

**METODE LOCAL BINARY PATTERN HISTOGRAM (LBPH)
PENGENALAN WAJAH PADA SISTEM ABSENSI ONLINE
KARYAWAN RADAR CIREBON**



SKRIPSI

Program Strata Satu (S1)

Program Studi Teknik Informatika

Disusun Oleh :

DANNY FACHRUL ALIANSYAH NURDIN

14517408

SEKOLAH TINGGI ILMU KOMPUTER

(STIKOM) POLTEK CIREBON

2021

LEMBAR PENGESAHAN

METODE LOCAL BINARY PATTERN HISTOGRAM (LBPH)
Pengenalan Wajah pada Sistem Absensi Online
Karyawan Radar Cirebon



DANNY FACHRUL ALIANSYAH NURDIN
NRP 14516864

DIKETAHUI DAN DISETUJUI OLEH :

Pembimbing I

Pembimbing II

Faisal Akbar, M.T

Yuhano, M.Kom

Pembantu Ketua I

Ketua Program Studi

Faisal Akbar, M.T

Cucu Handayani, M.Kom

Ketua
STIKOM Poltek Cirebon

Yuhano, M.Kom

LEMBAR PENGESAHAN SIDANG

METODE LOCAL BINARY PATTERN HISTOGRAM (LBPH)
Pengenalan Wajah pada Sistem Absensi Online
Karyawan Radar Cirebon



DANNY FACHRUL ALIANSYAH NURDIN
NRP 14517408

DIKETAHUI DAN DISETUJUI OLEH :

Pembimbing I

Pembimbing II

Faisal Akbar, M.T

Yuhano, M.Kom

Pembantu Ketua I

Ketua Program Studi

Faisal Akbar, M.T

Cucu Handayani, M.Kom

Ketua
STIKOM Poltek Cirebon

Yuhano, M.Kom

PERNYATAAN KEASLIAN TUGAS AKHIR

Sebagai Mahasiswa/i STIKOM Poltek Cirebon yang bertanggung jawab, di bawah ini saya:

Nama : Danny Fachrul Aliansyah Nurdin

NRP : 14517408

Menyatakan bahwa tugas akhir (skripsi) saya yang berjudul:

“METODE LOCAL BINARY PATTERN HISTOGRAM (LBPH) PENGENALAN WAJAH PADA SISTEM ABSENSI OLINE KARYAWAN RADAR CIREBON” merupakan tugas akhir saya (kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya dan perangkat pendukung).

Apabila di kemudian hari, tugas akhir saya disinyalir bukan merupakan karya asli saya, yang disertai dengan bukti-bukti yang cukup, maka saya bersedia untuk membatalkan gelar saya beserta hak dan kewajiban yang melekat pada gelar tersebut.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Cirebon
Pada tanggal : 14 September 2021

Yang Menyatakan,

DANNY FACHRUL ALIANSYAH
NURDIN
NRP: 14517408

PERNYATAAN PERSETUJUAN PUBLIKASI ILMIAH UNTUK KEPENTINGAN AKADEMIK

Sebagai maha siswa STIKOM Poltek Cirebon, yang bertanda tangan di bawah ini saya:

Nama : Danny Fachrul Aliansyah Nurdin

NRP : 14517408

Demi mengembangkan ilmu pengetahuan, menyetujui untuk memberikan kepada STIKOM Poltek Cirebon Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas tugas akhir (skripsi) saya yang berjudul:

“METODE LOCAL BINARY PATTERN HISTOGRAM (LBPH) PENGENALAN WAJAH PADA SISTEM ABSENSI ONLINE KARYAWAN RADAR CIREBON”. Dengan Hak Bebas Royalti Non-Eksklusif ini STIKOM Poltek Cirebon berhak untuk menyimpan, men-copy ulang (memperbanyak), menggunakan, mengelolanya dalam bentuk pangkalan data (*database*), mendistribusikan dan mempublikasannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Cirebon
Pada tanggal : 14 September 2021

Yang Menyatakan,

DANNY FACHRUL ALIANSYAH
NURDIN
NRP: 14517897

MOTTO

***“Jadilah Besar Meskipun Kecil Dan Tetaplah Hidup
Walau Tak Berguna”***

DAFTAR RIWAYAT HIDUP

NAMA : DANNY FACHRUL ALIANSYAH NURDIN
N R P : 14517408
PROGRAM STUDI : TEKNIK INFORMATIKA
TEMPAT LAHIR : KUNINGAN
TANGGAL LAHIR : 27 JANUARI 1999
JENIS KELAMIN : LAKI-LAKI
AGAMA : ISLAM
ALAMAT : JL. JENDERAL SUDIRMAN, RT 04 RW 02, DESA
WANAYASA, KECAMATAN BEBER,
KABUPATEN CIREBON, JAWA BARAT 45172

NAMA ORANG TUA

IBU : YETI NURHAYATI
AYAH : CAHARUDIN
EMAIL : dannyfachrul@gmail.com
NO. HP : 0895155155490

RIWAYAT PENDIDIKAN

1.SD : SDN 1 WANAYASA (2005 – 2011)
2.SMP : SMPN 1 CILIMUS (2011 – 2014)
3.SMA : SMAN 1 CILIMUS (2014 – 2017)

JUDUL SKRIPSI : METODE LOCAL BINARY PATTERN
HISTOGRAM (LBPH) PENGENALAN WAJAH
PADA SISTEM ABSENSI ONLINE KARYAWAN
RADAR CIREBON

Cirebon, 14 September 2021

DANNY FACHRUL ALIANSYAH NURDIN

ABSTRAK

ABSTRACT

KATA PENGANTAR

DAFTAR ISI

LEMBAR PENGESAHAN	ii
LEMBAR PENGESAHAN SIDANG	iii
PERNYATAAN KEASLIAN TUGAS AKHIR	iv
PERNYATAAN PERSETUJUAN PUBLIKASI ILMIAH UNTUK KEPENTINGAN AKADEMIK	v
MOTTO	vi
DAFTAR RIWAYAT HIDUP	vii
ABSTRAK	viii
ABSTRACT	ix
KATA PENGANTAR.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL	xv
DAFTAR GAMBAR.....	xvi
DAFTAR LAMPIRAN	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	4
1.3 Batasan Masalah.....	5
1.4 Tujuan dan Manfaat.....	5
1.4.1 Tujuan	5
1.4.2 Manfaat	6
1.5 Metode dan Teknik Penelitian.....	7
1.5.1 Metode Penelitian.....	7

1.5.2	Teknik Penelitian	7
1.6	Anggapan Dasar	8
1.7	Kerangka Pemikiran	10
1.8	Sistematika Penulisan	12
1.9	Lokasi Penelitian dan Jadwal Penelitian	13
1.9.1	Lokasi Penelitian	13
1.9.2	Jadwal Penelitian	13
BAB II LANDASAN TEORI		15
2.1	Tinjauan Pustaka	15
2.2	Teori Utama Penelitian	20
2.2.1	Kecerdasan Buatan (Artificial Intelligence)	20
2.2.2	Local Binary Pattern Histogram (LBPH)	21
2.2.3	OpenCV	26
2.2.4	Computer Vision	27
2.2.5	Python	27
2.2.6	Amazon Web Service (AWS)	27
2.3	Metode Pengembangan Perangkat Lunak	28
2.3.1	Waterfall	28
2.3.2	Tahapan-tahapan Waterfall	29
2.4	Tools Perancangan	31
2.4.1	Flowchart	31
2.4.2	Flowmap	33
2.4.3	Data Flow Diagram (DFD)	34
2.5	Tools Perangkat Lunak	36
2.5.1	Visual Studio Code	36

2.5.2	AWS Lambda.....	36
2.5.3	Amazon Simple Cloud Storage (S3).....	36
2.5.4	Amazon DynamoDB	37
2.5.5	Amazon Elastic Compute Cloud (EC2)	38
2.5.6	NodeJs	39
2.5.7	React Js.....	39
BAB III ANALISIS SISTEM.....		40
3.1	Gambaran Organisasi	40
3.1.1	Sejarah Singkat.....	40
3.1.2	Struktur Organisasi dan Uraian Tugas	41
3.2	Analisis Sistem Berjalan	43
3.2.1	Uraian Prosedur.....	43
3.2.2	Diagram Sistem Prosedur.....	44
3.2.3	Analisa Dokumen.....	47
3.2.4	Desain Informasi	48
BAB IV PERANCANGAN SISTEM.....		50
4.1	Desain Prosedur.....	50
4.1.1	Prosedur Pendaftaran Pengguna.....	50
4.1.2	Prosedur Login Pengguna	51
4.1.3	Prosedur Kelola Akun Pengguna	51
4.1.4	Prosedur Melatih Model.....	52
4.1.5	Prosedur Memprediksi Wajah.....	58
4.2	Desain Dokumen dan informasi	60
4.2.1	Desain Dokumen.....	60
4.2.2	Desain Informasi	61

4.3	Desain Aliran Data	61
4.3.1	Data Flow Diagram	62
4.3.2	Flowmap.....	63
4.3.3	Flowchart	67
4.3.4	Skema Database	69
4.4	Desain Interface dan Struktur Menu.....	71
4.4.1	Desain Interface	71
4.4.2	Struktur Menu	75
4.5	Implementasi Sistem	75
4.5.1	Konfigurasi Perangkat Lunak	75
4.5.2	Konfigurasi Perangkat Keras	76
4.5.3	Implementasi Program	76
4.5.4	Pedoman Pengoprasian Perangkat Lunak	81
4.5.5	Pengujian.....	85
4.5.6	Evaluasi Hasil Pengujian.....	96
BAB V KESIMPULAN DAN SARAN		98
5.1	Kesimpulan.....	98
5.2	Saran	98
DAFTAR PUSTAKA		99
LAMPIRAN-LAMPIRAN		102

DAFTAR TABEL

Tabel 1.1 Jadwal Penelitian.....	14
Tabel 2. 1 Simbol Flowchart Program	31
Tabel 2. 2 Simbol Flowchart Sistem	32
Tabel 2. 3 Simbol Flowmap	34
Tabel 2. 4 Simbol Data Flow Diagram (DFD).....	35
Tabel 4. 1 Tingkat Kecerahan Gambar	54
Tabel 4. 2 Jendela 3 x 3.....	55
Tabel 4. 3 Hasil Operator LBP.....	57
Tabel 4. 4 Konstanta LBP	57
Tabel 4. 5 Hasil Operator LBP Setelah Dikalikan	57
Tabel 4. 6 Skema Tabel Pengguna.....	69
Tabel 4. 7 Skema Tabel Data Absensi	70
Tabel 4. 8 Skema Model	70
Tabel 4. 9 Sintaks Membuat Dataset.....	76
Tabel 4. 10 Sintaks Membuat Model	78
Tabel 4. 11 Sintaks Pengenalan Wajah	78
Tabel 4. 12 Sintaks Server Pengenalan Wajah.....	79
Tabel 4. 13 Durasi Video	86
Tabel 4. 14 Hasil Pengujian Muka Senyum.....	Error! Bookmark not defined.
Tabel 4. 15 Hasil Pengujian Muka Datar	88
Tabel 4. 16 Hasil Pengujian Gambar Acak	89
Tabel 4. 17 Hasil Pengujian Dengan Id Diketahui.....	90
Tabel 4. 18 Z Index	93

DAFTAR GAMBAR

Gambar 1. 1 Kerangka Pemikiran	11
Gambar 2. 1 Operator LBP Asli.....	22
Gambar 2. 2 Flowchart deteksi wajah.....	24
Gambar 2. 3 Pembagian Gambar LBP	25
Gambar 2. 4 Histogram LBPH.....	25
Gambar 2. 5 Contoh gambar dalam dataset	26
Gambar 2. 6 Model Pengembangan Waterfall (Pressman 2002).....	29
Gambar 3. 1 Struktur Organisasi Radar Cirebon .com	41
Gambar 3. 2 Flowmap Daftar Karyawan	45
Gambar 3. 3 Flowmap Absensi.....	46
Gambar 3. 4 Flowmap Membuat Laporan	47
Gambar 4. 1 Hasil Deteksi Wajah.....	53
Gambar 4. 2 Merubah Warna Gambar	53
Gambar 4. 3 Pembagian Wilayah Wajah	54
Gambar 4. 4 Diagram Konteks.....	62
Gambar 4. 5 DFD Level 1	63
Gambar 4. 6 Flowmap Pendaftaran.....	64
Gambar 4. 7 Flowmap Login	65
Gambar 4. 8 Flowmap Memprediksi Wajah	66
Gambar 4. 9 Flowmap Melatih Model.....	67
Gambar 4. 10 Flowchart Program.....	68
Gambar 4. 11 Desain Halaman Login.....	72
Gambar 4. 12 Desain Halaman Dashboard	72
Gambar 4. 13 Desain Halaman Setting	73
Gambar 4. 14 Desain Halaman Absensi	74
Gambar 4. 15 Desain Halaman Laporan	74
Gambar 4. 16 Struktur Menu Sistem	75
Gambar 4. 17 Halaman Pendaftaran	82
Gambar 4. 18 Halaman Login.....	83

Gambar 4. 19 Tampilan Setting	84
Gambar 4. 20 Halaman Absensi.....	85

DAFTAR LAMPIRAN

Lampiran 1 Sintaks Membuat Dataset	103
Lampiran 2 Sintaks Membuat Model.....	107
Lampiran 3 Sintaks Pengenalan Wajah.....	109
Lampiran 4 Sintaks Server Pengenalan Wajah	111
Lampiran 5 Hasil Pengujian Gambar Acak	116
Lampiran 6 Pengujian Dengan Id Diketahui.....	118
Lampiran 7 Hasil pengujian Muka Senyum.....	120
Lampiran 8 Hasil pengujian Muka Datar.....	121

BAB I

PENDAHULUAN

1.1 Latar Belakang

Menurut UU. RI No. 88 Th. 2019, Kesehatan Kerja adalah upaya yang ditujukan untuk melindungi setiap orang yang berada di tempat kerja agar hidup sehat dan terbebas dari gangguan kesehatan serta pengaruh buruk yang diakibatkan dari pekerjaan. Berdasarkan undang-undang UU. RI No. 88 Th. 2019, bahwa untuk menghadirkan lingkungan kerja yang aman maka perlu dilakukan upaya Kesehatan Kerja.

Maret 2020 merupakan awal penyebaran virus Covid-19 di Indonesia. Kasus Covid-19 di Indonesia hingga saat ini masih sulit dibendung, menurut data dari *Our World In Data* per tanggal 27 April 2021 kasus positif di negara kita mencapai 1,65 juta jiwa [1]. WHO memberikan pernyataan bahwa Covid-19 dapat menyebar dalam beberapa kemungkinan termasuk transmisi secara langsung, *airbone*, *droplets* (percikan), *formit*, *fekal-oral*, melalui darah, binatang ke manusia, dan ibu ke anak [2]. Untuk menghadapi penyebaran Covid-19 pemerintah mengeluarkan Kebijakan Permenkes Nomor 9 Tahun 2020 tentang Pembatasan Sosial Berskala Besar (PSBB). Kebijakan ini sangat memengaruhi berbagai kegiatan seperti kegiatan bekerja yang biasanya dilaksanakan di kantor sekarang harus dilaksanakan dirumah atau *work from home* (WFH).

Work from home yang sering disingkat dengan WFH memiliki arti bekerja dari rumah. *Work from home* digambarkan dengan kegiatan atau pekerjaan karyawan yang berada di luar kantor atau dengan kata lain bekerja dari rumah. Sistem kerja WFH memiliki fleksibilitas yang tinggi namun sulit dalam pengawasannya. Salah satu bentuk pengawasan terhadap karyawan adalah dengan menggunakan absensi. Saat ini sudah banyak teknologi yang dipakai untuk sistem absensi seperti absensi untuk memindai biometrik manusia. Teknologi biometrik berdasarkan ciri kodrati manusia yaitu ciri fisiologis dan ciri tingkah laku, seperti wajah, sidik jari, suara, telapak tangan, iris mata dan retina mata, DNA dan tanda tangan.

Wajah manusia mengungkapkan banyak informasi kepada siapa saja yang melihatnya. Wajah tidak hanya dapat memberi tahu tentang suasana hati, niat atau perhatian, tetapi juga dapat berfungsi sebagai identitas dari seseorang [3]. Tentu seseorang juga dapat dikenali selain dari wajah. Suara, bentuk tubuh, gaya berjalan, atau bahkan gaya pakaian dapat menjadi identitas dari seseorang. Tetapi dari semuanya, wajah manusia yang paling unik, manusia tidak mungkin berwajah serupa bahkan pada kasus anak kembar.

Pada era modern ini dimana teknologi berkembang dengan pesatnya, wajah digunakan sebagai bagian yang dapat dikenali oleh komputer. Pendeteksian wajah dan pengenalan wajah merupakan teknik yang digunakan untuk melakukan proses pengenalan wajah pada komputer. Saat ini sudah banyak teknologi yang digunakan untuk mendeteksi kehadiran

yaitu teknologi absensi sidik jari dan juga wajah. Melalui teknologi ini seseorang dapat diketahui waktu kehadirannya pada suatu tempat dengan cara melakukan proses absen pada mesin absensi. Pada sistem yang memanfaatkan sidik jari, pengguna wajib menyentuh detektor sidik jari pada proses absensi. Namun untuk mesin absensi wajah, pengguna wajib melakukan absensi melalui proses perekam wajah dengan cara berdiri didepan detektor kamera.

Ada banyak metode untuk pengenalan wajah, diantaranya Local Binary Pattern (LBP) dan Local Binary Pattern Histogram (LBPH). Pada LBP ada yang disebut LBP Operator yang digunakan untuk menggambarkan nilai kontras suatu piksel dan piksel didekatnya. LBP asli mendefinisikan ukuran piksel 3×3 . Menggunakan nilai piksel median sebagai ambang, ini membandingkan dengan nilai abu-abu dari 8 piksel yang berdekatan. Jika nilai piksel yang didekatnya lebih besar atau sama dibandingkan dengan nilai piksel median, maka nilai posisi piksel diberi tanda 1, jika tidak diberi tanda (0) seperti pada Gambar 2.1.

Metode LBPH membagi citra wajah menjadi beberapa sel, dan setiap sel diberikan operasi LBP sehingga menghasilkan sebuah nilai biner. Dalam metode LBPH semua nilai yang dihasilkan oleh setiap sel akan disatukan menjadi sebuah histogram yang kemudian dapat dibaca dan dijadikan referensi wajah oleh mesin.

Dalam pengenalan wajah akurasi adalah hal penting yang harus diperhatikan. Menurut riset yang dilakukan oleh Qadrisa Mutiara Detila dan Eri Prasetyo Wibowo, LBPH merupakan metode pengenalan wajah paling akurat disbanding dengan metode *Eigenface* dan *Fisherface* dengan akurasi rata-rata 83% [4].

Berdasarkan uraian tersebut penerapan sistem pengenalan wajah menggunakan metode LBPH dapat menjadi solusi yang baik untuk membuat sistem absensi online. Oleh karena itu penulis membuat tugas akhir dengan judul “**METODE LOCAL BINARY PATTERN HISTOGRAM (LBPH) PENGENALAN WAJAH PADA SISTEM ABSENSI ONLINE KARYAWAN RADAR CIREBON**”. Dengan sistem absensi online yang dibuat diharapkan dapat membantu dan berguna bagi pihak yang membutuhkan.

1.2 Identifikasi Masalah

Dari latar belakang tersebut, penulis mengidentifikasi masalah penelitian sebagai berikut:

1. Karyawan yang sedang bekerja diluar seperti wartawan tidak dapat melakukan absensi, karena alat detektor sidik jari berada di kantor.
2. Tidak menerapkan protokol kesehatan saat pandemi secara maksimal karena karyawan harus menempelkan jari di alat detektor sidik jari.

1.3 Batasan Masalah

Dalam pengembangan sistem absen menggunakan pengenalan wajah penulis membuat batasan masalah agar tujuan yang dari sistem yang dibuat dapat tercapai. Adapun batasan masalahnya adalah sebagai berikut:

1. Sistem absen ini hanya mengambil data wajah dan lokasi pengguna.
2. Sistem tidak memperhitungkan jarak antara pengguna dan lokasi kantor, lokasi pengguna hanya akan disimpan di database.
3. Sistem ini tidak menghasilkan laporan apapun dalam bentuk pdf/csv seperti laporan absensi harian/bulanan, laporan tiap karyawan. Sistem hanya menyimpan data dalam *database*.
4. *Training* dataset dan proses pengenalan wajah dilakukan secara otomatis ketika sistem berjalan di Amazon Web Service.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan utama dari penelitian ini adalah:

1. Membuat sistem absensi *online* yang mampu mengenali wajah sebagai media verifikasinya sehingga dapat membuat karyawan yang memiliki mobilitas tinggi dapat melakukan absensi dimanapun dan menerapkan protokol kesehatan dengan baik.

2. Membuat lingkungan yang menerapkan protokol kesehatan dengan baik, karena mengurangi interaksi antar karyawan.

1.4.2 Manfaat

Setelah mengetahui tujuan penelitian tersebut di atas, maka diharapkan penelitian ini dapat dikembangkan dan diamalkan dengan baik. Maka manfaat penelitian antara lain, yaitu:

1. Bagi Penulis

Dapat memahami bagaimana dasar-dasar proses pengenalan wajah menggunakan bahasa pemrograman python dan menerapkan ke aplikasi *serverless* berbasis website menggunakan Amazon Web Service.

2. Bagi Lembaga Pendidikan

Manfaat penelitian ini bagi lembaga pendidikan diharapkan dapat menjadi bahan pembelajaran dan referensi bagi rekan-rekan yang akan melakukan penelitian lebih lanjut dengan topik yang berhubungan dengan judul penelitian di atas.

3. Bagi Perusahaan

Sistem absensi online yang dibangun dapat memudahkan pengguna ketika melakukan absensi di manapun secara *online* sehingga memungkinkan pengguna tidak perlu lagi ke kantor hanya untuk melakukan absensi.

1.5 Metode dan Teknik Penelitian

1.5.1 Metode Penelitian

Metode yang digunakan dalam penelitian adalah metode penelitian dan pengembangan (*Research and Development*). Menurut Sugiyono metode penelitian dan pengembangan adalah metode penelitian yang digunakan untuk menghasilkan produk dan mengujinya [5]. Agar dapat menghasilkan produk digunakan penelitian yang bersifat analisis kebutuhan dan untuk menguji keefektifan produk tersebut agar dapat berfungsi di masyarakat luas, maka diperlukan penelitian untuk menguji keefektifan produk tersebut.

Salah satu keuntungan yang merupakan alasan penulis memilih metode penelitian dan pengembangan (*Research and Development*) adalah mampu memenuhi dan mengatasi kebutuhan yang sifatnya mendesak melalui pengembangan solusi dari sebuah masalah sekaligus menghasilkan pengetahuan yang berguna bagi masa yang akan datang.

1.5.2 Teknik Penelitian

Teknik penelitian yang digunakan penulis untuk perancangan dan pengembangan sistem diantaranya adalah:

1. Observasi

Observasi salah satu kegiatan ilmiah yang berdasarkan pada fakta-fakta lapangan maupun dokumen, melalui pengalaman sendiri tanpa menggunakan manipulasi apapun [6].

Dalam penelitian ini penulis melakukan observasi di Radar Cirebon guna mendapatkan data-data pendukung yang diperlukan dalam penelitian ini seperti bagaimana sistem absensi yang sedang berjalan.

2. Studi kepustakaan

Studi kepustakaan adalah suatu studi yang digunakan untuk mengumpulkan informasi dan data dengan bantuan berbagai macam material yang ada di perpustakaan seperti dokumen, buku, majalah, kisah-kisah sejarah, dan lainnya [7].

Dalam penelitian ini penulis melakukan studi kepustakaan dengan cara mencari berbagai informasi yang berkaitan dengan penelitian pada buku atau jurnal maupun mencari dokumentasi *software* melalui internet.

1.6 Anggapan Dasar

Penulis melakukan studi kepustakaan guna mengumpulkan teori-teori dari buku maupun penelitian terdahulu. Teori-teori tersebut dinamakan

anggapan dasar dari sebuah penelitian, anggaran dasar adalah sesuatu yang sudah jelas kebenarannya.

Pengenalan wajah merupakan salah satu ilmu yang berhubungan dengan *computer vision*, dimana komputer menganalisis citra wajah yang ada pada citra tersebut dan membandingkannya dengan data pada citra wajah tersebut, dimana komputer dapat menemukan identitas atau data pribadi dari gambar wajah dalam *database*. Biasanya, pengenalan wajah dilakukan dari depan dengan menerangi 10 wajah secara merata. Namun, ada beberapa masalah seperti posisi wajah, ukuran wajah dan jarak, orientasi, usia dan ekspresi wajah.

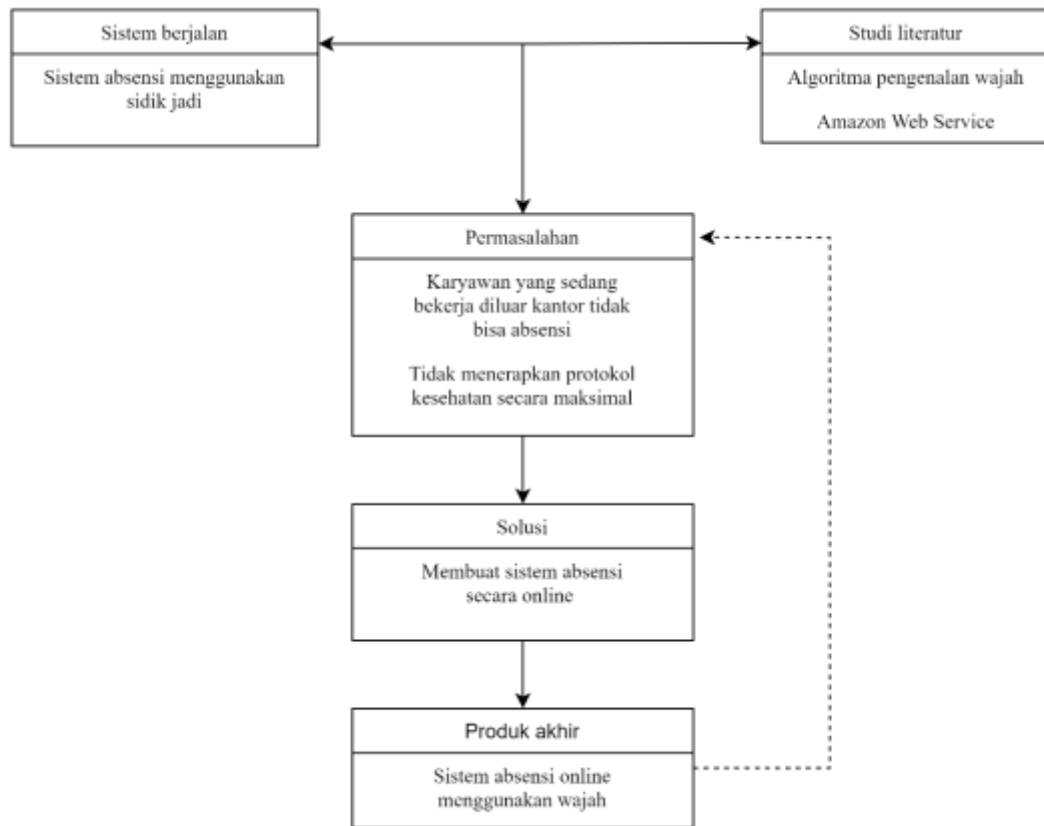
Menurut Vicki Bruce dan Andy Young wajah manusia mengungkapkan banyak informasi kepada siapa saja yang melihatnya. Wajah tidak hanya dapat memberi tahu tentang suasana hati, niat atau perhatian, tetapi juga dapat berfungsi sebagai identitas dari seseorang. Tentu seseorang juga dapat dikenali selain dari wajah. Suara, bentuk tubuh, gaya berjalan, atau bahkan gaya pakaian dapat menjadi identitas dari seseorang. Tetapi dari semuanya, wajah manusia yang paling unik, manusia tidak mungkin berwajah serupa bahkan pada kasus anak kembar [3].

Dalam buku yang berjudul "*Handbook of Face Recognition*" yang ditulis oleh Stan Z. Li Anil K. Jain mengemukakan bahwa pengenalan wajah atau *face recognition* adalah tugas yang dilakukan manusia secara rutin dan mudah dalam kehidupan sehari-hari. Ketersediaan komputasi komputer

yang luas dan mudah memunculkan minat yang besar untuk pemrosesan otomatis gambar dan video digital di sejumlah aplikasi termasuk otentikasi biometrik, pengawasan, interaksi manusia dengan komputer dan lainnya [8].

1.7 Kerangka Pemikiran

Pada Gambar 1.1 memaparkan pemikiran penulis secara kasar mengenai penelitian yang dilakukan untuk membuat aplikasi absensi *online* menggunakan pengenalan wajah. Sistem ini diharapkan dapat mengenali objek wajah secara akurat. Objek gambar tersebut di ambil dan direkam menjadi sebuah citra/gambar. Metode Local Binary Patterm Histogram (LBPH) diterapkan untuk mengambil tekstur dari objek gambar. Hasil pengambilan tekstur dari LBPH adalah sebuah histogram yang terdiri dari gabungan beberapa histogram kecil yang berisi informasi nilai operasi operator Local Binary Pattern (LBP).



Gambar 1. 1 Kerangka Pemikiran

Tahapan-tahapan pembangunan atau pengembangan perangkat lunak (*software*) sebagai berikut:

1. Menentukan perangkat lunak yang akan digunakan.
2. Membuat desain tampilan program.
3. Membuat kode program.
4. Melakukan pengujian dan pemeliharaan.

Setelah aplikasi selesai dibuat maka dilakukan pengujian dengan cara pengguna mencoba aplikasi secara langsung. Hasil dari penelitian ini berupa *website* absensi *online* berbasis pengenalan wajah.

1.8 Sistematika Penulisan

Sistematika penulisan merupakan langkah-langkah yang digunakan untuk menyusun laporan penelitian, penulis menggunakan sistematika sebagai berikut:

BAB I PENDAHULUAN

Bab ini merupakan pendahuluan yang berisi latar belakang, identifikasi masalah, batasan masalah, tujuan dan manfaat penelitian, metode dan teknik penelitian, kerangka pemikiran, anggapan dasar, sistematika penulisan serta lokasi penelitian dan jadwal penelitian.

BAB II LANDASAN TEORI

Dalam BAB II akan dijelaskan mengenai landasan teori sebagai acuan primer yang digunakan penulis dalam penelitian ini, teori pendukung serta tinjauan pustaka yang ditambahkan untuk menjelaskan informasi secara lebih detail.

BAB III ANALISIS SISTEM

Dalam BAB III membahas tentang gambaran organisasi serta analisis dari sistem yang sedang berjalan, baik dalam bentuk uraian, diagram atau keduanya.

BAB IV PERANCANGAN SISTEM

Dalam BAB IV akan dijelaskan tentang prosedur yang akan berjalan untuk sistem yang baru, desain *interface*, implementasi maupun pengujian pada sistem tersebut.

BAB V KESIMPULAN DAN SARAN

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

DAFTAR PUSTAKA

LAMPIRAN

1.9 Lokasi Penelitian dan Jadwal Penelitian

1.9.1 Lokasi Penelitian

Penelitian ini dilakukan di Radar Cirebon yang beralamat di Grha Pena Radar Cirebon, Jl. Perjuangan No.9 Kota Cirebon 45135 Telp. (0231) 483531, 483532 Fax: (0231) 483533, Indonesia dengan jadwal penelitian dari Maret – Agustus 2021.

1.9.2 Jadwal Penelitian

Pada Tabel 1.1 menjelaskan waktu penelitian yang dilakukan penulis yaitu dari bulan Maret 2021 hingga bulan Agustus 2021.

Tabel 1.1 Jadwal Penelitian

[illegible]

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini penulis menggunakan beberapa penelitian terdahulu yang membahas pengenalan wajah dengan metode Local Binary Pattern (LBP) atau Local Binary Pattern Histogram (LBPH). Untuk menambahkan gambaran penelitian penulis dibandingkan dengan penelitian yang sudah dilakukan, maka dibawah ini diuraikan tentang penelitian terdahulu yang relevan. Beberapa penelitian yang berkaitan dengan masalah penelitian ini adalah:

1. PERBANDINGAN METODE EIGENFACE, FISHERFACE, DAN LBPH PADA SISTEM PENGENALAN WAJAH.

Qadrisa Mutiara Detila dan Eri Prasetyo Wibowo, Jurnal Ilmiah KOMPUTASI, Volume 18 No. 4, Desember 2019. Penelitian ini menggunakan *dataset* yang diperoleh dari pengambilan citra menggunakan *webcam* dan *Haar cascade classifier* dalam bentuk *xml* file untuk deteksi wajah dan proses pengenalan wajah. Tahap awal dalam sistem pengenalan wajah adalah proses deteksi wajah, citra yang diperlukan dalam penelitian diambil dari kamera yang menghasilkan citra dinamis kemudian dikonversi menjadi citra statis, lalu disimpan ke dalam folder *dataset* untuk dijadikan citra *training*.

Qadrisa Mudiara Detila dkk. melakukan perbandingan dengan cara membandingkan gambar wajah dengan 4 kondisi yang berbeda

yaitu tingkat kecerahan rendah, tingkat kecerahan normal, wajah tidak berekspresi, dan wajah tersenyum. Hasil dari jurnal ini adalah nilai persentase keberhasilan dari 4 kondisi yang ditentukan sebelumnya dengan rata-rata 46% untuk *Eigenface*, 54,25% untuk *Fisherface* dan 88,75% untuk LBPH. Dilihat dari nilai persentase keberhasilan dari tiap kondisi yang ditentukan, Metode LBPH dapat mengenali wajah lebih baik dibandingkan metode *Eigenface* dan *Fisherface*. Berdasarkan hasil tersebut, metode LBPH juga lebih unggul dibandingkan dengan kedua metode lainnya dalam mengenali wajah yang datar atau tanpa ekspresi. Sedangkan untuk ekspresi senyum, metode *Eigenface* masih sulit untuk mengenali wajah dibandingkan kedua metode lainnya yaitu *Fisherface* dan LBPH yang masih bisa mengenali wajah apabila terjadi perubahan ekspresi.

2. **SISTEM PENGENALAN WAJAH DENGAN METODE LOCAL BINARY PATTERN HISTOGRAM PADA FIREBASE BERBASIS OPENCV.**

Fajar Setiawan dan Dewi Agushinta, Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK) Volume 4 Nomor 1, 23 September 2020. Penelitian ini menggunakan LBPH sebagai metode untuk pengolahan fitur wajah dan *Haar Cascade* untuk pengambilan frame wajah. Metode LBPH mengubah tekstur dari suatu citra wajah menjadi nilai biner, dan nilai tersebut mewakili bagian dari piksel-piksel suatu wajah yang membentuk sebuah lingkaran dan memiliki pusat sebagai acuan terhadap nilai-nilai tersebut. *Database*

yang digunakan adalah *database online* dari firebase yang dinaungi oleh Google, sehingga dengan terciptanya *dataset* wajah secara online, pengoperasian pengenalan wajah dapat lebih efisien.

Fajar Setiawan dkk. melakukan penelitian untuk mengukur tingkat keberhasilan metode LBPH dalam mengenali wajah. Hasil yang diperoleh dari penelitian ini sistem pengenalan wajah yang memiliki tingkat keberhasilan 93,5%. Hasil tersebut diperoleh dengan cara menguji sistem untuk mengenali 62 citra wajah.

3. FACIAL RECOGNITION SYSTEM USING LBPH FACE RECOGNIZER FOR ANTI-THEFT AND SURVEILLANCE APPLICATION BASED ON DRONE TECHNOLOGY.

Li Wang dan Ali Akbar Siddique, *Original Paper Measurement and Control (United Kingdom)*, 15 Mei 2020. Penelitian ini mengusulkan sistem pengenalan wajah menggunakan Local Binary Patterns Histogram (LBPH) *Face recognition* yang dipasang pada teknologi *drone*. Selama bertahun-tahun, para peneliti telah menggunakan algoritma yang berbeda untuk pengenalan wajah seperti *Haar cascade*, fitur TPLBP/HOG, *DBN Depth Model* dan *Fisherface* untuk keamanan mesin ATM atau untuk autentikasi seseorang. Peneliti juga menggunakan *neural network* bersama dengan *sparse auto-encoder* untuk melatih model pengenalan wajah.

Pengenal wajah LBPH juga dapat melacak dan mengidentifikasi tindakan manusia beserta fitur dominan dan klasifikasinya. Makalah ini mengusulkan sistem tertanam berbasis teknologi *drone*, yang mampu

mengidentifikasi seseorang menggunakan pengenalan wajah LBPH. Kamera yang terpasang pada drone menangkap bingkai secara *real time* dan mengidentifikasi orang yang menjadi target. Kecanggihan Bagian ini menawarkan beberapa penelitian mutakhir tentang teknologi pengenalan wajah menggunakan *drone*. Studi ini berkonsentrasi pada pengenalan wajah orang-orang dengan latar belakang kriminal. Jika drone mengidentifikasi orang yang menjadi target, *drone* akan menginformasikan otoritas terkait dengan mengirimkan gambar yang menampilkan orang yang diidentifikasi. Karena, model yang diusulkan menggunakan pengenalan wajah algoritma pembelajaran mesin (LBPH), ini juga merupakan aplikasi berbasis *AI*. Model perangkat lunak Aplikasi berbasis *Machine learning* seperti pengenalan wajah memerlukan kumpulan data yang sangat besar untuk pelatihan sebelum kami mengimplementasikannya. Hasil dari penelitian tersebut adalah sistem pengenalan wajah yang memiliki tingkat akurasi sebesar 89,1%.

4. LBPH BASED IMPROVED FACE RECOGNITION AT LOW RESOLUTION.

Aftab Ahmed, Jiandong Guon Fayaz Ali, Farha Deebea, dan Awais Ahmed, *International Conference on Artificial Intelligence and Big Data, 2018*. Penelitian ini menggunakan metode algoritma Local Binary Patterns Histogram (LBPH) untuk mengatasi pengenalan wajah manusia secara *real time* pada tingkat resolusi yang rendah. Ekstraksi Fitur Operator LBP diterapkan untuk menggambarkan informasi kontras suatu piksel ke piksel tetangganya lalu mengubahnya menjadi

angka desimal sehingga diperoleh nilai LBP dari titik piksel tengah. Algoritma LBPH saat ini menggunakan operator LBP melingkar yang ditingkatkan. Algoritma ini membuat operator LBP tidak lagi terbatas pada radius dan lingkungan tetap dan dapat memenuhi kebutuhan fitur ukuran dan tekstur yang lebih berbeda. Untuk setiap piksel gambar, ia menghitung nilai *eigen* LBP-nya kemudian nilai *eigen* ini dapat membentuk spektrum fitur LBP. Algoritme LBPH menggunakan histogram spektrum karakteristik LBP sebagai vektor fitur untuk klasifikasi dan membagi gambar menjadi beberapa *sub-region*, kemudian mengekstrak fitur LBP dari setiap piksel *sub-region*, membentuk histogram statistik spektrum karakteristik LBP di setiap *sub-region*, sehingga setiap *sub-region* dapat menggunakan histogram statistik untuk gambarkan keseluruhan gambar melalui sejumlah komponen histogram.

Penelitian ini menggunakan *dataset* LR500 dan menguji dengan 2500 gambar yang diambil oleh *webcam* beresolusi 1080px. Dari 2500 gambar 98.8% berhasil dikenali. Untuk hasil akurasi ukuran wajah 45px sebesar 94% dan 35px sebesar 90%.

5. SISTEM PENGENALAN WAJAH DENGAN ALGORITMA HAAR CASCADE DAN LOCAL BINARY PATTERN HISTOGRAM.

Sayeed Al-Aidid dan Daniel S. Pamungkas, Jurnal Rekayasa Elektrika, Volume 1 No.1, April 2018. Penelitian ini menggunakan webcam dan mampu membedakan objek wajah dan bukan wajah, serta

mengenai orang yang telah terdaftar pada database secara *real time*. Sayeed Al-Aidid dan Daniel S. Pamungkas menggunakan *Haar cascade classifier* dan metode Local Binary Pattern Histogram (LBPH) untuk mengenali citra wajah. Metode LBPH digunakan karena mampu mendeteksi wajah dari sebuah citra gambar meskipun di sekitar wajah manusia terdapat gangguan objek. Ada beberapa tahapan yang dilakukan peneliti untuk membuat sistem pengenalan wajah, diantaranya deteksi wajah, pembuatan *database*, dan pengenalan wajah. Dalam penelitian ini juga dibahas seberapa jauh sistem dapat mengenali wajah. Hasil dari penelitian Sayeed Al-Aidid dkk. adalah sistem pengenalan wajah dapat mendeteksi wajah meskipun terdapat gangguan objek lain dan dapat mendeteksi wajah paling jauh sekitar 150cm.

2.2 Teori Utama Penelitian

2.2.1 Kecerdasan Buatan (Artificial Intelligence)

Kecerdasan buatan adalah cabang ilmu komputer yang menjelaskan bagaimana komputer dapat meniru pemikiran manusia. Kecerdasan buatan memungkinkan komputer untuk menarik kesimpulan dan mengidentifikasi masalah dengan cara manusia. Dalam buku yang ditulis oleh Dr. Hendra Jaya, S.Pd., M.T., dkk. Dengan judul “Kecerdasan Buatan” mengemukakan bahwa kecerdasan buatan merupakan salah satu bidang ilmu komputer yang

membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik apa yang manusia lakukan [9].

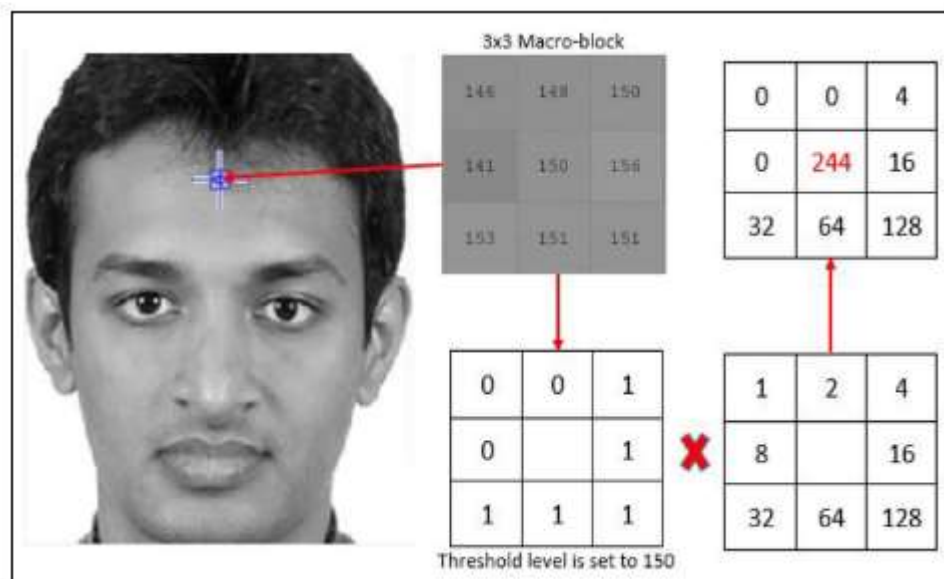
Menurut Dr. Hendra Jaya, S.Pd., M.T., dkk. Kecerdasan buatan dapat dipandang dalam 4 sudut pandang yaitu sudut pandang kecerdasan, penelitian, bisnis, dan pemrograman. Berikut beberapa contoh penerapan kecerdasan buatan (*Artificial Intelligence*), diantaranya:

1. Sistem kemudi otomatis pada mobil, contohnya TESLA pabrikan mobil dari Amerika Serikat memproduksi mobil *autopilot*.
2. Asisten virtual seperti Siri pada Apple, Bixbi pada Samsung, dan Alexa pada Amazon.
3. Sistem rekomendasi, seperti rekomendasi produk di *e-commerce* dan rekomendasi iklan pada youtube.

2.2.2 Local Binary Pattern Histogram (LBPH)

Untuk memahami apa itu Local Binary Pattern Histogram (LBPH) kita harus memahami dahulu apa itu Local Binary Pattern (LBP) karena LBPH merupakan gabungan nilai yang dihasilkan LBP operator dari sebuah citra gambar. Operator LBP diterapkan untuk menggambarkan kontras informasi dari suatu piksel ke piksel tetangganya. Operator LBP asli didefinisikan di jendela 3×3 . Menggunakan nilai piksel median sebagai ambang jendela,

dibandingkan dengan nilai abu-abu dari 8 piksel yang berdekatan. Jika nilai piksel tetangga lebih besar atau sama dengan nilai median piksel, nilai posisi piksel ditandai sebagai 1, sebaliknya ditandai sebagai (0) [9]. Fungsi didefinisikan seperti yang ditunjukkan pada persamaan 1. 1. Dapat diilustrasikan seperti pada Gambar 2.1.



Gambar 2. 1 Operator LBP Asli

Sumber: (Li Wang dan Ali Akbar Siddique, 2020)

Setelah diubah menjadi nilai biner maka akan menghasilkan deret biner yang kemudian jika diubah menjadi nilai desimal akan menghasilkan nilai baru untuk nilai tengah piksel selanjutnya, deret biner tersebut yang disebut dengan kode LBP. Kode LBP yang dihasilkan tersebut direpresentasikan dengan histogram. Histogram akan menunjukkan frekuensi kejadian dari berbagai nilai LBP. Secara umum perhitungan LBP ini bisa dilakukan seperti persamaan 1.1, hasil dari persamaan 1.1

kemudian dihitung menggunakan persamaan dua sehingga hasil akhirnya angka nol atau satu yang dapat dilihat pada persamaan 1.2 [10]:

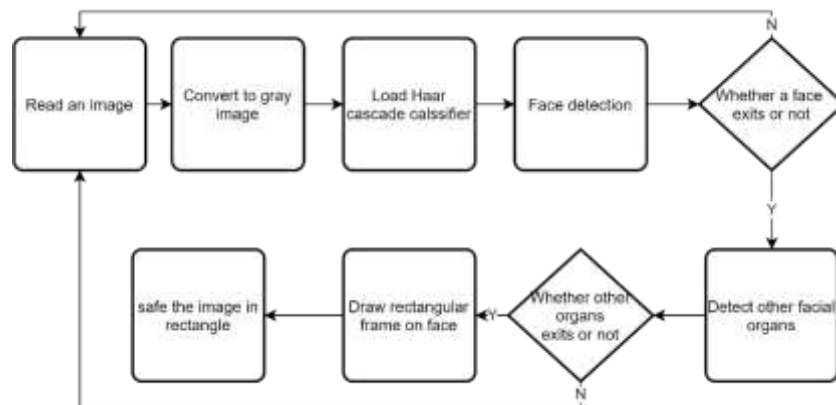
$$LPB(x_c, y_c) = \sum_{n=0}^7 (i_n - i_c) 2^n \quad (1.1)$$

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (1.2)$$

Pada persamaan 1.1 menjelaskan operator LBP asli dengan i_c adalah nilai piksel dari titik pusat (x_c, y_c) , sedangkan i_n merupakan nilai piksel tetangga dari i_c . Persamaan 1.2 dengan x merupakan nilai LBP. Berikut adalah alur pengenalan wajah menggunakan metode LBPH [11]:

1. Deteksi Wajah (*Face Detection*)

Untuk deteksi wajah, penulis menggunakan OpenCV dan *Haar cascade classifier*. Pada Gambar 2.2 menjelaskan tentang alur dari deteksi wajah. Langkah pertama yaitu membaca gambar kemudian dirubah ke gambar abu-abu, setelah itu lakukan deteksi wajah menggunakan *Haar cascade classifier* hingga wajah ditemukan dan disimpan menjadi gambar wajah.



Gambar 2. 2 Flowchart deteksi wajah

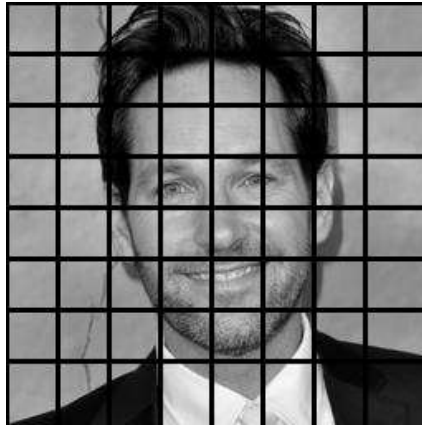
Sumber: (A.Singh et al 2019)

2. Ekstraksi Fitur (*Feature Extraction*)

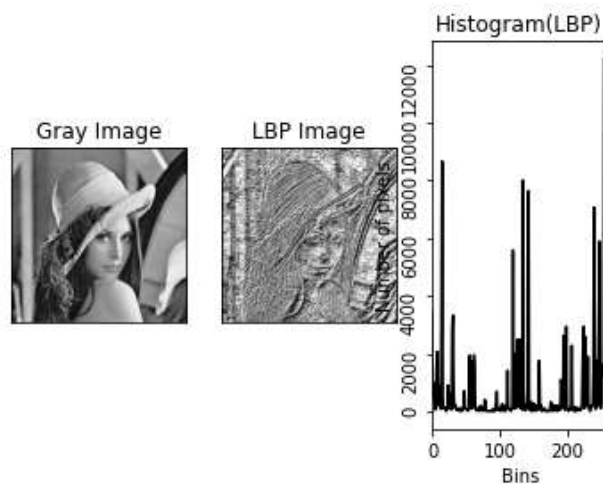
Untuk mengekstrak fitur wajah dari citra digunakan operasi LBP yang membandingkan nilai intensitas setiap komponen dengan nilai 8 piksel tetangga terdekat. Jika nilai piksel tetangga lebih besar dari nilai piksel tengah, itu akan menetapkan 1 ke piksel tetangganya, jika tidak maka akan menetapkan 0. Untuk setiap piksel, tugas ini menyediakan string 8-bit. Nilai desimal dari string piksel 8-bit menentukan nilai LBP. Gambar 2.1 menunjukkan operasi ini.

Gambar input dibagi menjadi banyak sub-gambar kecil seperti pada Gambar 2.3. Setelah penerapan operator LBP dan histogram nilai LBP dari setiap sub-gambar diekstraksi. Kemudian semua histogram dihubungkan seperti pada Gambar 2.4 untuk membuat vektor fitur yang

merepresentasikan citra dan digunakan untuk melatih pengklasifikasi pengenalan wajah.



Gambar 2. 3 Pembagian Gambar LBP
Sumber: (Li Wang dan Ali Akbar Siddique, 2020)



Gambar 2. 4 Histogram LBPH
Sumber: (Li Wang dan Ali Akbar Siddique, 2020)

3. Membuat Dataset

Penulis merancang sendiri *dataset* yang akan dibuat, *dataset* berisi 3 gambar untuk setiap wajah. *Dataset* akan dilatih setiap

ada pengguna yang mendaftarkan wajahnya ke sistem absen ini. Gambar 2.5 menunjukkan contoh gambar dalam *dataset*.



Gambar 2. 5 Contoh gambar dalam dataset
Sumber: (A.Singh et al, 2019)

4. Pengenalan Wajah (*Face Recognition*)

Dalam langkah ini gambar akan diproses menggunakan operator LBP seperti pada Gambar 2.1. Setelah perhitungan menggunakan operator LBP selesai dan menghasilkan Histogram, maka akan dibandingkan dengan *dataset* yang sudah ada.

2.2.3 OpenCV

OpenCV adalah kependekan dari *opensource computer vision*. OpenCV adalah sebuah kumpulan *tools*, *library*, dan *module* yang berkaitan dengan *computer vision*. Dalam OpenCV terdapat kumpulan algoritma yang dinamakan “*Haar Cascade Classifier*” untuk mendeteksi objek khususnya wajah manusia. Dengan menggunakan *Haar Cascade Classifier* penulis tidak kesulitan untuk mendeteksi wajah dalam sebuah gambar atau video.

2.2.4 Computer Vision

Computer Vision adalah bagian dari komputer yang menggambarkan bagaimana komputer terlihat layaknya manusia, dan karena ini terkait erat dengan penglihatan, cahaya juga merupakan faktor penting dalam hal ini. [12].

2.2.5 Python

Python adalah bahasa pemrograman berorientasi objek yang ditafsirkan, interaktif. Ini menggabungkan modul, pengecualian, pengetikan dinamis, tipe data dinamis tingkat sangat tinggi, dan kelas. Ini mendukung beberapa paradigma pemrograman di luar pemrograman berorientasi objek, seperti pemrograman prosedural dan fungsional. Python menggabungkan kekuatan yang luar biasa dengan sintaks yang sangat jelas. Ini memiliki antarmuka ke banyak panggilan sistem dan perpustakaan, serta berbagai sistem jendela, dan dapat dikembangkan dalam C atau C++. Ini juga dapat digunakan sebagai bahasa ekstensi untuk aplikasi yang membutuhkan *programmable* antarmuka. Terakhir, Python bersifat portable, ia berjalan di banyak varian Unix termasuk Linux dan macOS, dan di Windows.

2.2.6 Amazon Web Service (AWS)

Ada banyak definisi dan interpretasi komputasi awan yang dapat ditemukan dari berbagai sumber. Istilah "komputasi awan"

itu sendiri kemungkinan berasal dari diagram jaringan di mana *cloud* bentuk digunakan untuk menggambarkan jenis jaringan tertentu, baik internet atau jaringan internal. Beberapa sumber menyebut komputasi awan sebagai sekumpulan aplikasi yang dikirimkan sebagai gabungan layanan dengan perangkat keras dan perangkat lunak pusat data yang memungkinkan aplikasi. Yang lain mengatakan awan itu komputasi adalah model bisnis daripada teknologi atau layanan tertentu [13].

Amazon Web Service (AWS) adalah penyedia layanan *cloud computing*. Menurut situs resminya yaitu <https://aws.amazon.com/>, AWS menyediakan lebih dari 200 service seperti penyimpanan, *compute*, *internet of things*, *machine learning*, dan masih banyak lainnya.

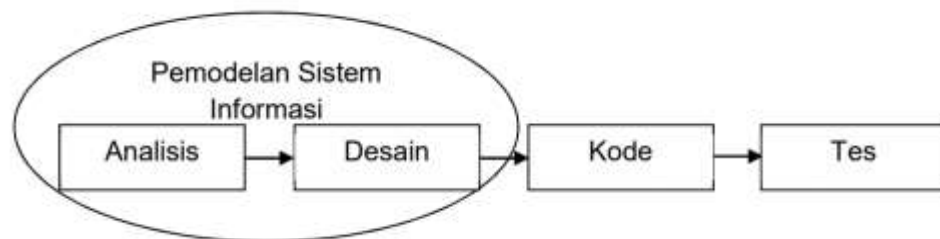
2.3 Metode Pengembangan Perangkat Lunak

2.3.1 Waterfall

Pengembangan perangkat lunak meliputi metode, proses, dan alat yang memudahkan dalam proses pengembangan perangkat lunak dari tahap satu ke tahap lainnya agar menghasilkan perangkat lunak yang berkualitas [14].

Model pengembangan perangkat lunak merupakan gambaran dari proses membuat suatu *software*. Menurut Pressman [14] Model Waterfall merupakan suatu model pengembangan

secara sekuensial. Model Waterfall bersifat sistematis dan berurutan dalam membangun sebuah perangkat lunak. Proses pembuatannya mengikuti alur dari mulai analisis, desain, kode, pengujian dan pemeliharaan. Model pengembangan waterfall memiliki beberapa kelebihan, antara lain: dapat mudah dipahami dan dapat diterapkan dalam proses pengembangan perangkat lunak. Gambar 2.6. merupakan model pengembangan Waterfall.



Gambar 2. 6 Model Pengembangan Waterfall (Pressman 2002)
Sumber: (Pressman, 2002)

2.3.2 Tahapan-tahapan Waterfall

Tahapan-tahapan dalam metode waterfall adalah sebagai berikut [14]:

1. Analisis kebutuhan perangkat lunak.

Tahap ini merupakan tahap untuk mengumpulkan kebutuhan perangkat lunak. Tujuan dari tahap ini yaitu untuk mengetahui apa saja kebutuhan untuk membuat sebuah perangkat lunak atau *software*. Analisis kebutuhan perangkat lunak ini perlu

didokumentasikan untuk dilihat kembali oleh pengguna apakah kebutuhan sudah sesuai atau tidak.

2. Desain

Desain merupakan tahapan yang berfokus pada perangkat lunak yang meliputi arsitektur, struktur data, antarmuka perangkat lunak, dan prosedur perngkodean. Ada beberapa pemodelan perangkat lunak yang umum digunakan, salah satu pemodelan perangkat lunak yang digunakan dalam tahapan ini yaitu Unified Modeling Language (UML), yang merupakan gambaran mengenai perangkat lunak yang akan dibuat. UML dibuat untuk mempermudah pengembang dalam membuat suatu perangkat lunak.

3. Pembuatan Kode

Dalam tahap ini desain yang sebelumnya telah dibuat akan diterapkan kedalam sebuah kode. Penerjemahan desain ke dalam kode haruslah lengkap dan berjalan dengan baik, Hasil dari tahap ini adalah sebuah sistem informasi yang sesuai dengan desain yang telah dibuat.

4. Pengujian

Tahapan biasanya ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. pengujian melibatkan pembetulan kesalahan-kesalahan yang tidak

ditemukan pada tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

2.4 Tools Perancangan

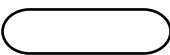


2.4.1 Flowchart

Flowchart menurut Drs. Suarga, M.Sc, Math, Ph.D, adalah untaian simbol atau gambar yang menunjukkan aliran dari data. Dalam bukunya juga yang berjudul ALGORITMA (LOGIKA PEMROGRAMAN), Drs. Suarga, M.Sc, Math, Ph.D, membagi simbol *flowchart* menjadi 2 jenis [15], yaitu:

1. Simbol *flowchart* program




Flowchart Program merupakan simbol-simbol *flowchart* yang digunakan untuk menggambarkan logika dari pemrosesan terhadap data ditunjukan pada Tabel 2.1.

Tabel 2. 1 Simbol Flowchart Program

Simbol	Nama Simbol	Keterangan
	terminator	Menunjukan awal/akhir sistem
	proses	Menunjukan operasi yang dilakukan sistem
	<i>Input/output</i>	Menerima input atau menampilkan output

Sumber: (Suarga, 2002)

Tabel 2.1 Simbol Flowchart Program (Lanjutan)

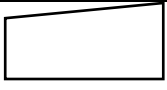


Simbol	Nama Simbol	Keterangan
	<i>Predefined process</i>	Menunjukkan fungsi atau <i>sub-program</i>
	penghubung	Menunjukkan penghubung pada halaman yang sama
	penghubung	Menunjukkan penghubung pada halaman yang lain

Sumber: (Suarga, 2002)

2. Simbol *flowchart* sistem






Flowchart sistem merupakan simbol-simbol peralatan sistem komputer yang digunakan untuk menyatakan proses pengolahan data ditunjukkan pada Tabel 2.2.

Tabel 2. 2 Simbol Flowchart Sistem

Simbol	Nama Simbol
	<i>keyboard</i>
	<i>printer</i>
	<i>File/storage</i>

Sumber: (Suarga, 2002)

Tabel 2.2 Simbol Flowchart Sistem (Lanjutan)

Simbol	Nama Simbol
	<i>Magnetic tape</i>
	<i>Magnetic disk</i>
	<i>sorting</i>
	<i>extract</i>
	<i>merge</i>

Sumber: (Suarga, 2002)

2.4.2 Flowmap




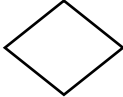




Flowmap adalah campuran peta dan *flowchart* yang menunjukkan pergerakan benda dari satu lokasi ke lokasi lain [16]. *Flowmap* sendiri sering disebut diagram alir atau aliran data yang berbentuk dokumen dalam sebuah aktifitas yang saling berkaitan antara kebutuhan data dan informasi. *Flowmap* dapat membantu programmer untuk memecahkan masalah yang besar ke dalam segmen atau bagian yang lebih kecil.

Fungsi *flowmap* adalah mendefinisikan hubungan antara bagian (pelaku proses), proses (manual/berbasis komputer) dan

aliran data (dalam bentuk dokumen masukan dan keluaran) [16].

Simbol Flowmap dalam Tabel 2.3.

Tabel 2. 3 Simbol Flowmap

Simbol	Keterangan
	<i>Terminator Symbol</i> (Simbol yang menunjukkan awal/akhir dari sistem)
	<i>Process Symbol</i> (Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer)
	<i>Simbol Document</i> (Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak kekertas)
	<i>Decision Symbol</i> (Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau aksi)
	<i>Predefined process Symbol</i> (Simbol untuk menjelaskan proses dalam <i>chart</i> tersendiri)
	<i>Connector Symbol</i> (Simbol untuk keluar atau masuk prosedur atau proses dalam lembar atau halaman)
	<i>Off-line Connector Symbol</i> (Simbol untuk keluar atau masuk proses pada halaman yang lain)
	<i>Input/output Symbol</i> (Simbol yang dapat menerima input atau menampilkan output)


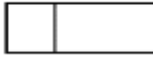



Sumber: (Haidar Bagir1 & Bramantiyo Eko Putro, 2018)

2.4.3 Data Flow Diagram (DFD)

Pada akhir 1970an diagram aliran data (DFD) diperkenalkan dan dipopulerkan untuk analisis dan desain terstruktur (Gane dan Sarson 1979). DFD menunjukkan aliran data dari entitas eksternal ke dalam sistem, menunjukkan bagaimana data dipindahkan dari satu proses ke proses lainnya, serta penyimpanan logisnya [17].

Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem. Sistem dibatasi oleh boundary. Dalam diagram konteks hanya ada satu proses. Tidak boleh ada *store* dalam diagram konteks [16]. Diagram konteks terdiri dari beberapa simbol [16], yang ditunjukkan pada Tabel 2.4

Tabel 2. 4 Simbol Data Flow Diagram (DFD)

Simbol	Keterangan
	<i>External entity</i> adalah sumber atau tujuan aliran data suatu sistem
	<i>Data store</i> adalah tempat penyimpanan data
	<i>Process</i> , menggambarkan bagaimana <i>input</i> diubah menjadi <i>output</i>
	<i>Boundary</i> , merupakan Batasan dari sebuah sistem
	Aliran data, menggambarkan aliran dari suatu proses ke proses lainnya

Sumber: (Suarga, 2002)

2.5 Tools Perangkat Lunak

2.5.1 Visual Studio Code

Visual Studio Code adalah *text editor* yang dikembangkan oleh Microsoft yang dapat berjalan di Windows, Linux dan MacOS. Ini termasuk dukungan untuk *debugging*, *GIT Control* yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode *refactoring*. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema *editor*, *shortcut keyboard*, dan preferensi. Visual Studio Code gratis dan *opensource*, meskipun unduhan resmi berada di bawah lisensi MIT.

2.5.2 AWS Lambda

AWS Lambda atau biasa disebut Lambda merupakan layanan dari Amazon Web Service yang memungkinkan kita menjalankan kode program tanpa harus memikirkan infrastruktur dari server seperti tidak perlu memilih Sistem Operasi. Lambda berjalan oleh sebuah *trigger*, *trigger* tersebut dapat berupa *event* dari *endpoint*, *event* dari *S3 bucket*, maupun dari sumber lainnya.

2.5.3 Amazon Simple Cloud Storage (S3)

Amazon S3 atau biasa disebut S3 adalah kependekan dari *Simple Cloud Storage* yang merupakan layanan penyimpanan dari Amazon Web Service. Amazon S3 memiliki kelebihan yaitu

memiliki ketahanan yang tinggi, sangat mudah digunakan dan memiliki penyimpanan dan *bandwidth* yang tidak terbatas [18].

Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja terdepan di industri. Ini berarti pelanggan dari semua ukuran dan industri dapat menggunakannya untuk menyimpan dan melindungi data dalam jumlah berapa pun untuk berbagai kasus penggunaan, seperti data lake, situs web, aplikasi seluler, pencadangan dan pemulihan, arsip, aplikasi perusahaan, perangkat IoT, dan besar analitik data. Amazon S3 menyediakan fitur manajemen yang mudah digunakan sehingga Anda dapat mengatur data dan mengonfigurasi kontrol akses yang disesuaikan untuk memenuhi persyaratan bisnis, organisasi, dan kepatuhan spesifik Anda. Amazon S3 dirancang untuk ketahanan 99,999999999% (11 9), dan menyimpan data untuk jutaan aplikasi untuk perusahaan di seluruh dunia.

2.5.4 Amazon DynamoDB

Amazon DynamoDB atau biasa disebut Dynamodb merupakan layanan penyimpanan data milik AWS. DynamoDB merupakan database NoSql yang memiliki ketahanan yang tinggi. Amazon DynamoDB adalah database nilai kunci dan dokumen yang memberikan kinerja milidetik satu digit pada skala apa pun. Ini

adalah database yang terkelola sepenuhnya, multi-wilayah, multi-aktif, dan tahan lama dengan keamanan bawaan, pencadangan dan pemulihan, dan penyimpanan dalam memori untuk aplikasi skala internet. DynamoDB dapat menangani lebih dari 10 triliun permintaan per hari dan dapat mendukung puncak lebih dari 20 juta permintaan per detik.

2.5.5 Amazon Elastic Compute Cloud (EC2)

Amazon EC2 atau biasa disebut EC2 adalah kependekan dari *Elastic Compute Cloud* yang merupakan layanan dari Amazon Web Service yang memungkinkan kita memiliki paket komputer lengkap dalam *cloud* dalam hitungan detik [18].

Amazon Elastic Compute Cloud (Amazon EC2) adalah layanan web yang menyediakan kapasitas komputasi yang aman dan dapat diubah ukurannya di cloud. Ini dirancang untuk membuat komputasi awan skala web lebih mudah bagi pengembang. Antarmuka layanan web sederhana Amazon EC2 memungkinkan Anda memperoleh dan mengonfigurasi kapasitas dengan gesekan minimal. Ini memberi Anda kendali penuh atas sumber daya komputasi Anda dan memungkinkan Anda berjalan di lingkungan komputasi Amazon yang telah terbukti.

2.5.6 NodeJs

Node.JS adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi web. Node.JS dapat juga disebut sebagai *runtime environment*. Aplikasi ini ditulis dalam campuran Bahasa C++ dan juga JavaScript, mempunyai model *event driven* (basis event) dan *asynchronous I/O*. Tidak seperti kebanyakan bahasa JavaScript yang dijalankan pada web browser, Node.JS dieksekusi sebagai aplikasi server. Node.JS dapat berjalan di server karena dukungan dari *V8 Engine* buatan Google dan beberapa modul bawaan yang terintegrasi seperti modul *http*, modul *filesystem*, modul *security* dan beberapa modul penting lainnya.

2.5.7 React Js

React.js adalah salah satu web *framework* paling populer di dunia javascript, Dokumentasinya yang lengkap dan penggunaannya yang cukup mudah, dapat membuat kita mengembangkan berbagai produk seperti aplikasi web ataupun RESTful API dan dapat digunakan menjadi pijakan untuk membangun web *framework* yang lebih kompleks.

BAB III

ANALISIS SISTEM

3.1 Gambaran Organisasi

3.1.1 Sejarah Singkat

Radar Cirebon adalah surat kabar harian pagi yang terbit di Cirebon, Jawa Barat. Harian ini masih satu grup dengan Jawa Pos. Lahir dan terbentuk 20 Desember 1999. Radar Cirebon memiliki sirkulasi di Cirebon, Indramayu, Kuningan, dan Majalengka. Saat ini memiliki 28 halaman, terbit setiap hari. kecuali hari minggu terbit dengan 20 halaman.

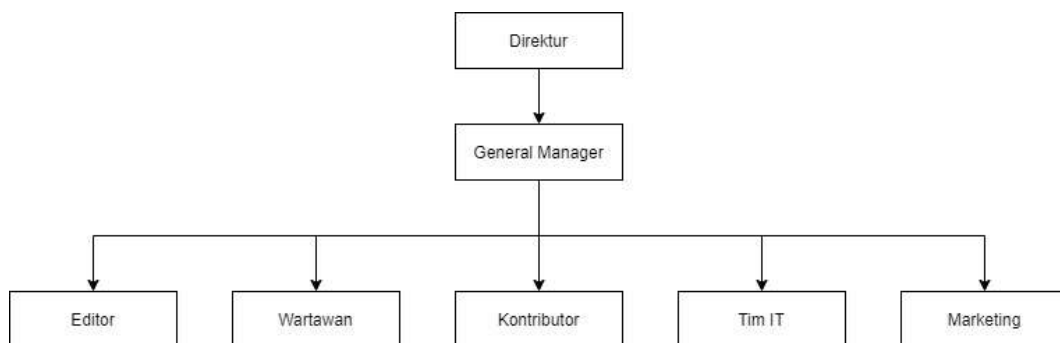
Tahun 2011, Radar Cirebon membangun kantor baru dengan sebutan Grha Pena yang bermarkas di Jalan Perjuangan No 9 Kota Cirebon. Saat ini Radar Cirebon mengembangkan Radar Indramayu, Radar Kuningan dan Radar Majalengka, guna menyajikan berita-berita lokal yang lebih banyak.

Selain koran harian Radar Cirebon juga memiliki stasiun tv lokal dengan sebutan Radar Cirebon TV atau lebih dikenal dengan RCTV, dengan jangkauan wilayah siar Kota/Kabupaten Cirebon dan Indramayu. Untuk masyarakat yang berada di luar wilayah edar Radar Cirebon dapat juga membaca berita-berita wilayah Cirebon dan sekitarnya secara online di www.radarcirebon.com.

3.1.2 Struktur Organisasi dan Uraian Tugas

1. Struktur Organisasi

Berikut adalah struktur organisasi dari Radar Cirebon .com yang dapat dilihat pada Gambar 3.1:



Gambar 3. 1 Struktur Organisasi Radar Cirebon .com

2. Uraian Tugas

a. Direktur

- (1) Menyusun strategi bisnis.
- (2) Melakukan evaluasi.
- (3) Melakukan rapat rutin.
- (4) Implementasi dan melaksanakan Visi dan Misi perusahaan.

b. General Manager

- (1) Mengatur sekaligus memimpin jalannya operasional kantor.

- (2) Mengawasi dan membina manager di tingkat yang lebih rendah.

c. Editor

- (1) Membuat dan memastikan berita yang akan ditayangkan memiliki struktur yang baik.

d. Wartawan

- (1) Melaporkan suatu peristiwa kepada tim editor.
- (2) Melakukan wawancara dengan narasumber.

e. Kontributor

Tugas dari kontributor sama dengan karyawan, yang membedakan hanyalah kontributor berdomisili di daerah-daerah.

f. Tim IT

- (1) Mengelola dan memastikan *Server* Radar Cirebon berjalan dengan semestinya.
- (2) Mengelola dan memastikan *website* radarcirebon.com dan jaringannya berjalan dengan semestinya.

g. Marketing

- (1) Merencanakan promosi.

(2) Meningkatkan Search Engine Optimization (SEO).

3.2 Analisis Sistem Berjalan

Analisis sistem yang sedang berjalan bertujuan untuk mengetahui secara detail prosedur apa saja yang dilakukan.

3.2.1 Uraian Prosedur

Analisa prosedur bertujuan untuk mengetahui cara kerja dari sistem yang sedang berjalan maupun sistem yang sedang dalam pengembangan dan perawatan sehingga kelebihan serta kelemahan sistem tersebut dapat diketahui, hal ini tentu sangat penting dalam proses pengembangan atau perawatan sistem tersebut.

Adapun uraian prosedur sistem absensi di Radar Cirebon dalam penelitian ini adalah sebagai berikut:

1. Prosedur Daftar Karyawan

Tahapan untuk prosedur ini adalah sebagai berikut:

- a. Admin masuk ke aplikasi khusus admin.
- b. Admin memasukan data karyawan yang terdaftar kedalam *database*. Data yang diinputkan meliputi data personal karyawan dan sidik jari.
- c. Admin mengklik tombol simpan.

2. Prosedur Absensi

Tahapan untuk prosedur ini adalah sebagai berikut:

- a. Karyawan menempelkan ibu jari pada mesin deteksi.
- b. Mesin deteksi akan memindai sidik jari.
- c. Mesin deteksi akan mencocokkan sidik jari karyawan dengan sidik jari yang ada di *database*.
- d. Jika sidik jari cocok, maka mesin akan berwarna hijau
- e. Jika sidik jari tidak terdaftar maka mesin akan berwarna merah.

3. Prosedur Membuat Laporan

Tahapan untuk prosedur ini adalah sebagai berikut:

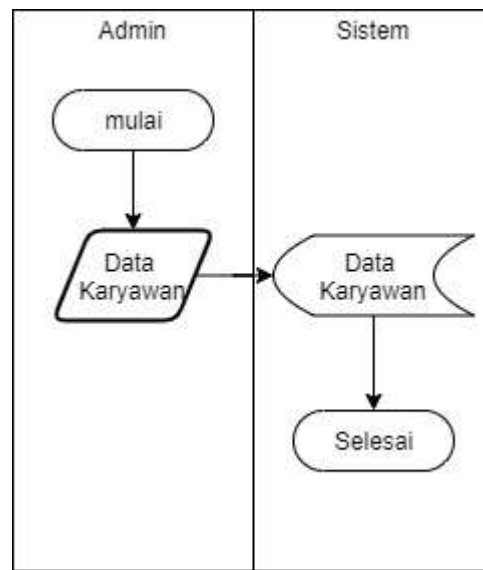
- a. Admin masuk ke aplikasi khusus admin.
- b. Admin mencetak laporan absensi tiap karyawan.
- c. Laporan diberikan ke bagian lain untuk pertimbangan pemberian gaji.

3.2.2 Diagram Sistem Prosedur

Diagram sistem prosedur yang ada pada sistem absensi di Radar Cirebon adalah sebagai berikut:

1. Flowmap Daftar Karyawan

Berikut adalah *Flowmap* Daftar Karyawan yang dapat dilihat pada Gambar 3.2.

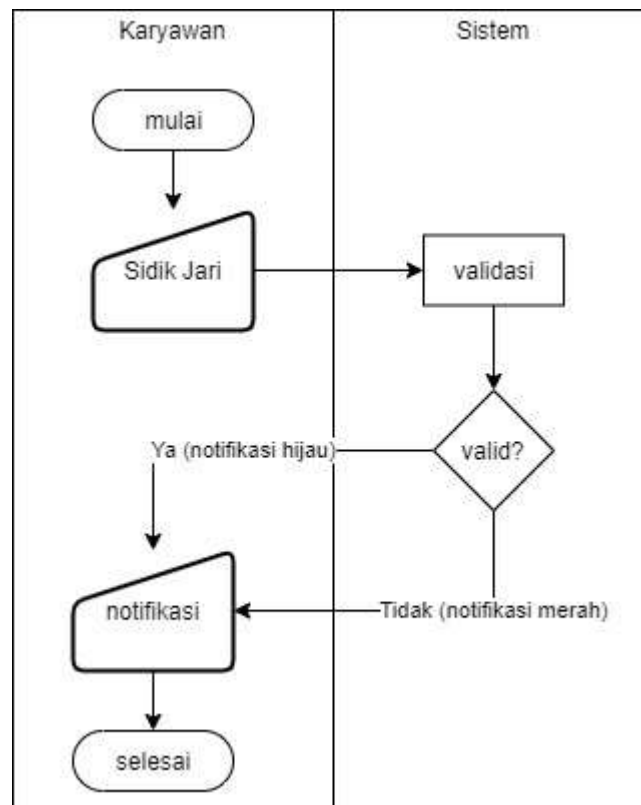


Gambar 3. 2 Flowmap Daftar Karyawan

2. Flowmap Absensi

Berikut adalah *Flowmap* Absensi yang dapat dilihat pada Gambar

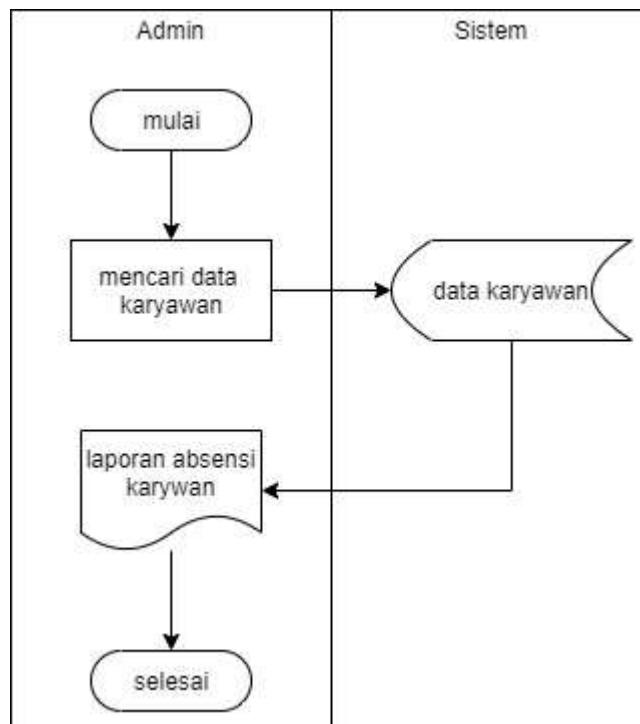
3.3.



Gambar 3. 3 Flowmap Absensi

3. Flowmap Membuat Laporan

Berikut adalah *Flowmap* Membuat Laporan yang dapat dilihat pada Gambar 3.4.



Gambar 3. 4 Flowmap Membuat Laporan

3.2.3 Analisa Dokumen

Analisa dokumen memiliki tujuan yaitu untuk mempelajari dokumen yang terdapat pada sistem, dalam hal ini adalah sistem absensi di Radar Cirebon.

Adapun dokumen yang terdapat pada sistem absensi di Radar Cirebon yang sedang berjalan adalah sebagai berikut:

1. Dokumen Data Karyawan

Nama dokumen : Data Karyawan

Fungsi : Untuk mencatat data karyawan

Sumber : Admin

Bentuk : Formulir

Atribut : nama, *email*, nomor *handphone*, id dan sidik jari

2. Dokumen Absensi

Nama dokumen : Data Absensi

Fungsi : Untuk mencatat data absensi karyawan

Sumber : Pengguna

Bentuk : Formulir

Atribut : Id, *timestamp*, status, nama

3.2.4 Desain Informasi

Hasil Analisa dari dokumen yang ada menghasilkan informasi sebagai berikut:

1. Informasi Data Karyawan

Nama dokumen : Data Karyawan

Fungsi : Untuk mencatat data karyawan

Sumber : Admin

Bentuk : Formulir

Atribut : nama, *email*, nomor *handphone*, id dan sidik jari

2. Informasi Absensi

Nama dokumen	: Data Absensi
Fungsi	: Untuk mencatat data absensi karyawan
Sumber	: Pengguna
Bentuk	: Formulir
Atribut	: Id, <i>timestamp</i> , status, nama

BAB IV

PERANCANGAN SISTEM

4.1 Desain Prosedur

Berdasarkan uraian pada bab sebelumnya, untuk mengatasi permasalahan absensi dimasa pandemi covid-19 di Radar Cirebon, maka pada penelitian ini penulis akan mengimplementasikan pengenalan wajah menggunakan metode Local Binary Pattern Histogram berbasis *website* untuk mempermudah karyawan Radar Cirebon melakukan absensi sekaligus menerapkan protokol kesehatan.

4.1.1 Prosedur Pendaftaran Pengguna

Berikut desain prosedur pendaftaran pengguna yang diajukan oleh penulis, adapapun langkah-langkahnya sebagai berikut:

1. Pengguna melakukan pendaftaran dengan menggunakan *email* dan *password*.
2. Pengguna akan menerima kode *verifikasi* yang akan dikirimkan melalui *email* yang digunakan untuk pendaftaran, masukan kode *verifikasi* ke *form* yang tersedia.
3. Setelah pendaftaran akun selesai pengguna dapat masuk ke aplikasi.

4.1.2 Prosedur Login Pengguna

Berikut desain prosedur *login* pengguna yang diajukan oleh penulis, adapun langkah-langkahnya sebagai berikut:

1. Pengguna melakukan login dengan memasukan alamat *email* yang sudah terdaftar dan *password*.
2. Jika alamat *email/password* salah, silahkan masukan alamat *email/password* yang benar.
3. Jika pengguna berhasil *login* maka pengguna sudah bisa menggunakan aplikasi

4.1.3 Prosedur Kelola Akun Pengguna

Berikut desain prosedur kelola akun pengguna yang diajukan oleh penulis, adapun Langkah-langkahnya sebagai berikut:

1. Pengguna memilih menu untuk melengkapi data diri pengguna.
2. Pengguna mengisi *form* yang tersedia.
3. Pengguna mengisi nama, alamat, nomor *handphone*, dan video yang berisi wajah pengguna.
4. Pengguna menekan tombol *send data* untuk menyimpan perubahan ke *database*.

4.1.4 Prosedur Melatih Model

Berikut desain prosedur melatih model yang diajukan oleh penulis, adapun Langkah-langkahnya sebagai berikut:

1. Pengguna memilih video yang berisi gambar wajahnya untuk di unggah ke server.
2. Membuat dataset wajah dari video menggunakan *library OpenCV*.
 - a. Langkah pertama yaitu mengunduh file video yang disimpan di S3.
 - b. Langkah selanjutnya adalah menambahkan nama video ke database sesuai dengan data pengguna.
 - c. Kemudian muat video menggunakan perintah '*cv2.CaptureVideo*' yang bertujuan untuk memisahkan video *perframe* sehingga dapat menghasilkan foto dari sebuah video.
3. Membuat model dari *dataset* wajah menggunakan metode Local Binary Pattern Histogram.
 - a. Langkah pertama adalah deteksi wajah.

Pada Langkah ini penulis mendeteksi wajah menggunakan bantuan dari *library OpenCV*. Gambar dari dataset akan dibandingkan dengan *Cascade Classifier* sehingga dapat

diketahui bagian wajahnya. Gambar 4.1 merupakan contoh hasil yang diperoleh dari deteksi wajah.



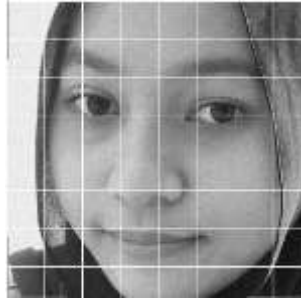
Gambar 4. 1 Hasil Deteksi Wajah

Setelah wajah terdeteksi maka gambar tersebut akan dirubah menjadi gambar abu-abu agar dapat dilatih menjadi sebuah model seperti pada gambar 4.2.



Gambar 4. 2 Merubah Warna Gambar

Langkah selanjutnya yaitu membagi wajah menjadi 64 bagian atau 8x8 seperti terlihat pada Gambar 4.3.



Gambar 4. 3 Pembagian Wilayah Wajah

b. Langkah kedua yaitu ekstraksi fitur.

Pada Langkah ini Gambar 4.3 akan dirubah menjadi bentuk array yang terdiri dari kepadatan/kecerahan sebuah piksel atau bila dituliskan dalam angka akan dihasilkan angka antara 0-255. Penulis mengambil sample sebanyak 54 piksel dari keseluruhan 16384 piksel, Tabel 4.1 merupakan nilai dari 54 piksel tersebut.

Tabel 4. 1 Tingkat Kecerahan Gambar

212	209	207	206	205	202	198	197	195
211	209	208	207	205	201	198	198	195
209	209	209	208	205	201	198	197	197
208	208	209	208	206	202	199	197	198
209	209	209	209	209	206	201	196	197
210	210	209	209	209	207	201	196	195

Langkah selanjutnya akan dicari nilai LBP menggunakan operator LBP yang telah dijelaskan pada BAB II. Dari Tabel 4.1 penulis mengambil nilai 212, 209, 207, 211, 209, 208, 209, 209, 209 sebagai jendela piksel 3x3 yang dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Jendela 3 x 3

212	209	207
211	209	208
209	209	209

Untuk mengetahui nilai tengah jendela 3x3 dari Tabel 4.2, maka perlu dicari nilai LBP dari setiap piksel. Berikut cara mencari nilai LBP dari setiap piksel sesuai dengan Persamaan 1.1 dan Persamaan 1.2.

Mencari nilai LBP pada piksel pertama:

$$LBP(212) = (212 - 209)2^0 = 3 \quad (1.3)$$

$$S(3) = 1 \quad (1.4)$$

Mencari nilai LBP pada piksel kedua:

$$LBP(209) = (209 - 209)2^1 = 0 \quad (1.5)$$

$$S(0) = 1 \quad (1.6)$$

Mencari nilai LBP pada piksel ketiga:

$$LBP(207) = (207 - 209)2^2 = -8 \quad (1.7)$$

$$S(-8) = 0 \quad (1.8)$$

Mencari nilai LBP pada piksel keempat:

$$LBP(211) = (211 - 209)2^3 = 16 \quad (1.9)$$

$$S(16) = 1 \quad (1.10)$$

Mencari nilai LBP pada piksel kelima:

$$LBP(208) = (208 - 209)2^4 = -16 \quad (1.11)$$

$$S(-16) = 0 \quad (1.12)$$

Mencari nilai LBP pada piksel keenam:

$$LBP(209) = (209 - 209)2^5 = 0 \quad (1.13)$$

$$S(0) = 1 \quad (1.14)$$

Mencari nilai LBP pada piksel ketujuh:

$$LBP(209) = (209 - 209)2^6 = 0 \quad (1.15)$$

$$S(0) = 1 \quad (1.16)$$

Mencari nilai LBP pada piksel delapan:

$$LBP(209) = (209 - 209)2^7 = 0 \quad (1.17)$$

$$S(0) = 1 \quad (1.18)$$

Dari hasil persamaan 1.3 hingga persamaan 1.18 diketahui nilai LBP yang jika dibuatkan tabel maka akan seperti Tabel 4.3.

Tabel 4. 3 Hasil Operator LBP

1	1	0
1		0
1	1	1

Pada Tabel 4.3 bisa didapatkan bilangan biner dengan cara mengurutkan nilai dari ujung kiri atas searah jaum jam segingga didapatkan nilai 11001111 atau 207 pada basis 10. Nilai 255 disebut nilai tengah dari Tabel 4.2.

Tabel 4. 4 Konstanta LBP

1	2	4
8		16
32	64	128

Tabel 4.4 merupakan nilai konstanta LBP yang akan dikalikan dengan Tabel 4.3 dan menghasilkan Tabel 4.5.

Tabel 4. 5 Hasil Operator LBP Setelah Dikalikan

1	2	0
8	207	0
32	64	128

Selanjutnya untuk jendela piksel 3x3 yang lain perhitungannya dapat diulang dari mulai Tabel 4.3 hingga Tabel 4.5. Setelah semua piksel sudah dihitung menggunakan Operator LBP, maka diperoleh histogram seperti pada Gambar 2.4.

- c. Membuat *dataset* yang berisi nilai LBP dari setiap gambar wajah.

Dalam tahap ini penulis menggunakan *library OpenCV* untuk membuat dataset yang berisi Histogram LBP dari setiap gambar wajah yang ada di *dataset* wajah.

4.1.5 Prosedur Memprediksi Wajah

Berikut desain prosedur memprediksi wajah yang diajukan oleh penulis, adapun Langkah-langkahnya sebagai berikut:

1. Pengguna mengambil gambar wajah.
2. Pengguna mengunggah foto wajah ke s3.
3. Sistem akan mengunduh gambar yang dikirimkan pengguna.
4. Pengenalan wajah menggunakan metode Local Binary Pattern Histogram.
 - a. Mendeteksi wajah.

Pada Langkah ini penulis mendeteksi wajah menggunakan bantuan dari *library OpenCV*. Gambar yang diunduh akan

dibandingkan dengan *Cascade Classifier* sehingga dapat diketahui bagian wajahnya. Gambar 4.1 merupakan contoh hasil yang diperoleh dari deteksi wajah.

Setelah wajah terdeteksi maka gambar tersebut akan dirubah menjadi gambar abu-abu agar dapat dilatih menjadi sebuah model seperti pada gambar 4.2. Langkah selanjutnya yaitu membagi wajah menjadi 64 bagian atau 8x8 seperti terlihat pada Gambar 4.3. Langkah selanjutnya sama seperti Langkah-langkah ekstraksi fitur pada Prosedur Memprediksi Wajah.

Selanjutnya setelah melakukan ekstraksi fitur dan mendapatkan nilai Histogram LBP maka dilakukan pengukuran jarak antara histogram LBP dengan histogram dari dataset yang sudah dilatih sebelumnya. Mengukur jarak dari dua histogram dapat menggunakan persamaan *euclidean distance* seperti pada persamaan 1.19.

$$D = \sqrt{\sum_{i=1}^n (hist1_i - hist2_i)^2} \quad (1.19)$$

Hasil dari persamaan 1.19 merupakan jarak antara kedua histogram, nilai ini tidak memiliki batas namun semangin kecil (mendekati 0) nilainya maka tingkat kemiripan semakin tinggi maupun sebaliknya jika nilainya semakin besar (menjauhi 0) maka tingkat kemiripannya semakin kecil.

4.2 Desain Dokumen dan informasi

4.2.1 Desain Dokumen

Desain dokumen baru pada *website* website Absensi Online Radar Cirebon setelah diimplementasikan dengan teknik pengenalan wajah adalah sebagai berikut:

1. Dokumen pengguna

Nama dokumen : Data Pengguna
 Fungsi : Untuk melengkapi data pengguna
 Sumber : Pengguna
 Bentuk : Formulir
 Atribut : nama, *email*, nomor *handphone*, id wajah dan alamat

2. Dokumen absensi

Nama dokumen : Data Absensi
 Fungsi : Untuk bukti absensi
 Sumber : Pengguna/ Sistem
 Bentuk : Formulir
 Atribut : *_typename*, *confidence*, *face_id*, *time*, *user_id*, *year*, dan *location*.

4.2.2 Desain Informasi

Berdasarkan desain dokumen yang baru menghasilkan informasi sebagai berikut:

1. Informasi Data Pengguna

Nama dokumen : Data Pengguna
 Fungsi : Untuk melengkapi data pengguna
 Sumber : Pengguna
 Bentuk : Formulir
 Atribut : nama, *email*, nomor *handphone*, id wajah dan alamat

2. Informasi Absensi

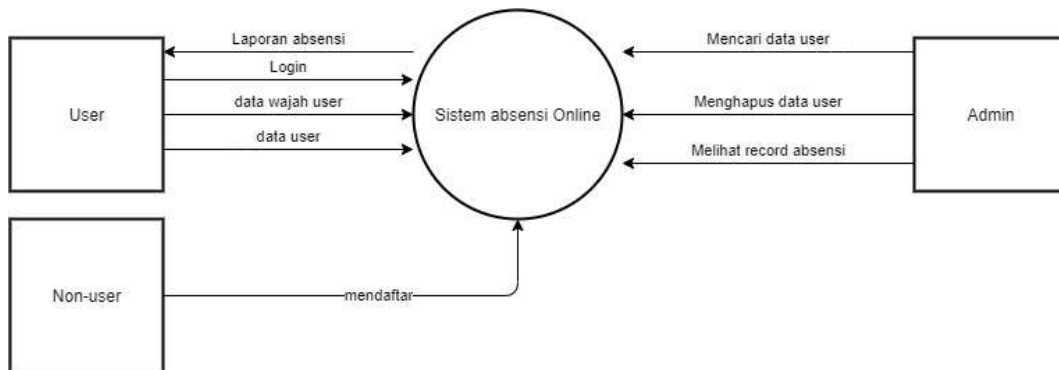
Nama dokumen : Data Absensi
 Fungsi : Untuk bukti absensi
 Sumber : Pengguna/ Sistem
 Bentuk : Formulir
 Atribut : *_typename*, *confidence*, *face_id*, *time*, *user_id*, *year*, dan *location*.

4.3 Desain Aliran Data

Pada penelitian ini digunakan *flowmap* dan *data flow diagram* untuk menggambarkan aliran data dari sistem yang akan dibuat. *Flowmap* dan *data flow diagram* adalah sebagai berikut:

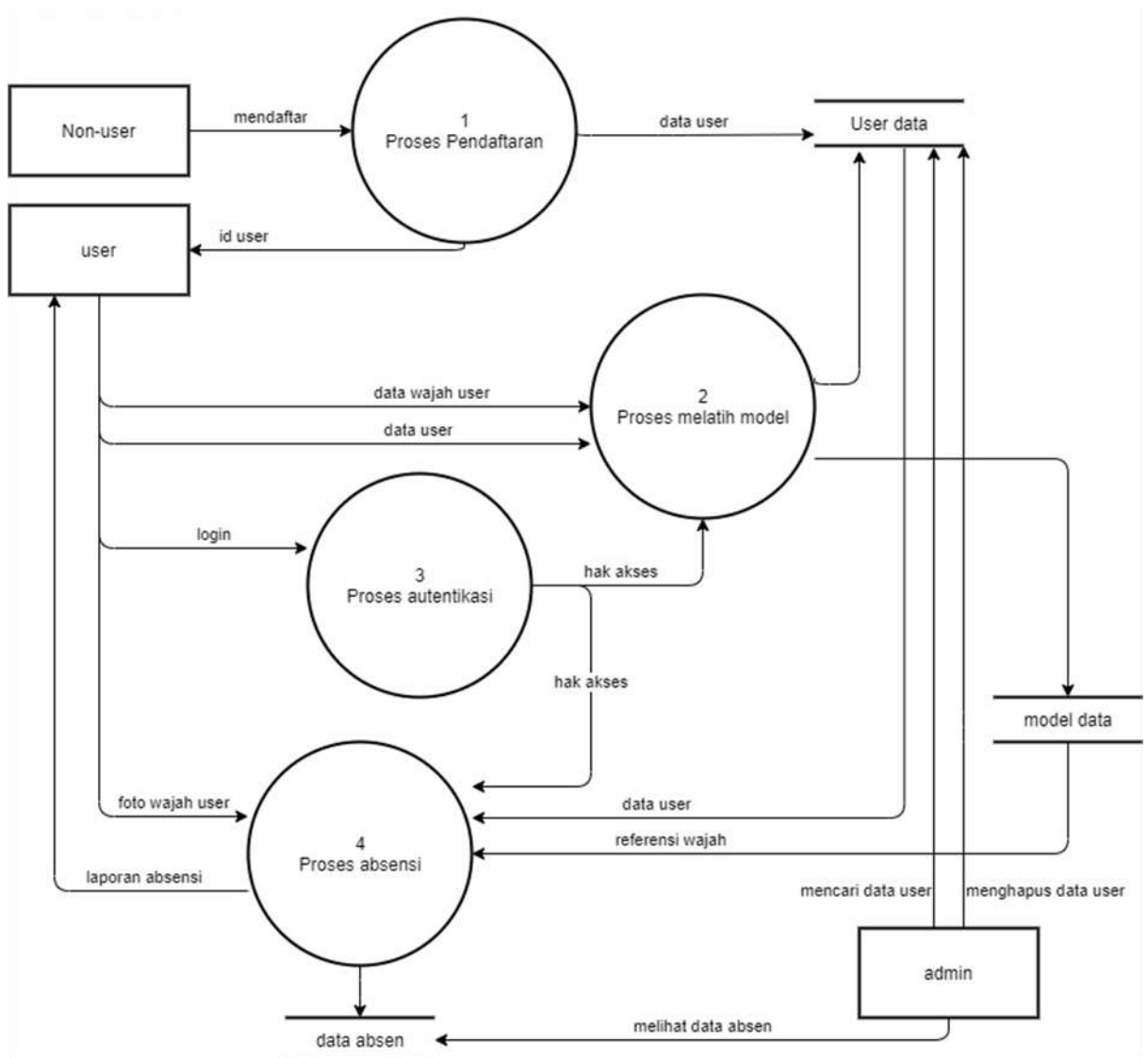
4.3.1 Data Flow Diagram

Berikut adalah Diagram Konteks yang penulis buat pada penelitian ini yang dapat dilihat pada Gambar 4.4.



Gambar 4. 4 Diagram Konteks

Dari Gambar 4.4 diketahui ada 3 *external entity* yang berinteraksi dengan sistem absensi online yaitu *User*, *Non-User*, dan *Admin*. Ketiga *external entity* tersebut memiliki dapat melakukan beberapa kegiatan. Kegiatan yang dilakukan oleh *external entity* dijelaskan lebih detail pada Gambar 4.5. Pada Gambar 4.5 terdapat 4 proses yaitu proses pendaftaran, proses melatih model, proses autentikasi, dan proses absensi.



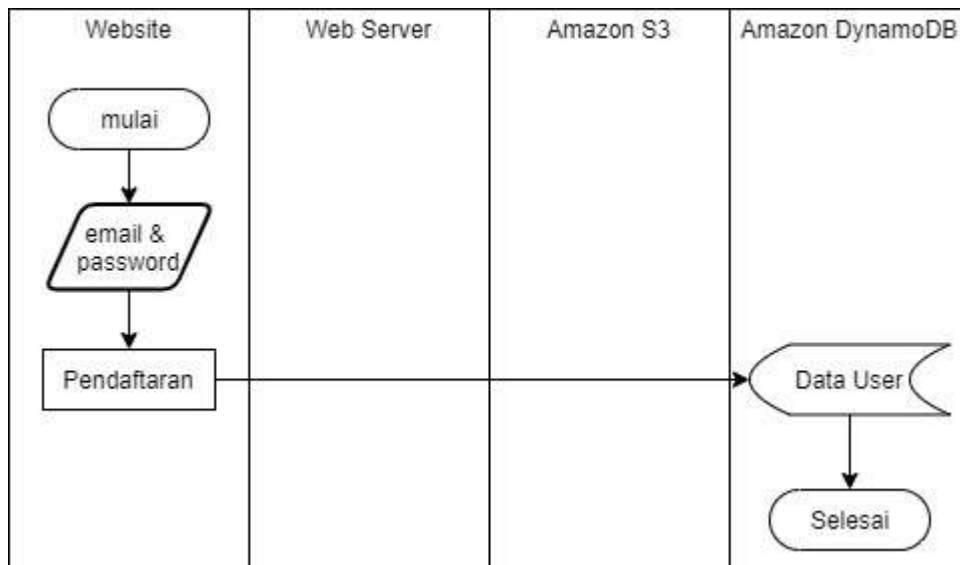
Gambar 4. 5 DFD Level 1

4.3.2 Flowmap

Untuk mempermudah dalam pembuatan program penulis menggunakan *flowmap* sebagai acuan aliran data dalam sebuah program. *Flowmap* aliran data program dibagi berdasarkan prosedur sebagai berikut:

1. Flowmap Pendaftaran

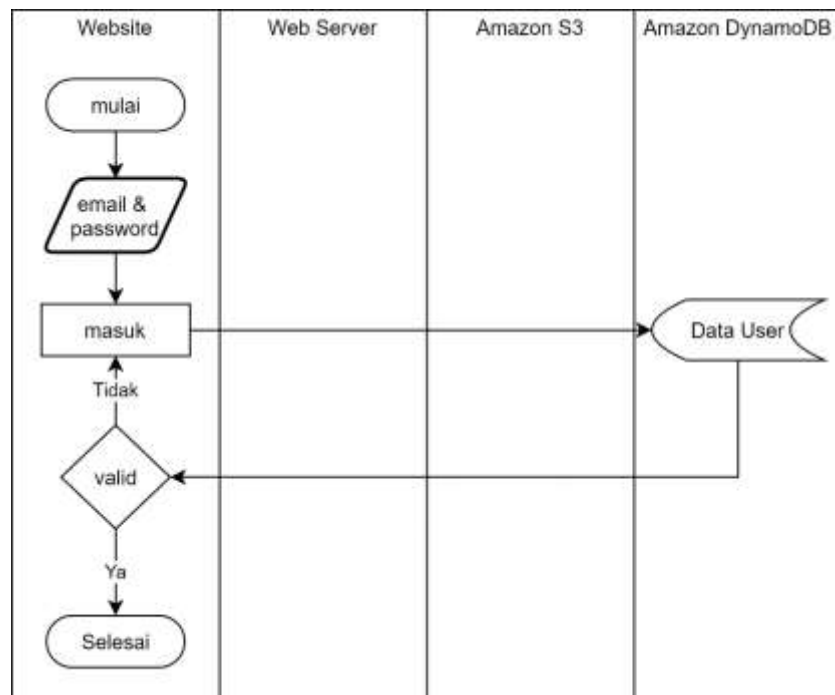
Berikut adalah *Flowmap* Pendaftaran yang dapat dilihat pada Gambar 4.6.



Gambar 4. 6 Flowmap Pendaftaran

2. Flowmap Login

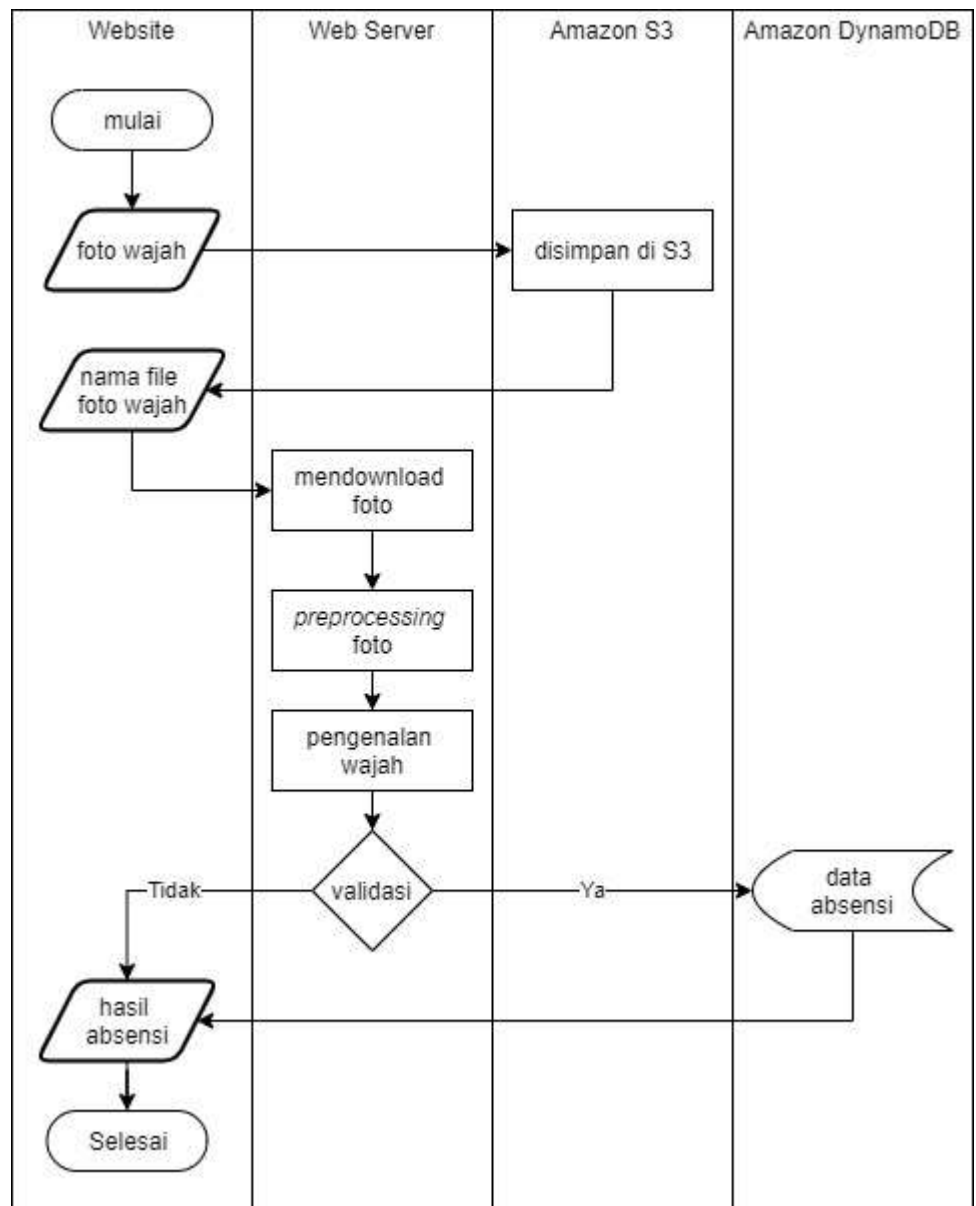
Berikut adalah *flowmap login* yang dapat dilihat pada Gambar 4.7.



Gambar 4. 7 Flowmap Login

3. Flowmap Memprediksi Wajah (Absensi)

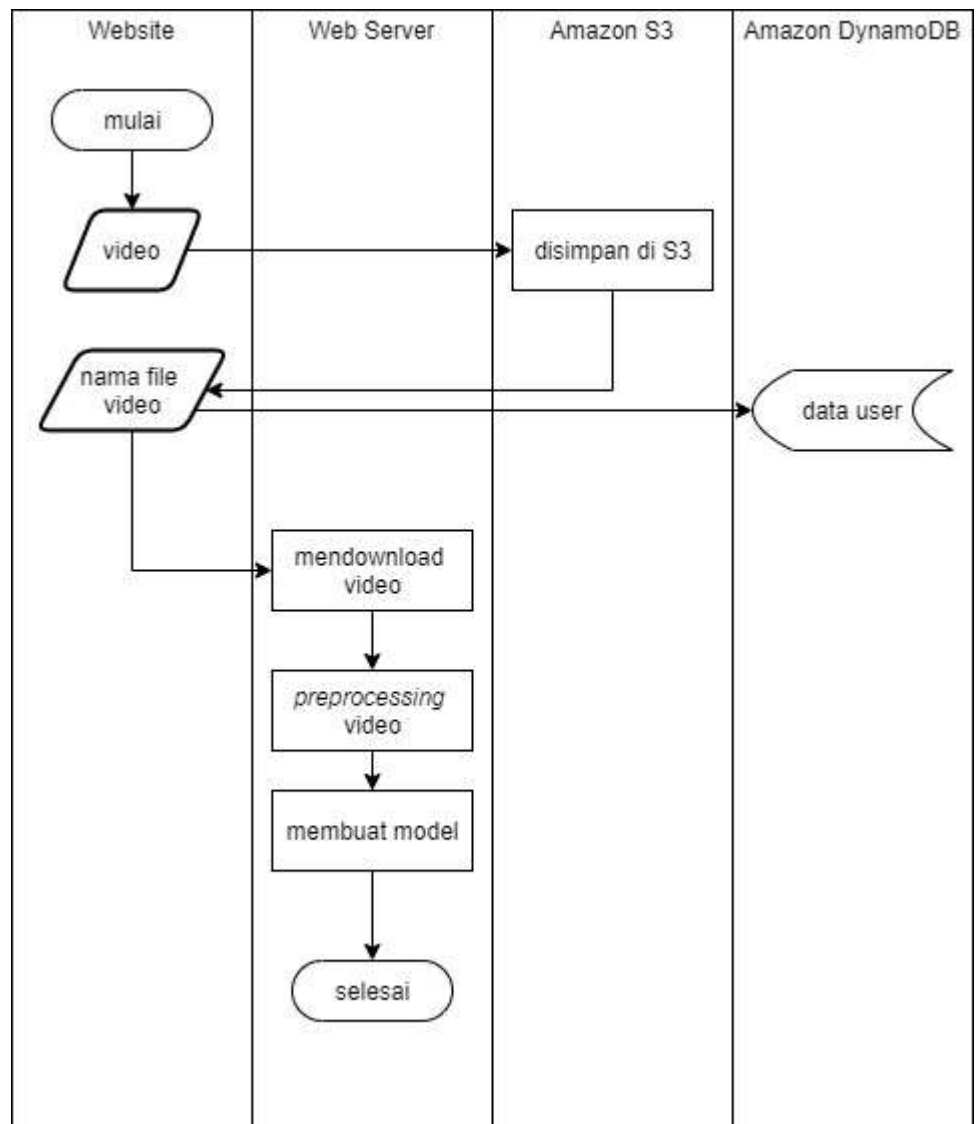
Berikut adalah *flowmap* memprediksi wajah yang dapat dilihat pada Gambar 4.8.



Gambar 4. 8 Flowmap Memprediksi Wajah

4. Flowmap Melatih Model

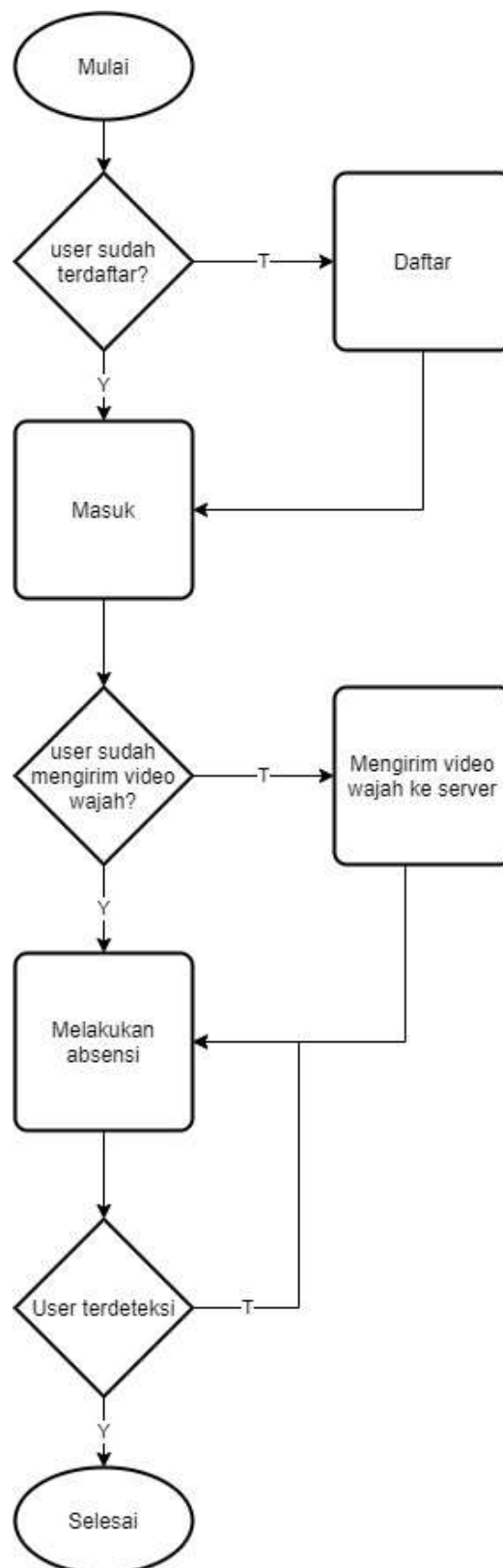
Berikut adalah *flowmap* melatih model yang dapat dilihat pada Gambar 4.9.



Gambar 4. 9 Flowmap Melatih Model

4.3.3 Flowchart

Untuk mempermudah dalam pembuatan program penulis menggunakan *flowchart* sebagai acuan alur program. *Flowchart* alur program ditunjukkan oleh Gambar 4.10 berikut:



Gambar 4. 10 Flowchart Program

4.3.4 Skema Database

Database nosql adalah *database* yang tidak memiliki perintah SQL dan konsep penyimpanannya semistuktural atau tidak struktural dan tidak harus memiliki relasi layaknya tabel-tabel MySQL. Dalam penelitian ini penulis memilih DynamoDB sebagai *database nosql* karena DynamoDB memiliki kecepatan baca yang konstan sehingga cocok untuk aplikasi *realtime*.

1. Skema Tabel Pengguna

Berikut adalah skema yang penulis buat dalam bentuk json untuk tabel pengguna yang dapat dilihat pada Tabel 4.15.

Tabel 4. 6 Skema Tabel Pengguna

```
{
  "address": {
    "S": ""
  },
  "comolete": {
    "BOOL": ""
  },
  "edited_at": {
    "S": ""
  },
  "face_id": {
    "S": ""
  },
  "id": {
    "S": ""
  },
  "name": {
    "S": ""
  },
  "phone_number": {
    "S": ""
  },
  "picture": {
    "S": ""
  }
}
```

2. Skema Tabel Data Absensi

Berikut adalah skema yang penulis buat dalam bentuk json untuk tabel data absensi yang dapat dilihat pada Tabel 4.16.

Tabel 4. 7 Skema Tabel Data Absensi

```
{
  "_typename": {
    "S": ""
  },
  "confidence": {
    "BOOL": ""
  },
  "face_id": {
    "S": ""
  },
  "time": {
    "S": ""
  },
  "id": {
    "S": ""
  },
  "user_id": {
    "S": ""
  },
  "year": {
    "S": ""
  },
  "location": {
    "S": ""
  }
}
```

3. Skema Model

Berikut adalah skema yang dihasilkan oleh OpenCV dalam bentuk yaml yang dapat dilihat pada Tabel 4.17.

Tabel 4. 8 Skema Model

```
opencv_lbphfaces:
  treshold:
  radius:
  neighbors:
  grid_x:
```



```

grid_y:
  histogram:
    -!!opencv-matrix
      rows:
      cols:
      dt:
      data:
  labels: !!opencv-matrix
    rows:
    cols:
    dt:
    data:
  labelsInfo:
    []

```

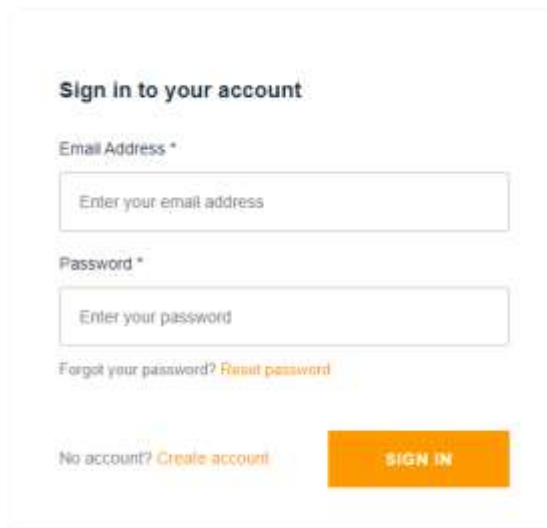
4.4 Desain Interface dan Struktur Menu

4.4.1 Desain Interface

Desain adalah sesuatu yang sangat dibutuhkan dalam membuat sebuah sistem, tujuannya agar sistem tersebut mudah untuk digunakan oleh pengguna. Adapun desain *interface* untuk *website* Absensi Online Radar Cirebon setelah diimplementasikan dengan teknik pengenalan wajah adalah sebagai berikut:

1. Desain Halaman Login

Berikut adalah Desain Halaman *Login* yang dapat dilihat pada Gambar 4.11:

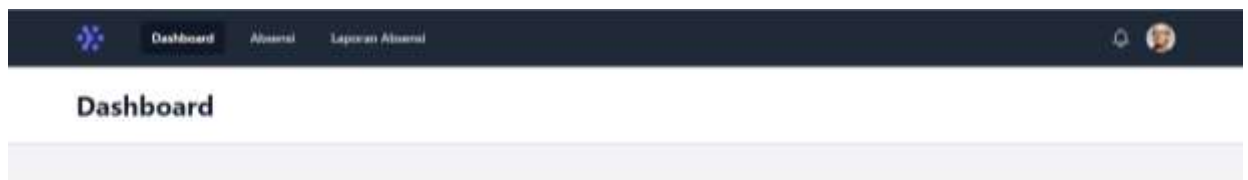


The image shows a login form titled "Sign in to your account". It contains two input fields: "Email Address *" with a placeholder "Enter your email address" and "Password *" with a placeholder "Enter your password". Below the password field is a link "Forgot your password? Reset password". At the bottom left is a link "No account? Create account", and at the bottom right is an orange "SIGN IN" button.

Gambar 4. 11 Desain Halaman Login

2. Desain Halaman Dashboard

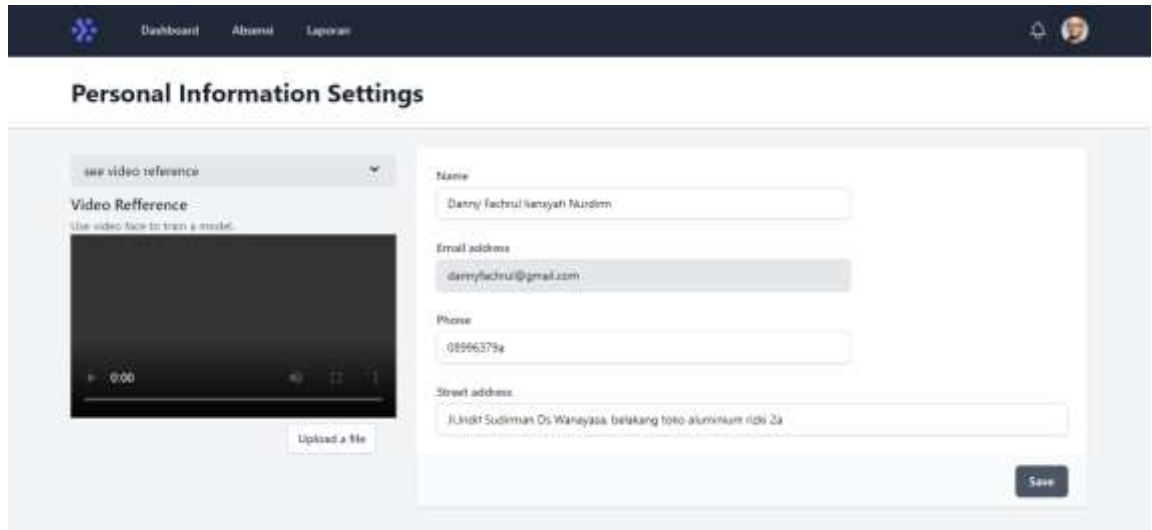
Berikut adalah Desain Halaman Setting yang dapat dilihat pada Gambar 4.12:



Gambar 4. 12 Desain Halaman Dashboard

3. Desain Halaman Setting

Berikut adalah Desain Halaman *Setting* yang dapat dilihat pada Gambar 4.13:



Dashboard Absensi Laporan

Personal Information Settings

see video reference

Video Reference
Use video face to train a model.

0:00

Upload a file

Name
Danny Fachrul Ihsanah Mardim

Email address
dannyfachrul@gmail.com

Phone
08506379g

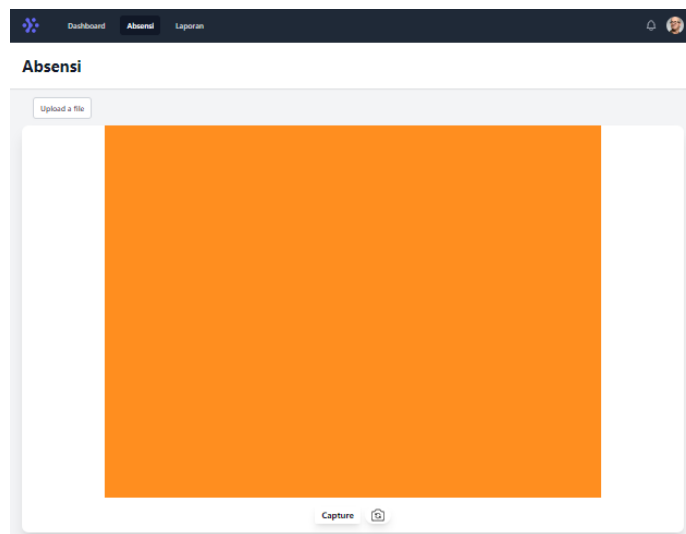
Street address
Jl.Indit Sudirman Di Wanayasa betakang foto aluminium rido Za

Save

Gambar 4. 13 Desain Halaman Setting

4. Desain Halaman Absensi

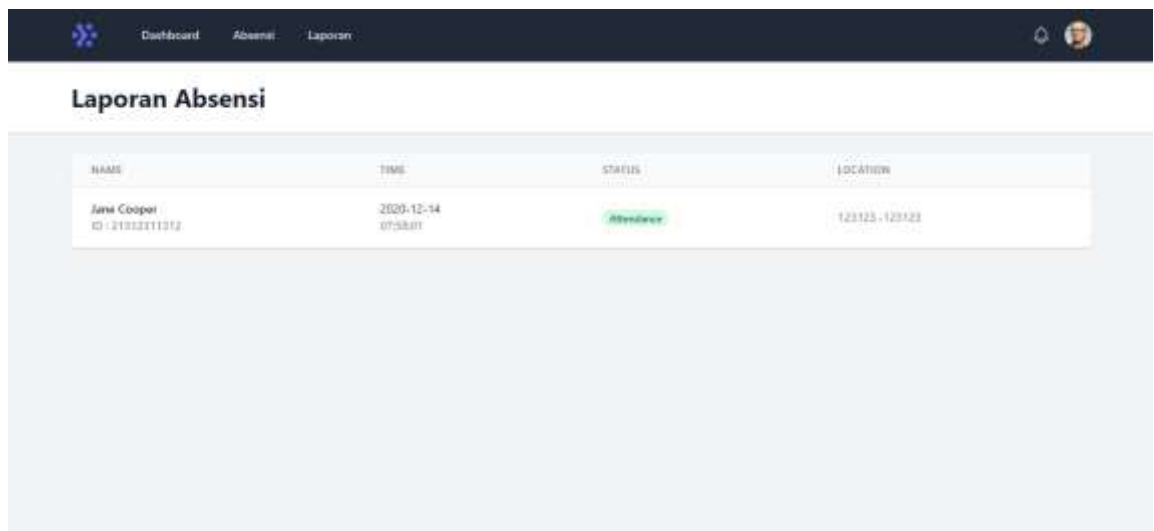
Berikut adalah Desain Halaman Absensi yang dapat dilihat pada Gambar 4.14:



Gambar 4. 14 Desain Halaman Absensi

5. Desain Halaman Laporan

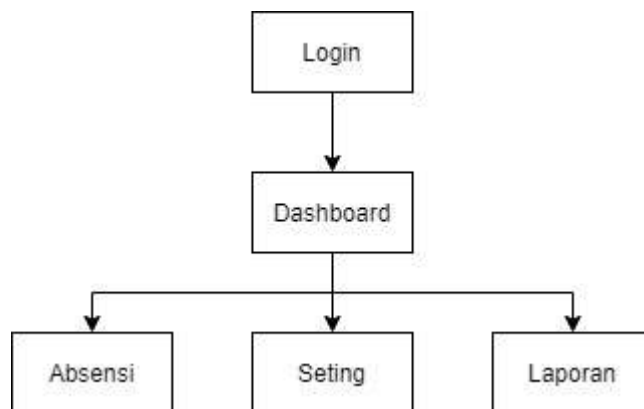
Berikut adalah Desain Halaman Laporan yang dapat dilihat pada Gambar 4.15:



Gambar 4. 15 Desain Halaman Laporan

4.4.2 Struktur Menu

Berikut adalah struktur dari website Absensi Online Radar Cirebon setelah diimplementasikan dengan teknik pengenalan wajah dapat dilihat pada Gambar 4.16:



Gambar 4. 16 Struktur Menu Sistem

4.5 Implementasi Sistem

4.5.1 Konfigurasi Perangkat Lunak

Kebutuhan perangkat lunak untuk mengelola server pengenalan wajah guna mendukung *website* Absensi Online Radar Cirebon yang dijalankan dalam Amazon EC2 adalah sebagai berikut:

1. Linux Ubuntu 18.04.
2. Nginx.
3. Flask.
4. Gunicorn.

5. Sedangkan perangkat lunak yang digunakan untuk pengembangan website adalah Visual Studio Code dan Postman.

4.5.2 Konfigurasi Perangkat Keras

Dalam pembuatan *website* Absensi Online Radar Cirebon penulis hanya menggunakan perangkat keras pada saat pengembangan *website*, karena penulis menggunakan Amazon Web Service sebagai sarana *hosting* sehingga tidak membutuhkan perangkat keras secara langsung. Berikut perangkat keras yang digunakan pada saat pengembangan *website*.

1. Processor Intel Core i3 7100.
2. Ram 4 GB.
3. Penyimpanan 250 GB.
4. Mouse, keyboard, monitor, dan webcam.

4.5.3 Implementasi Program

1. Sintaks Pembuatan Dataset

Berikut adalah Sintaks Pembuatan Dataset yang dapat dilihat pada Tabel 4.18:

Tabel 4. 9 Sintaks Membuat Dataset

```
def
createDataset(self,samples,cam,dataset_name):
    fig, axs =
plt.subplots(10,5,figsize=(20,20),
facecolor='w', edgecolor='k')
    fig.subplots_adjust(hspace=.5,
wspace=.001)
    self.dataset_path(dataset_name)
```

Tabel 4. 9 Sintaks Membuat Dataset

```

count = 0
face_id = self.AddUser()
print('\n[INFO] Membuat dataset')
while(True):
    success, image = cam.read()
    # convert image to grayscale
    gray = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY)
    faces =
self._Face_Cascade.detectMultiScale(gray,
scaleFactor = 1.098, minNeighbors = 6, minSize
= (50, 50))
    if(len(faces)> 1):
        print('\n[WARNING] Terdeteksi
lebih dari 1 wajah')
        continue
    try:
        for _,face in enumerate(faces):
            x, y, w, h = face
            gray_chunk = gray[y-30: y +
h + 30, x-30: x + w + 30]
            image_chunk = image[y: y +
h, x: x + w]
            self.create_Rect(image,
face, [0,255,0])

            print("\n[INFO] Adding
image number {} to the dataset".format(count))
            # Save image
            cv2.imwrite("dataset/User."
+ str(face_id) + '.' + str(count) + ".jpg " ,
                image)
            axs[int(count/5)][count%5].imshow(image,cmap='g
ray', vmin=0, vmax=255)

            axs[int(count/5)][count%5].set_title("Person."
+ str(face_id) + '.' + str(count) + ".jpg ",
                fontdict={'fontsize':
15,'fontweight': 'medium'})
            axs[int(count/5)][count%5].axis('off')
            count += 1
        except Exception as e:
            print(e)
            print('[WARNING] Ada error')
            continue
        if cv2.waitKey(1) & 0xff == 27:
            break
        elif count >= samples:
            break

```

*Sintaks selengkapnya pada Lampiran 1

2. Sintaks Membuat Model

Berikut adalah Sintaks Membuat Model yang dapat dilihat pada

Tabel 4.19:

Tabel 4. 10 Sintaks Membuat Model

```
def train_classifier(faces, faceID):

    face_recognizer=cv2.face.LBPHFaceRecognizer_create()

    face_recognizer.train(faces,np.array(faceID))

    face_recognizer.write(f'{os.path.dirname(os.path.realpath(__file__))}/model.yml')
    return face_recognizer
```

*Sintaks selengkapnya pada Lampiran 2

3. Sintaks Pengenalan Wajah

Berikut adalah Sintaks Pengenalan Wajah yang dapat dilihat pada

Tabel 4.20:

Tabel 4. 11 Sintaks Pengenalan Wajah

```
def predict(self):
    logging.info('Recognizer start')
    recognizer =
cv2.face.LBPHFaceRecognizer_create()
    recognizer.read('model.yml')
    faceCascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    font = cv2.FONT_HERSHEY_SIMPLEX
    ID = None
    CONF = None
    ### resize image
    imgs =
cv2.resize(img, (0,0), None, 0.25, 0.25)

    # convert to gray scale
    gray = cv2.cvtColor(imgs,
cv2.COLOR_BGR2GRAY)
```


Tabel 4. 11 Sintaks Pengenalan Wajah

```

    ### detect the face
    faces =
faceCascade.detectMultiScale(gray,scaleFactor=1
.05,minNeighbors=4,minSize=(30, 30))
    threshold =
cv2.face_LBPHFaceRecognizer.getThreshold
    print(threshold)

    for (x,y,w,h) in faces:
        Id,conf =
recognizer.predict(gray[y:y+h,x:x+w])
    print('ID {0}, confidence {1}'.format(Id,
conf))

        ID = Id
        CONF = conf
        logging.info('Predict result: {} with
{} distance'.format(ID, CONF))
        print('ID {0}, confidence
{1}'.format(ID, CONF))
    return ID,CONF

```

*Sintaks selengkapnya pada Lampiran 3

4. Sintaks Server Pengenalan Wajah

Berikut adalah Sintaks Server Pengenalan Wajah yang dapat dilihat pada Tabel 4.21:

Tabel 4. 12 Sintaks Server Pengenalan Wajah

```

@app.route('/recognize/predict',
methods=['POST'])
def predict():
    # download video from s3
    downloader_client =
downloader(key=image,bucket='dannynurdin',
destination=image_path)
    downloader_client.download()

    model = Predict(image_path)

    res = model.print()
    id,c = model.predict()
    logging.info('DATA => id from model ==
', id)
    print('DATA => id from model == ', id)

```

Tabel 4. 12 Sintaks Server Pengenalan Wajah

```

        logging.info('success {} -
{}'.format(id, c))
        if c:
            disMax = 140.0
            simMax = 100
            similarity = simMax -
simMax/disMax*c
            conf = similarity

@app.route('/recognize/train',
methods=['POST'])
def train():

    # download video from s3
    downloader_client =
downloader(key=key,bucket=bucket_name,
destination=video_path)
    downloader_client.download()
    # get data user
    dynamodb_client = updateData(id =
username, key = key)
    ress = dynamodb_client.update()
    print(ress)
    # start recognize using opencv
    model =
Train(face_cascade,var,username) # create
instance train
    video = cv2.VideoCapture(video_path) #
load video

model.createDataset(samples,video,dataset_name)
# create dataset
    # id =
model.train(dataset_name,file_name)

    faces,faceID =
re.labels_for_training_data('dataset')

face_recognizer=re.train_classifier(faces,faceI
D)

face_recognizer.save(f'{os.path.dirname(os.path
.realpath(__file__))}/model.yml')
    print(f'faces: {len(faces)} , id:
{len(faceID)}')
'

@app.route('/update', methods=['POST'])
def update():
    expression = 'Set ' + ','.join([str(elem)
for elem in Expression])

```

Tabel 4. 12 Sintaks Server Pengenalan Wajah

```

dynamodb_client = updateDataNew(id)
ress =
dynamodb_client.update(AttrName,AttrValue,
expression)

```

*Sintaks selengkapnya pada Lampiran 4

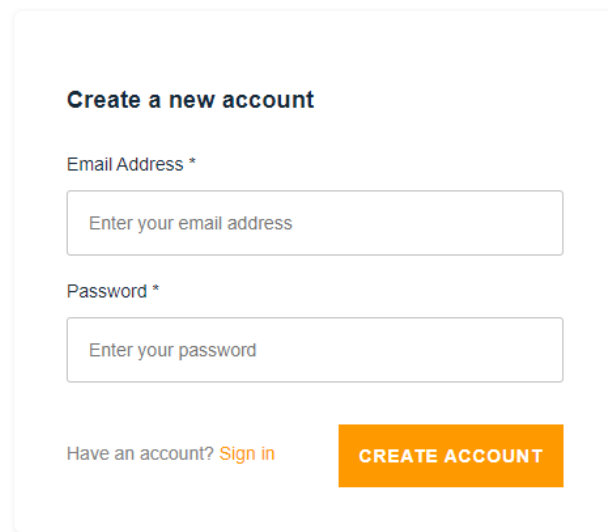
4.5.4 Pedoman Pengoprasian Perangkat Lunak

Pedoman pengoperasian berfungsi tatacara untuk membantu pengguna dalam menggunakan sebuah sistem, pedoman lebih membatu pengguna dibandingkan harus mencoba semua menu yang ada. Adapun pedoman dalam menggunakan *website* Absensi Online Radar Cirebon setelah diimplementasikan dengan teknik pengenalan wajah adalah sebagai berikut:

1. Proses Pendaftaran

- a. Masukan *email* yang akan didaftarkan.
- b. Masukan *password* yang akan anda gunakan, pastikan *password* harus merupakan *alphanumeric*.
- c. Setelah pendaftaran pengguna harus mengkonfirmasi akun menggunakan kode yang diberikan ke *email* yang terdaftar.

Berikut adalah tampilan halaman pendaftaran yang dapat dilihat pada Gambar 4.17:



Create a new account

Email Address *

Enter your email address

Password *

Enter your password

Have an account? [Sign in](#)

CREATE ACCOUNT

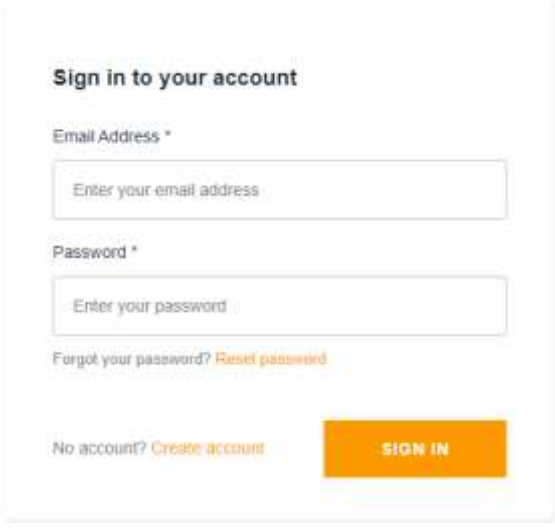
Gambar 4. 17 Halaman Pendaftaran

2. Proses *Login*

- a. Masukkan *email* dan *password* yang telah terdaftar.
- b. Klik tombol masuk
- c. Jika *email* dan *password* benar, maka akan dialihkan ke halaman *dashboard*.

Berikut adalah tampilan halaman *login* yang dapat dilihat pada

Gambar 4.18:



The image shows a login form titled "Sign in to your account". It contains two input fields: "Email Address *" with a placeholder "Enter your email address" and "Password *" with a placeholder "Enter your password". Below the password field is a link "Forgot your password? Reset password". At the bottom left is a link "No account? Create account", and at the bottom right is an orange button labeled "SIGN IN".

Gambar 4. 18 Halaman Login

3. Proses Menambahkan Informasi Pengguna

Untuk menambahkan informasi pengguna maka pilih menu *setting*. Pada halaman *setting* berisi informasi dari pengguna seperti nama, alamat *email*, alamat dan nomer telepon. Di halaman ini juga pengguna bisa memasukan data wajahnya yang berupa file video. Tampilan halaman *setting* dapat dilihat pada Gambar 4.19 berikut:

Personal Information Settings

see video reference

Video Reference
Use video face to train a model.

0:00

Upload a file

Name
Denny Fachrud Alanzah Nordin

Email address
dennyfachu@gmail.com

Phone
0896379200

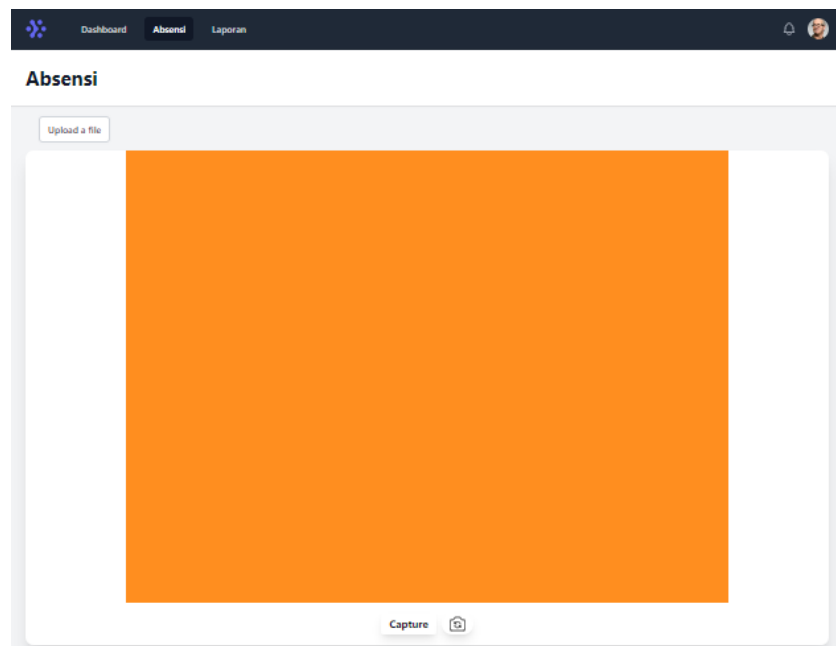
Street address
A.Indit Sudirman Di Widyajaya

Save

Gambar 4. 19 Tampilan Setting

4. Proses Absensi

Untuk melakukan absensi maka pilih menu absensi. Pada halaman absensi terdapat tombol untuk mengunggah foto dan kamera yang bisa digunakan untuk menangkap gambar wajah pengguna. Tampilan halaman absensi dapat dilihat pada Gambar 4.20 berikut:



Gambar 4. 20 Halaman Absensi

4.5.5 Pengujian

Pengujian sistem dilakukan dengan maksud mengetahui sistem bekerja sesuai dengan harapan atau tidak. Berikut Pengujian yang dilakukan

1. Prosedur I

Pada prosedur pertama akan dilakukan pengujian terhadap lamanya durasi video. Pengujian ini bertujuan untuk mengetahui berapa lama durasi video minimal untuk mendapatkan jumlah *frame* yang diinginkan. Dalam penelitian ini penulis menetapkan jumlah *frame* yang diinginkan yaitu 15 *frame* per video dan hasil dari pengujian dapat dilihat pada Tabel 4.13.

Tabel 4. 13 Durasi Video

Durasi Video	Jumlah Frame
3 detik	6
5 detik	10
10 detik	17
15 detik	21

Dari proses pengujian yang dilakukan pada Tabel 4.13 terlihat bahwa durasi video 3 detik dan 5 detik menghasilkan jumlah *frame* kurang dari 15, sedangkan video dengan durasi 10 detik dan 15 detik menghasilkan jumlah *frame* diatas 15. Hasil pengujian ini tergantung banyak aspek seperti *fps* video, spesifikasi komputer, dan program yang dibuat.

2. Prosedur II

Pada prosedur kedua akan dilakukan pengujian akurasi model terhadap berbagai kriteria gambar diantaranya gambar wajah dengan berbagai ekspresi dan pose. Hasil pengujian akurasi model dengan gambar seseorang yang sedang tersenyum dapat dilihat pada Tabel 4.14.

Tabel 4. 14 Hasil Pengujian Muka Senyum

No	Id Pengguna	Hasil Prediksi	Jarak	Prediksi
1	1	1	110	Benar
2	1	1	111	Benar
3	1	1	109	Benar
4	1	1	122	Benar
5	1	1	116	Benar
6	1	-	-	Salah
7	1	1	90	Benar
8	1	1	79	Benar
9	1	-	-	Salah
10	1	-	-	Salah

*Tabel selengkapnya pada Lampiran 7

Keterangan:

- Jika id pengguna sama dengan hasil prediksi maka prediksi pengenalan wajah berhasil.
- Jika hasil prediksi kosong, maka saat pengujian tidak ada wajah yang terdeteksi.

Dari proses pengujian yang dilakukan pada Tabel 4.14 terlihat bahwa 14 dari 20 pengujian berhasil memprediksi wajah dengan benar. Selanjutnya dilakukan pengujian dengan gambar seseorang dengan wajah datar, hasilnya dapat dilihat pada Tabel 4.15.

Tabel 4. 15 Hasil Pengujian Muka Datar

No	Id Pengguna	Hasil Prediksi	Jarak	Prediksi
1	1	1	76	Benar
2	1	-	-	Salah
3	1	1	95	Benar
4	1	1	163	Benar
5	2	2	85	Benar
6	1	1	77	Benar
7	2	2	113	Benar
8	2	2	96	Benar
9	1	1	79	Benar
10	1	-	-	Salah

*Tabel selengkapnya pada Lampiran 8

Keterangan:

- Jika id pengguna sama dengan hasil prediksi maka prediksi pengenalan wajah berhasil.
- Jika hasil prediksi kosong, maka saat pengujian tidak ada wajah yang terdeteksi.

Dari proses pengujian yang dilakukan pada Tabel 4.15 terlihat bahwa sistem dapat memprediksi dengan benar 17 dari 20 gambar dengan wajah datar.

3. Prosedur III

Dalam prosedur ini penulis akan menguji sistem untuk memprediksi 50 gambar acak. Hasil pengujian dapat dilihat pada Tabel 4.16.

Tabel 4. 16 Hasil Pengujian Gambar Acak

Pengujian ke	Id Pengguna	Hasil prediksi	Jarak	Prediksi
1	Tidak diketahui	1	89	Salah
2	Tidak diketahui	1	113	Salah
3	Tidak diketahui	-	-	benar
4	2	2	121	Benar
5	1	1	97	Benar
6	Tidak diketahui	1	78	Salah
7	1	1	76	Benar
8	2	2	117	Benar
9	1	-	-	Salah
10	Tidak diketahui	1	83	Salah
...
50	Tidak diketahui	1	132	Salah

*Tabel selengkapnya pada Lampiran 5

Keterangan:

- Jika id pengguna sama dengan hasil prediksi maka prediksi pengenalan wajah berhasil.

- Jika hasil prediksi kosong, maka saat pengujian tidak ada wajah yang terdeteksi.

Dari proses pengujian pada Tabel 4.16 terlihat bahwa banyak dari pengujian yang tidak diketahui id pengguna menghasilkan hasil prediksi. Hal ini dikarenakan sistem hanya memperhitungkan jarak dari wajah pengguna dengan model, jadi model tidak bisa membedakan mana wajah yang sudah dilatih dan mana wajah yang belum dilatih. Oleh karena itu penulis hanya memperhitungkan id yang diketahui saja, Tabel 4.17 menampilkan pengujian terhadap id yang diketahui.

Tabel 4. 17 Hasil Pengujian Dengan Id Diketahui

Pengujian ke	Id Pengguna	Hasil prediksi	Jarak	Prediksi
1	2	2	121	Benar
2	1	1	97	Benar
3	1	1	76	Benar
4	2	2	117	Benar
5	1	-	-	Salah
6	1	-	-	Salah
7	2	2	85	Benar
8	1	1	95	Benar
9	1	1	110	Benar
10	1	1	118	Benar
...
36	1	-	-	Salah

*Tabel selengkapnya pada Lampiran 6

Keterangan:

- Jika id pengguna sama dengan hasil prediksi maka prediksi pengenalan wajah berhasil.
- Jika hasil prediksi kosong, maka saat pengujian tidak ada wajah yang terdeteksi.

Dari Tabel 4.17 diketahui bahwa dari 36 pengujian, prediksi yang benar ada 28 dan prediksi yang salah ada 8. Jika dihitung menggunakan persentase keberhasilan sistem untuk memprediksi wajah seseorang yang sudah dilatih pada model yaitu sebesar 77,78% dan kesalahannya sebesar 22,22%.

Setelah pengujian dilakukan, langkah selanjutnya adalah menentukan *Confidence Interval* dari data prediksi yang benar (28 data) dengan cara sebagai berikut:

- (1) Hitung nilai rata-rata dari jarak (x) dengan menambahkan semua jarak hasil prediksi dan membagi dengan banyak sample (n), perhitungan ini dapat menggunakan Persamaan 1.20.

$$\bar{x} = \sum x / n \quad (1.20)$$

Keterangan:

x = variable untuk nilai rata-rata
 n = variable untuk banyak sampel

$$\begin{aligned}\bar{x} &= 3018/28 \\ \bar{x} &= 107,79\end{aligned}\tag{1.21}$$

(2) Hitung *Standart Deviasi* sesuai denngan persamaan 1.22 berikut:

$$\begin{aligned}\sigma &= \sqrt{\frac{\sum(x - \bar{x})^2}{n}} \\ \sigma &= \sqrt{\frac{625,45}{28}} \\ \sigma &= 25,01\end{aligned}\tag{1.22}$$

Keterangan:

σ = variabel untuk nilai *Standard Deviation*

x = variabel untuk nilai sampel

\bar{x} = variabel untuk nilai rata-rata (*mean*)

n = variabel untuk nilai banyak sampel

(3) Tentukan nilai kepercayaan. Dalam pengujian ini penulis menentukan tingkat kepercayaan 95%.

(4) Hitung margin kesalahan (margin of error). Dengan Persamaan 1.23 berikut:

$$Z_{\alpha/2} \times \frac{\sigma}{\sqrt{(n)}} \quad (1.23)$$

Keterangan:

$Z_{\alpha/2}$ = variabel untuk nilai Koefisien Kepercayaan

α = variabel untuk nilai Tingkat Kepercayaan

σ = variabel untuk nilai *Standard Deviation*

n = variabel untuk nilai banyak sampel

- Untuk menghitung margin kesalahan, Langkah pertama yang harus dilakukan adalah mencari koefisien kepercayaan, Pada pengujian ini tingkat kepercayaan adalah 95%. Konversikan persentase menjadi desimal, 0,95.
- Kemudian bagi 0,95 dengan 2. Dan mendapatkan hasil 0,475.
- Selanjutnya periksa Z Tabel untuk mencari nilai yang sesuai dengan 0,475. Z Tabel dapat dilihat pada Tabel 4.18 Berikut:

Tabel 4. 18 Z Index

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06
0.0	0.0000	0.0040	0.0080	0.0120	0.0160	0.0199	0.0239
0.1	0.0398	0.0438	0.0478	0.0517	0.0557	0.0596	0.0636
0.2	0.0793	0.0832	0.0871	0.0910	0.0948	0.0987	0.1026
0.3	0.1179	0.1217	0.1255	0.1293	0.1331	0.1368	0.1406

0.4	0.1554	0.1591	0.1628	0.1664	0.1700	0.1736	0.1772
0.5	0.1915	0.1950	0.1985	0.2019	0.2054	0.2088	0.2123
0.6	0.2257	0.2291	0.2324	0.2357	0.2389	0.2422	0.2454
0.7	0.2580	0.2611	0.2642	0.2673	0.2704	0.2734	0.2764
0.8	0.2881	0.2910	0.2939	0.2967	0.2995	0.3023	0.3051
0.9	0.3159	0.3186	0.3212	0.3238	0.3264	0.3289	0.3315
1.0	0.3413	0.3438	0.3461	0.3485	0.3508	0.3531	0.3554
1.1	0.3643	0.3665	0.3686	0.3708	0.3729	0.3749	0.3770
1.2	0.3849	0.3869	0.3888	0.3907	0.3925	0.3944	0.3962
1.3	0.4032	0.4049	0.4066	0.4082	0.4099	0.4115	0.4131
1.4	0.4192	0.4207	0.4222	0.4236	0.4251	0.4265	0.4279
1.5	0.4332	0.4345	0.4357	0.4370	0.4382	0.4394	0.4406
1.6	0.4452	0.4463	0.4474	0.4484	0.4495	0.4505	0.4515
1.7	0.4554	0.4564	0.4573	0.4582	0.4591	0.4599	0.4608
1.8	0.4641	0.4649	0.4656	0.4664	0.4671	0.4678	0.4686
1.9	0.4713	0.4719	0.4726	0.4732	0.4738	0.4744	0.4750

- Melihat nilai pada Tabel 4.18, titik terdekat untuk nilai 0,475 adalah 1,96. Pada persimpangan 1,9 dan 0,06. Masukkan titik 1,9 dan 0,06 ke Persamaan 1.23 sehingga menghasilkan Persamaan 1.24

$$z_{0,475} = 1,96 \quad (1.24)$$

- Selanjutnya hitung kesalahan standar. Masukkan nilai *Standart Deviasi* dan akar dari jumlah data ke

Persamaan 1.23 sehingga menghasilkan Persamaan 1.25.

$$\frac{\sigma}{\sqrt{n}} = \frac{25,01}{\sqrt{28}} \quad (1.25)$$

$$\frac{\sigma}{\sqrt{n}} = 4,73$$

- Selanjutnya masukan semua nilai dari Persamaan 1.24 dan Persamaan 1.25 ke Persamaan 1.23 sehingga menghasilkan Persamaan 1.26 berikut:

$$Z_{a/2} \times \frac{\sigma}{\sqrt{(n)}} = 1,96 \times 4,73 \quad (1.26)$$

$$Z_{a/2} \times \frac{\sigma}{\sqrt{(n)}} = 9,27$$

- (5) Mencari interval kepercayaan. Untuk menghitung interval kepercayaan dapat menggunakan Persamaan 1.27 berikut:

$$\bar{x} \pm Z_{a/2} \times \frac{\sigma}{\sqrt{(n)}}$$

$$107,79 + 9,27 = 116,56 \quad (1.27)$$

$$107,79 - 9,27 = 98,02$$

- (6) Dari Persamaan 1.27 didapatkan hasil bahwa nilai *lower* (jarak terdekat) adalah 98,02 dan nilai *upper* (jarak terjauh)

adalah 116,56. Kesimpulan dari hasil pengujian dengan id pengguna yang sudah diketahui adalah dari 28 gambar wajah maka *range* jarak hasil prediksi yaitu antara 98.02 sampai dengan 116,56 atau jika dirubah menjadi persen dengan batasan 140 akan menghasilkan Persamaan 1.28.

$$\text{similarity} = 100 - 100 \div 140 \times \text{jarak}$$

$$100 - 100 \div 140 \times 116,56 = 16,74 \quad (1.28)$$

$$100 - 100 \div 140 \times 98,02 = 29,99$$

(7) Kesimpulan pengujian jika dengan id pengguna yang sudah diketahui jika dirubah menjadi bentuk persen adalah dari 28 gambar wajah maka akurasi yang diperoleh antara 16,74% hingga 29,99%.

4.5.6 Evaluasi Hasil Pengujian

Berdasarkan dari pengujian tersebut, maka berikut ini evaluasi hasil pengujian yang telah dilakukan:

1. Berdasarkan pengujian sistem absensi tersebut dapat mengenali lebih baik terhadap gambar dengan wajah datar dikarenakan dataset yang terbatas. Namun jika menginginkan sistem absensi yang dapat mengenali wajah dengan ekspresi tertentu, model pengenalan wajahnya perlu dilatih untuk mengenali ekspresi tersebut.

2. Berdasarkan pengujian sistem absensi menggunakan gambar secara acak, sistem masih belum mampu membedakan mana pengguna yang sudah terdaftar di sistem/model dan mana pengguna yang belum terdaftar. Oleh karena itu sistem yang saat ini dibuat akan membandingkan setiap wajah yang terdeteksi dengan model yang ada, sehingga pengguna yang tidak terdaftar pun dapat terdeteksi sebagai pengguna yang terdaftar.
3. Berdasarkan hasil pengujian terhadap durasi video, diketahui bahwa durasi video yang kurang dari 10 detik akan menghasilkan jumlah *frame* kurang dari apa yang penulis harapkan. Dan jika video kurang dari 10 detik diproses maka akan menghasilkan *error* namun tidak berpengaruh terhadap sistem secara keseluruhan hanya berpengaruh terhadap jumlah *dataset* saja.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

5.2 Saran

DAFTAR PUSTAKA

- [1] “Indonesia: Coronavirus Pandemic Country Profile - Our World in Data.”
<https://ourworldindata.org/coronavirus/country/indonesia> (accessed Apr. 30, 2021).
- [2] WHO, “Transmisi SARS-CoV-2: implikasi terhadap kewaspadaan pencegahan infeksi,” pp. 1–10, 2020.
- [3] V. Bruce and A. Young, “Understanding face recognition,” *Br. J. Psychol.*, vol. 77, no. 3, pp. 305–327, 1986, doi: 10.1111/j.2044-8295.1986.tb02199.x.
- [4] Q. Mutiara and E. Prasetyo, “Perbandingan Metode Eigenface, Fisherface, dan LBPH pada Sistem Pengenalan Wajah,” *J. Ilm. Komputasi*, vol. 18, no. 4, 2019, doi: 10.32409/jikstik.18.4.2675.
- [5] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif, dan Tindakan*. 2012.
- [6] H. Hasanah, “TEKNIK-TEKNIK OBSERVASI (Sebuah Alternatif Metode Pengumpulan Data Kualitatif Ilmu-ilmu Sosial),” *At-Taqaddum*, vol. 8, no. 1, p. 21, 2017, doi: 10.21580/at.v8i1.1163.
- [7] A. Mirzaqon, “Studi Kepustakaan Mengenai Landasan Teori Dan Praktik Konseling Expressive Writing Library,” *J. BK UNESA*, no. 1, pp. 1–8, 2018.
- [8] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*. 2005.
- [9] H. Jaya *et al.*, *Kecerdasan Buatan*, vol. 53, no. 9. 2018.

- [10] A. Ahmed, F. Ali, and A. Ahmed, "LBPH based improved face recognition at low resolution LBPH Based Improved Face Recognition At Low Resolution," *2018 Int. Conf. Artif. Intell. Big Data*, no. October 2019, pp. 144–147, 2018, doi: 10.1109/ICAIBD.2018.8396183.
- [11] A. P. Singh, S. S. Manvi, P. Nimbal, and G. K. Shyam, "Face recognition system based on LBPH algorithm," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 5 Special Issue, pp. 26–30, 2019.
- [12] D. A. Wahyudi and I. H. Kartowisastro, "Menghitung Kecepatan Menggunakan Computer Vision," *J. Teknik Komput.*, vol. 19, no. 2, pp. 89–101, 2011.
- [13] E. Gorelik, "Cloud Computing Models," 2013.
- [14] R. S. Pressman, *Software Quality Engineering: A Practitioner's Approach*, vol. 9781118592. 2014.
- [15] D. S. M. Sc, M. Math, and D. Ph, "LOGIKA PEMROGRAMAN Disusun," no. September, 2002.
- [16] H. Bagir and B. E. Putro, "Analisis Perancangan Sistem Informasi Pergudangan di CV. Karya Nugraha," *J. Media Tek. dan Sist. Ind.*, vol. 2, no. 1, p. 30, 2018, doi: 10.35194/jmtsi.v2i1.274.
- [17] "Data Flow Diagram (DFD)s: An Agile Introduction."
<http://www.agilemodeling.com/artifacts/dataFlowDiagram.htm> (accessed Jun. 14, 2021).

- [18] D. Vassallo and J. Pschorr, “The Good Parts of AWS.”

LAMPIRAN-LAMPIRAN

Lampiran 1 Sintaks Membuat Dataset

```

import cv2
import os
import numpy as np
import time
import matplotlib.pyplot as plt
import sys
import argparse as arg
class Train():

    def __init__(self, face_cascade,config,username):
        self.username = username
        # self.current_dir = current_dir
        self._Face_Cascade =
cv2.CascadeClassifier(face_cascade)
        self.dataset_path("dataset/")
        self.recognizer =
cv2.face.LBPHFaceRecognizer_create(config[0], config[1],
config[2], config[3],config[4])
        def dataset_path(self, path):
            dir = os.path.dirname(path)
            if not os.path.exists(dir):
                os.makedirs(dir)
        def ReadName(self):
            NAME = []
            with open ("users.txt", "r") as f:
                for line in f:
                    NAME.append(line.split(",")[1].rstrip())
            return NAME
        def AddUser(self):
            # Name = input('\n[INFO] Masukkan nama user : ')
            Name = self.username
            info = open('users.txt', "a+")
            ID = len(open("users.txt").readlines( ))
            info.write(str(ID) + "," + Name + "\n")
            print("\n[INFO] Tambah user berhasil, ID:" + str(ID))
            info.close
            return ID
        def getImageWithLabels(self,path):
            imagePath = [os.path.join(path, f) for f in
os.listdir(path)]
            faceSamples = []
            ids = []
            for imagePath in imagePath:
                img = cv2.imread(imagePath, 0)
                img_numpy = np.array(img, 'uint8')
                id = int(os.path.split(imagePath)[-
1].split('.')[1])
                faceSamples.append(img_numpy)
                ids.append(id)
            return faceSamples, ids
        def train(self, path, file_name):

```

```

# os.chdir('tmp')
# real_path = '{0}/{1}'.format(self.current_dir,path)
print("\n[INFO] Training wajah sedang dimulai...")
time.sleep(1)
faces, ids = self.getImageWithLabels(path)
self.recognizer.update(faces,np.array(ids))
self.recognizer.write(file_name)
print("\n[INFO] Model sukses melatih user ID:
{0}".format(len(np.unique(ids))))
print("\n[INFO] Menutup program")
return len(np.unique(ids))

def create_Rect(self, Image, face, color):
    x,y,w,h = face
    cv2.line(Image, (x, y), (int(x + (w/5)),y), color, 2)
    cv2.line(Image, (int(x+((w/5)*4)), y), (x+w, y),
color, 2)
    cv2.line(Image, (x, y), (x,int(y+(h/5))), color, 2)
    cv2.line(Image, (x+w, y), (x+w, int(y+(h/5))), color,
2)
    cv2.line(Image, (x, int(y+(h/5*4))), (x, y+h), color,
2)
    cv2.line(Image, (x, int(y+h)), (x + int(w/5) ,y+h),
color, 2)
    cv2.line(Image, (x+int((w/5)*4), y+h), (x + w, y +
h), color, 2)
    cv2.line(Image, (x+w, int(y+(h/5*4))), (x+w, y+h),
color, 2)

    def createDataset(self,samples,cam,dataset_name):
        fig, axs = plt.subplots(10,5,figsize=(20,20),
facecolor='w', edgecolor='k')
        fig.subplots_adjust(hspace=.5, wspace=.001)
        self.dataset_path(dataset_name)
        count = 0
        face_id = self.AddUser()
        print('\n[INFO] Membuat dataset')
        while(True):
            success, image = cam.read()
            # convert image to grayscale
            gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
            faces = self._Face_Cascade.detectMultiScale(gray,
scaleFactor = 1.098, minNeighbors = 6, minSize = (50, 50))
            if(len(faces)> 1):
                print('\n[WARNING] Terdeteksi lebih dari 1
wajah')
                continue
            try:
                for _,face in enumerate(faces):
                    x, y, w, h = face
                    gray_chunk = gray[y-30: y + h + 30, x-30:
x + w + 30]

                    image_chunk = image[y: y + h, x: x + w]
                    self.create_Rect(image, face, [0,255,0])
                    # cv2.imshow("Video", image)

```

```

        #get center image
        # image_center =
tuple(np.array(gray_chunk.shape) / 2)
        # rot_mat =
cv2.getRotationMatrix2D(image_center, angle_degree, 1.0)
        # rotated_image =
cv2.warpAffine(gray_chunk, rot_mat, gray_chunk.shape,
flags=cv2.INTER_LINEAR)
        print("\n[INFO] Adding image number {} to
the dataset".format(count))
        # Save image
        cv2.imwrite("dataset/User." +
str(face_id) + '.' + str(count) + ".jpg " ,
            image)

    axs[int(count/5)][count%5].imshow(image,cmap='gray', vmin=0,
vmax=255)

    axs[int(count/5)][count%5].set_title("Person." + str(face_id)
+ '.' + str(count) + ".jpg ",
        fontdict={'fontsize':
15,'fontweight': 'medium'})
        axs[int(count/5)][count%5].axis('off')
        count += 1
    except Exception as e:
        print(e)
        print('[WARNING] Ada error')
        continue
    if cv2.waitKey(1) & 0xff == 27:
        break
    elif count >= samples:
        break
    print('\n[INFO] Dataset berhasil dibuat')
    # cam.release()
    # cv2.destroyAllWindows()
    # plt.show()
def Arg_Parse():
    Arg_Par = arg.ArgumentParser()
    Arg_Par.add_argument("-v", "--video",
                        help = "path of the video or if
not then webcam")
    Arg_Par.add_argument("-c", "--camera",
                        help = "Id of the camera")
    arg_list = vars(Arg_Par.parse_args())
    return arg_list
if __name__ == "__main__":
    if len(sys.argv) == 1:
        print("Masukan argumen yang valid!")
        sys.exit()
    Arg_list = Arg_Parse()
    face_cascade = 'lib/haarcascade_frontalface_default.xml'
    if not (os.path.isfile(face_cascade)):
        raise RuntimeError("%s: not found" % face_cascade)
    samples = 2

```

```

dataset_name = 'dataset/'
file_name = 'train.yml'
radius = 1
neighbour = 8
grid_x = 8
grid_y = 8
threshold = 140
var = list([radius,neighbour,grid_x,grid_y,threshold])
model = Train(face_cascade,var)
if Arg_list["video"] != None :
    video = cv2.VideoCapture(Arg_list["video"])
    #create a dataset for further model training
    print('{0} {1}
{2}'.format(samples,video,dataset_name))
    model.createDataset(samples,video,dataset_name)
    #Training the model
    model.train(dataset_name,file_name)
if Arg_list["camera"] != None :
    camera = cv2.VideoCapture(eval(Arg_list["camera"]))
    camera.set(3, 640)
    camera.set(4, 480)
    model.createDataset(samples,camera,dataset_name)
    #Training the model
    model.train(dataset_name,file_name)

```

Lampiran 2 Sintaks Membuat Model

```

import numpy as np
import cv2
import os
#Face detection is done
def faceDetection(test_img):
    gray_img=cv2.cvtColor(test_img,cv2.COLOR_BGR2GRAY)

    face_haar=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    faces=face_haar.detectMultiScale(gray_img,
    scaleFactor=1.2,
        minNeighbors=4,
        minSize=(30, 30))
    return faces,gray_img
# labeling dataset
def labels_for_training_data(directory):
    faces=[]
    faceID=[]
    file_count = 0
    for path,subdirnames,filenames in os.walk(directory):
        for filename in filenames:
            if filename.startswith("."):
                print("skipping system file")
                continue
            file_count += 1
            print(f'{filename} with id:
{filename.split(".")[1]}')
            id = filename.split(".")[1]
            image = cv2.imread(f'dataset/{filename}')
            if image is None:
                print(f'{filename} not exist!')
                continue
            faces_rect,gray_img=faceDetection(image)
            if len(faces_rect)!=1:
                print(f'{filename} -> no/multiple face
detected')

                continue
            (x,y,w,h)=faces_rect[0]
            tiny_face=gray_img[y:y+w,x:x+h]
            faces.append(tiny_face)

            faceID.append(int(id))
    print(f'{file_count=}')
    return faces,faceID
#Here training Classifier is called
def train_classifier(faces,faceID):
    face_recognizer=cv2.face.LBPHFaceRecognizer_create()
    face_recognizer.train(faces,np.array(faceID))

```

```
face_recognizer.write(f'{os.path.dirname(os.path.realpath(__f
ile__))}/model.yml')
    return face_recognizer
#Drawing a Rectangle on the Face Function
def draw_rect(test_img, face):
    (x,y,w,h)=face

    cv2.rectangle(test_img, (x,y), (x+w,y+h), (0,255,0), thickness=3)
#Putting text on images
def put_text(test_img, text, x, y):

    cv2.putText(test_img, text, (x,y), cv2.FONT_HERSHEY_DUPLEX, 3, (25
5,0,0), 6)
```

Lampiran 3 Sintaks Pengenalan Wajah

```

import cv2
import numpy as np
import csv
import os
import logging
logging.basicConfig(filename='predict.log',
format='% (asctime)s % (levelname)-8s
% (message)s', level=logging.DEBUG, datefmt='%Y-%m-%d
%H:%M:%S')
# global recognizer = cv2.face.LBPHFaceRecognizer_create()
# recognizer.read('train.yml')
# global faceCascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
# global font = cv2.FONT_HERSHEY_SIMPLEX
class Predict():
    def __init__(self, image):
        self.image = image
        # self.recognizer =
cv2.face.LBPHFaceRecognizer_create()
        # self.recognizer.read('train.yml')
        # self.faceCascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
        # self.font = cv2.FONT_HERSHEY_SIMPLEX
    def print(self):
        return os.path.join('dataset', self.image)
    def predict(self):
        logging.info('-----')
        logging.info('Predict Start')
        # print(self.image)
        # print(os.path.join('dataset', self.image))
        # print('{} '.format(self.image))
        # logging.info('{} '.format(os.path.join('dataset',
self.image)))
        # logging.info('{} '.format('dataset/{}
'.format(self.image)))

        # img = cv2.imread('image/59b53e2d-0c03-44ec-a4a9-
23d496caf6e0.jpg')
        img = cv2.imread(str(self.image))
        # img = cv2.imread(os.path.join('dataset',
self.image)+ " ")

        logging.info('Recognizer start')
        recognizer = cv2.face.LBPHFaceRecognizer_create()
        recognizer.read('model.yml')

```

```

        faceCascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
        font = cv2.FONT_HERSHEY_SIMPLEX
        ID = None
        CONF = None
        ### resize image
        imgs = cv2.resize(img, (0,0),None,0.25,0.25)
        # convert to gray scale
        gray = cv2.cvtColor(imgs, cv2.COLOR_BGR2GRAY)
        ### detect the face
        faces =
faceCascade.detectMultiScale(gray,scaleFactor=1.05,minNeighbo
rs=4,minSize=(30, 30))
        threshold = cv2.face_LBPHFaceRecognizer.getThreshold
        print(threshold)
        for (x,y,w,h) in faces:
            Id,conf = recognizer.predict(gray[y:y+h,x:x+w])
            # cv2.rectangle(imgS, (x,y), (x+w, y+h),
(255,0,0), 2)
            # cv2.rectangle(img, (x, y), (x + w, y + h), (0,
260, 0), 2)
            # cv2.putText(imgS, str(Id), (x,y-40),font, 2,
(255,255,255), 3)
            print('ID {0}, confidence {1}'.format(Id, conf))
            ID = Id
            CONF = conf
            logging.info('Predict result: {} with {}
distance'.format(ID, CONF))
            print('ID {0}, confidence {1}'.format(ID, CONF))
            return ID,CONF

```


Lampiran 4 Sintaks Server Pengenalan Wajah

```

from flask import Flask, request, make_response
from flask_cors import CORS
from file_downloader import downloader, getData, updateData,
updateDataNew, putData
import cv2
from train import Train
from predict import Predict
import uuid
import os
import datetime
import json
import logging
import math
import recognize as re
logging.basicConfig(filename='skripsi.log',
format='%(asctime)s %(levelname)-8s %(message)s',
level=logging.INFO, datefmt='%Y-%m-%d %H:%M:%S')
app = Flask(__name__)
cors = CORS(app, resources={r"/*": {"origins": "*"}})
@app.route('/')
def test():
    current_time = datetime.datetime.now()
    response = make_response('<h1>Success at {} (server
time)</h1>'.format(current_time))
    response.headers['Content-Security-Policy'] = 'upgrade-
insecure-requests'
    return response
@app.route('/recognize/predict', methods=['POST'])
def predict():
    logging.info('halo')
    request_data = request.get_json()

    if request_data:
        image = request_data['image']
        expect_username = request_data['username']
        location = request_data['location']
        username = None
        conf = None

        debug_id = None
        debug_sim = None
        debug_dis = None
        debug_username = None
        result_id = None
        result_sim = None
        result_dis = None
        result_username = None

```

```

debug_user_data = None
result_user_data = None
minSim = 40 # 75%

image_path = 'image/{}.jpg'.format(uuid.uuid4())
# download video from s3
downloader_client =
downloader(key=image,bucket='dannynurdin',
destination=image_path)
downloader_client.download()

model = Predict(image_path)

res = model.print()
id,c = model.predict()
logging.info('DATA => id from model == ', id)
print('DATA => id from model == ', id)
logging.info('success {} - {}'.format(id, c))
if c:
    disMax = 140.0
    simMax = 100
    similarity = simMax - simMax/disMax*c
    conf = similarity

with open('users.txt') as f:
    datafile = f.readlines()
    for line in datafile:
        Id = line.split(',')[0]
        print('DATA => Id from model == ', Id)
        logging.info('DATA => Id from model == ', Id)
        if Id == id:
            dump= line.split(',')[1]
            debug_username = dump.split('\n')[0]
            debug_id = id
            debug_sim = conf
            debug_dis = c

            dynamodb_client = getData(id =
debug_username)

            debug_response = dynamodb_client.get()
            debug_user_data = debug_response['Item']
or None

            # add record to database test-v2
            record = putData(id =
debug_username,face_id=debug_id,conf=debug_sim,location='test
location')

            record.put()

            if conf and conf >= minSim:
                result_username = dump.split('\n')[0]
                result_sim = conf
                result_id = id
                result_dis = c

```

```

        dynamodb_client = getData(id =
result_username)
        result_response =
dynamodb_client.get()
        result_user_data =
result_response['Item'] or None

    response = make_response({
        "debug": {
            "id": debug_id,
            "dis": debug_dis,
            'sim': debug_sim,
            "username": debug_username,
            "user_data": debug_user_data,
            'expect_user': expect_username
        },
        'result': {
            "username": result_username,
            "dis": result_dis,
            "sim": result_sim,
            "id": result_id,
            "user_data": result_user_data
        },
    })
    response.headers['Content-Security-Policy'] =
'upgrade-insecure-requests'
    return response

    return 'kosong'
@app.route('/recognize/train', methods=['POST'])
def train():
    # grab data from post request
    request_data = request.get_json()
    key = None
    username = None
    # config
    face_cascade = 'lib/haarcascade_frontalface_default.xml'
    dataset_name = 'dataset/'
    samples = 15
    file_name = 'model.yml'
    video_path = 'video/{}.mp4'.format(uuid.uuid4())

    # LBPH variables
    radius = 1
    neighbour = 8
    grid_x = 8
    grid_y = 8
    treshold = 140
    var = list([radius,neighbour,grid_x,grid_y,treshold])
    if request_data:
        if 'key' in request_data:

```

```

        key = request_data['key']

        if 'username' in request_data:
            username = request_data['username']
        if 'from' in request_data:
            status = request_data['from']
        if status == 'development':
            bucket_name = 'skripsi200053-dev'
        else:
            bucket_name = 'skripsi132739-prod'

        # download video from s3
        downloader_client =
downloader(key=key,bucket=bucket_name,
destination=video_path)
        downloader_client.download()
        # get data user
        dynamodb_client = updateData(id = username, key =
key)
        ress = dynamodb_client.update()
        print(ress)
        # start recognize using opencv
        model = Train(face_cascade,var,username) # create
instance train
        video = cv2.VideoCapture(video_path) # load video
        model.createDataset(samples,video,dataset_name) #
create dataset
        # id = model.train(dataset_name,file_name)
        faces,faceID = re.labels_for_training_data('dataset')
        face_recognizer=re.train_classifier(faces,faceID)

face_recognizer.save(f'{os.path.dirname(os.path.realpath(__fi
le__))}/model.yml')
        print(f'faces: {len(faces)} , id: {len(faceID)}')
        response = {
            "success": True,
            "face_id": id,
            "username": username
        }
        response = make_response({
            "success": True,
            "face_id": id,
            "username": username
        })
        response.headers['Content-Security-Policy'] =
'upgrade-insecure-requests'
        return response
    else:
        return 'Key required!'
@app.route('/update', methods=['POST'])
def update():
    request_data = request.get_json()
    AttrName = {}
    AttrValue = {}

```

```

Expression = []
if request_data:
    if 'id' in request_data:
        id = request_data['id']

    if 'name' in request_data:
        # key = request_data['name']
        AttrName['#N'] = 'name'
        AttrValue[':N'] = { 'S' : request_data['name']}
        Expression.append('#N = :N')
    if 'phone' in request_data:
        # status = request_data['phone']
        AttrName['#P'] = 'phone_number'
        AttrValue[':P'] = { 'S' : request_data['phone']}
        Expression.append('#P = :P')
    if 'address' in request_data:
        # status = request_data['address']
        AttrName['#A'] = 'address'
        AttrValue[':A'] = { 'S' :
request_data['address']}
        Expression.append('#A = :A')

    expression = 'Set ' + ','.join([str(elem) for elem in
Expression])
    dynamodb_client = updateDataNew(id)
    res = dynamodb_client.update(AttrName,AttrValue,
expression)
    response = make_response({
        "res": request_data,
        'AttrName': json.dumps(AttrName),
        'AttrValue': json.dumps(AttrValue),
        'Expression': expression

    })
    response.headers['Content-Security-Policy'] = 'upgrade-
insecure-requests'
    return response
if __name__ == '__main__':
    # run app in debug mode on port 5000
    app.run(host='0.0.0.0',port=80)

```

Lampiran 5 Hasil Pengujian Gambar Acak

Pengujian ke	Id Pengguna	Hasil prediksi	Jarak	Prediksi
1	Tidak diketahui	1	89	Salah
2	Tidak diketahui	1	113	Salah
3	Tidak diketahui	-	-	benar
4	2	2	121	Benar
5	1	1	97	Benar
6	Tidak diketahui	1	78	Salah
7	1	1	76	Benar
8	2	2	117	Benar
9	1	-	-	Salah
10	Tidak diketahui	1	83	Salah
11	1	-	-	Salah
12	2	2	85	Benar
13	1	1	95	Benar
14	1	1	110	Benar
15	1	1	118	Benar
16	1	1	111	Benar
17	2	2	76	Benar
18	Tidak diketahui	1	89	salah
19	1	1	167	Benar
20	2	2	125	Benar
21	2	2	117	Benar
22	1	1	109	Benar
23	1	0	163	Benar
24	1	-	-	Salah
25	Tidak diketahui	1	84	Salah
26	Tidak diketahui	1	117	Salah
27	Tidak diketahui	2	108	Salah
28	2	2	159	Benar
29	2	2	85	Benar
30	1	1	122	Benar
31	1	-	-	Salah
32	1	1	116	Benar

33	1	-	-	Salah
34	1	1	77	Benar
35	2	2	113	Benar
36	2	2	96	Benar
37	1	1	90	Benar
38	1	1	79	Benar
39	2	2	122	Benar
40	Tidak diketahui	1	143	Salah
41	1	-	-	Salah
42	1	1	82	Benar
43	1	1	79	Benar
44	Tidak diketahui	1	66	Salah
45	1	-	-	Salah
46	1	1	111	Benar
47	1	-	-	Salah
48	Tidak diketahui	1	65	Salah
49	Tidak diketahui	1	89	Salah
50	Tidak diketahui	1	132	Salah

Lampiran 6 Pengujian Dengan Id Diketahui

Pengujian ke	Id Pengguna	Hasil prediksi	Jarak	Prediksi
1	2	2	121	Benar
2	1	1	97	Benar
3	1	1	76	Benar
4	2	2	117	Benar
5	1	-	-	Salah
6	1	-	-	Salah
7	2	2	85	Benar
8	1	1	95	Benar
9	1	1	110	Benar
10	1	1	118	Benar
11	1	1	111	Benar
12	2	2	76	Benar
13	1	1	167	Benar
14	2	2	125	Benar
15	2	2	117	Benar
16	1	1	109	Benar
17	1	0	163	Benar
18	1	-	-	Salah
19	2	2	159	Benar
20	2	2	85	Benar
21	1	1	122	Benar
22	1	-	-	Salah
23	1	1	116	Benar
24	1	-	-	Salah
25	1	1	77	Benar
26	2	2	113	Benar
27	2	2	96	Benar
28	1	1	90	Benar
29	1	1	79	Benar
30	2	2	122	Benar
31	1	-	-	Salah
32	1	1	82	Benar

33	1	1	79	Benar
34	1	-	-	Salah
35	1	1	111	Benar
36	1	-	-	Salah

Lampiran 7 Hasil pengujian Muka Senyum

No	Id Pengguna	Hasil Prediksi	Jarak	Prediksi
1	1	1	110	Benar
2	1	1	111	Benar
3	1	1	109	Benar
4	1	1	122	Benar
5	1	1	116	Benar
6	1	-	-	Salah
7	1	1	90	Benar
8	1	1	79	Benar
9	1	-	-	Salah
10	1	-	-	Salah
11	1	1	112	Benar
12	1	1	103	Benar
13	1	1	100	Benar
14	1	1	118	Benar
15	1	1	114	Benar
16	1	-	-	Salah
17	1	1	81	Benar
18	1	1	68	Benar
19	1	-	-	Salah
20	1	-	-	Salah

Lampiran 8 Hasil pengujian Muka Datar

No	Id Pengguna	Hasil Prediksi	Jarak	Prediksi
1	1	1	76	Benar
2	1	-	-	Salah
3	1	1	95	Benar
4	1	1	163	Benar
5	2	2	85	Benar
6	1	1	77	Benar
7	2	2	113	Benar
8	2	2	96	Benar
9	1	1	79	Benar
10	1	-	-	Salah
11	1	1	88	Benar
12	1	1	152	Benar
13	2	2	81	Benar
14	1	1	66	Benar
15	2	2	102	Benar
16	2	2	82	Benar
17	1	1	69	Benar
18	2	2	111	Benar
19	2	2	92	Benar
20	1	-	-	Salah