.FONT_HERSHEY_SIMPLEX, 0.7, color_kedip, 2)

def generate_frames():
    global justscanned, pause_cnt, cnt, stream_stop

    # Reset variables at the start of frame generation
    reset_variables()

    faceCascade = cv2.CascadeClassifier('aplikasi/resources/haarcascade_frontalface_default.xml')
    idfr = cv2.Face_LBPHFaceRecognizer_create()

    identifier_path = "aplikasi/identifier.xml"
    if os.path.exists(identifier_path):
        idfr.read(identifier_path)
        identifier_loaded = True
    else:
        identifier_loaded = False

    cap = cv2.VideoCapture(0)
    cap.set(3, 480)
    cap.set(4, 480)

```

```

while True:
    success, frame = cap.read()
    if not success:
        break

    if identifier_loaded:
        coords = draw_boundary(frame, faceCascade, 1.1, 10, (255, 255, 0), "Face", idfr)
    else:
        coords = draw_boundary(frame, faceCascade, 1.1, 10, (255, 255, 0), "Face")

    detect_eye_blink(frame)

    # Convert the frame to JPEG format
    ret, buffer = cv2.imencode('.jpg', frame)
    if not ret:
        continue

    frame = buffer.tobytes()

    # Yield the frame as a multipart response
    yield (b'--frame\r\n'
          + b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

    if stream_stop:
        break

cap.release()

```

- e) generate_frames() mengelola aliran video dari kamera, mendeteksi wajah dan mata, serta mengirimkan frame ke antarmuka pengguna dalam format multipart. Memuat model pengenalan wajah identifier.xml jika tersedia untuk memverifikasi wajah. Membaca frame dari webcam memproses frame untuk ditampilkan di antarmuka. Menambahkan kotak di sekitar wajah, mendeteksi kedipan mata, dan memverifikasi wajah.

```

def draw_boundary(img, faceCascade, scaleFactor, minNeighbors, color, text, idfr=None):
    global justscanned, pause_cnt, cnt, stream_stop
    gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    features = faceCascade.detectMultiScale(gray_image, scaleFactor, minNeighbors)

    pause_cnt += 1
    coords = []

    for (x, y, w, h) in features:
        cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)

        if idfr:
            id, pred = idfr.predict(gray_image[y:y + h, x:x + w])
            confidence = int(100 * (1 - pred / 300))

            # Jika confidence tinggi, wajah dianggap dikenali
            if confidence > 87 and not justscanned:
                cnt += 1
                n = (100 / 30) * cnt
                w_filled = (cnt / 30) * w

                cv2.putText(img, str(int(n)) + ' %', (x + 20, y + h + 25), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (153, 255, 255), 1, cv2.LINE_AA)
                cv2.rectangle(img, (x, y + h + 40), (x + w, y + h + 50), color, 2)
                cv2.rectangle(img, (x, y + h + 40), (x + int(w_filled), y + h + 50), (153, 255, 255), cv2.FILLED)

            try:
                # Cek data wajah di database ImgDataset
                img_data = ImgDataset.objects.get(id_img=id)

```

```

data = img_data.kode_img
dkode, dnama, dkategori = data.kode_dp, data.nama_dp, data.kategori_dp
except ImgDataset.DoesNotExist:
    # Jika data wajah tidak ditemukan di database
    dkode = 'Unknown in Dataset'
    dnama = 'Unknown in Dataset'
    dkategori = 'Unknown in Dataset'
    cv2.putText(img, 'Not Found in Database', (x - 10, y + 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA)

# Jika pengenalan wajah berhasil, baru deteksi kedipan mata dilakukan
if cnt == 10:
    cnt = 0
    if dkode != 'Unknown in Dataset': # Hanya lakukan absensi jika wajah ditemukan
        try:
            data_instance = DataPekerja.objects.get(kode_dp=dkode)
            keterangan = tentukan_keterangan_waktu()

            # Hanya jika kedipan mata terdeteksi sebagai 'Berhasil'
            if kedip == 'Berhasil':
                Absensi.objects.create(
                    tanggal_ab=data.today(),
                    kode_ab=data_instance.kode_dp,
                    nama_ab=data_instance.nama_dp,
                    ktdp_ab=data_instance.kategori_dp,
                    waktu_ab=timezone.localtime(),
                    keterangan_ab=keterangan
                )

            cv2.putText(img, dnama + ' ' + dkategori, (x - 10, y + 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (153, 255, 255))
            cv2.putText(img, keterangan, (x - 10, y + 40), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (153, 255, 255), 2, cv2.LINE_AA)
            justscanned = True
            pause_cnt = 0
            stream_stop = True
        except DataPekerja.DoesNotExist:
            cv2.putText(img, 'Not Found in Worker DB', (x - 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA)
        else:
            # Jika wajah tidak dikenali di dataset
            cv2.putText(img, 'Unknown in Dataset', (x - 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA)
            stream_stop = True
    else:
        # Jika confidence rendah (wajah tidak dikenali), tampilkan pesan
        if not justscanned:
            cv2.putText(img, 'Unknown (Low Confidence)', (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA)

coords.append((x, y, w, h))
return coords

def video_feed(request):
    reset_variables() # Reset variables before starting the feed
    return StreamingHttpResponse(generate_frames(), content_type='multipart/x-mixed-replace; boundary=frame')

```

- f) `draw_boundary()` mendeteksi wajah dalam frame, menggambar kotak di sekitarnya, dan memverifikasi wajah menggunakan model LBPH. Jika confidence level wajah melebihi 87%, data diproses lebih lanjut. Sistem memverifikasi kedipan mata untuk memastikan kehadiran fisik dan mencegah spoofing. Jika wajah dan kedipan terverifikasi, data absensi akan disimpan ke database Absensi.
- g) `video_feed(request)` menyediakan endpoint untuk streaming video feed. mengembalikan aliran frame dalam format `multipart/x-mixed-replace`. Fungsi ini juga memanggil `reset_variables()` sebelum memulai streaming.

4.2.3. Implementasi Program Rekap Absensi

```
# Fungsi untuk menghitung jumlah hari kerja dalam satu bulan tertentu
def hitung_hari_kerja(bulan_tahun):
    tahun, bulan = map(int, bulan_tahun.split('-')) # Memisahkan tahun dan bulan
    hari_kerja = 0 # Inisialisasi jumlah hari kerja
    for week in calendar.monthcalendar(tahun, bulan): # Mendapatkan kalender bulanan dalam bentuk matriks
        for day in week[:5]: # Iterasi dari hari Senin hingga Jumat
            if day != 0: # Hanya hitung hari kerja yang valid (bukan 0)
                hari_kerja += 1
    return hari_kerja # Mengembalikan jumlah hari kerja
```

- a) `hitung_hari_kerja` menghitung jumlah hari kerja Senin sampai Jumat dalam bulan tertentu berdasarkan format input YYYY-MM. Memisahkan input `bulan_tahun` menjadi tahun dan bulan dengan `split`. Menggunakan `calendar.monthcalendar` untuk membuat kalender bulan dalam bentuk matriks mingguan. Iterasi dilakukan pada hari kerja Senin hingga Jumat, kolom 0-4. Hari kerja yang valid bukan nol dihitung. Mengembalikan total hari kerja dalam bulan tersebut.

```
# Fungsi untuk menghitung total absensi dan mengupdate tabel RekapAbsensi
def hitung_total_absensi(request):
    # Mengambil data absensi dari database
    data_absensi = Absensi.objects.values('nama_ab', 'kode_ab_kode_dp', 'tanggal_ab', 'ktdp_ab')

    # Debug: Memeriksa data absensi yang diambil
    print("Data Absensi:", list(data_absensi)) # Baris debugging

    # Membuat DataFrame dari data absensi
    df = pd.DataFrame(list(data_absensi))

    # Debug: Memeriksa kolom dalam DataFrame
    print("Columns in DataFrame:", df.columns) # Baris debugging

    if df.empty or 'tanggal_ab' not in df.columns:
        # Menangani kasus jika tidak ada data atau kolom tidak ada
        return render(request, 'rekap_absensi.html', {'rekap_list': []})

    # Mengubah kolom 'tanggal_ab' menjadi tipe datetime
    df['tanggal_ab'] = pd.to_datetime(df['tanggal_ab'])
    # Menghapus duplikat berdasarkan kombinasi tanggal, nama, dan kode
    df = df.drop_duplicates(subset=['tanggal_ab', 'nama_ab', 'kode_ab_kode_dp'], keep='first')
    # Memfilter hanya hari kerja (Senin hingga Jumat)
    df = df[df['tanggal_ab'].dt.dayofweek < 5]
    # Menambahkan kolom 'bulan_tahun' dengan format periode 'YYYY-MM'
    df['bulan_tahun'] = df['tanggal_ab'].dt.to_period('M')
```

```

# Menghitung jumlah absensi per kombinasi nama, kode, bulan-tahun, dan kategori
jumlah_data = df.groupby(['nama_ab', 'kode_ab_kode_dp', 'bulan_tahun', 'ktdp_ab']).size().reset_index(name='total_absensi')

for idx, row in jumlah_data.iterrows():
    # Mengonversi periode ke format string 'YYYY-MM'
    bulan_tahun_str = row['bulan_tahun'].strftime('%Y-%m')
    # Menghitung total hari kerja untuk bulan dan tahun tersebut
    total_hari_kerja = hitung_hari_kerja(bulan_tahun_str)
    kategori = row['ktdp_ab']
    total_absensi = row['total_absensi']

    # Calculate deductions
    gaji_perbulan = salary_data[kategori]['gaji_perbulan']
    potongan_perhari = salary_data[kategori]['potongan_perhari']
    potongan = (total_hari_kerja - total_absensi) * potongan_perhari
    gaji_final = gaji_perbulan - potongan

    # Membuat atau memperbarui entri RekapAbsensi dengan data baru
    obj, created = RekapAbsensi.objects.update_or_create(
        kode=row['kode_ab_kode_dp'],
        nama=row['nama_ab'],
        bulan_tahun=bulan_tahun_str,
        kategori=row['kategori'],
        defaults={
            'total_absensi': total_absensi,
            'total_hari_kerja': total_hari_kerja,
            'gaji_final': gaji_final # Save final salary
        }
    )

# Mengambil semua data RekapAbsensi untuk ditampilkan di template
rekap_list = RekapAbsensi.objects.all()
return render(request, 'rekap_absensi.html', {'rekap_list': rekap_list})

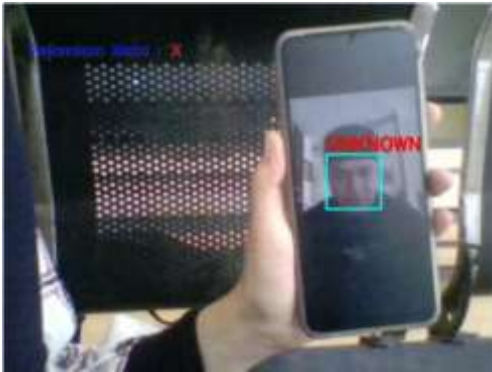
```

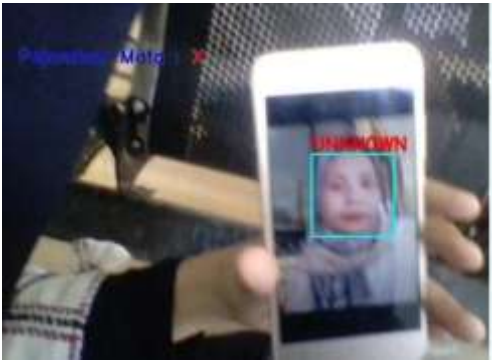
- b) `hitung_total_absensi` data diambil dari model `Absensi` dengan informasi seperti nama, kode pekerja, tanggal absensi, dan kategori jabatan. Data diubah menjadi `DataFrame` menggunakan `pandas`. Melakukan pengecekan apakah data absensi kosong atau kolom `tanggal_ab` tidak ada. Jika tidak ada data, fungsi akan merender template kosong tanpa melanjutkan proses. Kolom `tanggal_ab` diubah ke format `datetime` untuk memudahkan manipulasi. Data duplikat dihapus berdasarkan kombinasi `tanggal_ab`, `nama_ab`, dan `kode_ab_kode_dp`. Data difilter untuk hanya menyertakan hari kerja Senin hingga Jumat. Ditambahkan kolom `bulan_tahun` untuk menandai bulan dan tahun. Data di group berdasarkan `nama_ab`, `kode_ab_kode_dp`, `bulan_tahun`, dan `ktdp_ab`. Setiap kelompok dihitung jumlah total absensi dalam bulan tersebut. Total hari kerja dihitung dengan `hitung_hari_kerja`. Data disimpan atau diperbarui di tabel `RekapAbsensi` menggunakan `update_or_create`. Data yang disimpan meliputi nama, kode, bulan-tahun, kategori, total absensi, total hari kerja, dan gaji final. Data dari tabel `RekapAbsensi` diambil dan dirender ke template `rekap_absensi.html`.

4.3. Pengujian Sistem

No	Skenario	Bukti Pengujian	Hasil
1.	Jodi Septian, Kepala Dusun. Wajah dapat dikenali Kondisi wajah menghadap kamera Kondisi pecahayaannya yang terang		✓
2.	Rita Anggraeni, Kepala Seksi. Wajah dapat dikenali Kondisi wajah menghadap kamera Kondisi pecahayaannya yang terang		✓
3.	Dian Putri Rahayu, Kepala Seksi. Wajah dapat dikenali Kondisi wajah menghadap kamera Kondisi pecahayaannya yang terang		✓

4.	Jodi Septian, Kepala Dusun. Kondisi wajah menghadap kamera Kondisi pecahayaannya yang terang Terdapat dua wajah yaitu wajah asli dan wajah palsu.		✓
5.	Rita Anggraeni, Kepala Seksi. Kondisi wajah menghadap kamera Kondisi pecahayaannya yang terang Terdapat dua wajah yaitu wajah asli dan wajah palsu.		✓
6.	Dian Putri Rahayu, Kepala Seksi. Kondisi wajah menghadap kamera		✓

	Kondisi pecahayaan yang terang Terdapat dua wajah yaitu wajah asli dan wajah palsu.		
7.	Jodi Septian, Kepala Dusun. Kondisi wajah menghadap kamera Kondisi pecahayaan yang terang Menggunakan wajah palsu atau penitipan absensi, wajah tidak dikenali		✓
8.	Rita Anggraeni, Kepala Seksi. Kondisi wajah menghadap kamera Kondisi pecahayaan yang terang Menggunakan wajah palsu atau penitipan		✓

	absensi, wajah tidak dikenali		
9.	Dian Putri Rahayu, Kepala Seksi. Kondisi wajah menghadap kamera Kondisi pecahaya- an yang terang Menggunakan wajah palsu atau penitipan absensi		✓

Tabel 4. 1 Pengujian Blackbox

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, kesimpulan yang dapat diambil adalah sebagai berikut:

1. Merancang bangun absensi pengenalan wajah dapat memastikan pegawai hadir pada hari kerja dan deteksi kedipan mata dapat menghindari kecurangan penitipan absensi.
2. Penerapan absensi pengenalan wajah menggunakan algoritma LBPH dapat mengidentifikasi wajah pegawai pada hari kerja, sementara library dlib dengan facial landmark untuk deteksi kedipan mata memastikan kehadiran fisik pegawai yang terdeteksi benar-benar hadir.
3. Rekapitulasi absensi dapat langsung diketahui secara real-time dan diunduh dalam format CSV tanpa perlu menunggu waktu.

5.2. Saran

Adapun saran yang dapat diberikan untuk pengembangan selanjutnya adalah sebagai berikut:

1. Menambahkan tampilan grafik untuk menunjukkan jumlah keterlambatan dan hari absen setiap bulan membantu pengguna melihat pola kehadiran dengan lebih cepat dan mudah.
2. Mengembangkan fitur laporan bulanan detail untuk meningkatkan fungsionalitas aplikasi seperti pengajian.
3. Feedback pengguna mengumpulkan umpan balik dari pengguna untuk perbaikan berkelanjutan pada aplikasi, sehingga dapat memenuhi kebutuhan pengguna dengan lebih baik.

DAFTAR PUSTAKA

- Clara, C. (2022). *Implementasi Algoritma Local Binary Pattern Dan Backpropagation Untuk Face Recognition*.
- Detila, Q. M., Eri, D., & Wibowo, P. (2019). *Perbandingan Metode Eigenface, Fisherface, dan LBPH pada Sistem Pengenalan Wajah*.
- Diana, I. (2021). *Analisa Pengaruh Nilai Bobot Heuristik Pada Algoritma Weighted A*.
- Dimasmewa Widyatmoko. (2022). *Prototipe Sistem Pendataan Barang Peralatan Kantor Dengan Menggunakan Bootstrap Berbasis MySql*.
- Dionisa Dero, A. (2019). *Situs Komunitas Dan Unit Kegiatan Mahasiswa Universitas Atma Jaya Yogyakarta Berbasis Web*.
- Dona Marleny, F. (2021). *Pengolahan Citra Digital Menggunakan Python*.
<https://www.researchgate.net/publication/358220979>
- Dwi Kristianto, G. A. (2019). *Perancangan Aplikasi Sistem Informasi Migrasi Data Berbasis Web Framework Django*.
- Irzan, M., & Sutriyono, D. (2021). *Rancang Bangun Sistem Informasi Inventaris Barang Dinas Komunikasi dan Informatika Indragiri Hulu*.
- Ludia Ramadini, F., & Haryatmi, E. (2022). *Penggunaan Metode Haar Cascade classifier dan LBPH Untuk Pengenalan Wajah Secara Realtime*. 6(2).
<https://doi.org/10.30743/infotekjar.v6i2.4714>
- Pratama, E. A., & Febriani, O. M. (2021). *Sistem Informasi Unit Kegiatan Mahasiswa Darmajaya Basketball Asosiasi Berbasis Web*.
- Santi. (2019). *Perancangan Dan Implementasi Sistem Kasir Untuk Toko Pioner*.
- Santo, A. (2022). *Implementasi Time Based One Time Password Menggunakan Challenge Response Authentication Mechanism Untuk Keamanan Transaksi Digital Aplikasi Android*.

Wiratna Eka, R., & Sahputro Aghil. (2021). *Pembuatan Aplikasi Sistem Informasi Penduduk Kelurahan Petungasri*.

Yanu F, M., Yuwono, B., & Boedi P, D. (2022). *Dasar Pengolahan Digital Edisi 2022*.

Zein, A., Raya, J., Serpong, P., 10 Tangerang, N., & Banten, S. (2018). *Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OpenCv dan Dlib Python Real Time Sleepiness Detection Using Opencv Library and Python Dlib* (Vol. 28, Issue 2).

Lampiran 1 Source Code Program

generate_dataset.py

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

from django.shortcuts import render, redirect
from django.http import StreamingHttpResponse, HttpResponse
from aplikasi.models import ImgDataset, DataPekerja, Absensi
from django.db import models
from django.db.models import Max
from django.core.files.base import ContentFile

def data_pekerja(request):
    max_kode_dp = DataPekerja.objects.aggregate(max_kode_dp=Max('kode_dp'))['max_kode_dp'] or 100
    new_kode = str(int(max_kode_dp) + 1)

    return render(request, 'data_pekerja.html', {'newkode': new_kode})

def data_pekerja_submit(request):
    if request.method == 'POST':
        namadp = request.POST.get('namadp')
        kategoridp = request.POST.get('kategoridp')

        max_kode_dp = DataPekerja.objects.aggregate(max_kode_dp=Max('kode_dp'))['max_kode_dp'] or 100
        new_kode = str(int(max_kode_dp) + 1)

        DataPekerja.objects.create(kode_dp=new_kode, nama_dp=namadp, kategori_dp=kategoridp)

    # Redirect menggunakan templat URL Django
    return redirect('vfdataset_page', kode=new_kode)

def vfdataset_page(request, kode):
    return render(request, 'gendataset.html', {'kode': kode})

def generate_dataset(nbr):
    # Inisialisasi Cascade Classifier
    face_cascade = cv2.CascadeClassifier('aplikasi/resources/haarcascade_frontalface_default.xml')

    # Fungsi internal untuk mendeteksi dan memotong wajah dari gambar
    def face_cropped(img):
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)

        if len(faces) == 0:
            return None

        x, y, w, h = faces[0]
        cropped_face = img[y:y+h, x:x+w]

        return cropped_face

    # Mulai tangkapan video dari kamera
    cap = cv2.VideoCapture(0)
    try:
        # Dapatkan ID terakhir dari dataset gambar
        lastid = ImgDataset.objects.aggregate(max_id=models.Max('id_img'))['max_id'] or 0

        id_img = lastid
        max_idimg = id_img + 100
        count_img = 0

        # Peroleh objek DataPekerja berdasarkan nomor yang diberikan
        dp_instance = DataPekerja.objects.get(kode_dp=nbr)

        base_dir = os.path.join("media", "dataset")
        if not os.path.exists(base_dir):
            os.makedirs(base_dir)

        # Loop utama untuk menghasilkan dataset
        while True:
            # img = cap.read()
            face = face_cropped(img)
            if face is not None:
                count_img += 1
                id_img += 1
                face_resized = cv2.resize(face, (280, 280))
                face_gray = cv2.cvtColor(face_resized, cv2.COLOR_BGR2GRAY)
```

```

        # Simpan gambar tanpa teks
        file_name = f'{nbr}.{id_img}.jpg'
        file_name_path = os.path.join(base_dir, file_name)
        cv2.imwrite(file_name_path, face_gray)

        with open(file_name_path, 'rb') as f:
            content = ContentFile(f.read())
            img_instance = ImgDataset.objects.create(id_img=id_img, kode_img=dp_instance)
            img_instance.foto.save(file_name, content)

        # Tambahkan teks pada gambar untuk tampilan streaming
        face_with_text = cv2.putText(face_resized.copy(), str(count_img), (50, 50), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
        frame = cv2.cvtColor(face_with_text, cv2.COLOR_BGR2GRAY)
        encoded_frame = cv2.imencode('.jpg', frame)[1].tobytes()
        yield (b'--frame\r\nb'Content-Type: image/jpeg\r\n\r\n' + encoded_frame + b'\r\n\r\n')

        # Periksa apakah perlu menghentikan proses
        if cv2.waitKey(1) == 13 or int(id_img) == int(max_idimg):
            break

    except Exception as e:
        # Tangani pengecualian (cetak atau catat kesalahan, dll.)
        print(f'Terjadi kesalahan: {e}')

finally:
    # Stop tangkapan video dan bersihkan jendela
    cap.release()
    cv2.destroyAllWindows()

def vidfeed_dataset(request, nbr):
    return StreamingHttpResponse(generate_dataset(nbr), content_type='multipart/x-mixed-replace; boundary=frame')

def identifier(request, idr):
    dataset_dir = os.path.join("media", "dataset")

    paths = [os.path.join(dataset_dir, f) for f in os.listdir(dataset_dir)]
    faces = []
    ids = []

    for image_path in paths:
        file_name = os.path.basename(image_path)
        try:
            kode, id_img, _ = file_name.split("-")
            id_img = int(id_img)
        except ValueError:
            # Skip file if it does not match the expected format
            continue

        img = Image.open(image_path).convert('L')
        image_np = np.array(img, 'uint8')

        faces.append(image_np)
        ids.append(id_img)

    ids = np.array(ids)

    # Train the classifier and save
    idfr = cv2.Face_LBPHFaceRecognizer_create()
    idfr.train(faces, ids)

    # Lokasi penyimpanan file classifier.xml di dalam folder 'aplikasi'
    identifier_path = os.path.join("aplikasi", "identifier.xml")
    idfr.write(identifier_path)

    return redirect('home') # Sesuaikan dengan nama fungsi URL yang merender halaman utama

```

pengenalan_wajah.py

```
import cv2
import dlib
import os
import numpy as np
from numpy.linalg import norm
from datetime import date
from django.http import StreamingHttpResponse
from django.shortcuts import render
from django.utils import timezone
from aplikasi.models import IngDataset, DataPekerja, Absensi
from waktu_utils import tentukan_keterangan_waktu

# Initialize the face and eye detectors
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("aplikasi/resources/shape_predictor_68_face_landmarks.dat")

# Define constants and variables
threshold = 0.2
eye_closed = False
kedip = 'X'
color_kedip = (0, 0, 255)
justscanned = False
pause_cnt = 0
cnt = 0
stream_stop = False

def reset_variables():
    global eye_closed, kedip, color_kedip, justscanned, pause_cnt, cnt, stream_stop
    eye_closed = False
    kedip = 'X'
    color_kedip = (0, 0, 255)
    justscanned = False
    pause_cnt = 0
    cnt = 0
    stream_stop = False

def mid_line_distance(p1, p2, p3, p4):
    p5 = np.array([int((p1[0] + p2[0]) / 2), int((p1[1] + p2[1]) / 2)])
    p6 = np.array([int((p3[0] + p4[0]) / 2), int((p3[1] + p4[1]) / 2)])
    return norm(p5 - p6)

def aspect_ratio(landmarks, eye_range):
    eye = np.array([np.array([landmarks.part(i).x, landmarks.part(i).y]) for i in eye_range])
    B = norm(eye[0] - eye[3])
    A = mid_line_distance(eye[1], eye[2], eye[5], eye[4])
    ear = A / B
    return ear

def detect_eye_blink(frame):
    global eye_closed, kedip, color_kedip
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = detector(gray)

    for rect in faces:
        landmarks = predictor(gray, rect)
        left_aspect_ratio = aspect_ratio(landmarks, range(42, 48))
        right_aspect_ratio = aspect_ratio(landmarks, range(36, 42))
        ear = (left_aspect_ratio + right_aspect_ratio) / 2.0

        # Display the EAR value on the frame
        cv2.putText(frame, "EAR: {:.2f}".format(ear), (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 0, 0), 2)

        if ear < threshold:
            eye_closed = True
        elif ear >= threshold and eye_closed:
            kedip = 'Berhasil'
            color_kedip = (0, 255, 0)
            eye_closed = False

        for n in range(30, 48):
            x = landmarks.part(n).x
            y = landmarks.part(n).y
            cv2.putText(frame, "Pejamian Mata :", (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 0, 0), 2)
            cv2.putText(frame, "{}".format(kedip), (200, 60), cv2.FONT_HERSHEY_SIMPLEX, 0.7, color_kedip, 2)

def generate_frames():
    global justscanned, pause_cnt, cnt, stream_stop

    # Reset variables at the start of frame generation
    reset_variables()
```

```

faceCascade = cv2.CascadeClassifier('aplikasi/resources/haarcascade_frontalface_default.xml')
idfr = cv2.face.LBPHFaceRecognizer_create()

identifier_path = "aplikasi/identifier.xml"
if os.path.exists(identifier_path):
    idfr.read(identifier_path)
    identifier_loaded = True
else:
    identifier_loaded = False

cap = cv2.VideoCapture(0)
cap.set(3, 480)
cap.set(4, 480)

while True:
    success, frame = cap.read()
    if not success:
        break

    if identifier_loaded:
        coords = draw_boundary(frame, faceCascade, 1.1, 10, (255, 255, 0), "Face", idfr)
    else:
        coords = draw_boundary(frame, faceCascade, 1.1, 10, (255, 255, 0), "Face")

    detect_eye_blink(frame)

    # Convert the frame to JPEG format
    ret, buffer = cv2.imencode('.jpg', frame)
    if not ret:
        continue

    frame = buffer.tobytes()

    # Yield the frame as a multipart response
    yield (b'--frame\r\n'
          b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

    if stream_stop:
        break

cap.release()

def draw_boundary(img, faceCascade, scaleFactor, minNeighbors, color, text, idfr=None):
    global justscanned, pause_cnt, cnt, stream_stop
    gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    features = faceCascade.detectMultiScale(gray_image, scaleFactor, minNeighbors)

    pause_cnt += 1
    coords = []

    for (x, y, w, h) in features:
        cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)

    if idfr:
        id, pred = idfr.predict(gray_image[y:y + h, x:x + w])
        confidence = int(100 * (1 - pred / 360))

        # Jika confidence tinggi, wajah dianggap dikenal
        if confidence > 87 and not justscanned:
            cnt += 1
            n = (100 / 30) * cnt
            w_filled = (cnt / 30) * w

            cv2.putText(img, str(int(n)) + ' %', (x + 10, y + h + 20), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (155, 255, 255), 2, cv2.LINE_AA)
            cv2.rectangle(img, (x, y + h + 40), (x + w, y + h + 50), color, 2)
            cv2.rectangle(img, (x, y + h + 40), (x + int(w_filled), y + h + 50), (155, 255, 255), cv2.FILLED)

            try:
                # Cek data wajah di database IngDataset
                img_data = IngDataset.objects.get(id_img=id)
                data = img_data.kode_img
                dkode, dnama, dkategori = data.kode_dp, data.nama_dp, data.kategori_dp
            except IngDataset.DoesNotExist:
                # Jika data wajah tidak ditemukan di database
                dkode = 'Unknown in Dataset'
                dnama = 'Unknown in Dataset'
                dkategori = 'Unknown in Dataset'
            cv2.putText(img, 'Not Found in Database', (x - 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA)

```

```

# Jika pengenalan wajah berhasil, baru deteksi kedipan mata dilakukan
if cnt == 30:
    cnt = 0
    if dkode != 'Unknown in Dataset': # Hanya lakukan absensi jika wajah ditemukan
        try:
            data_instance = DataPekerja.objects.get(kode_dp=dkode)
            keterangan = tentukan_keterangan_waktu()

            # Hanya jika kedipan mata terdeteksi sebagai 'Berhasil'
            if kadip == 'Berhasil':
                Absensi.objects.create(
                    tanggal_ab=date.today(),
                    kode_ab=data_instance.kode_dp,
                    nama_ab=data_instance.nama_dp,
                    ktdp_ab=data_instance.kategori_dp,
                    waktu_ab=timezone.localtime(),
                    keterangan_ab=keterangan
                )

                cv2.putText(img, dnama + ' | ' + dkategori, (x - 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (155, 255, 255), 2, cv2.LINE_AA)
                cv2.putText(img, keterangan, (x - 10, y + 40), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (155, 255, 255), 2, cv2.LINE_AA)
                justscanned = True
                pause_cnt = 0
                stream_stop = True
            else:
                cv2.putText(img, 'Eye Verification Failed', (x - 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA)
                stream_stop = True

        except DataPekerja.DoesNotExist:
            cv2.putText(img, 'Not Found in Worker DB', (x - 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA)
        else:
            # Jika wajah tidak dikenali di dataset
            cv2.putText(img, 'Unknown in Dataset', (x - 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA)
            stream_stop = True

    else:
        # Jika confidence rendah (wajah tidak dikenali), tampilkan pesan
        if not justscanned:
            cv2.putText(img, 'Unknown (Low Confidence)', (x, y + 5), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA)

    coords.append((x, y, k, h))
    return coords

def video_feed(request):
    reset_variables() # Reset variables before starting the feed
    return StreamingHttpResponse(generate_frames(), content_type='multipart/x-mixed-replace; boundary=frame')

```

rekap_absensi.py

```

from django.shortcuts import render
from aplikasi.models import Absensi, RekapAbsensi
import pandas as pd
import calendar

# Fungsi untuk menghitung jumlah hari kerja dalam satu bulan tertentu
def hitung_hari_kerja(bulan_tahun):
    tahun, bulan = map(int, bulan_tahun.split('-')) # Memisahkan tahun dan bulan
    hari_kerja = 0 # Inisialisasi jumlah hari kerja
    for week in calendar.monthcalendar(tahun, bulan): # Mendapatkan kalender bulanan dalam bentuk matriks
        for day in week[:5]: # Iterasi dari hari Senin hingga Jumat
            if day != 0: # Hanya hitung hari kerja yang valid (bukan 0)
                hari_kerja += 1
    return hari_kerja # Mengembalikan jumlah hari kerja

salary_data = {
    "Kamu": {"gaji_perbulan": 4000000, "potongan_perhari": 150000},
    "Sekertaris Desa": {"gaji_perbulan": 2000000, "potongan_perhari": 100000},
    "Kepala Seksi": {"gaji_perbulan": 2250000, "potongan_perhari": 70000},
    "Kepala Urusan": {"gaji_perbulan": 2250000, "potongan_perhari": 70000},
    "Kepala Dusun": {"gaji_perbulan": 2250000, "potongan_perhari": 70000},
    "Staff": {"gaji_perbulan": 2250000, "potongan_perhari": 70000},
}

# Fungsi untuk menghitung total absensi dan mengupdate tabel RekapAbsensi
def hitung_total_absensi(request):
    # Mengambil data absensi dari database
    data_absensi = Absensi.objects.values('nama_ab', 'kode_ab_kode_dp', 'tanggal_ab', 'ktdp_ab')

```

```

# Debug: Periksa data absensi yang diambil
print("Data Absensi:", list(data_absensi)) # Baris debugging

# Membuat DataFrame dari data absensi
df = pd.DataFrame(list(data_absensi))

# Debug: Periksa kolom dalam DataFrame
print("Columns in DataFrame:", df.columns) # Baris debugging

if df.empty or 'tanggal_ab' not in df.columns:
    # Menangani kasus jika tidak ada data atau kolom tidak ada
    return render(request, 'rekap_absensi.html', {'rekap_list': []})

# Mengubah kolom 'tanggal_ab' menjadi tipe datetime
df['tanggal_ab'] = pd.to_datetime(df['tanggal_ab'])
# Menghapus duplikat berdasarkan kombinasi tanggal, nama, dan kode
df = df.drop_duplicates(subset=['tanggal_ab', 'nama_ab', 'kode_ab_kode_dp'], keep='first')
# Memfilter hanya hari kerja (Senin hingga Jumat)
df = df[df['tanggal_ab'].dt.dayofweek < 5]
# Menambahkan kolom 'bulan_tahun' dengan format periode 'YYYY-MM'
df['bulan_tahun'] = df['tanggal_ab'].dt.to_period('M')

# Menghitung jumlah absensi per kombinasi nama, kode, bulan-tahun, dan kategori
jumlah_data = df.groupby(['nama_ab', 'kode_ab_kode_dp', 'bulan_tahun', 'ktdp_ab']).size().reset_index(name='total_absensi')

for idx, row in jumlah_data.iterrows():
    # Mengonversi periode ke format string 'YYYY-MM'
    bulan_tahun_str = row['bulan_tahun'].strftime('%Y-%m')
    # Menghitung total hari kerja untuk bulan dan tahun tersebut
    total_hari_kerja = hitung_hari_kerja(bulan_tahun_str)
    kategori = row['ktdp_ab']
    total_absensi = row['total_absensi']

    # Calculate deductions
    gaji_perbulan = salary_data[kategori]['gaji_perbulan']
    potongan_perhari = salary_data[kategori]['potongan_perhari']
    potongan = (total_hari_kerja - total_absensi) * potongan_perhari
    gaji_final = gaji_perbulan - potongan

    # Membuat atau memperbarui entri RekapAbsensi dengan data baru
    obj, created = RekapAbsensi.objects.update_or_create(
        kode_barrow=['kode_ab_kode_dp'],
        nama_barrow=['nama_ab'],
        bulan_tahun=bulan_tahun_str,
        kategori_rekategorikategori,
        defaults={
            'total_absensi': total_absensi,
            'total_hari_kerja': total_hari_kerja,
            'gaji_final': gaji_final # Save final salary
        }
    )

# Mengambil semua data RekapAbsensi untuk ditampilkan di template
rekap_list = RekapAbsensi.objects.all()
return render(request, 'rekap_absensi.html', {'rekap_list': rekap_list})

```

Penjelasan Program LBPH Dan Deteksi Kedipan Mata

```
import cv2

# Memuat classifier Haar Cascade untuk deteksi wajah
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Memuat gambar dari file (ganti path-nya sesuai dengan file gambar yang
ingin diuji)
img = cv2.imread('103rgb.jpg')

# Mengonversi gambar menjadi grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Mendeteksi wajah pada gambar
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30))

# Memproses setiap wajah yang terdeteksi
for (x, y, w, h) in faces:
    # Memotong bagian wajah dari gambar berdasarkan koordinat yang
ditemukan
    face = gray[y:y+h, x:x+w]

    # Mengubah ukuran wajah menjadi 200x200 piksel
    resized_face = cv2.resize(face, (200, 200))

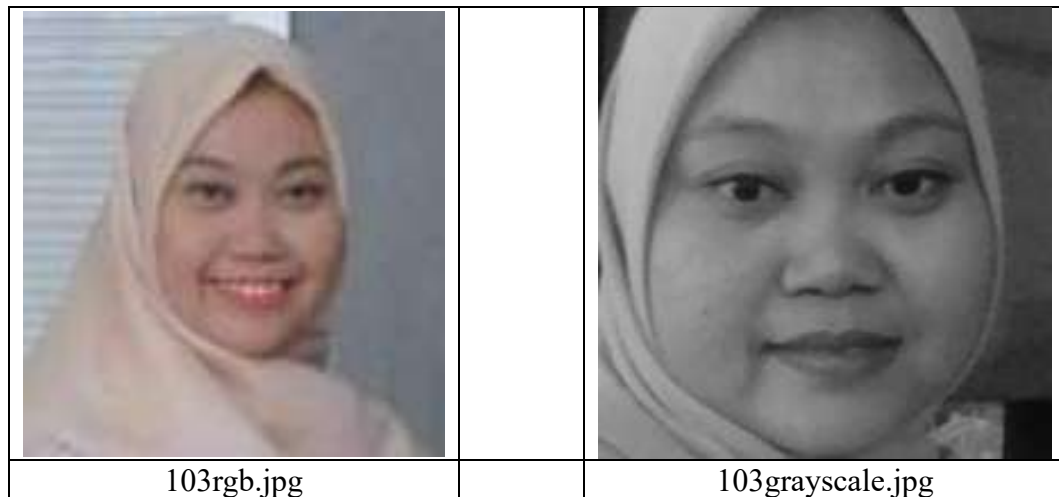
    # Menampilkan gambar wajah yang telah diubah ukurannya
    cv2.imshow('Resized Face', resized_face)

    # Menyimpan gambar wajah yang telah diubah ukurannya ke file
    cv2.imwrite('103grayscale.jpg', resized_face)

# Menunggu input dari pengguna sebelum menutup jendela tampilan
cv2.waitKey(0)
cv2.destroyAllWindows()
```

haarcascade_frontalface_default.xml file XML yang berisi model deteksi classifier yang telah dilatih menggunakan algoritma *Haar Cascade* untuk mendeteksi wajah manusia dari tampak depan. File ini merupakan bagian dari pustaka OpenCV yang sering digunakan dalam deteksi wajah.

Wajah RGB yang terdeteksi diubah menjadi gambar grayscale dan dipotong menjadi ukuran 200x200 piksel.



```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Membaca gambar dalam format grayscale
# Parameter cv2.IMREAD_GRAYSCALE memastikan gambar dibaca sebagai grayscale
gambar = cv2.imread('103grayscale.jpg', cv2.IMREAD_GRAYSCALE)

# Memastikan ukuran gambar adalah 200x200 piksel
# Jika tidak, akan muncul pesan error untuk memberitahu bahwa ukuran gambar tidak sesuai
if gambar.shape != (200, 200):
    raise ValueError("Gambar harus berukuran 200x200 piksel.")

# Menentukan ukuran sel atau blok yang diinginkan
# Dalam hal ini, ukuran blok yang diambil adalah 3x3 piksel
blok_size = 3

# Fungsi untuk menampilkan dan mencetak sel-sel (blok) dari gambar
# Fungsi ini akan membagi gambar ke dalam blok-blok kecil berukuran blok_size x blok_size
def tampilkan_sel_sel(gambar, blok_size):
    tinggi, lebar = gambar.shape # Mengambil dimensi gambar (tinggi dan lebar)

    # Iterasi untuk setiap blok dalam gambar berdasarkan blok_size
```

```

# Loop ini akan memproses blok secara horizontal dan vertikal
for y in range(0, tinggi - blok_size + 1, blok_size):
    for x in range(0, lebar - blok_size + 1, blok_size):
        # Mengambil area blok berukuran blok_size x blok_size dari koordinat
        (x, y)
        area_blok = gambar[y:y+blok_size, x:x+blok_size]

        # Menampilkan informasi blok berupa koordinat awal dari blok
        print(f"Blok mulai dari koordinat ({y}, {x}):")
        print(area_blok) # Menampilkan nilai piksel di dalam blok

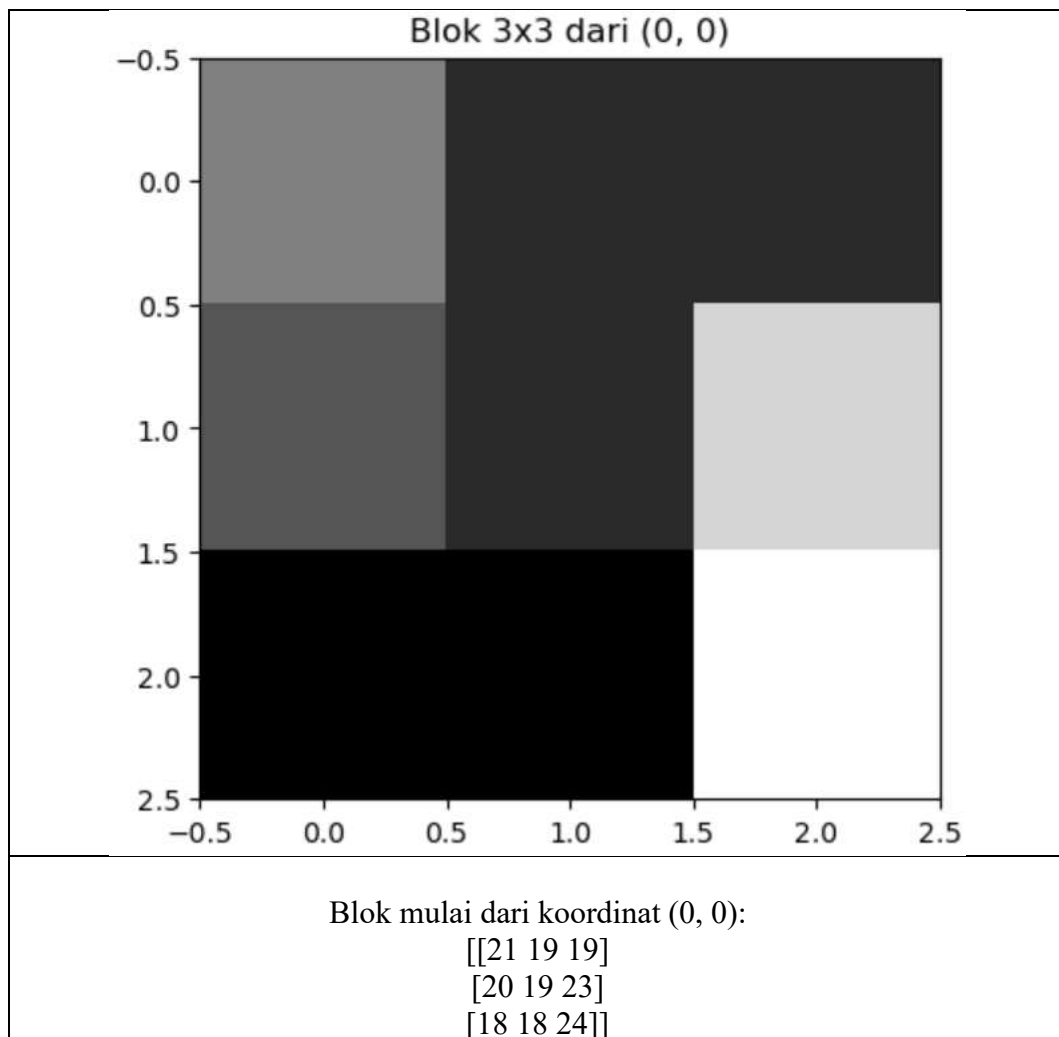
        # Menampilkan blok menggunakan matplotlib
        plt.imshow(area_blok, cmap='gray') # Menampilkan blok dalam skala
        abu-abu
        plt.title(f'Blok {blok_size}x{blok_size} dari ({y}, {x})') # Judul berisi
        koordinat blok
        plt.show() # Menampilkan jendela blok

# Memanggil fungsi untuk menampilkan seluruh blok 3x3 dari gambar
tampilkan_sel_sel(gambar, blok_size)

```

Membagi citra wajah menjadi bagian-bagian kecil yang disebut dengan sel, dimana tiap sel berukuran 3x3 piksel.

Tentukan nilai pusat dari tiap sel yaitu piksel tengah untuk digunakan sebagai nilai ambang. Nilai tersebut digunakan untuk menentukan nilai-nilai baru dari ke 8 pixel disekitarnya piksel tetangga, nilai matrix pixel 3x3 sebagai berikut



```
import numpy as np

# Definisikan matriks 3x3
matrix = np.array([[21, 19, 19],
                  [20, 19, 23],
                  [18, 18, 24]])

# Ambil nilai tengah (piksel tengah) sebagai ambang
threshold = matrix[1, 1] # Piksel tengah (baris 1, kolom 1) adalah 19

# Buat matriks biner baru berdasarkan aturan
binary_matrix = np.zeros((3, 3), dtype=int)

# Loop untuk memeriksa setiap tetangga
for i in range(3):
```

```

for j in range(3):
    if matrix[i, j] >= threshold:
        binary_matrix[i, j] = 1
    else:
        binary_matrix[i, j] = 0

# Cetak hasilnya
print("Matrix asli:")
print(matrix)
print("Matrix biner berdasarkan aturan:")
print(binary_matrix)

```

Untuk setiap piksel tetangga dari piksel tengah, akan ditetapkan nilai biner baru dengan aturan 1 untuk nilai yang sama atau lebih tinggi dari ambang dan 0 untuk nilai lebih rendah dari ambang.

```

Matrix asli:
[[21 19 19]
 [20 19 23]
 [18 18 24]]

```

```

Matrix biner berdasarkan aturan:
[[1 1 1]
 [1 1 1]
 [0 0 1]]

```

```

import numpy as np

# Matriks biner
binary_matrix = np.array([[1, 1, 1],
                          [1, 1, 1],
                          [0, 0, 1]])

# Definisikan urutan tetangga sesuai pola (7, 6, 5, 8, 4, 1, 2, 3)
# Ambil nilai biner dari posisi tetangga
neighbors = [binary_matrix[0, 0], binary_matrix[0, 1], binary_matrix[0, 2], # 7,
            6, 5
            binary_matrix[1, 0], binary_matrix[1, 2], # 8, 4
            binary_matrix[2, 0], binary_matrix[2, 1], binary_matrix[2, 2]] # 1, 2, 3

```

```
# Gabungkan nilai biner menjadi satu string
binary_string = ".join(map(str, neighbors))

# Konversi string biner ke desimal
decimal_value = int(binary_string, 2)

# Tampilkan hasilnya
print("Nilai biner:", binary_string)
print("Nilai desimal:", decimal_value)
```

Nilai binery 11111001 dan desimalnya 249. Setelah diubah menjadi nilai biner maka akan menghasilkan deret biner yang kemudian jika diubah menjadi nilai decimal yang akan menghasilkan nilai baru untuk nilai tengah pixel selanjutnya. Hasil dari deret biner ini disimpan didalam file identifier.xml yang akan digunakan untuk proses pengenalan wajah.

Nilai biner: 11111001
Nilai desimal: 249

7 6 5 4 3 2 1 0

1 1 1 1 1 0 0 1 → 1111001

Setiap angka 1 atau 0 dalam bilangan biner ini mempresentasikan apakah pangkat dari 2 di posisi itu dihitung atau tidak. Jika angka 1, maka pangkat 2 di posisi itu dihitung. Jika angka 0 maka tidak dihitung.

$$\text{Bit ke 7} \rightarrow 1 \quad 2^7 = 128 \times 1 = 128$$

$$\text{Bit ke 6} \rightarrow 1 \quad 2^6 = 64 \times 1 = 64$$

$$\text{Bit ke 5} \rightarrow 1 \quad 2^5 = 32 \times 1 = 32$$

$$\text{Bit ke 4} \rightarrow 1 \quad 2^4 = 16 \times 1 = 16$$

$$\text{Bit ke 3} \rightarrow 1 \quad 2^3 = 8 \times 1 = 8$$

$$\text{Bit ke 2} \rightarrow 0 \quad \text{tidak dihitung} = 0 = 0$$

$$\text{Bit ke 1} \rightarrow 0 \quad \text{tidak dihitung} = 0 = 0$$

$$\text{Bit ke 0} \rightarrow 1 \quad 2^0 = 1 \times 1 = 1$$

$2^2 \cdot 1$

299

Jadi hasil konversi dari biner 1111001 adalah 299 dalam desimal.

```

import cv2
import dlib

# Memuat detektor wajah dan prediktor bentuk wajah
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

def get_eye_landmarks(image):
    # Mengonversi gambar ke grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Mendeteksi wajah dalam gambar grayscale
    faces = detector(gray)

    # Daftar untuk menyimpan koordinat mata
    eye_landmarks = {'left_eye': [], 'right_eye': []}

    # Mengiterasi setiap wajah yang terdeteksi
    for face in faces:
        # Mendapatkan titik-titik landmark wajah
        landmarks = predictor(gray, face)

        # Menyaring landmark mata (mata kiri: 42-47, mata kanan: 36-41)
        left_eye = [landmarks.part(i) for i in range(42, 48)]
        right_eye = [landmarks.part(i) for i in range(36, 42)]

        # Mengonversi landmark menjadi koordinat
        eye_landmarks['left_eye'] = [(point.x, point.y) for point in left_eye]
        eye_landmarks['right_eye'] = [(point.x, point.y) for point in right_eye]

        # Menggambar landmark pada gambar (untuk visualisasi)
        for (x, y) in eye_landmarks['left_eye']:
            cv2.circle(image, (x, y), 2, (0, 255, 0), -1) # Titik mata kiri dengan warna
hijau
        for (x, y) in eye_landmarks['right_eye']:
            cv2.circle(image, (x, y), 2, (0, 0, 255), -1) # Titik mata kanan dengan
warna merah

    return eye_landmarks, image

# Memuat gambar contoh
image = cv2.imread("103grayscale.jpg")

# Mendapatkan landmark mata
eye_landmarks, image_with_landmarks = get_eye_landmarks(image)

```

```
# Menampilkan gambar dengan landmark
cv2.imshow("Eye Landmarks", image_with_landmarks)
cv2.waitKey(0)
cv2.destroyAllWindows()

# Mencetak landmark mata dengan indeks
print("Landmark Mata Kiri (42-47):", eye_landmarks['left_eye'])
print("Landmark Mata Kanan (36-41):", eye_landmarks['right_eye'])
```

shape_predictor_68_face_landmarks.dat adalah file model yang digunakan oleh dlib untuk mendeteksi 68 titik landmark pada wajah manusia.

```
Landmark Mata Kiri (42-47): [(122, 81), (131, 74), (141, 73), (151, 77), (143, 82), (132, 83)]
Landmark Mata Kanan (36-41): [(52, 79), (61, 76), (71, 76), (81, 83), (71, 85), (60, 85)]
```

```
import cv2
import dlib

# Memuat detektor wajah dan prediktor bentuk wajah
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

def get_eye_landmarks(image):
    # Mengonversi gambar ke grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Mendeteksi wajah dalam gambar grayscale
    faces = detector(gray)

    # Daftar untuk menyimpan koordinat mata
    eye_landmarks = {'right_eye': []}

    # Mengiterasi setiap wajah yang terdeteksi
    for face in faces:
        # Mendapatkan titik-titik landmark wajah
        landmarks = predictor(gray, face)
```

```

# Menyaring landmark mata kanan (indeks 36-41)
right_eye = [landmarks.part(i) for i in range(36, 42)]

# Mengonversi landmark menjadi koordinat
eye_landmarks['right_eye'] = [(point.x, point.y) for point in right_eye]

# Menggambar landmark pada gambar (untuk visualisasi)
for (x, y) in eye_landmarks['right_eye']:
    cv2.circle(image, (x, y), 2, (0, 0, 255), -1) # Titik mata kanan dengan
warna merah

return eye_landmarks, image

# Memuat gambar contoh
image = cv2.imread("103grayscale.jpg")

# Mendapatkan landmark mata
eye_landmarks, image_with_landmarks = get_eye_landmarks(image)

# Menampilkan gambar dengan landmark
cv2.imshow("Eye Landmarks", image_with_landmarks)
cv2.waitKey(0)
cv2.destroyAllWindows()

# Mencetak landmark mata kanan dengan indeks
print("Landmark Mata Kanan (36-41):")
print("36:", eye_landmarks['right_eye'][0]) # Koordinat landmark 36
print("37:", eye_landmarks['right_eye'][1]) # Koordinat landmark 37
print("38:", eye_landmarks['right_eye'][2]) # Koordinat landmark 38
print("39:", eye_landmarks['right_eye'][3]) # Koordinat landmark 39
print("40:", eye_landmarks['right_eye'][4]) # Koordinat landmark 40
print("41:", eye_landmarks['right_eye'][5]) # Koordinat landmark 41

```

EAR dihitung dari jarak antara landmark mata. Rumusnya menggunakan jarak vertikal dan horizontal mata. Jika nilai EAR turun di bawah threshold (0.2), mata dianggap tertutup. Jika EAR kembali di atas threshold setelah sebelumnya di bawah, maka mata dianggap telah berkedip.

Landmark Mata Kanan (36-41):
36: (52, 79)

```
37: (61, 76)
38: (71, 76)
39: (81, 83)
40: (71, 85)
41: (60, 85)
```

```
import numpy as np
from numpy.linalg import norm

# Fungsi untuk menghitung jarak antara dua garis tengah
def mid_line_distance(p1, p2, p3, p4):
    # Menghitung titik tengah dari p1 dan p2
    p5 = np.array([int((p1[0] + p2[0]) / 2), int((p1[1] + p2[1]) / 2)])
    # Menghitung titik tengah dari p3 dan p4
    p6 = np.array([int((p3[0] + p4[0]) / 2), int((p3[1] + p4[1]) / 2)])
    # Menghitung jarak antara titik-titik tengah
    return norm(p5 - p6)

# Fungsi untuk menghitung Aspect Ratio Mata (EAR)
def aspect_ratio(landmarks, eye_range):
    # Mengonversi landmarks mata menjadi array NumPy
    eye = np.array([np.array([landmarks[i][0], landmarks[i][1]]) for i in
eye_range])
    # Jarak antara titik-titik 36 dan 39 (horizontal)
    B = norm(eye[0] - eye[3])
    # Jarak vertikal antara titik-titik tengah dari 37-38 dan 40-41
    A = mid_line_distance(eye[1], eye[2], eye[5], eye[4])
    # Menghitung Aspect Ratio Mata (EAR) sebagai perbandingan A/B
    ear = A / B
    return ear

# Landmark Mata Kanan
right_eye_landmarks = [(52, 79), (61, 76), (71, 76), (81, 83), (71, 85), (60, 85)]

# Menghitung EAR
ear = aspect_ratio(right_eye_landmarks, range(0, 6)) # Menggunakan indeks 0-
5 untuk mata kanan
print(f'EAR: {ear:.2f}')
```

EAR bernilai 0.31 dengan threshold 0.2. Semakin tinggi nilai EAR, maka mata semakin terbuka. Sebaliknya, semakin rendah nilai EAR, mata semakin tertutup.

EAR: 0.31

$$EAR = \frac{9.01}{29.28} = 0.31$$
$$EAR = \frac{|(37+38) - (41+40)|}{2 \times |36 - 39|} \rightarrow \frac{A}{B} =$$

36 $\rightarrow$ (52, 79) Sudut $\rightarrow B$
37 $\rightarrow$ (61, 76) } Atas $> A$
38 $\rightarrow$ (71, 76) }
39 $\rightarrow$ (81, 83) Sudut $\rightarrow B$
40 $\rightarrow$ (71, 85) } titik bawah $> A$
41 $\rightarrow$ (60, 85) }

$$B = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
$$= \sqrt{(81 - 52)^2 + (83 - 79)^2}$$
$$= \sqrt{(29)^2 + (4)^2}$$
$$= \sqrt{841 + 16}$$
$$= \sqrt{857}$$
$$= 29.28$$

$29 \times 29 = 841$
 $29.28 \times 29.28 = 857.1584$
 $30 \times 30 = 900$

$$\textcircled{g} \Rightarrow = \frac{61+71}{2}, \frac{76+76}{2}$$
$$= \frac{132}{2}, \frac{152}{2} = (66, 76)$$
$$\textcircled{f} = \frac{60+71}{2}, \frac{85+85}{2}$$
$$= \frac{131}{2}, \frac{170}{2} = (65.5, 85)$$
$$A = \sqrt{(66 - 65.5)^2 + (76 - 85)^2}$$
$$= \sqrt{(0.5)^2 + (-9)^2}$$
$$= \sqrt{0.25 + 81}$$
$$= \sqrt{81.25}$$
$$= 9.01$$

Lampiran 2 Halaman Website





Kelurahan Kertawinangun Cirebon

Registrasi AbsensiAbsensiRekap AbsensiLogout

Hari Ini

Terlambat/Jam Kerja

Dian Putri Rahayu | Kepala Se

100%



Kode	Nama	Kategori	Tanggal	Waktu	Keterangan
101	Dian Putri Rahayu	Kepala Seksi	2024-08-21	10:20:52.428	Tertambat/Jam Kerja

01

Kelurahan Kertawinangun Cirebon

Registrasi AbsensiAbsensiRekap AbsensiLogout

Rekap Absensi

Kode	Nama	Kategori	Bulan-Tahun	Total Absensi	Total Hari Kerja
101	Dian Putri Rahayu	Kepala Seksi	2024-08	1	22
104	Jodi Septian	Kepala Dusun	2024-08	1	22
103	Rita Anggraeni	Kepala Dusun	2024-08	1	22

Administration Kelurahan Kertawinangun

WELCOME ADMINVIEW SITE / CHANGE PASSWORD / LOG OUT

HomeAplikasiRekap absensi

Start typing to filter...

APLIKASI

AbsensiAdd

Data pekerjaAdd

Img datasetsAdd

PekerjaAdd

Rekap absensiAdd

Select rekap absensi to change

ADD REKAP ABSENSI

Action:Export to CSVGo0 of 3 selected

☐	ID RA	KODE RA	NAMA RA	BULAN TAHUN	KATEGORI RA	TOTAL ABSENSI	TOTAL HARI KERJA	GAJI FINAL
☐	4	103	Rita Anggraeni	2024-08	Kepala Dusun	1	22	780000
☐	3	104	Jodi Septian	2024-08	Kepala Dusun	1	22	780000
☐	1	101	Dian Putri Rahayu	2024-08	Kepala Seksi	1	22	780000

Lampiran 3 Dokumentasi Penelitian

