



**SISTEM PRESENSI MAHASISWA BERDASARKAN
PENGENALAN WAJAH MENGGUNAKAN METODE LBP DAN
K-NEAREST NEIGHBOR BERBASIS MINI PC**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:

Meidiana Adinda Prasanty

NIM: 165150301111037



**PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2020

PENGESAHAN

SISTEM PRESENSI MAHASISWA BERDASARKAN PENGENALAN WAJAH
MENGUNAKAN METODE LBP DAN *K-NEAREST NEIGHBOR* BERBASIS MINI PC

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :
Meidiana Adinda Prasanty
NIM: 165150301111037

Skripsi ini telah diuji dan dinyatakan lulus pada
11 Juni 2020
Telah diperiksa dan disetujui oleh:

Menyetujui,

Dosen Pembimbing I



Fitri Utamingrum, Dr.Eng., S.T., M.T
NIP. 19820710 200812 2 001

Mengetahui,

Ketua Jurusan Teknik Informatika



Dr. Astoto Kurniawan, S.T., M.T., Ph.D.
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, didalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 11 Juni 2020



Meidiana Adinda Prasanty
NIM. 165150301111037



ABSTRAK

Meidiana Adinda Prasanty, Sistem Presensi Mahasiswa Berdasarkan Pengenalan Wajah Menggunakan Metode LBP dan *K-Nearest Neighbor* Berbasis *Mini PC*.

Pembimbing: Fitri Utamingrum, Dr. Eng., S.T, M.T

Di era modern sekarang ini, dimana teknologi sudah maju dengan pesat ternyata beberapa institusi di Indonesia masih mengandalkan sistem presensi yang lama yaitu dengan cara manual, seperti menggunakan kertas dan membuat paraf. Penelitian ini memungkinkan untuk mengurangi tindak kecurangan dengan memanfaatkan citra digital yaitu pengenalan wajah guna melakukan presensi agar menjadi lebih praktis, efisien, cepat dan tentunya aman serta tidak terjadi hal-hal yang merugikan setiap institusi. Pada penelitian ini, dikembangkan sebuah sistem presensi mahasiswa berdasarkan pengenalan wajah dengan metode *Local Binary Pattern* dan *K-Nearest Neighbor*. Sistem ini menggunakan *webcam* Logitech C270 dan *mouse* Alcatroz Stealth 5 sebagai penghasil input, *Mini PC* Intel NUC5i7RYH sebagai pemroses utama, dan monitor Waveshare 7-inch sebagai output. Citra yang ditangkap oleh *webcam* berupa citra dari para mahasiswa yang sedang duduk di kelas. Dari citra tersebut akan diproses oleh *Mini PC* untuk dilakukan pendeteksian dan pengenalan wajah dari tiap mahasiswa sehingga hasilnya berupa nama mahasiswa yang ditampilkan pada aplikasi sistem presensi. Selanjutnya aplikasi sistem presensi akan ditampilkan pada monitor. Nama-nama mahasiswa yang sudah dikenali oleh sistem dapat masuk pada daftar presensi apabila pengguna (dosen atau peneliti) menekan tombol konfirmasi presensi pada aplikasi menggunakan *mouse*. Rata-rata akurasi sistem dalam pendeteksian wajah menggunakan *Haar Cascade Classifier* dari seluruh percobaan yaitu 88.88% sedangkan untuk rata-rata akurasi sistem dalam pengenalan wajah dari seluruh percobaan menggunakan *Local Binary Pattern* dan *K-Nearest Neighbor* untuk nilai $k = 3$ yaitu 78.125%, untuk nilai $k = 5$ yaitu 74.375%, dan untuk nilai $k = 7$ yaitu 68.125%. Rata-rata akurasi tertinggi dalam melakukan pengenalan wajah dari seluruh percobaan dapat dicapai menggunakan nilai $k = 3$ yaitu sebesar 78.125%. Rata-rata waktu komputasi deteksi wajah dari seluruh percobaan deteksi wajah adalah 26.2 ms sedangkan untuk rata-rata waktu komputasi pengenalan wajah dari seluruh percobaan pengenalan wajah adalah 371.675 ms. Setiap satu kenaikan jumlah wajah dapat memperlambat waktu komputasi deteksi wajah 1 milisekon hingga 4 milisekon sedangkan untuk pengenalan wajah dapat melambatkan 28 milisekon hingga 160 milisekon.

Kata kunci: Sistem Presensi, *Haar Cascade Classifier*, *Local Binary Pattern*, *K-Nearest Neighbor*, *Webcam*, *Mini PC*

ABSTRACT

Meidiana Adinda Prasanty, Student Presence System Based on Face Recognition Using LBP and K-Nearest Neighbor Methods on Mini PC.

Supervisors: Fitri Utamingrum, Dr. Eng., S.T, M.T

Every institution such as the education system in Indonesia, even for offices certainly requires a system that can record the entire community of it's members. But in today's modern era, where technology has advanced rapidly it turns out that in some institutions in Indonesia still rely on the old presence system that is manually, such as using paper and initialing. This research makes it possible to reduce fraud by utilizing digital imagery that is face recognition in order to make a presence so that it becomes more practical, efficient, fast and certainly safe and does not happen to the detriment of any institution. In this study, a student presence system was developed based on face recognition using Local Binary Pattern and K-Nearest Neighbor method. By using the Logitech C270 webcam and the Alcatroz Stealth 5 mouse as an input producer, the Intel NUC5i7RYH Mini PC as the main processor, and 7-inch Waveshare monitor as output. The image captured by the webcam is the image of students sitting in class. That image will be processed by the Mini PC for face detection and recognition of each student so that the result is the name of the student displayed on the presence system application. Then the presence system application will be displayed on the monitor. The names of students who have been recognized by the system can enter the attendance list if the user (lecturer or researcher) presses the attendance confirmation button on the application using the mouse. The average accuracy of the system in face detection of all experiments using Local Binary Pattern and Haar Cascade Classifier is 88.88%. The average accuracy of the system in face recognition of all experiments using K-Nearest Neighbor for the value of $k = 3$ is 78.125%, for the value of $k = 5$ is 74.375%, and for the value of $k = 7$ is 68.125%. The highest average accuracy in facial recognition of all experiments is to use the value $k = 3$ that is equal to 78.125%. The average face detection computation of all face detection experiments is 26.2 ms while the average facial recognition computation time of all face recognition experiments is 371.675 ms. Every single increase in the number of faces can slower the face detection computation time by 1 ms to 4 ms while for facial recognition it can slowed by 28 ms to 160 ms.

Keyword: Presence System, Haar Cascade Classifier, Local Binary Pattern, K-Nearest Neighbor, Webcam, Mini PC



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Manfaat	2
1.5 Batasan Masalah	3
1.6 Sistematika Penulisan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori	7
2.2.1 Pemrosesan Citra	7
2.2.2 Operasi Konversi Citra RGB ke <i>Grayscale</i>	7
2.2.3 Normalisasi <i>Min Max</i>	8
2.2.4 Basis Data	8
2.2.5 Sistem Manajemen Basis Data	8
2.2.6 Metode <i>Haar Cascade Classifier</i>	8
2.2.7 Metode <i>Local Binary Pattern (LBP)</i>	9
2.2.8 Metode <i>K-Nearest Neighbor</i>	9
2.2.9 <i>Structured Query Language (SQL)</i>	11
BAB 3 METODOLOGI PENELITIAN	12
3.1 Tipe Penelitian	12
3.2 Strategi dan Rancangan Penelitian	12



3.2.1 Metode Secara Umum	12
3.2.2 Subjek Penelitian	14
3.2.3 Lokasi Penelitian	14
3.2.4 Teknik Pengumpulan Data	14
3.2.5 Teknik Analisis Data dan Pembahasan	14
3.2.6 Peralatan Pendukung yang digunakan	14
3.2.6.1 Perangkat Keras	14
3.2.6.2 Perangkat Lunak	14
BAB 4 REKAYASA KEBUTUHAN	16
4.1 Gambaran Umum Sistem	16
4.2 Analisis Kebutuhan Sistem	16
4.2.1 Kebutuhan Sistem	17
4.2.1.1 Kebutuhan Perangkat Keras	17
4.2.1.2 Kebutuhan Perangkat Lunak	18
4.2.2 Kebutuhan Fungsional	19
4.2.3 Kebutuhan Non Fungsional	20
4.2.3.1 Karakteristik Pengguna	20
4.2.3.2 Lingkungan Operasi	20
4.2.3.3 Batasan Desain Sistem	20
BAB 5 PERANCANGAN DAN IMPLEMENTASI	21
5.1 Perancangan Sistem	21
5.1.1 Perancangan Prototipe Sistem	21
5.1.2 Perancangan Perangkat Keras	22
5.1.3 Perancangan Perangkat Lunak	23
5.1.3.1 Perancangan Program Utama	23
5.1.3.2 Perancangan <i>Database</i> Presensi Mahasiswa	25
5.1.3.3 Perancangan Program Pelatihan Wajah Mahasiswa	25
5.1.3.4 Perancangan Tampilan Aplikasi Sistem Presensi	26
5.1.3.5 Perancangan Deteksi Wajah Mahasiswa Menggunakan Metode <i>Haar Cascade Classifier</i>	28
5.1.3.6 Perancangan Ekstraksi Fitur Wajah Mahasiswa Menggunakan Metode <i>Local Binary Pattern</i>	28



5.1.3.7 Perancangan Normalisasi <i>Min Max</i> pada Fitur <i>Local Binary Pattern</i> Wajah Mahasiswa	30
5.1.3.8 Perancangan Klasifikasi Wajah Mahasiswa Menggunakan Metode <i>K-Nearest Neighbor</i>	30
5.1.3.9 Perancangan Konfirmasi Hasil Klasifikasi Wajah Mahasiswa ke Daftar Presensi	31
5.2 Implementasi Sistem	32
5.2.1 Implementasi Prototipe Sistem	32
5.2.2 Implementasi Perangkat Keras	33
5.2.3 Implementasi Perangkat Lunak	34
5.2.3.1 Implementasi Program Utama	34
5.2.3.2 Implementasi <i>Database</i> Presensi Mahasiswa	38
5.2.3.3 Implementasi Program Pelatihan Wajah Mahasiswa	39
5.2.3.4 Implementasi Tampilan Aplikasi Sistem Presensi	44
5.2.3.5 Implementasi Deteksi Wajah Mahasiswa Menggunakan Metode <i>Haar Cascade Classifier</i>	44
5.2.3.6 Implementasi Ekstraksi Fitur Wajah Mahasiswa Menggunakan Metode <i>Local Binary Pattern</i>	45
5.2.3.7 Implementasi Normalisasi <i>Min Max</i> pada Fitur <i>Local Binary Pattern</i> Wajah Mahasiswa	48
5.2.3.8 Implementasi Klasifikasi Wajah Mahasiswa Menggunakan Metode <i>K-Nearest Neighbor</i>	48
5.2.3.9 Implementasi Konfirmasi Hasil Klasifikasi Wajah Mahasiswa ke Daftar Presensi	51
BAB 6 PENGUJIAN DAN ANALISIS	53
6.1 Pengujian dan Analisis Rata-rata Akurasi Deteksi Wajah menggunakan <i>Haar Cascade Classifier</i>	53
6.1.1 Tujuan Pengujian	53
6.1.2 Prosedur Pengujian	53
6.1.3 Hasil Pengujian	54
6.1.4 Analisis Pengujian	56
6.2 Pengujian dan Analisis Rata-rata Akurasi Pengenalan Wajah menggunakan <i>Local Binary Pattern</i> dan <i>K-Nearest Neighbor</i>	57
6.2.1 Tujuan Pengujian	57
6.2.2 Prosedur Pengujian	57



6.2.3 Hasil Pengujian.....	57
6.2.4 Analisis Pengujian.....	64
6.3 Pengujian dan Analisis Rata-rata Waktu Komputasi Deteksi Wajah menggunakan <i>Haar Cascade Classifier</i>	65
6.3.1 Tujuan Pengujian.....	65
6.3.2 Prosedur Pengujian	65
6.3.3 Hasil Pengujian.....	65
6.3.4 Analisis Pengujian.....	68
6.4 Pengujian dan Analisis Rata-rata Waktu Komputasi Pengenalan Wajah menggunakan <i>Local Binary Pattern</i> dan <i>K-Nearest Neighbor</i>	69
6.4.1 Tujuan Pengujian.....	69
6.4.2 Prosedur Pengujian	69
6.4.3 Hasil Pengujian.....	69
6.4.4 Analisis Pengujian.....	72
BAB 7 PENUTUP	73
7.1 Kesimpulan.....	73
7.2 Saran	73
DAFTAR REFERENSI.....	74
LAMPIRAN	76
A.1 Percobaan Aplikasi Sistem Presensi Dari Tengah.....	76
A.2 Kode Program Utama Keseluruhan.....	86



DAFTAR TABEL

Tabel 2.1 Jurnal Terkait	5
Tabel 5.1 Kode Program Utama	34
Tabel 5.2 Penjelasan Kode Program Utama	36
Tabel 5.3 Kode Program Pelatihan Wajah Mahasiswa	40
Tabel 5.4 Penjelasan Kode Program Pelatihan Wajah Mahasiswa	42
Tabel 5.5 Kode Program Deteksi Wajah Mahasiswa Menggunakan Metode <i>Haar Cascade Classifier</i>	45
Tabel 5.6 Penjelasan Kode Program Deteksi Wajah Mahasiswa Menggunakan Metode <i>Haar Cascade Classifier</i>	45
Tabel 5.7 Kode Program Ekstraksi Fitur Wajah Mahasiswa Menggunakan <i>Local Binary Pattern</i>	45
Tabel 5.8 Penjelasan Kode Program Ekstraksi Fitur Wajah Mahasiswa Menggunakan <i>Local Binary Pattern</i>	47
Tabel 5.9 Kode Program Normalisasi <i>Min Max</i> pada Fitur <i>Local Binary Pattern</i> Wajah Mahasiswa	48
Tabel 5.10 Penjelasan Kode Program Normalisasi <i>Min Max</i> pada Fitur <i>Local Binary Pattern</i> Wajah Mahasiswa	48
Tabel 5.11 Kode Program Klasifikasi Wajah Mahasiswa Menggunakan Metode <i>K-Nearest Neighbor</i>	49
Tabel 5.12 Penjelasan Kode Program Klasifikasi Wajah Mahasiswa Menggunakan Metode <i>K-Nearest Neighbor</i>	50
Tabel 5.13 Kode Program Konfirmasi Pengenalan Wajah Mahasiswa ke Daftar Presensi	51
Tabel 5.14 Penjelasan Kode Program Konfirmasi Pengenalan Wajah Mahasiswa ke Daftar Presensi	52
Tabel 6.1 Hasil Pendeteksian Wajah	55
Tabel 6.2 Hasil Pengenalan Wajah dengan nilai $k = 3$	59
Tabel 6.3 Hasil Pengenalan Wajah dengan nilai $k = 5$	61
Tabel 6.4 Hasil Pengenalan Wajah dengan nilai $k = 7$	63
Tabel 6.5 Hasil Waktu Komputasi Deteksi Wajah	67
Tabel 6.6 Hasil Waktu Komputasi Pengenalan Wajah	71
Tabel A.1.1 Tabel Percobaan Pertama Aplikasi Sistem Presensi Dari Tengah	76
Tabel A.2.1 Kode Program Utama	86



DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Matriks Citra Digital.....	7
Gambar 2.2 Proses Kalkulasi Piksel LBP.....	9
Gambar 2.3 Ilustrasi <i>K-Nearest Neighbor</i>	10
Gambar 2.4 Logo SQL.....	11
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	13
Gambar 4.1 Gambaran Umum Sistem.....	16
Gambar 4.2 <i>Webcam</i> Logitech C270 HD.....	17
Gambar 4.3 Mini PC NUC5i7RYH.....	17
Gambar 4.4 <i>Mouse</i> Alcatroz Stealth 5.....	18
Gambar 4.5 Monitor Waveshare 7-inch.....	18
Gambar 5.1 Diagram Blok Sistem.....	21
Gambar 5.2 Desain Prototipe Sistem.....	22
Gambar 5.3 Skema Perancangan Perangkat Keras.....	22
Gambar 5.4 Diagram Alir Perancangan Program Utama.....	24
Gambar 5.5 Diagram Alir Perancangan Program Pelatihan Wajah Mahasiswa.....	26
Gambar 5.6 Rancangan Desain Tampilan Aplikasi Sistem Presensi.....	27
Gambar 5.7 Diagram Alir Deteksi Wajah Mahasiswa menggunakan Metode <i>Haar Cascade Classifier</i>	28
Gambar 5.8 Diagram Alir Ekstraksi Fitur Wajah Mahasiswa menggunakan Metode <i>Local Binary Pattern</i>	29
Gambar 5.9 Diagram Alir Normalisasi <i>Min Max</i> pada Fitur <i>Local Binary Pattern</i> Wajah Mahasiswa.....	30
Gambar 5.10 Diagram Alir Klasifikasi Wajah Mahasiswa menggunakan Metode <i>K-Nearest Neighbor</i>	31
Gambar 5.11 Diagram Alir Konfirmasi Pengenalan Wajah Mahasiswa ke Daftar Presensi.....	32
Gambar 5.12 Implementasi Prototipe Sistem.....	33
Gambar 5.13 Implementasi Perangkat Keras.....	33
Gambar 5.14 Struktur <i>Database</i> Sistem Presensi Mahasiswa.....	39
Gambar 5.15 Tampilan dari Seluruh <i>Record</i> dalam <i>table</i> 'mahasiswa'.....	39
Gambar 5.16 Implementasi Desain Tampilan Aplikasi Sistem Presensi.....	44



Gambar 6.1 Tampilan Aplikasi Sistem Presensi Saat Pengujian Pendeteksian Tiga Wajah Mahasiswa	54
Gambar 6.2 Tampilan Aplikasi Sistem Presensi Saat Pengujian Pendeteksian Delapan Wajah Mahasiswa	54
Gambar 6.3 Tampilan Aplikasi Sistem Presensi Saat Pengujian Pengenalan Tiga Wajah Mahasiswa	58
Gambar 6.4 Tampilan Aplikasi Sistem Presensi Saat Pengujian Pengenalan Delapan Wajah Mahasiswa	58
Gambar 6.5 Tampilan Aplikasi Sistem Presensi Saat Pengujian Waktu Komputasi Deteksi Tiga Wajah Mahasiswa	66
Gambar 6.6 Tampilan Aplikasi Sistem Presensi Saat Pengujian Waktu Komputasi Deteksi Delapan Wajah Mahasiswa	66
Gambar 6.7 Tampilan Aplikasi Sistem Presensi Saat Pengujian Waktu Komputasi Pengenalan Tiga Wajah Mahasiswa	70
Gambar 6.8 Tampilan Aplikasi Sistem Presensi Saat Pengujian Waktu Komputasi Pengenalan Delapan Wajah Mahasiswa	70



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Setiap institusi seperti pendidikan tingkat SD, SMP, SMA, perkuliahan, bahkan untuk perkantoran pun pasti memerlukan sebuah sistem yang dapat mendaftarkan seluruh civitas para anggotanya. Sistem yang dimaksud tersebut adalah presensi dimana menjadi faktor yang penting dalam aspek penilaian. Pada umumnya sistem presensi yang terdapat pada suatu institusi terutama universitas dilakukan secara manual, yaitu dengan menulis nama atau membuat paraf. Dan kendala lainnya jika sistem presensi masih manual adalah form presensi dapat hilang, rusak atau terbawa dan mahasiswa juga dapat memalsukan paraf atau dikenal dengan istilah titip presensi.

Pada era modern sekarang ini, tidak menutup kemungkinan untuk dapat mengembangkan sebuah sistem presensi dengan memanfaatkan teknologi pada berbagai macam metode agar menjadi lebih praktis, efisien, cepat dan tentunya aman serta tidak terjadi lagi hal-hal yang merugikan setiap institusi. Terutama jika pengambilan data presensi menggunakan pengenalan wajah manusia dimana setiap orang telah memiliki data presensi mereka masing-masing sesuai dengan data yang telah di set sebelumnya pada sistem. Pengenalan wajah yang dimaksud adalah membandingkan suatu citra wajah yang diatur sebagai masukan pada *database* yang berisi data-data presensi kemudian proses selanjutnya menemukan data dan mencocokkan wajah yang paling sesuai dengan masukan citra tersebut (Suprianto, et al., 2013).

Pada penelitian sebelumnya yang dilakukan oleh Puspaningrum (2017), metode Haar Cascade Classifier digunakan sebagai metode deteksi wajah karena memiliki akurasi sebesar 81.6% sedangkan pada penelitian yang dilakukan oleh Ajib Susanto (2018) yang berjudul "*A High Performance of Local Binary Pattern on Classify Javanese Character Classification*" terdapat metode *Local Binary Pattern* (LBP) sebagai metode ekstraksi fitur dari karakter aksara Jawa dan metode *K-Nearest Neighbor* (KNN) sebagai metode klasifikasi untuk mengenali karakter aksara Jawa tersebut. Hasil penelitian tersebut memiliki akurasi 82.5% dengan nilai $k = 3$ menggunakan ukuran citra 64×64 piksel. Menurut Huang et al. (2011) metode *Local Binary Pattern* memiliki keunggulan yaitu komputasinya yang sederhana dan memiliki sifat invarian terhadap perubahan *grayscale*, artinya apabila pada suatu citra digital mengalami perubahan intensitas cahaya maka fitur yang dihasilkan masih tetap sama. Sedangkan metode *K-Nearest Neighbor* memiliki keunggulan yaitu mudah diimplementasikan dan komputasinya sederhana serta tidak membutuhkan waktu untuk pelatihan.

Normalisasi *Min Max* juga diterapkan pada fitur hasil ekstraksi metode LBP karena menurut Hassanien & Olivia (2017), normalisasi ini dapat meningkatkan hasil klasifikasi dengan menutup celah antara fitur yang memiliki nilai tertinggi dengan nilai terendah sehingga simpangan data berkurang. Selanjutnya diimplementasikan metode KNN karena memiliki keunggulan yaitu



pengimplementasian yang mudah dan komputasinya sederhana serta tidak membutuhkan waktu untuk pelatihan (*training*).

Dari penelitian-penelitian tersebut, penulis ingin mengembangkan sebuah ide yaitu dalam pengambilan presensi yang pada awalnya hanya memakai kertas, kartu identitas atau *fingerprnt* diubah menjadi mengandalkan pengenalan wajah yang didukung dengan metode deteksi wajah *Haar Cascade Classifier* dan metode ekstraksi fitur yaitu *Local Binary Pattern* serta metode klasifikasi *K-Nearest Neighbor* yang berjudul "Sistem Presensi Mahasiswa berdasarkan Pengenalan Wajah menggunakan Metode LBP dan *K-Nearest Neighbor* berbasis Mini PC". Diharapkan dengan terciptanya ide ini dapat memudahkan kegiatan presensi yang ada di setiap instansi.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka rumusan masalah untuk penelitian ini antara lain:

1. Bagaimana rata-rata akurasi sistem dalam melakukan pendeteksian wajah menggunakan metode *Haar Cascade Classifier*?
2. Bagaimana rata-rata akurasi sistem dalam melakukan pengenalan wajah mahasiswa menggunakan metode *Local Binary Pattern* dan *K-Nearest Neighbor* sebagai metode pengenalan wajah?
3. Bagaimana rata-rata waktu komputasi sistem dari proses deteksi wajah menggunakan metode *Haar Cascade Classifier*?
4. Berapa rata-rata waktu komputasi sistem dari proses pengenalan wajah menggunakan metode *Local Binary Pattern* dan *K-Nearest Neighbor*?

1.3 Tujuan

Berdasarkan latar belakang dan rumusan masalah tersebut, penelitian ini mempunyai beberapa tujuan, antara lain:

1. Mengetahui rata-rata akurasi sistem dalam melakukan pendeteksian wajah menggunakan metode *Haar Cascade Classifier*.
2. Mengetahui rata-rata akurasi sistem dalam melakukan pengenalan wajah menggunakan *Local Binary Pattern* dan *K-Nearest Neighbor*.
3. Mengetahui rata-rata waktu komputasi sistem dari proses deteksi wajah menggunakan metode *Haar Cascade Classifier*?
4. Mengetahui rata-rata waktu komputasi sistem dari proses pengenalan wajah menggunakan metode *Local Binary Pattern* dan *K-Nearest Neighbor*?

1.4 Manfaat

Manfaat dari adanya penelitian ini, penulis berharap karya yang telah dibuat dapat memudahkan sistem presensi yang ada pada setiap instansi terutama yang masih memakai sistem lama agar dalam melakukan presensi untuk anggotanya dapat lebih cepat, efisien, relevan dan tidak banyak membuang waktu.



1.5 Batasan Masalah

Batasan penelitian disini dibuat untuk membatasi pembahasan pada penelitian ini sehingga penelitian terfokus sesuai rumusan masalah. Batasan masalah penelitian ini diantara lain:

1. Pengambilan citra wajah harus dilakukan dengan ketentuan wajah menghadap langsung ke *webcam* dengan pencahayaan yang cukup.
2. Digunakan folder data latih untuk menyimpan data latih dari mahasiswa, berupa citra wajah mahasiswa dengan fitur LBP yang telah dinormalisasi.
3. *Database* SQLite digunakan untuk menyimpan informasi mahasiswa berupa id, nama, nim, dan waktu terdeteksinya mahasiswa.
4. Ukuran citra wajah yang terdeteksi yaitu 128 x 128 piksel.
5. Digunakan tiga nilai parameter k yang berbeda pada *K-Nearest Neighbor* yaitu 3, 5, dan 7.
6. Pengujian dilakukan pada kombinasi variasi wajah dengan citra sekali tangkap dari *webcam* sebanyak delapan subjek.
7. Hasil pengenalan wajah mahasiswa yang akan dimasukkan pada *database* presensi SQLite adalah hasil klasifikasi dari KNN dengan nilai $k = 3$ karena memiliki rata-rata akurasi yang paling baik yaitu 88.88%.
8. Sampel yang digunakan untuk satu individu hanya sebanyak 8 sampel.

1.6 Sistematika Penulisan

Sistematika penulisan pada skripsi ini terdiri dari beberapa bab sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika penulisan skripsi.

BAB II LANDASAN KEPUSTAKAAN

Bab ini berisi teori-teori pendukung tentang implementasi sistem presensi mahasiswa menggunakan metode *Local Binary Pattern* dan *K-Nearest Neighbor*.

BAB III METODOLOGI PENELITIAN

Bab ini berisi metode penelitian implementasi sistem presensi mahasiswa menggunakan metode *Local Binary Pattern* dan *K-Nearest Neighbor*.

BAB IV REKAYASA KEBUTUHAN

Bab ini berisi tentang kebutuhan yang diperlukan oleh sistem yang akan dibuat.



BAB V PERANCANGAN DAN IMPLEMENTASI

Pada bab ini berisi tentang perancangan dan implementasi dari sistem yang berada pada bab sebelumnya.

BAB VI PENGUJIAN

Bab ini berisi pelaksanaan pengujian dan analisis penerapan pengenalan wajah menggunakan metode *Local Binary Pattern* dan *K-Nearest Neighbor* pada sistem presensi.

BAB VII PENUTUP

Bab ini berisi kesimpulan dan saran untuk pengembangan selanjutnya dari pelaksanaan penelitian ini.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Pada penelitian ini menggunakan beberapa jurnal penelitian yang sudah ada, yang berfungsi sebagai pembandingan dengan penelitian yang akan dilakukan pada penelitian ini. Berikut ini merupakan penelitian-penelitian yang akan dijadikan sebagai acuan:

Tabel 2.1 Jurnal Terkait

No	Nama Penulis (Tahun), Judul Penelitian	Kelebihan	Kelemahan	Rencana Penelitian
1.	Ajib Susanto (2018), <i>A High Performance of Local Binary Pattern on Classify Javanese Character Classification.</i> Universitas Dian Nuswantoro, Semarang.	Penggunaan metode <i>Local Binary Pattern</i> yang dikombinasikan dengan metode <i>K-Nearest Neighbor</i> menghasilkan akurasi 82.5% dengan nilai $k = 3$.	Objek berupa dataset yang berasal dari foto hasil tulisan tangan oleh peneliti, bukan dari buku.	Mengadopsi metode <i>Local Binary Pattern</i> untuk ekstraksi fitur wajah dan metode <i>K-Nearest Neighbor</i> untuk melakukan klasifikasi atau pengenalan wajah.
2.	Eva Y Puspaningrum (2018), <i>Deteksi Wajah Dengan Boosted Cascade Classifier.</i> Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya.	Tingkat akurasi sistem dalam pendeteksian wajah menggunakan <i>Haar Cascade Classifier</i> yaitu 81.6%.	Pendeteksian wajah dilakukan pada citra berwarna atau bukan <i>grayscale</i> sehingga warna dari wajah dapat memengaruhi hasil deteksi wajah.	Menerapkan metode <i>Haar Cascade Classifier</i> pada citra berformat <i>grayscale</i> untuk pendeteksian wajah.
3.	Muhammad Mukti Raharjo (2020), <i>Sistem Deteksi Wajah Dan Sebuah Benda Menggunakan Algoritma Viola-</i>	Tingkat akurasi sistem dalam pendeteksian wajah menggunakan <i>Haar Cascade</i>	Ketentuan data uji hanya menggunakan 1 variasi posisi wajah.	Penggunaan data latih <i>Haar Cascade Classifier</i> untuk wajah yang menghadap



	Jones Berbasis OpenCV. Sekolah Tinggi Manajemen Informatika dan Komputer Amikom, Yogyakarta.	<i>Classifier</i> yaitu 77.8%.		langsung ke kamera.
4	Rabiuldien Amat (2017), Implementasi Metode <i>Local Binary Patterns</i> Untuk Pengenalan Pola Huruf Hiragana Dan Katakana Pada <i>Smartphone</i> . Universitas Halu Oleo, Kendari.	Rata-rata akurasi sistem dalam mengenali pola huruf menggunakan <i>Local Binary Pattern</i> sebesar 81.1%	Dibutuhkan proses <i>cropping</i> untuk mendeteksi huruf yang akan dikenali.	Mengadopsi metode <i>Local Binary Pattern</i> sebagai metode untuk ekstraksi fitur wajah.

Berikut ini merupakan ringkasan dalam bentuk deskripsi dari kajian pustaka dengan tujuan agar dapat memperjelas pemahaman terhadap semua pustaka yang telah dipaparkan sebelumnya pada Tabel 2.1.

Pada penelitian yang berjudul “*A High Performance of Local Binary Pattern on Classify Javanese Character Classification*” yang dilakukan oleh Ajib Susanto (2018), peneliti membuat sistem yang dapat mengenali tulisan karakter aksara jawa dengan menggunakan metode *Local Binary Pattern* sebagai metode untuk ekstraksi fitur sedangkan metode *K-Nearest Neighbor* digunakan sebagai metode untuk melakukan klasifikasi dari hasil ekstraksi fitur. Dari pengujian yang dilakukan didapatkan akurasi pengenalan tulisan aksara jawa terbaik sebesar 82.5% dengan nilai $k = 3$.

Pada penelitian yang berjudul “Deteksi Wajah Dengan *Boosted Cascade Classifier*” yang dilakukan oleh Eva Y Puspaningrum (2018), peneliti membuat sistem untuk mendeteksi wajah menggunakan metode *Haar Cascade Classifier* yang telah dilatih menggunakan *dataset* MIT CBCL lalu dilakukan deteksi wajah pada data uji berwarna dengan resolusi 300 x 400 piksel sehingga dihasil tingkat akurasi deteksi wajah sebesar 81.6%.

Pada penelitian yang berjudul “Sistem Deteksi Wajah Dan Sebuah Benda Menggunakan Algoritma Viola-Jones Berbasis Open CV” yang dilakukan oleh Muhammad Mukti Raharjo (2020), peneliti membuat sistem yang dapat menentukan apakah pada suatu citra tersebut terdapat wajah atau bukan menggunakan *Haar Cascade Classifier* sehingga didapatkan akurasi sebesar 77.8%

Pada penelitian yang berjudul “Implementasi Metode *Local Binary Patterns* Untuk Pengenalan Pola Huruf Hiragana Dan Katakana Pada *Smartphone*” yang dilakukan oleh Rabiuldin Amat (2017), peneliti membuat sistem yang dapat melakukan pengenalan pola huruf Hiragana dan Katakana melalui *smartphone*



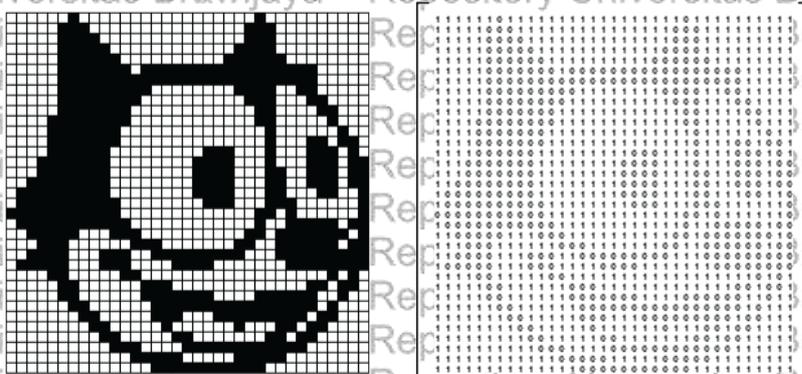
dengan sistem operasi Android. Pengenalan pola dilakukan dengan menggunakan metode *Local Binary Pattern* sebagai metode untuk ekstraksi fitur dari pola huruf tersebut. Didapatkan akurasi pengenalan pola huruf yaitu sebesar 81.1%

2.2 Dasar Teori

Pada dasar teori menjelaskan literatur yang terkait dengan penelitian yang dilakukan saat ini.

2.2.1 Pemrosesan Citra

Metode pemrosesan citra digital berasal dari dua area utama yaitu peningkatan informasi bergambar untuk menggambarkan interpretasi dari manusia dan pengolahan data citra untuk penyimpanan, transmisi, dan lain-lain (Dewi & Supianto, 2015).



Gambar 2.1 Ilustrasi Matriks Citra Digital

Sumber: <http://blog.kleinproject.org/?p=588>

2.2.2 Operasi Konversi Citra RGB ke *Grayscale*

Merupakan operasi pemrosesan citra yang mengubah citra dengan format 3 *channel* (RGB atau *Red Green Blue*) menjadi 1 *channel grayscale*. Operasi konversi ke RGB ke *grayscale* dirumuskan pada Persamaan 2.1 sebagai berikut:

$$Gray = 0.299 R + 0.587 G + 0.114 B \quad (2.1)$$

Keterangan:

Gray = Nilai keabuan

R = Nilai intensitas dari *channel* merah

G = Nilai intensitas dari *channel* hijau

B = Nilai intensitas dari *channel* biru



2.2.3 Normalisasi *Min Max*

Normalisasi *Min Max* merupakan salah satu teknik normalisasi data untuk mempersempit celah antara nilai tertinggi dan terendah dari fitur yang telah diekstraksi sehingga didapatkan data dengan deviasi atau simpangan yang lebih kecil serta meningkatkan hasil klasifikasi (Hassanien & Oliva, 2017).

Normalisasi *Min Max* dirumuskan pada Persamaan 2.2 sebagai berikut:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.2)$$

Keterangan:

x_{norm} = nilai x yang sudah dinormalisasi

x = nilai x dalam dataset

x_{max} = nilai terbesar dari x dalam dataset

x_{min} = nilai terkecil dari x dalam dataset

2.2.4 Basis Data

Basis data atau *database* adalah sekumpulan data yang telah disusun sedemikian rupa menggunakan aturan tertentu yang saling berkaitan sehingga memudahkan pengguna dalam melakukan pengelolaan data dan mempermudah dalam memperoleh informasi. Selain itu basis data juga dapat disebut sebagai kumpulan file, tabel, maupun arsip yang saling terhubung yang disimpan dalam bentuk media elektronik (Rozaq, 2019).

2.2.5 Sistem Manajemen Basis Data

Sistem Manajemen Basis Data atau biasa disebut dengan DBMS (*Database Management System*) merupakan perangkat lunak yang memungkinkan pengguna untuk membuat dan memelihara basis data. DBMS memungkinkan pengguna untuk menentukan tipe data, struktur, dan batasan untuk data, menyimpan data, maupun memanipulasi data seperti melakukan fungsi seperti menanyakan basis data, memperbarui basis data, dan menghasilkan laporan dari data yang disimpan dalam basis data (Group, 2005). Perangkat lunak yang merupakan DBMS contohnya yaitu MySQL, SQLite, Oracle, dan sebagainya.

2.2.6 Metode *Haar Cascade Classifier*

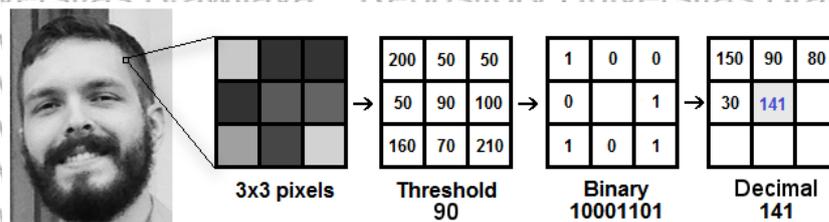
Metode *Haar Cascade Classifier* adalah sebuah metode dimana ketika ada wajah yang terdeteksi, maka biasanya bagian wajah tersebut akan ditandai dengan bentuk kotak atau lingkaran. *Haar Cascade* memiliki beberapa tahapan yaitu terdapat fitur *Haar Wavelet* merupakan gelombang tunggal bujur sangkar (satu interval tinggi dan satu rendah). Kemudian kombinasi-kombinasi kotak yang digunakan untuk pendeteksian objek visual lebih dikenal dengan nama fitur *Haar* dimana setiap gambar terdiri dari ratusan fitur *Haar* yang mempunyai nilai tertentu di setiap fiturnya. *Haar Cascade Classifier* terdiri atas beberapa tahapan yaitu mendeteksi objek-objek yang diinginkan dengan cara menggerakkan *window* melalui suatu citra. Setiap tahapan akan dilabeli sebagai positif atau negatif



tergantung pada hasil apabila objek spesifik telah ditemukan atau tidak. Apabila label negatif maka klasifikasi wilayah tersebut dinyatakan selesai dan *window* akan berpindah ke lokasi berikutnya.

2.2.7 Metode *Local Binary Pattern* (LBP)

Local Binary Pattern (LBP) pertama kali didefinisikan oleh Ojala *et al*, sebagai ukuran dari tekstur *grayscale* yang invarian karena hampir tidak dipengaruhi oleh intensitas cahaya yang berbeda. Maka dari itu metode ini digunakan untuk mendeskripsikan suatu tekstur, mempunyai daya pembeda yang akurat dan mampu menoleransi perubahan dari *grayscale* atau monotonik (Turiyanto, et al., 2014).



Gambar 2.2 Proses Kalkulasi Pikel LBP

Sumber: <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>

Ekstraksi fitur dimulai dari piksel ujung kiri atas sampai dengan ujung kanan bawah. Pada gambar 2.2, setiap satu siklus ekstraksi fitur LBP, satu piksel pada citra input dengan koordinat x, y (piksel tengah) akan diubah menjadi satu piksel fitur LBP dengan koordinat yang sama. Piksel pada koordinat tersebut dijadikan sebagai patokan untuk melakukan threshold. Apabila nilai piksel disekitar piksel tengah tersebut lebih besar atau sama dengan nilai piksel tengah, maka pada posisi piksel disekitarnya akan menjadi satu, sebaliknya maka nol. Nilai-nilai biner tersebut akan diubah menjadi bentuk desimal sebagai hasil dari ekstraksi fitur untuk satu piksel.

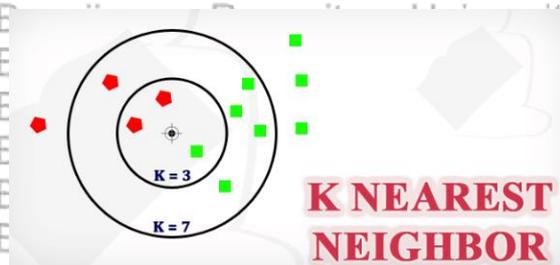
2.2.8 Metode *K-Nearest Neighbor*

Algoritme *K-Nearest Neighbor* (KNN) adalah sebuah algoritme yang berfungsi untuk melakukan suatu klasifikasi terhadap objek, dimana data pembelajaran atau bisa disebut variabel k sebagai data latih yang jaraknya paling dekat dengan objek tersebut. Adapun persyaratan dari metode KNN ini yaitu nilai variabel k tidak boleh lebih besar dari jumlah data latih yang sudah disediakan dan nilai variabel k harus bersifat ganjil atau lebih dari satu. Untuk menentukan dekat atau jauhnya jarak data, maka dapat dihitung dengan menggunakan metode *cosine similarity*. Dekat jauhnya tetangga biasanya dihitung berdasarkan *Euclidean Distance*, yaitu dengan rumus:

$$D(a, b) = \sqrt{\sum_{i=1}^n (b_i - a_i)^2} \quad (2.3)$$



Dimana $D(a,b)$ adalah jarak skalar dari dua buah vektor data a dan b yang berupa matrik berukuran d dimensi. Apabila nilai $D(a,b)$ semakin kecil maka dapat dinyatakan bahwa data a dengan data b adalah mirip, sebaliknya apabila nilai $D(a,b)$ semakin besar maka data a dengan data b adalah tidak mirip.



Gambar 2.3 Ilustrasi *K-Nearest Neighbor*

Sumber: <http://assignmentsolutionguru.com/article/intro-to-knn-in-python>



2.2.9 Structured Query Language (SQL)

SQL singkatan dari *Structured Query Language* merupakan bahasa *query standar* yang digunakan untuk mengakses basis data relasional. Standarisasi internasional terhadap SQL pertama kali dilakukan oleh ANSI (American National Standards Institution), melalui publikasi *Database Language SQL* (ANSI X3.136-1986). Saat ini, ANSI dan ISO (International Standards Organization) merupakan dua organisasi yang membuat standarisasi terhadap SQL (Kadir, 1999).



SQL

Structured Query Language

Gambar 2.4 Logo SQL

Sumber: <https://trainingnoida.in/best-sql-training>



BAB 3 METODOLOGI PENELITIAN

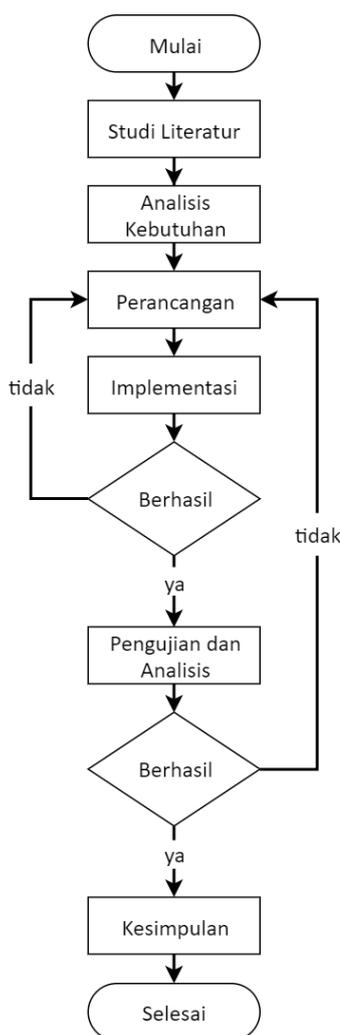
3.1 Tipe Penelitian

Penelitian ini merupakan penelitian dengan tipe Implementatif-Pengembangan Lanjut (*Enhancement*). Pengembangan Lanjut dilakukan dengan mengadopsi metode *Local Binary Pattern* dan *K-Nearest Neighbor* dari penelitian yang dilakukan oleh Ajib Susanto (2018) dengan judul “*A High Performance of Local Binary Pattern on Classify Javanese Character Classification*” dengan akurasi 82.5% menggunakan nilai $k = 3$ dan ukuran citra 64×64 piksel. Penelitian tersebut bertujuan untuk melakukan pengenalan karakter huruf aksara jawa sedangkan pada penelitian ini akan dikembangkan sistem presensi mahasiswa berdasarkan pengenalan wajah mahasiswa menggunakan metode *Local Binary Pattern* dengan ukuran citra 128×128 piksel dan metode *K-Nearest Neighbor* dengan nilai $k = 3, 5, \text{ dan } 7$. Kedua metode tersebut (*Local Binary Pattern* dan *K-Nearest Neighbor*) akan digunakan sebagai metode ekstraksi fitur dan klasifikasi untuk pengembangan lanjut dari penelitian sebelumnya yang dilakukan oleh Fitrihadi Surya Dharma (2019) berjudul “*Rekognisi Wajah pada Sistem Smart Class untuk Deteksi Kehadiran Mahasiswa menggunakan Metode Viola Jones dan Local Binary Patterns Histograms (LBPH) berbasis Raspberry Pi*” dengan rata-rata akurasi 56.94% dan rata-rata waktu komputasi pengenalan wajah untuk satu mahasiswa sebesar 2.17 detik.

3.2 Strategi dan Rancangan Penelitian

3.2.1 Metode Secara Umum

Metode secara umum yang digunakan dalam penelitian ini berupa tahapan yang telah disusun secara sistematis sebagai diagram alir pada Gambar 3.1 berikut ini:



Gambar 3.1 Diagram Alir Metodologi Penelitian

Penjelasan tiap komponen dari diagram alir metodologi penelitian:

1. Studi Literatur

Melakukan pencarian referensi teori yang relevan dengan masalah pada penelitian

2. Analisis Kebutuhan

Melakukan analisis kebutuhan yang diperlukan oleh sistem

3. Perancangan

Melakukan perancangan sistem yang akan dibuat

4. Implementasi

Melakukan implementasi dari rancangan sistem

5. Pengujian

Melakukan pengujian dari sistem yang sudah diimplementasikan



6. Analisis

Melakukan analisis dari hasil pengujian sistem

7. Kesimpulan

Mengambil kesimpulan berdasarkan analisis hasil pengujian sistem untuk menjawab rumusan masalah dari penelitian

3.2.2 Subjek Penelitian

Penelitian ini dilakukan dengan melibatkan subjek dengan karakteristik sebagai berikut:

1. Subjek adalah mahasiswa yang diambil citra wajahnya melalui *webcam* sebanyak delapan kali sebagai sampel data uji.
2. Subjek merupakan rekan dari peneliti yang memiliki fakultas yang sama.

3.2.3 Lokasi Penelitian

Penelitian ini dilakukan di area Fakultas Ilmu Komputer Universitas Brawijaya.

3.2.4 Teknik Pengumpulan Data

Pada penelitian ini, pengumpulan data dilakukan dengan cara mengambil citra wajah dari subjek sebanyak delapan kali dengan pose wajah tegak menghadap langsung ke *webcam*. Hasil pengambilan citra wajah berupa data citra lunak.

3.2.5 Teknik Analisis Data dan Pembahasan

Pada penelitian ini, analisis data kuantitatif dilakukan menggunakan metode jarak Euclidean untuk menentukan tingkat kemiripan citra wajah dari *webcam* dengan citra wajah pada *database*. Setelah itu nilai jarak Euclidean antara citra uji dan citra latih akan diurutkan dari yang paling kecil ke yang terbesar lalu diterapkan metode *K-Nearest Neighbor* untuk melakukan klasifikasi dari citra uji.

3.2.6 Peralatan Pendukung yang digunakan

3.2.6.1 Perangkat Keras

1. *Mini PC* Intel NUC5i7RYH
2. Monitor Waveshare 7-inch
3. *Mouse* Alcatroz Silent 5
4. *Webcam* Logitech C270 HD

3.2.6.2 Perangkat Lunak

1. NumPy 1.17.0
2. OpenCV 4.1.0.25



3. PyQt5 5.13.0

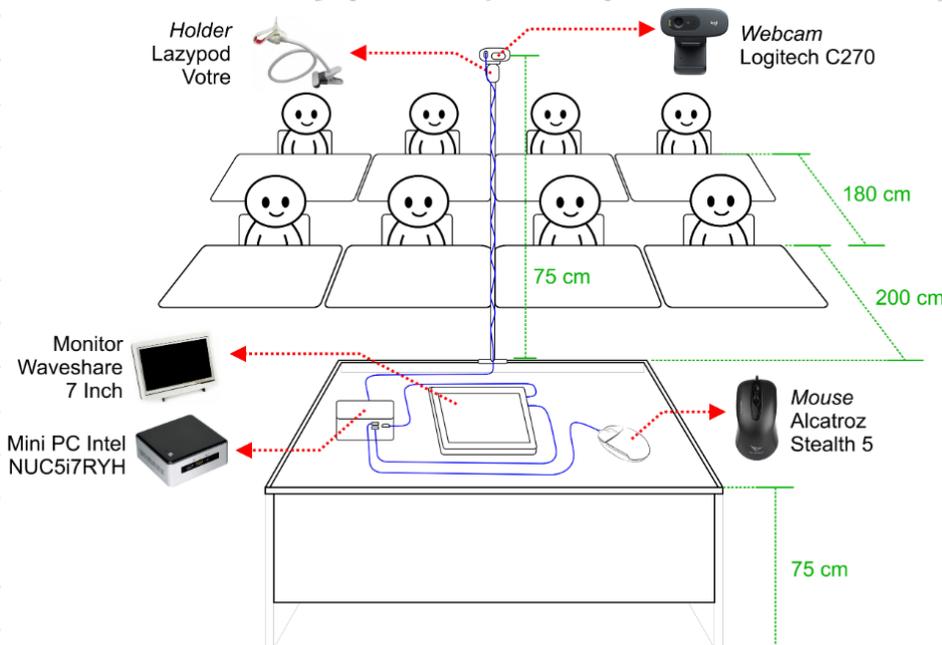
4. Python 3.7.4

5. SQLite3.3.21.0



BAB 4 REKAYASA KEBUTUHAN

4.1 Gambaran Umum Sistem



Gambar 4.1 Gambaran Umum Sistem

Sistem yang dikembangkan dalam penelitian ini sesuai pada Gambar 4.1 yaitu dengan mengandalkan citra yang ditangkap oleh *webcam*, sistem dapat mendeteksi wajah mahasiswa yang ada di kelas menggunakan *Haar Cascade Classifier* setelah itu dilakukan ekstraksi fitur menggunakan *Local Binary Pattern* sehingga *K-Nearest Neighbor* dapat melakukan klasifikasi (pengenalan wajah) berdasarkan fitur tersebut. Digunakan tiga nilai *k* yang berbeda pada *K-Nearest Neighbor* yaitu 3, 5, dan 7 untuk mencari nilai *k* manakah yang memiliki akurasi terbaik. Hasil klasifikasi dari *K-Nearest Neighbor* berupa nama mahasiswa yang nantinya akan masuk ke *database* bersamaan dengan waktu terdeteksinya berdasarkan nilai *k* yang akurasinya paling baik apabila tombol konfirmasi presensi ditekan.

4.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem menjelaskan tentang berbagai macam kebutuhan yang diperlukan sistem. Analisis kebutuhan sistem terdiri atas kebutuhan sistem, kebutuhan fungsional dan kebutuhan non fungsional.



4.2.1 Kebutuhan Sistem

4.2.1.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras pada sistem yang dikembangkan dalam penelitian ini adalah:

1. Webcam Logitech C270

Web Camera atau biasa disebut *webcam* adalah setiap kamera video yang yang dapat dihubungkan ke komputer dan menampilkan output pada sebuah halaman *web*. Istilah *webcam* merujuk pada teknologi secara umumnya, sehingga kata *web* kadang-kadang diganti dengan kata lain yang mendeskripsikan pemandangan yang ditampilkan di kamera, misalnya *StreetCam* yang memperlihatkan pemandangan jalan. Ada juga *Metrocam* yang memperlihatkan pemandangan panorama kota dan pedesaan, *TrafficCam* yang digunakan untuk memonitor keadaan jalan raya, *WeatherCam* untuk cuaca, bahkan keadaan gunung berapi dengan *VolcanoCam* (Shally, et al., 2006).



Gambar 4.2 Webcam Logitech C270 HD

Sumber: <https://www.harveynorman.com.au/logitech-c270-hd-webcam.html>

2. Mini PC Intel NUC5i7RYH

Mini PC ini hadir dengan spesifikasi yang sangat baik dan berkembang, ditenagai oleh prosesor Intel Core i7 3.1 Ghz. Dual-core i7-5557U dengan *Intel Turbo Boost Technology* yang memberikan tambahan Ghz sesuai dengan permintaan dan mempunyai 16GB RAM serta didukung oleh kartu Intel Iris Graphics 6100. *Mini PC* ini memiliki *port* pengisian daya USB 3.0 dan *mini-HDMI* untuk output grafis. Ukurannya sangat kecil dari komputer pada umumnya karena memiliki 115 x 111 x 49 mm.



Gambar 4.3 Mini PC NUC5i7RYH

Sumber: <https://www.intel.co.id/content/www/id/id/products/boards-kits/nuc/kits/nuc5i7ryh.html>



3. Mouse Alcatroz Stealth 5

Mouse Alcatroz Stealth 5 adalah sebuah mouse optik yang memiliki 3 tombol dengan switch yang berdurasi 5 juta klik serta tidak bersuara (senyap) apabila diklik.



Gambar 4.4 Mouse Alcatroz Stealth 5

Sumber: <https://leapfroglobal.com/alcatroz/alcatroz-sleath-5/alcatroz-mice-slient>

4. Monitor Waveshare 7-inch

Monitor Waveshare 7-inch merupakan monitor dengan layar LCD kapasitif (layar sentuh) berukuran 7 inci dan memiliki koneksi input HDMI. Monitor ini memiliki resolusi tampilan 800 x 480 piksel dengan fitur cahaya latar yang dapat diatur. Monitor ini juga dapat digunakan pada *Mini PC* yang populer seperti Raspberry Pi, BB Black, sama halnya dengan komputer umum yang lain.



Gambar 4.5 Monitor Waveshare 7-inch

Sumber: [https://www.waveshare.com/wiki/7inch_HDMI_LCD_\(B\)](https://www.waveshare.com/wiki/7inch_HDMI_LCD_(B))

4.2.1.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak pada sistem yang dikembangkan dalam penelitian ini adalah:

1. *Operating System (OS)* Ubuntu 16.04

Merupakan sistem operasi bawaan dari Mini PC.



2. Python 3.7.4
Merupakan bahasa pemrograman yang digunakan pada algoritme aplikasi sistem presensi.
3. OpenCV 4.0
Merupakan library dari Python yang menyediakan berbagai fungsi untuk mempermudah pemrosesan citra digital.
4. SQLite
Merupakan *database* yang digunakan untuk menyimpan informasi mahasiswa berupa ID mahasiswa (untuk pengolahan citra digital), NIM (Nomor Induk Mahasiswa), nama mahasiswa, serta waktu terdeteksinya.
5. PyQt5
PyQt5 merupakan modul untuk Python untuk membuat aplikasi antarmuka grafis pengguna.

4.2.2 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan dari sistem yang terdiri dari berbagai proses agar sistem mampu menghasilkan output sesuai dengan yang diinginkan. Berikut ini merupakan kebutuhan fungsional yang ada pada sistem:

1. *Webcam* Logitech C270 mampu menangkap kondisi kelas sehingga dihasilkan citra digital sebagai input untuk Mini PC.
2. *Mini PC* mampu memproses citra digital yang dihasilkan oleh *webcam*.
3. *Mouse* mampu mengarahkan kursor sehingga pengguna (dosen atau peneliti) dapat mengendalikan atau melakukan kontrol atas berjalannya sistem presensi.
4. Sistem mampu melakukan koneksi ke *database* SQLite dan melakukan modifikasi di dalamnya.
5. Sistem mampu mendeteksi wajah pada citra input baik tunggal maupun jamak.
6. Sistem mampu melakukan pengenalan wajah pada wajah terdeteksi menggunakan *K-Nearest Neighbor* dengan tiga nilai *k* yang berbeda.
7. Sistem dapat menampilkan antarmuka dari aplikasi sistem presensi.
8. Aplikasi sistem presensi mampu menyajikan berbagai informasi seperti tampilan citra input, jam berjalan, kumpulan hasil deteksi wajah dan pengenalannya, rata-rata waktu komputasi, tabel daftar presensi mahasiswa, serta tombol untuk melakukan kontrol aplikasi.
9. Tombol-tombol yang digunakan sebagai kontrol aplikasi sistem presensi dapat berfungsi sebagaimana mestinya seperti memulai atau menghentikan sistem presensi, melakukan konfirmasi presensi, serta mereset daftar presensi.



4.2.3 Kebutuhan Non Fungsional

Kebutuhan Non Fungsional adalah berbagai aspek mengenai batasan-batasan dalam perancangan sistem. Kebutuhan Non Fungsional dari sistem disebutkan sebagai berikut.

4.2.3.1 Karakteristik Pengguna

Subjek yang merupakan pengguna dari sistem ini adalah dosen atau peneliti, sedangkan mahasiswa berperan sebagai subjek yang akan dideteksi oleh sistem dan hasil pengenalannya akan masuk ke dalam daftar presensi. Mahasiswa hanya perlu duduk di kursi dan harus terdeteksi oleh sistem sehingga pengguna dapat melakukan presensi dengan menekan tombol konfirmasi yang ada pada aplikasi sistem presensi.

4.2.3.2 Lingkungan Operasi

Lingkungan operasi merupakan kebutuhan yang mendukung sistem agar dapat berjalan dengan baik, disebutkan sebagai berikut:

1. Subjek yang akan dideteksi (mahasiswa) oleh sistem harus terlihat jelas wajahnya (tidak terhalangi oleh objek apapun) dan menghadap tepat ke *webcam*.
2. Pengguna sistem (dosen ataupun peneliti) melakukan konfirmasi hasil pengenalan dengan menekan tombol konfirmasi pada aplikasi sistem presensi menggunakan *mouse* yang sudah disediakan oleh sistem.
3. Pencahayaan yang cukup.

4.2.3.3 Batasan Desain Sistem

Batasan desain sistem merupakan batasan-batasan yang digunakan sehingga pengembangan sistem pada penelitian ini menjadi terarah dan tidak meluas serta dapat berjalan baik sesuai yang diharapkan. Berikut ini merupakan batasan desain sistem pada penelitian ini:

1. Menggunakan *webcam* untuk mendapatkan input.
2. Menggunakan *Mini PC* sebagai perangkat pemroses utama.
3. Menggunakan *OpenCV* sebagai *library* untuk mengolah citra digital.
4. Setiap wajah yang terdeteksi akan diubah ukurannya ke 128 x 128 agar semua ukuran citra yang diproses sama dan menghasilkan waktu komputasi yang singkat.
5. Hasil pengujian pada sistem ini difokuskan pada pemilihan tiga nilai k yang berbeda pada *K-Nearest Neighbor* untuk mencari nilai k yang paling optimal agar sistem dapat melakukan presensi dengan nilai akurasi yang paling baik.

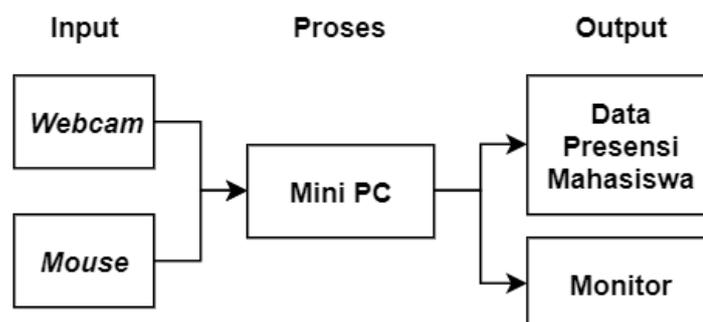


BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada Bab ini berisi proses-proses perancangan dan implementasi dari “Sistem Presensi Mahasiswa Berdasarkan Pengenalan Wajah Menggunakan Metode LBP Dan *K-Nearest Neighbor* Berbasis Mini PC” yang dapat memudahkan sistem presensi yang ada pada setiap instansi terutama yang masih memakai sistem lama agar dalam melakukan presensi untuk anggotanya dapat lebih cepat, efisien, relevan dan tidak banyak membuang waktu. Jika ada mahasiswa yang akan melakukan presensi didepan *webcam* kemudian wajah dan data cocok, maka pengguna akan menekan tombol yang ada pada tampilan aplikasi sistem presensi menggunakan *mouse* dan sehingga hasil pengenalan dari setiap wajah mahasiswa yang terdeteksi di kelas akan dimasukkan ke dalam *database* SQLite sehingga mahasiswa tersebut tercatat kehadirannya pada daftar presensi di hari tersebut.

5.1 Perancangan Sistem

Pada perancangan sistem berisi tiga macam dari perancangan yaitu perancangan prototipe alat, perangkat keras, dan perangkat lunak. Perancangan sistem pada dasarnya berisi hubungan antara input, proses, dan output yang biasa disebut diagram blok sistem yang disajikan pada Gambar 5.1 sebagai berikut:

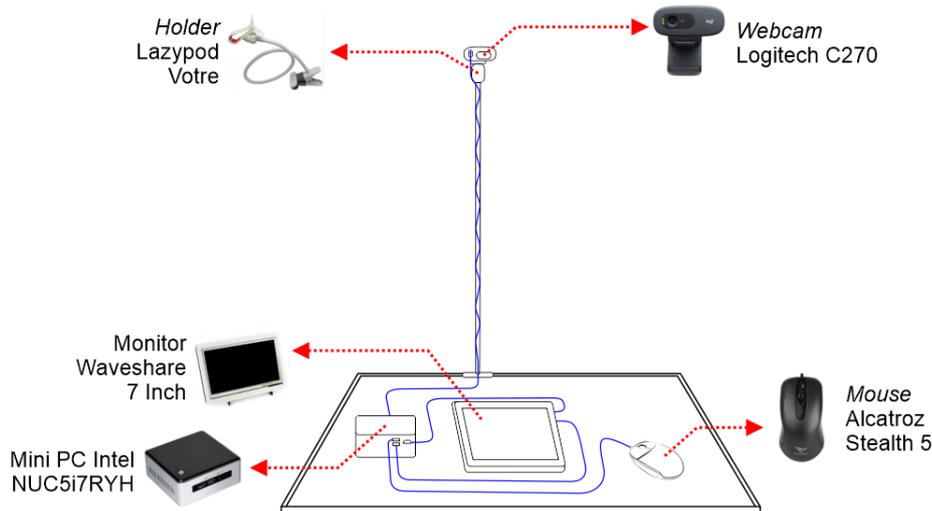


Gambar 5.1 Diagram Blok Sistem

Input utama pada sistem ini berasal dari *webcam* berupa citra mahasiswa dengan kondisi duduk di kelas sedangkan *mouse* digunakan untuk menerima inputan dari pengguna sistem untuk melakukan kontrol pada aplikasi sistem presensi. Pemrosesan data dilakukan oleh *Mini PC* sehingga hasilnya berupa data presensi mahasiswa yang sudah dikenali wajahnya dan data tersebut sudah masuk ke *database* SQLite. Monitor berguna untuk menampilkan aplikasi sistem presensi sehingga pengguna dari sistem dapat mengaksesnya dengan mudah.

5.1.1 Perancangan Prototipe Sistem

Perancangan prototipe dari sistem merupakan hal yang terpenting terutama dalam hal penempatan *webcam* karena membutuhkan lokasi yang tepat sehingga wajah dari para mahasiswa dapat terdeteksi dengan baik oleh sistem seperti ditunjukkan pada Gambar 5.2.

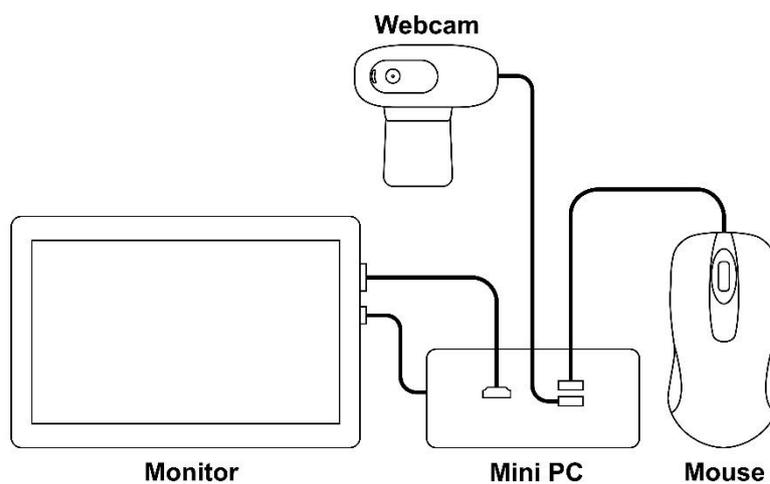


Gambar 5.2 Desain Prototipe Sistem

Mini PC, *Monitor*, dan *Mouse* berada di atas meja sedangkan *holder* ditancapkan di sisi depan meja dan dipasang lurus keatas. *Webcam* dikaitkan ke penjepit yang ada di bagian atas *holder* sehingga hasil citra dari *webcam* memiliki sudut pandang lebih baik dalam menangkap wajah mahasiswa, baik dibaris pertama maupun dibaris kedua. Posisi dari pengguna sistem yaitu tepat dibelakang meja sehingga memudahkan pengguna dalam memantau maupun mengendalikan aplikasi sistem presensi.

5.1.2 Perancangan Perangkat Keras

Pada perancangan perangkat keras ini, terdapat *Mini PC* sebagai pemroses utama, *webcam* Logitech C270 dan *mouse* sebagai input, dan monitor sebagai output.



Gambar 5.3 Skema Perancangan Perangkat Keras

Pada Gambar 5.3 terdapat *webcam* dan *mouse* sebagai perangkat input yang terhubung ke *Mini PC* pada slot USB. Sedangkan monitor sebagai perangkat output



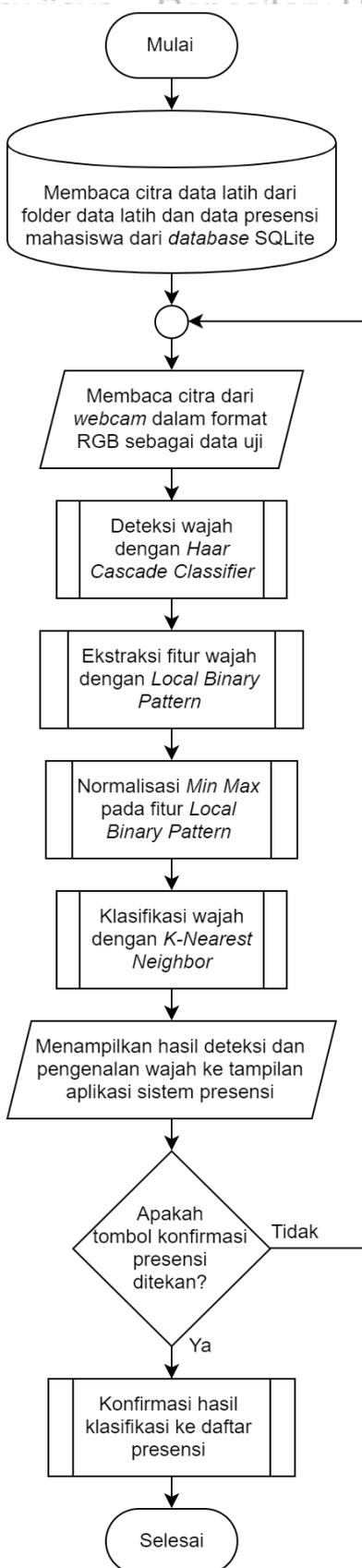
yang terhubung melalui slot HDMI dan membutuhkan sumber daya dari slot USB. *Mini PC* berperan penting dalam memproses input sehingga dihasilkan output yang nantinya ditampilkan lewat monitor.

5.1.3 Perancangan Perangkat Lunak

Pada perancangan perangkat lunak, terdapat algoritme jalannya perangkat lunak dari sistem. Perancangan perangkat lunak dimulai dengan pembacaan data citra latih (LBP) dari folder data latih dan data mahasiswa dari database SQLite serta data citra RGB dari *webcam*. Citra RGB diproses menjadi *grayscale* untuk mendeteksi wajah dari subjek dan dilakukan ekstraksi fitur menggunakan *Local Binary Pattern* sehingga dihasilkan citra uji *Local Binary Pattern* untuk dilakukan klasifikasi menggunakan *K-Nearest Neighbor*. Hasil klasifikasi beserta waktu terdeteksi akan dimasukkan ke dalam *database* SQLite sebagai presensi dari mahasiswa saat tombol konfirmasi presensi pada aplikasi sistem presensi ditekan oleh pengguna.

5.1.3.1 Perancangan Program Utama

Pada bagian ini, sistem akan melakukan inialisasi *interface* dari aplikasi sistem presensi lalu menampilkan *interface* tersebut. Setelah itu dilanjutkan dengan pembacaan data latih wajah mahasiswa beserta dengan informasi dari *database* berupa id (nomor urut), nama, NIM, dan waktu terdeteksi dari mahasiswa. Tahap berikutnya yaitu mendapatkan citra data uji melalui *webcam* lalu dilakukan pendeteksian, ekstraksi fitur, normalisasi fitur, dan klasifikasi (pengenalan) wajah mahasiswa dari citra data uji tersebut. Hasil pengenalan tiap wajah dari mahasiswa akan ditampilkan pada aplikasi sistem presensi dan akan dimasukkan ke daftar presensi apabila tombol konfirmasi pada aplikasi sistem presensi ditekan lalu sistem akan kembali ke tahap mendapatkan citra data uji melalui *webcam* lagi. Apabila tombol untuk mereset presensi ditekan maka waktu terdeteksi dari seluruh mahasiswa akan dikosongkan sedangkan apabila tombol untuk menghentikan program ditekan maka aplikasi sistem presensi akan berhenti. Alur dari berjalannya program utama dapat ditunjukkan pada Gambar 5.4 sebagai diagram alir perancangan program utama.



Gambar 5.4 Diagram Alir Perancangan Program Utama

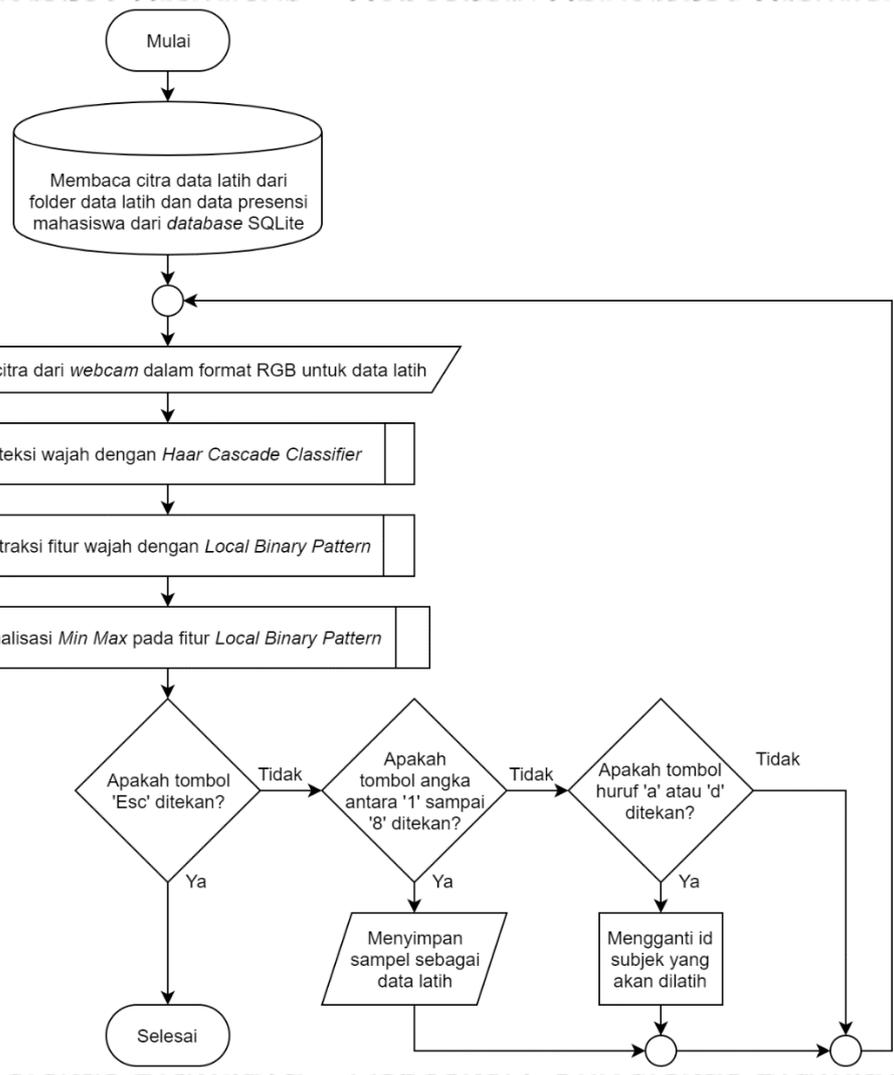


5.1.3.2 Perancangan Database Presensi Mahasiswa

Perancangan *database* presensi mahasiswa dimulai dengan pembuatan *database* SQLite dalam bentuk file dengan nama 'presensi.db'. Didalamnya ditambahkan *table* yang dinamai 'mahasiswa'. Dalam *table* tersebut, terdapat empat kunci yaitu id, NIM, Nama, dan waktu terdeteksi mahasiswa. Id pada *database* dimulai dari 1 dan merupakan kunci primer dari *database* sehingga tiap mahasiswa hanya memiliki satu id dalam *database*.

5.1.3.3 Perancangan Program Pelatihan Wajah Mahasiswa

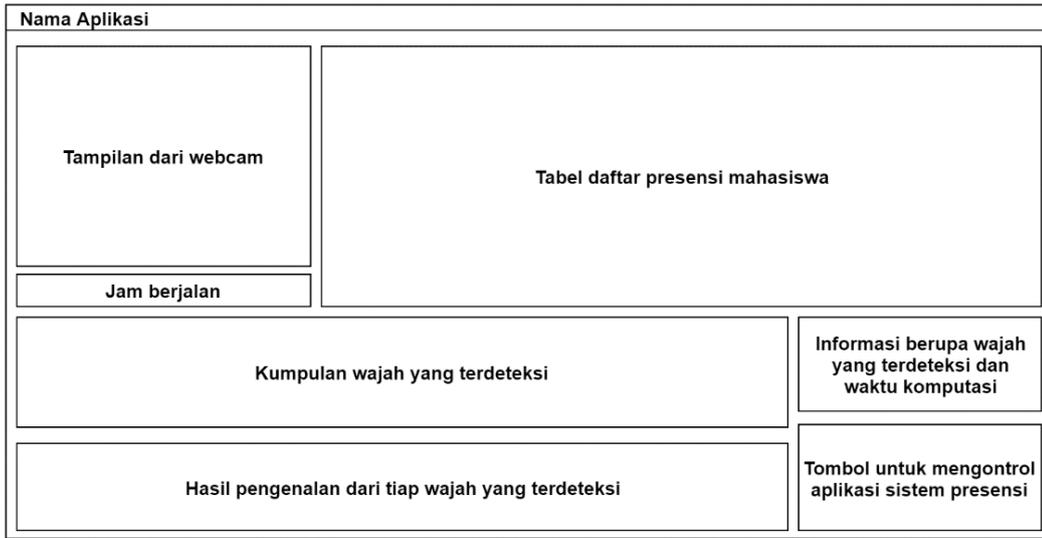
Perancangan program pelatihan wajah mahasiswa dimulai dengan inialisasi variabel untuk pemrosesan citra dilanjutkan dengan pembacaan citra data latih (apabila ada) dan juga mendapatkan data mahasiswa dari *database* SQLite. Lalu dilakukan pembacaan citra yang dihasilkan dari *webcam* yang akan diproses agar menjadi data latih. Dari citra tersebut dilakukan pendeteksian dan ekstraksi fitur dari wajah yang terdeteksi setelah itu dilakukan penyimpanan citra wajah yang telah diekstraksi fiturnya menggunakan *Local Binary Pattern* dan dinormalisasi apabila pengguna menekan tombol angka antara satu sampai delapan. Sedangkan tombol a atau d digunakan untuk mengganti id mahasiswa yang akan ditampilkan pada *window* pelatihan wajah mahasiswa. Setelah dilakukan penyimpanan data latih akan dilakukan pengenalan wajah dan hasilnya akan ditampilkan pada *window* pelatihan wajah. Alur dari berjalannya program pelatihan wajah mahasiswa dapat ditunjukkan pada Gambar 5.5 sebagai diagram alir perancangan program pelatihan wajah mahasiswa.



Gambar 5.5 Diagram Alir Perancangan Program Pelatihan Wajah Mahasiswa

5.1.3.4 Perancangan Tampilan Aplikasi Sistem Presensi

Perancangan tampilan aplikasi sistem presensi dilakukan sehingga sistem dapat menampilkan informasi-informasi penting dan juga tombol kontrol dalam desain yang tertata rapi pada saat sistem dijalankan. Dengan demikian pengguna dapat memahami input, proses, dan output yang dihasilkan oleh sistem serta dapat melakukan kontrol atas berjalannya sistem presensi. Untuk desain rancangan tampilan dari aplikasi sistem presensi dapat dilihat pada Gambar 5.6.

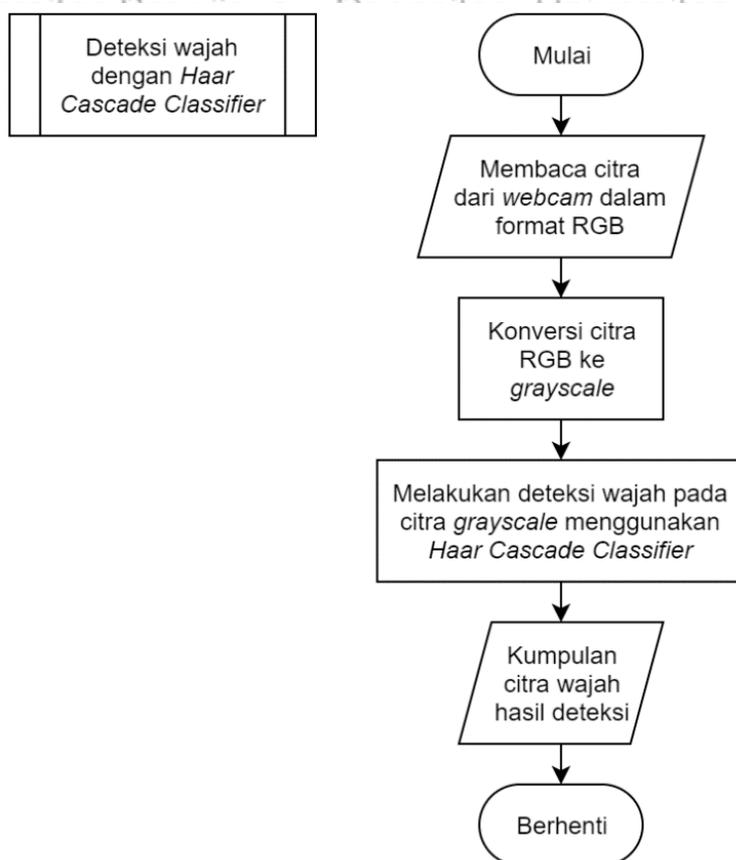


Gambar 5.6 Rancangan Desain Tampilan Aplikasi Sistem Presensi



5.1.3.5 Perancangan Deteksi Wajah Mahasiswa Menggunakan Metode *Haar Cascade Classifier*

Perancangan deteksi wajah mahasiswa dimulai dengan tahap pembacaan citra dengan format RGB yang dihasilkan oleh *webcam* lalu dikonversi kedalam format *grayscale*. Setelah itu dilakukan deteksi wajah pada citra *grayscale* tersebut sehingga dihasilkan kumpulan citra yang berisi wajah dari mahasiswa dalam format *grayscale* lalu dilanjutkan dengan pencatatan waktu pendeteksian wajah yang disebut dengan waktu terdeteksinya mahasiswa yang nantinya akan dimasukkan kedalam daftar presensi. Diagram alir dari perancangan deteksi wajah mahasiswa menggunakan metode *Haar Cascade Classifier* ditunjukkan pada Gambar 5.7 sebagai berikut:



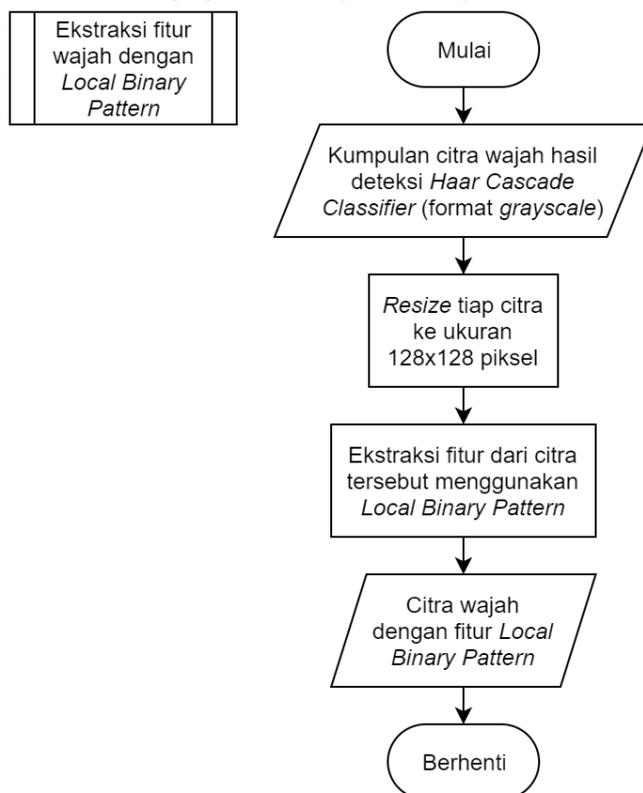
Gambar 5.7 Diagram Alir Deteksi Wajah Mahasiswa menggunakan Metode *Haar Cascade Classifier*

5.1.3.6 Perancangan Ekstraksi Fitur Wajah Mahasiswa Menggunakan Metode *Local Binary Pattern*

Perancangan ekstraksi fitur wajah mahasiswa dimulai dengan tahap pembacaan kumpulan citra wajah berformat *grayscale* hasil deteksi wajah menggunakan metode *Haar Cascade Classifier*. Karena kumpulan citra wajah tersebut biasanya memiliki ukuran yang berbeda, dilakukan pengubahan ukuran



(*resize*) menjadi 128 x 128 piksel sehingga semua citra wajah yang akan dikenali memiliki ukuran yang sama. Tahapan selanjutnya yaitu melakukan ekstraksi fitur dari kumpulan citra wajah yang terdeteksi. Dari tahapan-tahapan tersebut dihasilkan kumpulan citra wajah dalam bentuk fitur *Local Binary Pattern*. Diagram alir dari perancangan ekstraksi fitur wajah mahasiswa menggunakan metode *Local Binary Pattern* ditunjukkan pada Gambar 5.8 sebagai berikut:

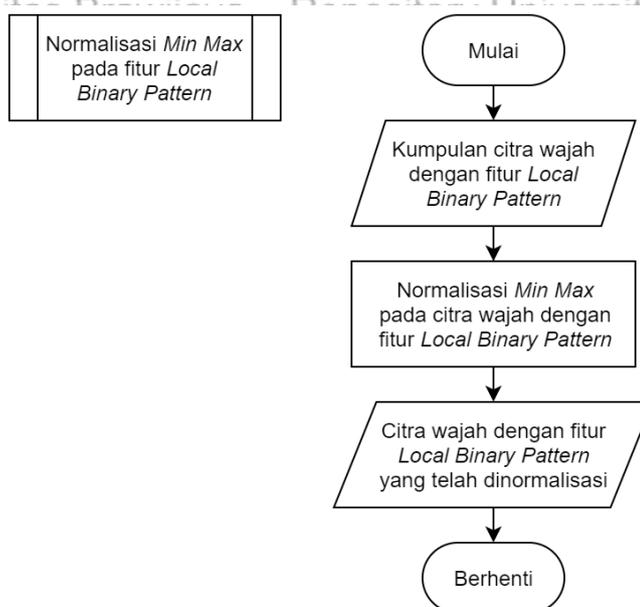


Gambar 5.8 Diagram Alir Ekstraksi Fitur Wajah Mahasiswa menggunakan Metode *Local Binary Pattern*



5.1.3.7 Perancangan Normalisasi *Min Max* pada Fitur *Local Binary Pattern* Wajah Mahasiswa

Perancangan normalisasi *min max* pada fitur *Local Binary Pattern* wajah mahasiswa dimulai dengan tahap pembacaan citra wajah yang sudah diekstraksi menggunakan *Local Binary Pattern*. Setiap citra tersebut akan dilakukan normalisasi sehingga dihasilkan fitur *Local Binary Pattern* yang sudah dinormalisasi menggunakan *min max*. Diagram alir dari perancangan normalisasi *min max* pada fitur *Local Binary Pattern* wajah mahasiswa ditunjukkan pada Gambar 5.9 sebagai berikut:



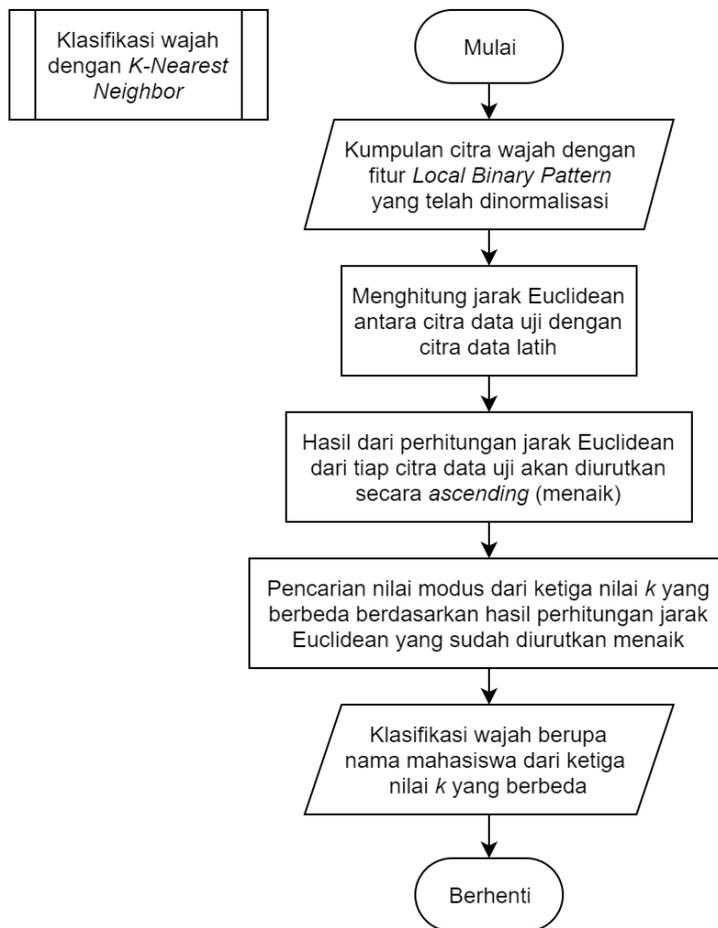
Gambar 5.9 Diagram Alir Normalisasi *Min Max* pada Fitur *Local Binary Pattern* Wajah Mahasiswa

5.1.3.8 Perancangan Klasifikasi Wajah Mahasiswa Menggunakan Metode *K-Nearest Neighbor*

Perancangan klasifikasi wajah mahasiswa dimulai dengan tahap pembacaan kumpulan citra wajah mahasiswa dalam bentuk fitur *Local Binary Pattern* yang telah dinormalisasi sebagai kumpulan data uji. Setelah itu dilakukan penghitungan jarak Euclidean antara tiap data uji dengan tiap data latih yang ada. Data latih tersebut merupakan data wajah mahasiswa yang sudah disimpan oleh program pelatihan wajah yang nantinya digunakan untuk klasifikasi menggunakan metode *K-Nearest Neighbor*. Hasil dari penghitungan jarak Euclidean antara tiap data uji dengan tiap data latih akan diurutkan secara *ascending* (menaik) sehingga didapatkan jarak Euclidean yang dimulai dari nilai terkecil hingga nilai terbesar. Untuk klasifikasi *K-Nearest Neighbor* dengan nilai $k = 3$ dilakukan dengan cara mencari nilai modus atau nilai yang paling sering muncul dari 3 (sesuai dengan nilai k) data latih yang memiliki jarak Euclidean terkecil pada data uji begitu juga dengan nilai $k = 5$ dan $k = 7$. Nilai modus tersebut adalah nilai kelas berupa nama mahasiswa dari data latih yang sering muncul berdasarkan jumlah nilai k -nya.



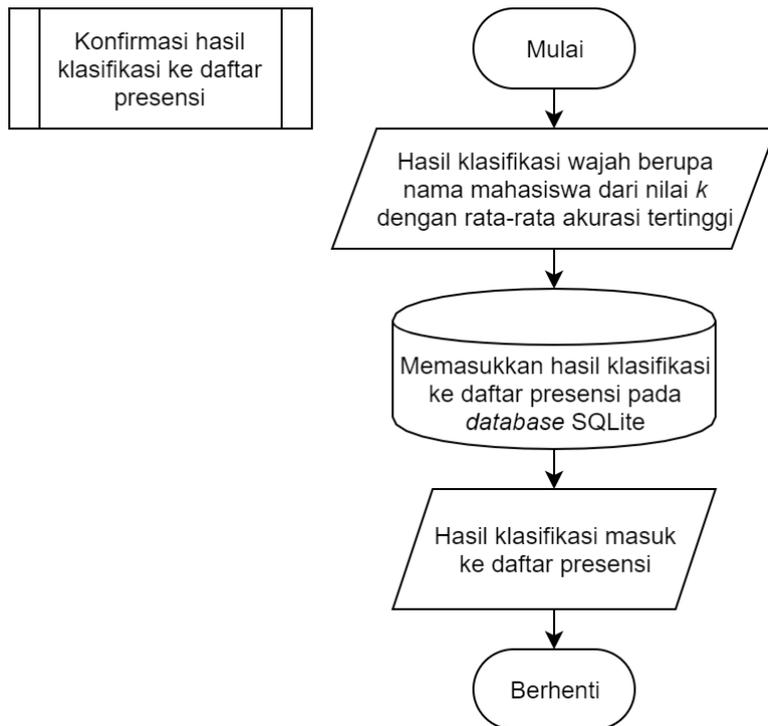
Didapatkan nama-nama mahasiswa sebagai hasil pengenalan dari tiap wajah mahasiswa. Diagram alir dari perancangan klasifikasi wajah mahasiswa menggunakan metode *K-Nearest Neighbor* ditunjukkan pada Gambar 5.10 sebagai berikut:



Gambar 5.10 Diagram Alir Klasifikasi Wajah Mahasiswa menggunakan Metode *K-Nearest Neighbor*

5.1.3.9 Perancangan Konfirmasi Hasil Klasifikasi Wajah Mahasiswa ke Daftar Presensi

Perancangan proses konfirmasi hasil pengenalan wajah mahasiswa dimulai dengan tahap pembacaan nama-nama mahasiswa yang dihasilkan oleh proses pengenalan wajah dan pembacaan waktu terdeteksinya mahasiswa yang dicatat saat proses deteksi wajah. Setelah itu dilakukan pengesetan waktu terdeteksi mahasiswa yang sudah dikenali wajahnya kedalam daftar presensi dengan cara mencari data *record* dari *database* SQLite dengan nama mahasiswa yang telah dikenali setelah itu dilakukan pengesetan waktu terdeteksinya sesuai dengan waktu yang dicatat pada saat wajah mahasiswa terdeteksi oleh sistem. Diagram alir dari proses konfirmasi hasil pengenalan wajah mahasiswa ke daftar presensi ditunjukkan pada Gambar 5.11 sebagai berikut:



Gambar 5.11 Diagram Alir Konfirmasi Pengenalan Wajah Mahasiswa ke Daftar Presensi

5.2 Implementasi Sistem

Pada tahap implementasi sistem ini, akan dibahas tiga macam dari implementasi sistem yang yaitu implementasi prototipe sistem, implementasi perangkat keras sistem, dan implementasi perangkat lunak sistem.

5.2.1 Implementasi Prototipe Sistem

Implementasi prototipe pada sistem ini membutuhkan jarak dan sudut pandang *webcam* yang pas sehingga wajah dari para mahasiswa terlihat dengan jelas. Hal tersebut dapat dicapai dengan cara meletakkan *webcam* diatas *holder* dengan tinggi 75 cm yang ditancapkan pada meja setinggi 75 cm sehingga tinggi dari lantai ke *webcam* adalah 150 cm. Untuk jarak terdekat antara baris pertama yang ditempati oleh mahasiswa dengan meja yang digunakan untuk implementasi prototipe sistem adalah 200 cm, sedangkan selisih antar baris pertama dengan baris kedua yaitu 180 cm. Gambar 5.12 merupakan foto hasil dari implementasi prototipe sistem.



Gambar 5.12 Implementasi Prototipe Sistem

Mini PC diletakkan diatas meja bersama dengan monitor dan *mouse*. *Holder* ditancapkan pada sisi depan meja dan diposisikan lurus keatas. Sedangkan *webcam* diapit oleh penjepit pada bagian atas *holder* dan dihadapkan ke tempat duduk mahasiswa.

5.2.2 Implementasi Perangkat Keras

Pada Gambar 5.13 ditunjukkan implementasi perangkat keras dari sistem. *Webcam* sebagai input perangkat berperan untuk menghasilkan input berupa citra digital yang dipasang diatas *holder* sehingga memiliki sudut pandang yang lebih baik serta terhubung ke *Mini PC* sehingga citra yang dihasilkannya akan diproses oleh *Mini PC* tersebut. Aplikasi sistem presensi berjalan pada *Mini PC* sehingga tampilan dari aplikasi tersebut akan ditampilkan melalui Monitor. *Mouse* digunakan untuk mempermudah pengguna dalam mengendalikan aplikasi sistem presensi.



Gambar 5.13 Implementasi Perangkat Keras



5.2.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak adalah tahapan implementasi dari kode program atau algoritme dari sistem. Penulisan kode program diimplementasikan menggunakan bahasa pemrograman Python 3. Modul-modul lainnya pada bahasa pemrograman Python 3 juga dibutuhkan untuk mendukung sistem dalam pembuatan aplikasi sistem presensi seperti OpenCV, NumPy, SQLite, dan PyQt5.

5.2.3.1 Implementasi Program Utama

Implementasi program utama dari sistem diawal dengan melakukan inisialisasi *interface* dari aplikasi sistem presensi lalu menampilkan *interface* tersebut. Setelah itu dilanjutkan dengan pembacaan data latih wajah mahasiswa beserta dengan informasi dari *database* berupa id (nomor urut), nama, NIM, dan waktu terdeteksi dari mahasiswa. Tahap berikutnya yaitu mendapatkan citra data uji melalui *webcam* lalu dilakukan pendeteksian, ekstraksi fitur, normalisasi fitur dan klasifikasi (pengenalan) wajah mahasiswa dari citra data uji tersebut. Hasil pengenalan tiap wajah dari mahasiswa akan ditampilkan pada aplikasi sistem presensi dan akan dimasukkan ke daftar presensi apabila tombol konfirmasi pada aplikasi sistem presensi ditekan lalu sistem akan kembali ke tahap mendapatkan citra data uji melalui *webcam* lagi. Apabila tombol untuk mereset presensi ditekan maka waktu terdeteksi dari seluruh mahasiswa akan dikosongkan sedangkan apabila tombol untuk menghentikan program ditekan maka aplikasi sistem presensi akan berhenti. Kode program program utama dapat dilihat pada Tabel 5.1 sedangkan untuk penjelasan kode program perbarisnya ada pada Tabel 5.2.

Tabel 5.1 Kode Program Utama

Baris	Kode Program
1	<code>n_subjek = 8</code>
2	<code>n_sampel = 8</code>
3	<code>n_data_latih = n_subjek*n_sampel</code>
4	<code>sampel_lbp_normal = np.zeros((128,128,n_data_latih), np.uint8)</code>
5	<code>distance_euclidean = [[0, "-", "-", 0, 0] for i in range(n_data_latih)]</code>
6	
7	<code>index_wajah = 0</code>
8	<code>for i in range(n_subjek):</code>
9	<code>for j in range(n_sampel):</code>
10	<code>if (os.path.exists('data_sampel/mhs_'+str(i)+'_'+str(j)+'_l bp.jpg')) == True:</code>
11	<code>sampel_lbp_normal[:, :, index_wajah] = cv2.imread('data_sampel/mhs_'+str(i)+'_'+str(j)+'_lbp.jpg', , cv2.IMREAD_UNCHANGED) # membaca sampel bertipe grayscale uint8</code>



```

12         else:
13             sampel_lbp_normal[:, :, index_wajah] =
14             np.zeros((128, 128), dtype=np.uint8)
15             index_wajah += 1
16         database_sqlite = 'database/presensi.db'
17         try:
18             sql_con = sqlite3.connect(database_sqlite)
19             cursor = sql_con.cursor()
20             for i in range(0, 8):
21                 cursor.execute("SELECT * FROM mahasiswa WHERE id = ?",
22                 (i+1,))
23                 hasil = cursor.fetchone()
24                 nama = hasil[2]
25                 nim = hasil[1]
26                 id_subjek = hasil[0]
27                 for j in range(0+(i*8), ((i+1)*8)):
28                     distance_euclidean[j][1] = nama
29                     distance_euclidean[j][2] = nim
30                     distance_euclidean[j][3] = id_subjek
31                     distance_euclidean[j][4] = j%8
32                 print("Koneksi SQLite berhasil!")
33             except sqlite3.Error as error:
34                 print("Koneksi SQLite error!")
35         self.cap = cv2.VideoCapture(0)
36         ret, image = self.cap.read()
37
38         gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
39         faces = face_cascade.detectMultiScale(gray, 1.3, 5)
40
41         for f in faces:
42             (x, y, w, h) = f
43             roi_gray = gray[y:y+h, x:x+w]
44             roi_gray = cv2.resize(roi_gray, (128, 128),
45             cv2.INTER_LINEAR)
46             roi_lbp_norm = ekstraksiFiturLBP(roi_gray)
47             for k in range(n_data_latih):

```



```

48     b = sampel_lbp_normal[:, :, k]
49     a_min_b = np.asarray(a-b)
50     distance_euclidean[k][0] = np.sqrt(np.sum(a_min_b**2))
51
52     sorted_distance_euclidean =
53     copy.copy(distance_euclidean)
54     sorted_distance_euclidean.sort(key =
55     operator.itemgetter(0))
56
57     try:
58         nama_k_3 =
59         statistics.mode([sorted_distance_euclidean[0][1],
60         sorted_distance_euclidean[1][1],
61         sorted_distance_euclidean[2][1],
62         sorted_distance_euclidean[3][1],
63         sorted_distance_euclidean[4][1]])
64         nama_k_5 =
65         statistics.mode([sorted_distance_euclidean[0][1],
66         sorted_distance_euclidean[1][1],
67         sorted_distance_euclidean[2][1],
68         sorted_distance_euclidean[3][1],
69         sorted_distance_euclidean[4][1],
70         sorted_distance_euclidean[5][1],
71         sorted_distance_euclidean[6][1]])
72     except:
73         pass

```

Tabel 5.2 Penjelasan Kode Program Utama

Baris	Keterangan Kode Program
1	Pendefinisian jumlah subjek
2	Pendefinisian jumlah sampel
3	Pendefinisian jumlah data latihan
4	Deklarasi variabel untuk menampung data latihan lbp normal
5	Pendefinisian jarak euclidean
6	



7	Pendefinisian indeks wajah
8	Loop for sebanyak jumlah subjek
9	Loop for sebanyak jumlah sampel
10	Mengecek apabila data latih ditemukan
11	Membaca data latih
12	Seleksi else
13	Mengeset data latih dengan warna putih.
14	Menaikkan indeks wajah
15	
16	Definisi lokasi database SQLite
17	Blok try
18	Melakukan koneksi ke database SQLite
19	Pendefinisian cursor database
20	Loop for sebanyak 8 kali
21	Mengeksekusi pencarian mahasiswa berdasarkan id-nya
22	Membaca hasil pencarian
23	Mendapatkan nama mahasiswa
24	Mendapatkan NIM mahasiswa
25	Mendapatkan id subjek
26	Loop sebanyak jumlah data latih
27	Menambahkan data nama ke list euclidean distance
28	Menambahkan data NIM ke list euclidean distance
29	Menambahkan data id subjek ke list euclidean distance
30	Menambahkan data id sampel ke list euclidean distance
31	Mencetak teks
32	Blok except
33	Mencetak teks
34	
35	Menyeting indeks kamera
36	Membaca citra dari webcam
37	
38	Konversi citra webcam ke grayscale
39	Mendeteksi wajah pada citra grayscale
40	
41	Loop untuk setiap wajah yang terdeteksi
42	Mendapatkan koordinat wajah yang terdeteksi
43	Mendapatkan citra wajah yang terdeteksi
44	Mereshize citra wajah ke 128 x 128 piksel



45	Melakukan ekstraksi fitur pada citra wajah
46	Loop for sebanyak jumlah data latih
47	Inisialisasi data uji
48	Inisialisasi data latih
49-50	Melakukan perhitungan jarak euclidean antara data uji dengan data latih
51	
52	Penyalinan jarak Euclidean saat awal inisialisasi
53	Pengurutan jarak Euclidean secara menaik (ascending)
54	
55	Blok try
56	Variabel nama hasil klasifikasi nilai $k = 3$ dan pencarian nilai modus dari jarak Euclidean indeks 0
57	Pencarian nilai modus dari jarak Euclidean indeks 1
58	Pencarian nilai modus dari jarak Euclidean indeks 2
59	Variabel nama hasil klasifikasi nilai $k = 5$ dan pencarian nilai modus dari jarak Euclidean indeks 0
60	Pencarian nilai modus dari jarak Euclidean indeks 1
61	Pencarian nilai modus dari jarak Euclidean indeks 2
62	Pencarian nilai modus dari jarak Euclidean indeks 3
63	Pencarian nilai modus dari jarak Euclidean indeks 4
64	Variabel nama hasil klasifikasi nilai $k = 7$ dan pencarian nilai modus dari jarak Euclidean indeks 0
65	Pencarian nilai modus dari jarak Euclidean indeks 1
66	Pencarian nilai modus dari jarak Euclidean indeks 2
67	Pencarian nilai modus dari jarak Euclidean indeks 3
68	Pencarian nilai modus dari jarak Euclidean indeks 4
69	Pencarian nilai modus dari jarak Euclidean indeks 5
70	Pencarian nilai modus dari jarak Euclidean indeks 6
71	Blok except
72	Melewati blok

5.2.3.2 Implementasi Database Presensi Mahasiswa

Hasil implementasi berupa struktur dari database tersebut yaitu ditambahkan satu *table* dengan nama 'mahasiswa' dengan empat macam kunci, yaitu id (angka), nim (teks), nama (teks), dan waktu_terdeteksi (teks) yang ditunjukkan pada Gambar 5.14 sebagai berikut:



Name	Type
Tables (1)	
mahasiswa	
id	INTEGER
nim	TEXT
nama	TEXT
waktu_terdeteksi	TEXT

Gambar 5.14 Struktur *Database* Sistem Presensi Mahasiswa

Record yang disimpan pada database disajikan dalam bentuk tabel seperti pada Gambar 5.15. Apabila pengguna melakukan konfirmasi hasil klasifikasi wajah mahasiswa ke dalam daftar presensi, maka waktu terdeteksi dari mahasiswa yang awalnya kosong (NULL) akan diperbarui dengan teks berisi waktu dengan format Tahun-Bulan-Tanggal Jam-Menit-Detik.

	id	nim	nama	waktu_terdeteksi
1	1	16515030011...	Ilham S	2020-03-09 10:05:10
2	2	16515030111...	Renita L S	NULL
3	3	16515030711...	Andika B	NULL
4	4	16515030111...	Leina A Z	NULL
5	5	16515030711...	Igo V F	NULL
6	6	16515030711...	Vedric A S	NULL
7	7	16515030011...	Rahma T P	NULL
8	8	16515070711...	Gani K W	NULL

Gambar 5.15 Tampilan dari Seluruh *Record* dalam *table* 'mahasiswa'

5.2.3.3 Implementasi Program Pelatihan Wajah Mahasiswa

Program pelatihan wajah mahasiswa diawali dengan inisialisasi variabel untuk pemrosesan citra lalu dilakukan pembacaan citra data latih dari folder data latih dan juga data mahasiswa dari *database* SQLite. Selanjutnya dilakukan pembacaan dari *webcam* berisi satu wajah mahasiswa yang akan dilakukan pelatihannya. Dari citra tersebut akan dideteksi dan diekstraksi fiturnya sehingga dihasilkan citra wajah mahasiswa dengan fitur *Local Binary Pattern* yang telah dinormalisasi. Citra tersebut apabila pengguna menekan tombol angka antara satu sampai delapan. Sedangkan tombol *a* atau *d* digunakan untuk mengganti id mahasiswa yang akan ditampilkan pada *window* pelatihan wajah mahasiswa. Setelah dilakukan penyimpanan data latih akan dilakukan pengenalan wajah dan hasilnya akan ditampilkan pada *window* pelatihan wajah. Kode program pelatihan wajah mahasiswa dapat dilihat pada Tabel 5.3 sedangkan untuk penjelasan kode program perbarisnya ada pada Tabel 5.4.



Tabel 5.3 Kode Program Pelatihan Wajah Mahasiswa

Baris	Kode Program
1	<code>n_subjek = 8</code>
2	<code>n_sampel = 8</code>
3	<code>n_data_latih = n_subjek*n_sampel</code>
4	<code>sampel_lbp_normal = np.zeros((128,128,n_data_latih), np.uint8)</code>
5	<code>distance_euclidean = [[0, "-", "-", 0, 0] for _ in range(n_data_latih)]</code>
6	<code>index_wajah = 0</code>
7	<code>for i in range(n_subjek):</code>
8	<code>for j in range(n_sampel):</code>
9	<code>if(os.path.exists('data_sampel/mhs_'+str(i)+'_'+str(j)+'_lbp.jpg')) == True:</code>
10	<code>sampel_lbp_normal[:, :, index_wajah] = cv2.imread('data_sampel/mhs_'+str(i)+'_'+str(j)+'_lbp.jpg' , cv2.IMREAD_UNCHANGED) # membaca sampel bertipe grayscale uint8</code>
11	<code>else:</code>
12	<code>sampel_lbp_normal[:, :, index_wajah] = ~np.zeros((128,128), dtype=np.uint8)</code>
13	<code>index_wajah += 1</code>
14	<code>database_sqlite = 'database/presensi.db'</code>
15	<code>try:</code>
16	<code>sql_con = sqlite3.connect(database_sqlite)</code>
17	<code>cursor = sql_con.cursor()</code>
18	<code>for i in range(0,8):</code>
19	<code>cursor.execute("SELECT * FROM mahasiswa WHERE id = ?", (i+1,))</code>
20	<code>hasil = cursor.fetchone()</code>
21	<code>nama = hasil[2]</code>
22	<code>nim = hasil[1]</code>
23	<code>id_subjek = hasil[0]</code>
24	<code>for j in range(0+(i*8), (i+1)*8):</code>
25	<code>distance_euclidean[j][1] = nama</code>
26	<code>distance_euclidean[j][2] = nim</code>
27	<code>distance_euclidean[j][3] = id_subjek</code>
28	<code>distance_euclidean[j][4] = j%8</code>



```
31 print("Koneksi SQLite berhasil!")
32 except sqlite3.Error as error:
33     print("Koneksi SQLite error!")
34
35 self.cap = cv2.VideoCapture(0)
36 ret, image = self.cap.read()
37
38 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
39 faces = face_cascade.detectMultiScale(gray, 1.3, 5)
40
41 for f in faces:
42     (x,y,w,h) = f
43     roi_gray = gray[y:y+h, x:x+w]
44     roi_gray = cv2.resize(roi_gray, (128,128),
45                           cv2.INTER_LINEAR)
46     roi_lbp_norm = ekstraksiFiturLBP(roi_gray)
47     tombol = cv2.waitKey(1) & 0xFF
48     if tombol >= ord('1') and tombol <= ord('8'):
49         index_tombol = (tombol-48)+1
50         cv2.imwrite('data_sampel/mhs_'+str(id_mhs)+'_'+str(index_t
51 ombol)+'_lbp.jpg', roi_lbp_norm)
52         cv2.imwrite('data_sampel/mhs_'+str(id_mhs)+'_'+str(index_t
53 ombol)+'_gray.jpg', roi_gray)
54         cv2.imwrite('data_sampel/mhs_'+str(id_mhs)+'_'+str(index_t
55 ombol)+'_rgb.jpg', roi_color)
56         sampel_lbp_normal[(id_mhs*8)+index_tombol-1,:] =
57         roi_lbp_norm[:,1]
58     elif tombol == ord('a'):
59         if id_mhs == 0:
60             id_mhs = 7
61         else:
62             id_mhs += 1
63     elif tombol == ord('d'):
64         if id_mhs == 7:
65             id_mhs = 0
66         else:
67             id_mhs -= 1
```



64	elif tombol == 27:
65	loop = False
66	break
67	
68	cursor.close()
69	sql_con.close()
70	cv2.destroyAllWindows()

Tabel 5.4 Penjelasan Kode Program Pelatihan Wajah Mahasiswa

Baris	Keterangan Kode Program
1	Pendefinisian jumlah subjek
2	Pendefinisian jumlah sampel
3	Pendefinisian jumlah data latih
4	Deklarasi variabel untuk menampung data latih lbp normal
5	Pendefinisian jarak euclidean
6	
7	Pendefinisian indeks wajah
8	Loop for sebanyak jumlah subjek
9	Loop for sebanyak jumlah sampel
10	Mengecek apabila data latih ditemukan
11	Membaca data latih
12	Seleksi else
13	Mengeset data latih dengan warna putih
14	Menaikkan indeks wajah
15	
16	Definisi lokasi database SQLite
17	Blok try
18	Melakukan koneksi ke database SQLite
19	Pendefinisian cursor database
20	Loop for sebanyak 8 kali
21	Mengeksekusi pencarian mahasiswa berdasarkan id-nya
22	Membaca hasil pencarian
23	Mendapatkan nama mahasiswa
24	Mendapatkan NIM mahasiswa
25	Mendapatkan id subjek
26	Loop sebanyak jumlah data latih
27	Menambahkan data nama ke list euclidean distance
28	Menambahkan data NIM ke list euclidean distance



```

29 Menambahkan data id subjek ke list euclidean distance
30 Menambahkan data id sampel ke list euclidean distance
31 Mencetak teks
32 Blok except
33 Mencetak teks
34
35 Menyeting indeks kamera
36 Membaca citra dari webcam
37
38 Konversi citra webcam ke grayscale
39 Mendeteksi wajah pada citra grayscale
40
41 Loop untuk setiap wajah yang terdeteksi
42 Mendapatkan koordinat wajah yang terdeteksi
43 Mendapatkan citra wajah yang terdeteksi
44 Meresize citra wajah ke 128 x 128 piksel
45 Melakukan ekstraksi fitur pada citra wajah
46
47 Membaca tombol yang ditekan
48 Apabila tombol yang ditekan adalah angka satu sampai
delapan
49 Mengubah nilai tombol menjadi nilai indeks data latih
50 Menyimpan data latih lbp yang telah dinormalisasi
51 Menyimpan data latih berformat grayscale
52 Menyimpan data latih berformat BGR
53 Memperbarui variabel data latih
54 Apabila tombol a yang ditekan
55 Apabila id mahasiswa adalah 0
56 Id mahasiswa = 7
57 Seleksi else
58 Menurunkan id mahasiswa
59 Apabila tombol d ditekan
60 Apabila id mahasiswa adalah 7
61 Id mahasiswa =
62 Seleksi else
63 Menaikkan id mahasiswa
64 Apabila tombol 'Esc' ditekan
65 Menghentikan loop
66 Menghentikan loop while

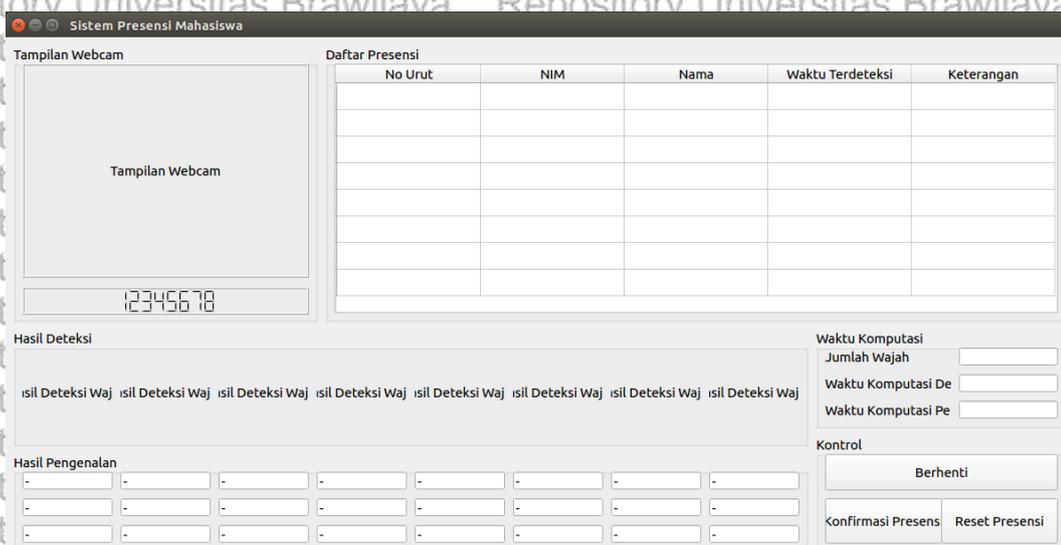
```



67	
68	Menutup koneksi ke database SQLite
69	Menutup kursor database SQLite
70	Menutup semua window

5.2.3.4 Implementasi Tampilan Aplikasi Sistem Presensi

Implementasi tampilan dari aplikasi sistem presensi dibuat menggunakan modul PyQt sehingga hasilnya dapat dilihat pada Gambar 5.16. Di sebelah kiri atas terdapat area tampilan webcam sedangkan di sebelah kanan atas disajikan daftar presensi dalam bentuk tabel. Pada bagian kiri bawah akan ditampilkan citra wajah mahasiswa hasil deteksi dan juga nama mahasiswa hasil pengenalan sedangkan pada kanan bawah disajikan informasi untuk jumlah wajah yang terdeteksi, waktu komputasi deteksi wajah dan waktu komputasi pengenalan wajah. Ditambahkan pula tombol untuk mengontrol berjalannya aplikasi sistem presensi, seperti menghentikan jalannya sistem presensi, melakukan konfirmasi presensi, dan mereset daftar presensi.



Gambar 5.16 Implementasi Desain Tampilan Aplikasi Sistem Presensi

5.2.3.5 Implementasi Deteksi Wajah Mahasiswa Menggunakan Metode Haar Cascade Classifier

Deteksi wajah mahasiswa diawali dengan cara mengubah citra yang ditangkap dari webcam dengan format RGB menjadi grayscale. Lalu dilakukan deteksi wajah pada citra dari webcam yang berformat grayscale. Setelah dilakukan deteksi, dicatatlah posisi waktu yang digunakan sebagai waktu terdeteksinya mahasiswa pada sistem presensi. Kode program dari deteksi wajah mahasiswa menggunakan metode Haar Cascade Classifier dapat dilihat pada Tabel 5.5 sedangkan untuk penjelasan kode program perbarisnya ada pada Tabel 5.6.



Tabel 5.5 Kode Program Deteksi Wajah Mahasiswa Menggunakan Metode Haar Cascade Classifier

Baris	Kode Program
1	<code>gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)</code>
2	<code>faces = face_cascade.detectMultiScale(gray, 1.3, 5)</code>
3	<code>waktu_terdeteksi = time.strftime('%Y-%m-%d %H:%M:%S')</code>

Tabel 5.6 Penjelasan Kode Program Deteksi Wajah Mahasiswa Menggunakan Metode Haar Cascade Classifier

Baris	Keterangan Kode Program
1	Mengubah format frame dari BGR menjadi <i>grayscale</i>
2	Melakukan deteksi wajah menggunakan <i>Haar Cascade Classifier</i>
3	Mencatat waktu terdeteksi wajah mahasiswa

5.2.3.6 Implementasi Ekstraksi Fitur Wajah Mahasiswa Menggunakan Metode *Local Binary Pattern*

Ekstraksi fitur pada wajah diawali dengan pendefinisian tiga fungsi utama yaitu fungsi untuk menghasilkan nilai biner dengan perbandingan piksel tengah dengan piksel disekitarnya, fungsi untuk menghasilkan nilai desimal dari nilai biner yang dihasilkan, fungsi untuk memproses citra wajah mahasiswa hasil deteksi menjadi citra dengan fitur *Local Binary Pattern* yang dinormalisasi. Kode program dari ekstraksi fitur wajah mahasiswa menggunakan metode *Local Binary Pattern* dapat dilihat pada Tabel 5.7 sedangkan untuk penjelasan kode program perbarisnya ada pada Tabel 5.8.

Tabel 5.7 Kode Program Ekstraksi Fitur Wajah Mahasiswa Menggunakan *Local Binary Pattern*

Baris	Kode Program
1	<code>def getPixelValue(img, tengah, y, x):</code>
2	<code> biner = 0</code>
3	<code> try:</code>
4	<code> if img[y][x] >= tengah:</code>
5	<code> biner = 1</code>
6	<code> except:</code>
7	<code> pass</code>
8	<code> return biner</code>
9	
10	<code>def hitungPixelLBP(img, y, x):</code>



```

11  def hitungPixelLBP(x, y):
12      | 128 | 64 | 32 |
13      |-----|
14      | 1 | 0 | 16 |
15      |-----|
16      | 2 | 4 | 8 |
17      |, , , |
18      tengah = img[y][x]
19      biner_ar = []
20      biner_ar.append(getPixelValue(img, tengah, y-1, x-1))
21      # atas kiri
22      biner_ar.append(getPixelValue(img, tengah, y-1, x)) #
23      # atas
24      biner_ar.append(getPixelValue(img, tengah, y-1, x+1))
25      # atas kanan
26      biner_ar.append(getPixelValue(img, tengah, y, x+1)) #
27      # kanan
28      biner_ar.append(getPixelValue(img, tengah, y+1, x+1))
29      # bawah kanan
30      biner_ar.append(getPixelValue(img, tengah, y+1, x)) #
31      # bawah
32      biner_ar.append(getPixelValue(img, tengah, y+1, x-1))
33      # bawah kiri
34      biner_ar.append(getPixelValue(img, tengah, y, x-1)) #
35      # kiri
36      power_biner = [128, 64, 32, 16, 8, 4, 2, 1]
37      desimal = 0
38      for i in range(len(biner_ar)):
39          desimal += biner_ar[i] * power_biner[i]
40      return desimal
41
42  def ekstraksiFiturLBP(frame):
43      h = frame.shape[0]
44      w = frame.shape[1]
45      frame_lbp = np.zeros((h, w), np.uint8)
46      for y in range(0, h):
47          for x in range(0, w):
48              frame_lbp[y, x] = hitungPixelLBP(frame, y, x)
49      frame_lbp_norm = cv2.normalize(frame_lbp, 0, alpha=0,
50      beta=255, norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_8UC1)

```



43	return frame_lbp_norm
----	-----------------------

Tabel 5.8 Penjelasan Kode Program Ekstraksi Fitur Wajah Mahasiswa Menggunakan Local Binary Pattern

Baris	Penjelasan Kode Program
1	Fungsi untuk membandingkan antara piksel tengah dengan piksel sekitar
2	Inisialisasi nilai biner baru dengan nilai 0
3	Blok try
4	Apabila disekitarnya lebih besar daripada dari piksel tengah
5	Nilai biner adalah satu
6	Blok except
7	Lewati
8	Mengembalikan nilai biner baru
9	
10	Fungsi untuk menghitung nilai LBP pada satu titik
11	Komentar
12-16	Ilustrasi LBP
17	Komentar
18	Piksel tengah
19	Deklarasi list untuk menampung semua nilai biner
20	Membandingkan piksel tengah dengan piksel atas kiri lalu ditambahkan ke list
21	Membandingkan piksel tengah dengan piksel atas
22	Membandingkan piksel tengah dengan piksel atas kanan
23	Membandingkan piksel tengah dengan piksel kanan
24	Membandingkan piksel tengah dengan piksel bawah kanan
25	Membandingkan piksel tengah dengan piksel bawah
26	Membandingkan piksel tengah dengan piksel bawah kiri
27	Membandingkan piksel tengah dengan piksel kiri
28	
29	Nilai kekuatan dari biner
30	Nilai desimal dari biner
31	Loop untuk konversi biner ke desimal
32	Penghitungan nilai kekuatan biner berdasarkan posisinya
33	Mengembalikan nilai desimal dari biner
34	
35	Fungsi untuk melakukan ekstraksi fitur LBP



36	Membaca tinggi dari citra wajah hasil deteksi
37	Membaca panjang dari citra wajah hasil deteksi
38	Inisialisasi citra LBP
39	Loop berdasarkan tinggi citra wajah
40	Loop berdasarkan tinggi citra panjang
41	Melakukan penghitungan LBP
42	Normalisasi min.max pada citra LBP
43	Mengembalikan citra LBP yang sudah dinormalisasi

5.2.3.7 Implementasi Normalisasi *Min Max* pada Fitur *Local Binary Pattern* Wajah Mahasiswa

Implementasi normalisasi *min max* diawali dengan pembacaan citra wajah dengan fitur *Local Binary Pattern* lalu dinormalisasi. Kode program dari normalisasi *min max* pada fitur wajah mahasiswa dapat dilihat pada Tabel 5.9 sedangkan untuk penjelasan kode program perbarisnya ada pada Tabel 5.10.

Tabel 5.9 Kode Program Normalisasi *Min Max* pada Fitur *Local Binary Pattern* Wajah Mahasiswa

Baris	Kode Program
1	<code>frame_lbp_norm = cv2.normalize(frame_lbp, 0, alpha=0, beta=255, norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_8UC1)</code>

Tabel 5.10 Penjelasan Kode Program Normalisasi *Min Max* pada Fitur *Local Binary Pattern* Wajah Mahasiswa

Baris	Keterangan Kode Program
1	Melakukan normalisasi min max pada citra dengan fitur <i>Local Binary Pattern</i>

5.2.3.8 Implementasi Klasifikasi Wajah Mahasiswa Menggunakan Metode *K-Nearest Neighbor*

Klasifikasi wajah mahasiswa diawali dengan menghitung jarak Euclidean antara data uji dengan tiap data latih. Setelah itu dilakukan pengurutan jarak Euclidean secara *ascending* (menaik) dari yang terkecil ke yang terbesar. Lalu dicari modus (berupa nama yang paling sering muncul) sesuai dengan nilai *k* yang digunakan pada penelitian ini ($k = 3, 5, 7$). Kode program dari klasifikasi wajah mahasiswa menggunakan metode *K-Nearest Neighbor* dapat dilihat pada Tabel 5.11 sedangkan untuk penjelasan kode program perbarisnya ada pada Tabel 5.12.



**Tabel 5.11 Kode Program Klasifikasi Wajah Mahasiswa Menggunakan Metode
K-Nearest Neighbor**

```

Baris Kode Program
1 for k in range(n_data_latih):
2     a = roi_lbp_norm[:, :]
3     b = sampel_lbp_normal[:, :, k]
4     a_min_b = np.asarray(a-b)
5     distance_euclidean[k][0] = np.sqrt(np.sum(a_min_b**2))
6
7     sorted_distance_euclidean = copy.copy(distance_euclidean)
8     sorted_distance_euclidean.sort(key = operator.itemgetter(0))
9     nama_k_3 = 'Tidak dikenal'
10    nama_k_5 = 'Tidak dikenal'
11    nama_k_7 = 'Tidak dikenal'
12
13    try:
14        nama_k_3 =
statistics.mode([sorted_distance_euclidean[0][1],
15                sorted_distance_euclidean[1][1],
16                sorted_distance_euclidean[2][1]])
17        nama_k_5 =
statistics.mode([sorted_distance_euclidean[0][1],
18                sorted_distance_euclidean[1][1],
19                sorted_distance_euclidean[2][1],
20                sorted_distance_euclidean[3][1],
21                sorted_distance_euclidean[4][1]])
22        nama_k_7 =
statistics.mode([sorted_distance_euclidean[0][1],
23                sorted_distance_euclidean[1][1],
24                sorted_distance_euclidean[2][1],
25                sorted_distance_euclidean[3][1],
26                sorted_distance_euclidean[4][1],
27                sorted_distance_euclidean[5][1],
28                sorted_distance_euclidean[6][1]])
29    except:
30        pass
31
32    frame_deteksi_lgr.append(roi_color)
33    nama_pengenalan.append([nama_k_3, nama_k_5, nama_k_7])

```



Tabel 5.12 Penjelasan Kode Program Klasifikasi Wajah Mahasiswa Menggunakan Metode *K-Nearest Neighbor*

Baris	Keterangan Kode Program
1	Loop for sebanyak jumlah dari data latih
2	Inisialisasi data uji
3	Inisialisasi data latih
4-5	Melakukan perhitungan jarak euclidean antara data uji dengan tiap data latih.
6	Penyalinan jarak Euclidean saat awal inisialisasi
7	Pengurutan jarak Euclidean secara menaik (<i>ascending</i>)
8	Inisialisasi nilai awal $k = 3$
9	Inisialisasi nilai awal $k = 5$
10	Inisialisasi nilai awal $k = 7$
11	
12	Blok try
13	Variabel nama hasil klasifikasi nilai $k = 3$ Pencarian nilai modus dari jarak Euclidean indeks 0
14	Pencarian nilai modus dari jarak Euclidean indeks 1
15	Pencarian nilai modus dari jarak Euclidean indeks 2
16	Variabel nama hasil klasifikasi nilai $k = 5$ Pencarian nilai modus dari jarak Euclidean indeks 0
17	Pencarian nilai modus dari jarak Euclidean indeks 1
18	Pencarian nilai modus dari jarak Euclidean indeks 2
19	Pencarian nilai modus dari jarak Euclidean indeks 3
20	Pencarian nilai modus dari jarak Euclidean indeks 4
21	Variabel nama hasil klasifikasi nilai $k = 7$ Pencarian nilai modus dari jarak Euclidean indeks 0
22	Pencarian nilai modus dari jarak Euclidean indeks 1
23	Pencarian nilai modus dari jarak Euclidean indeks 2
24	Pencarian nilai modus dari jarak Euclidean indeks 3
25	Pencarian nilai modus dari jarak Euclidean indeks 4
26	Pencarian nilai modus dari jarak Euclidean indeks 5
27	Pencarian nilai modus dari jarak Euclidean indeks 6
28	Blok except
29	Melewati bagian except
30	
31	Menampilkan citra rgb hasil deteksi wajah ke aplikasi
32	Menambahkan hasil klasifikasi tiap k ke nama hasil pengenalan



5.2.3.9 Implementasi Konfirmasi Hasil Klasifikasi Wajah Mahasiswa ke Daftar Presensi

Proses konfirmasi hasil pengenalan wajah mahasiswa ke daftar presensi diawali dengan membaca kumpulan nama-nama mahasiswa hasil dari pengenalan wajah serta membaca waktu terdeteksinya yang didapat saat proses pendeteksian wajah. Lalu melakukan koneksi ke *database* SQLite dan melakukan perubahan waktu terdeteksi dari mahasiswa didalamnya. Semua mahasiswa yang terdeteksi dan dikenali dengan benar oleh sistem akan di-*update* data waktu komputasinya sehingga waktu terdeteksi mereka masuk ke dalam daftar presensi. Kode program dari konfirmasi hasil klasifikasi wajah mahasiswa ke daftar presensi dapat dilihat pada Tabel 5.13 sedangkan untuk penjelasan kode program perbarisnya ada pada Tabel 5.14.

Tabel 5.13 Kode Program Konfirmasi Pengenalan Wajah Mahasiswa ke Daftar Presensi

Baris	Kode Program
1	daftarpresensi = []
2	nama_pengenalan = []
3	
4	def konfirmasiPresensi(self):
5	global nama_pengenalan, waktu_terdeteksi
6	if len(nama_pengenalan) != 0:
7	sql_con = sqlite3.connect(database_sqlite)
8	cursor = sql_con.cursor()
9	for i in range(len(nama_pengenalan)):
10	if (nama_pengenalan[i][2] != '-') and (nama_pengenalan[i][2] != 'Tidak dikenali'): # k=7
11	sql = "UPDATE mahasiswa SET waktu_terdeteksi = ? WHERE nama = ?"
12	val = (waktu_terdeteksi, nama_pengenalan[i][0]) # k=3
13	cursor.execute(sql, val)
14	sql_con.commit()
15	self.updateTabelPresensi()



Tabel 5.14 Penjelasan Kode Program Konfirmasi Pengenalan Wajah Mahasiswa ke Daftar Presensi

Baris	Keterangan Kode Program
1	Inisialisasi daftar presensi
2	Inisialisasi nama pengenalan
3	
4	Pendefinisian fungsi konfirmasi presensi
5	Mengakses variabel global nama pengenalan dan waktu terdeteksi
6	Apabila nama hasil dari pengenalan tidak kosong
7	Melakukan koneksi pada database
8	Deklarasi cursor untuk melakukan perintah pada database
9	Loop untuk setiap nama hasil pengenalan
10	Apabila nama hasil pengenalan merupakan nama mahasiswa
11	Perintah untuk meng-update waktu terdeteksi dari nama mahasiswa yang sudah dikenali
12	Variabel yang digunakan pada perintah SQL
13	Mengeksekusi perintah SQL pada database SQLite
14	Menerapkan perubahan pada database
15	Memperbarui tampilan daftar presensi pada aplikasi



BAB 6 PENGUJIAN DAN ANALISIS

Pengujian dilakukan untuk mencari tahu jawaban dari rumusan masalah yang telah dibuat. Data yang akan diujikan berupa rata-rata akurasi dan waktu komputasi dari pendeteksian maupun pengenalan wajah. Untuk mencari akurasi dari suatu prediksi dapat menggunakan rumus akurasi yang ditunjukkan pada Persamaan 6.1 sebagai berikut:

$$\text{Akurasi} = \frac{\text{jumlah data dengan prediksi benar}}{\text{jumlah data total}} \times 100\% \quad (6.1)$$

Sedangkan untuk mencari waktu komputasi dari suatu proses dapat digunakan rumus yang ditunjukkan pada Persamaan 6.2 sebagai berikut:

$$\text{Waktu komputasi} = \text{waktu berakhir} - \text{waktu mulai} \quad (6.2)$$

6.1 Pengujian dan Analisis Rata-rata Akurasi Deteksi Wajah menggunakan *Haar Cascade Classifier*

6.1.1 Tujuan Pengujian

Tujuan dari pengujian ini adalah untuk memastikan bahwa sistem mampu mendeteksi setiap wajah dari citra yang ditangkap oleh *webcam* menggunakan metode *Haar Cascade Classifier*.

6.1.2 Prosedur Pengujian

Prosedur dari pengujian ini memiliki beberapa tahap yang harus dipenuhi yaitu:

1. *Webcam* Logitech C270, *mouse*, dan monitor yang terhubung ke *Mini PC*, lalu ditempatkan diatas meja.
2. *Holder* dilekatkan ke meja dan ditegakkan ke atas lalu *webcam* ditaruh di bagian atas *holder* dan dihadapkan ke arah mahasiswa dengan ketentuan wajah mahasiswa baik di baris kedua maupun pertama harus tetap terlihat dengan jelas (tidak tertutupi oleh objek apapun).
3. Melakukan penempatan posisi duduk mahasiswa dimulai dari baris pertama sampai baris kedua. Pada tiap baris akan dilakukan delapan penempatan posisi duduk mahasiswa yang berbeda, dimulai dari pengujian pada satu mahasiswa lalu dilakukan penambahan jumlah mahasiswa secara bertahap satu-persatu sampai delapan mahasiswa.
4. Mengamati tampilan aplikasi sistem presensi yang berisi informasi berupa hasil citra yang ditangkap oleh *webcam*, kumpulan wajah yang terdeteksi dan hasil pengenalannya, serta waktu komputasinya.
5. Mencatat hasil pendeteksian wajah yang dihasilkan oleh sistem untuk dihitung rata-rata akurasinya.

6.1.3 Hasil Pengujian

Pengujian dilakukan berdasarkan jumlah wajah aktual atau jumlah mahasiswa yang ada dalam satu *frame*. Dimulai dari satu wajah sampai empat wajah. Pada Gambar 6.1 dan 6.2 akan disajikan tampilan aplikasi saat pengujian pendeteksian wajah sedangkan pada Tabel 6.1 akan disajikan tabel hasil pengujian dari pendeteksian wajah mahasiswa.

Sistem Presensi Mahasiswa

Tampilan Webcam



12:52:12

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi



Waktu Komputasi

Jumlah Wajah 3 subjek

Waktu Komputasi De 24 ms

Waktu Komputasi Pe 273 ms

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

Hasil Pengenalan

Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Renita L S	-	-	-	-	-

Gambar 6.1 Tampilan Aplikasi Sistem Presensi Saat Pengujian Pendeteksian Tiga Wajah Mahasiswa

Sistem Presensi Mahasiswa

Tampilan Webcam



12:55:14

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi



Waktu Komputasi

Jumlah Wajah 8 subjek

Waktu Komputasi De 31 ms

Waktu Komputasi Pe 743 ms

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

Hasil Pengenalan

Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S
Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S
Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S

Gambar 6.2 Tampilan Aplikasi Sistem Presensi Saat Pengujian Pendeteksian Delapan Wajah Mahasiswa



Tabel 6.1 Hasil Pendeteksian Wajah

Percobaan ke- <i>n</i>	Jumlah wajah (Aktual)	Jumlah wajah hasil deteksi		Hasil akurasi deteksi wajah (%)	Rata-rata akurasi deteksi wajah untuk <i>m</i> -jumlah wajah (%)
		Terdeteksi	Tidak terdeteksi		
1		1	0	100	
2		1	0	100	
3	1	1	0	100	100
4		1	0	100	
5		1	0	100	
6		2	0	100	
7		2	0	100	
8	2	2	0	100	100
9		2	0	100	
10		2	0	100	
11		3	0	100	
12		3	0	100	
13	3	3	0	100	100
14		3	0	100	
15		3	0	100	
16		4	0	100	
17		3	1	75	
18	4	3	1	75	85.00
19		4	0	100	
20		3	1	75	



Tabel 6.1 Hasil Pendeteksian Wajah (lanjutan)

21		5	0	100	
22		5	0	100	
23	5	5	0	100	96.00
24		5	0	100	
25		4	1	80	
26		5	1	83.33	
27		5	1	83.33	
28	6	6	0	100	83.33
29		5	1	83.33	
30		4	2	66.67	
31		4	3	57.14	
32		5	2	71.42	
33	7	5	2	71.42	80.00
34		7	0	100	
35		7	0	100	
36		7	1	87.50	
37		7	1	87.50	
38	8	7	1	87.50	90.00
39		8	0	100	
40		7	1	87.50	
Rata-rata akurasi dari seluruh percobaan deteksi wajah					88.88

Rata-rata akurasi dari pendeteksian satu wajah yaitu $\frac{5}{5} \times 100\% = 100\%$,
 untuk dua wajah $\frac{10}{10} \times 100\% = 100\%$, untuk tiga wajah $\frac{15}{15} \times 100\% = 100\%$,
 untuk empat wajah $\frac{17}{20} \times 100\% = 85\%$, untuk lima wajah $\frac{24}{25} \times 100\% = 96\%$,
 untuk enam wajah $\frac{25}{30} \times 100\% = 83.33\%$, untuk tujuh wajah $\frac{28}{35} \times 100\% = 80\%$,
 dan untuk delapan wajah $\frac{36}{40} \times 100\% = 90.00\%$. Sedangkan untuk rata-rata
 akurasi dari seluruh percobaan deteksi wajah adalah $\frac{160}{180} \times 100\% = 88.88\%$.

6.1.4 Analisis Pengujian

Dari 40 percobaan yang telah dilakukan terdapat jumlah wajah aktual dari mahasiswa yaitu 180 wajah sedangkan sistem hanya mampu mendeteksi wajah dengan benar sebanyak 160 wajah sehingga rata-rata akurasi deteksi wajah pada



seluruh percobaan yaitu sebesar 88.88%. Rata-rata akurasi terbaik didapat pada saat pendeteksian untuk satu hingga tiga wajah saja yaitu sebesar 100%. Hal yang dapat memengaruhi keberhasilan deteksi wajah yaitu jarak antara wajah dengan *webcam* yang terlalu jauh, intensitas cahaya yang kurang memadai, posisi wajah yang tidak menghadap ke arah *webcam*, wajah yang hanya terlihat sebagian karena terhalangi objek atau keluar dari *frame*.

6.2 Pengujian dan Analisis Rata-rata Akurasi Pengenalan Wajah menggunakan *Local Binary Pattern* dan *K-Nearest Neighbor*

6.2.1 Tujuan Pengujian

Tujuan dari pengujian ini adalah untuk memastikan bahwa sistem mampu mengenali setiap wajah yang sudah terdeteksi melalui metode *Haar Cascade Classifier*.

6.2.2 Prosedur Pengujian

Prosedur dari pengujian rata-rata akurasi pengenalan wajah ini memiliki beberapa tahap yang harus dipenuhi yaitu:

1. *Webcam* Logitech C270, *mouse*, dan monitor yang terhubung ke *Mini PC*, lalu ditempatkan diatas meja.
2. *Holder* dilekatkan ke meja dan ditegakkan ke atas lalu *webcam* ditaruh di bagian atas *holder* dan dihadapkan ke arah mahasiswa dengan ketentuan wajah mahasiswa baik di baris belakang maupun depan harus tetap terlihat dengan jelas (tidak tertutupi oleh objek apapun).
3. Melakukan penempatan posisi duduk mahasiswa dimulai dari baris pertama sampai baris kedua. Pada tiap baris akan dilakukan delapan penempatan posisi duduk mahasiswa yang berbeda, dimulai dari pengujian pada satu mahasiswa lalu dilakukan penambahan jumlah mahasiswa secara bertahap satu-persatu sampai delapan mahasiswa.
4. Mengamati tampilan aplikasi sistem presensi yang berisi informasi berupa hasil citra yang ditangkap oleh *webcam*, kumpulan wajah yang terdeteksi dan hasil pengenalannya, serta waktu komputasinya.
5. Mencatat hasil pengenalan wajah yang dihasilkan oleh sistem untuk dihitung rata-rata akurasi.

6.2.3 Hasil Pengujian

Pengujian pengenalan wajah mahasiswa menggunakan metode *K-Nearest Neighbor* dengan nilai *k* berturut-turut 3, 5, dan 7 dilakukan secara bersamaan karena dihasilkan jarak Euclidean antara citra wajah LBP data uji dengan citra wajah LBP data latih sehingga didapatkan citra wajah LBP data latih yang memiliki jarak Euclidean yang telah diurutkan secara menaik (*ascending*). Dari jarak Euclidean yang sudah diurutkan tersebut, hasil klasifikasi KNN dengan ketiga nilai *k* yang berbeda tersebut dapat dilakukan dalam waktu yang sama karena data



yang diproses merupakan data yang sama namun hanya berbeda pada rentang modulusnya saja. Pada Gambar 6.3 dan 6.4 akan disajikan tampilan aplikasi saat pengujian pengenalan wajah sedangkan pada Tabel 6.2 akan disajikan tabel hasil pengujian dari pengenalan wajah mahasiswa.

Sistem Presensi Mahasiswa

Tampilan Webcam

12:52:12

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Hasil Pengenalan

Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Renita L S	-	-	-	-	-

Waktu Komputasi

Jumlah Wajah: 3 subjek

Waktu Komputasi De: 24 ms

Waktu Komputasi Pe: 273 ms

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

Gambar 6.3 Tampilan Aplikasi Sistem Presensi Saat Pengujian Pengenalan Tiga Wajah Mahasiswa

Sistem Presensi Mahasiswa

Tampilan Webcam

12:55:14

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Hasil Pengenalan

Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S
Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S
Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S

Waktu Komputasi

Jumlah Wajah: 8 subjek

Waktu Komputasi De: 31 ms

Waktu Komputasi Pe: 743 ms

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

Gambar 6.4 Tampilan Aplikasi Sistem Presensi Saat Pengujian Pengenalan Delapan Wajah Mahasiswa

Tabel 6.2 Hasil Pengenalan Wajah dengan nilai $k = 3$

Percobaan ke- n	Jumlah wajah yang terdeteksi	Jumlah wajah hasil pengenalan dengan nilai $k = 3$		Hasil akurasi pengenalan wajah (%)	Rata-rata akurasi pengenalan untuk m -jumlah wajah (%)
		Dikenali dengan benar	Dikenali namun salah		
1	1	1	0	100	
2	1	1	0	100	
3	1	0	1	0	80
4	1	1	0	100	
5	1	1	0	100	
6	2	2	0	100	
7	2	2	0	100	
8	2	2	0	100	90
9	2	1	1	50	
10	2	2	0	100	
11	3	3	0	100	
12	3	3	0	100	
13	3	3	0	100	93.33
14	3	2	1	66.67	
15	3	3	0	100	
16	4	4	0	100	
17	3	3	0	100	
18	3	3	0	100	94.11
19	4	3	1	75	
20	3	3	0	100	
21	5	4	1	80	
22	5	4	1	80	
23	5	4	1	80	70.83
24	5	3	2	60	
25	4	2	2	50	

Tabel 6.2 Hasil Pengenalan Wajah dengan nilai $k = 3$ (lanjutan)

26	5	2	3	40	
27	5	3	2	60	
28	6	4	2	66.67	60
29	5	3	2	60	
30	4	3	1	75	
31	4	4	0	100	
32	5	4	1	80	
33	5	3	2	60	71.42
34	7	4	3	57.14	
35	7	5	2	71.42	
36	7	6	1	85.71	
37	7	5	2	71.42	
38	7	6	1	85.71	83.33
39	8	7	1	87.5	
40	7	6	1	85.71	
Rata-rata akurasi dari seluruh percobaan pengenalan wajah dengan $k = 3$					78.125

Dari 40 percobaan yang dilakukan untuk pengenalan wajah menggunakan metode *K-Nearest Neighbor* dengan $k = 3$, didapatkan rata-rata akurasi pengenalan untuk satu wajah yaitu $\frac{4}{5} \times 100\% = 80\%$, untuk dua wajah $\frac{9}{10} \times 100\% = 90\%$, untuk tiga wajah $\frac{14}{15} \times 100\% = 93.33\%$, untuk empat wajah $\frac{16}{17} \times 100\% = 94.11\%$, untuk lima wajah $\frac{17}{24} \times 100\% = 70.83\%$, untuk enam wajah $\frac{15}{25} \times 100\% = 60\%$, untuk tujuh wajah $\frac{20}{28} \times 100\% = 71.42\%$, dan untuk delapan wajah yaitu $\frac{18}{19} \times 100\% = 83.33\%$. Sedangkan untuk rata-rata akurasi dari seluruh percobaan pengenalan wajah mahasiswa dengan $k = 3$ yaitu $\frac{125}{160} \times 100\% = 78.125\%$.

Tabel 6.3 Hasil Pengenalan Wajah dengan nilai $k = 5$

Percobaan ke- n	Jumlah wajah yang terdeteksi	Jumlah wajah hasil pengenalan dengan nilai $k = 5$		Hasil akurasi pengenalan wajah (%)	Rata-rata akurasi pengenalan untuk m -jumlah wajah (%)
		Dikenali dengan benar	Dikenali namun salah		
1	1	1	0	100	
2	1	1	0	100	
3	1	0	1	0	60
4	1	1	0	100	
5	1	0	1	0	
6	2	2	0	100	
7	2	2	0	100	
8	2	2	0	100	90
9	2	1	1	50	
10	2	2	0	100	
11	3	3	0	100	
12	3	2	1	66.67	
13	3	3	0	100	86.67
14	3	2	1	66.67	
15	3	3	0	100	
16	4	3	1	75	
17	3	3	0	100	
18	3	3	0	100	88.23
19	4	3	1	75	
20	3	3	0	100	
21	5	4	1	80	
22	5	4	1	80	
23	5	3	2	60	70.83
24	5	3	2	60	
25	4	3	1	75	

Tabel 6.3 Hasil Pengenalan Wajah dengan nilai $k = 5$ (lanjutan)

26	5	2	3	40	
27	5	3	2	60	
28	6	3	3	50	56
29	5	3	2	60	
30	4	3	1	75	
31	4	4	0	100	
32	5	3	2	60	
33	5	3	2	60	67.85
34	7	4	3	57.14	
35	7	5	2	71.42	
36	7	6	1	85.71	
37	7	5	2	71.42	
38	7	6	1	85.71	80.56
39	8	6	2	75	
40	7	6	1	85.71	
Rata-rata akurasi dari seluruh percobaan pengenalan wajah dengan $k = 5$					74.375

Dari 40 percobaan yang dilakukan untuk pengenalan wajah menggunakan metode *K-Nearest Neighbor* dengan $k = 5$, didapatkan rata-rata akurasi pengenalan untuk satu wajah yaitu $\frac{3}{5} \times 100\% = 60\%$, untuk dua wajah $\frac{9}{10} \times 100\% = 90\%$, untuk tiga wajah $\frac{13}{15} \times 100\% = 86.67\%$, untuk empat wajah $\frac{10}{15} \times 100\% = 88.23\%$, untuk lima wajah $\frac{17}{24} \times 100\% = 70.83\%$, untuk enam wajah $\frac{14}{25} \times 100\% = 56\%$, untuk tujuh wajah $\frac{19}{28} \times 100\% = 67.85\%$, dan untuk delapan wajah yaitu $\frac{29}{36} \times 100\% = 80.56\%$. Sedangkan untuk rata-rata akurasi dari seluruh percobaan pengenalan wajah mahasiswa dengan $k = 5$ yaitu $\frac{119}{160} \times 100\% = 74.375\%$.

Tabel 6.4 Hasil Pengenalan Wajah dengan nilai $k = 7$

Percobaan ke- n	Jumlah wajah yang terdeteksi	Jumlah wajah hasil pengenalan dengan nilai $k = 7$		Hasil akurasi pengenalan wajah (%)	Rata-rata akurasi pengenalan untuk m -jumlah wajah (%)
		Dikenali dengan benar	Dikenali namun salah		
1	1	1	0	100	
2	1	0	1	0	
3	1	0	1	0	40
4	1	1	0	100	
5	1	0	1	0	
6	2	1	1	50	
7	2	2	0	100	
8	2	2	0	100	80
9	2	1	1	50	
10	2	2	0	100	
11	3	3	0	100	
12	3	2	1	66.67	
13	3	2	1	66.67	80
14	3	2	1	66.67	
15	3	3	0	100	
16	4	2	2	50	
17	3	3	0	100	
18	3	2	1	66.67	70.58
19	4	2	2	50	
20	3	3	0	100	
21	5	4	1	80	
22	5	3	2	60	
23	5	3	2	60	66.67
24	5	3	2	60	
25	4	3	1	75	

Tabel 6.4 Hasil Pengenalan Wajah dengan nilai $k = 7$ (lanjutan)

26	5	2	3	40	
27	5	2	3	40	
28	6	3	3	50	52
29	5	3	2	60	
30	4	3	1	75	
31	4	4	0	100	
32	5	3	2	60	
33	5	2	3	40	64.28
34	7	4	3	57.14	
35	7	5	2	71.42	
36	7	6	1	85.71	
37	7	5	2	71.42	
38	7	6	1	85.71	77.77
39	8	6	2	75	
40	7	5	2	71.42	
Rata-rata akurasi dari seluruh percobaan pengenalan wajah dengan $k = 7$					68.125

Dari 40 percobaan yang dilakukan untuk pengenalan wajah menggunakan metode *K-Nearest Neighbor* dengan $k = 7$, didapatkan rata-rata akurasi pengenalan untuk satu wajah yaitu $\frac{2}{5} \times 100\% = 40\%$, untuk dua wajah $\frac{8}{10} \times 100\% = 80\%$, untuk tiga wajah $\frac{12}{15} \times 100\% = 80\%$, untuk empat wajah $\frac{10}{12} \times 100\% = 70.58\%$, untuk lima wajah $\frac{16}{24} \times 100\% = 66.67\%$, untuk enam wajah $\frac{12}{25} \times 100\% = 52\%$, untuk tujuh wajah $\frac{18}{28} \times 100\% = 64.28\%$, dan untuk delapan wajah yaitu $\frac{28}{36} \times 100\% = 77.77\%$. Sedangkan untuk rata-rata akurasi dari seluruh percobaan pengenalan wajah mahasiswa dengan $k = 7$ yaitu $\frac{109}{160} \times 100\% = 68.125\%$.

6.2.4 Analisis Pengujian

Rata-rata akurasi sistem dalam melakukan pengenalan wajah dari seluruh percobaan menggunakan nilai $k = 3, 5$, dan 7 secara berturut-turut yaitu 78.125% , 74.375% , dan 68.125% . Sehingga rata-rata akurasi pengenalan wajah terbaik dapat dicapai menggunakan nilai $k = 3$.



6.3 Pengujian dan Analisis Rata-rata Waktu Komputasi Deteksi Wajah menggunakan *Haar Cascade Classifier*

Pengujian dan Analisis dari berbagai percobaan waktu komputasi deteksi untuk mahasiswa dengan jumlah tunggal maupun jamak menggunakan metode *Haar Cascade Classifier*.

6.3.1 Tujuan Pengujian

Tujuan dari pengujian ini yaitu untuk menguji rata-rata waktu komputasi sistem dalam bentuk milisekon saat melakukan proses pendeteksian wajah, diawali dengan pengambilan citra wajah dari *webcam*, sampai dengan hasil pendeteksian wajah.

6.3.2 Prosedur Pengujian

Prosedur dari pengujian rata-rata akurasi pengenalan wajah ini memiliki beberapa tahap yang harus dipenuhi yaitu:

1. *Webcam* Logitech C270, *mouse*, dan monitor yang terhubung ke *Mini PC*, lalu ditempatkan diatas meja.
2. *Holder* dilekatkan ke meja dan ditegakkan ke atas lalu *webcam* ditaruh di bagian atas *holder* dan dihadapkan ke arah mahasiswa dengan ketentuan wajah mahasiswa baik di baris belakang maupun depan harus tetap terlihat dengan jelas (tidak tertutupi oleh objek apapun).
3. Melakukan penempatan posisi duduk mahasiswa dimulai dari baris pertama sampai baris kedua. Pada tiap baris akan dilakukan delapan penempatan posisi duduk mahasiswa yang berbeda, dimulai dari pengujian pada satu mahasiswa lalu dilakukan penambahan jumlah mahasiswa secara bertahap satu-persatu sampai delapan mahasiswa.
4. Mengamati tampilan aplikasi sistem presensi yang berisi informasi berupa hasil citra yang ditangkap oleh *webcam*, kumpulan wajah yang terdeteksi dan hasil pengenalannya, serta waktu komputasinya.
5. Mencatat waktu komputasi sistem dalam melakukan deteksi wajah mahasiswa untuk mencari rata-ratanya.

6.3.3 Hasil Pengujian

Pengujian waktu komputasi deteksi wajah dilakukan untuk mencari rata-rata waktu komputasi sistem dalam melakukan deteksi wajah. Pada Gambar 6.5 dan 6.6 akan disajikan tampilan aplikasi saat pengujian waktu komputasi deteksi wajah sedangkan pada Tabel 6.5 akan ditunjukkan tabel hasil pengujian dari waktu komputasi pendeteksian wajah mahasiswa.



Sistem Presensi Mahasiswa

Tampilan Webcam

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi

Jumlah Wajah 3 subjek

Waktu Komputasi De 24 ms

Waktu Komputasi Pe 273 ms

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

Hasil Pengenalan

Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Renita L S	-	-	-	-	-

Gambar 6.5 Tampilan Aplikasi Sistem Presensi Saat Pengujian Waktu Komputasi Deteksi Tiga Wajah Mahasiswa

Sistem Presensi Mahasiswa

Tampilan Webcam

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi

Jumlah Wajah 8 subjek

Waktu Komputasi De 31 ms

Waktu Komputasi Pe 743 ms

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

Hasil Pengenalan

Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S
Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S
Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S

Gambar 6.6 Tampilan Aplikasi Sistem Presensi Saat Pengujian Waktu Komputasi Deteksi Delapan Wajah Mahasiswa



Tabel 6.5 Hasil Waktu Komputasi Deteksi Wajah

Percobaan ke- n	Jumlah wajah (Aktual)	Waktu komputasi deteksi wajah (ms)	Rata-rata waktu komputasi deteksi untuk m -jumlah wajah (ms)
1		18	
2		18	
3	1	16	16.8
4		16	
5		16	
6		21	
7		20	
8	2	20	21.2
9		23	
10		22	
11		2	
12		27	
13	3	29	25.2
14		23	
15		24	
16		24	
17		24	
18	4	34	26.4
19		24	
20		26	
21		30	
22		26	
23	5	26	27.6
24		30	
25		26	



Tabel 6.5 Hasil Waktu Komputasi Deteksi Wajah (lanjutan)

26		30	
27		27	
28	6	33	28.8
29		28	
30		26	
31		28	
32		37	
33	7	28	32.6
34		35	
35		35	
36		32	
37		30	
38	8	35	35.2
39		31	
40		48	

Dari hasil percobaan tersebut rata-rata waktu komputasi dalam pendeteksian satu wajah yaitu 16.8 ms, untuk dua wajah 21.2 ms, untuk tiga wajah 25.2 ms, untuk empat wajah 26.4 ms, untuk lima wajah 27.6 ms, untuk enam wajah 28.8, untuk tujuh wajah 32.6, dan untuk delapan wajah 35.2. Jadi rata-rata waktu komputasi dari seluruh percobaan deteksi wajah yaitu 26.2 ms.

6.3.4 Analisis Pengujian

Dari 40 percobaan yang telah dilakukan, didapatkan rata-rata waktu komputasi dalam pendeteksian satu hingga delapan wajah berturut-turut yaitu 16.8 ms, 21.2 ms, 25.2 ms, 26.4 ms, 27.6 ms, 28.8 ms, 32.6 ms, dan 35.2 ms. Setiap penambahan satu wajah akan memperlambat waktu komputasi deteksi wajah sebesar 1 milisekon hingga 4 milisekon saja sehingga perubahannya tidak terlalu signifikan.



6.4 Pengujian dan Analisis Rata-rata Waktu Komputasi Pengenalan Wajah menggunakan *Local Binary Pattern* dan *K-Nearest Neighbor*

6.4.1 Tujuan Pengujian

Tujuan dari pengujian ini yaitu untuk menguji rata-rata waktu komputasi sistem dalam melakukan proses pengenalan wajah menggunakan metode *Local Binary Pattern* dan *K-Nearest Neighbor*, diawali dengan pemrosesan citra wajah yang terdeteksi sampai dengan hasil pengenalan tiap wajah.

6.4.2 Prosedur Pengujian

Prosedur dari pengujian rata-rata akurasi pengenalan wajah ini memiliki beberapa tahap yang harus dipenuhi yaitu:

1. *Webcam* Logitech C270, *mouse*, dan monitor yang terhubung ke *Mini PC*, lalu ditempatkan diatas meja.
2. *Holder* dilekatkan ke meja dan ditegakkan ke atas lalu *webcam* ditaruh di bagian atas *holder* dan dihadapkan ke arah mahasiswa dengan ketentuan wajah mahasiswa baik di baris belakang maupun depan harus tetap terlihat dengan jelas (tidak tertutupi oleh objek apapun).
3. Melakukan penempatan posisi duduk mahasiswa dimulai dari baris pertama sampai baris kedua. Pada tiap baris akan dilakukan delapan penempatan posisi duduk mahasiswa yang berbeda, dimulai dari pengujian pada satu mahasiswa lalu dilakukan penambahan jumlah mahasiswa secara bertahap satu-persatu sampai delapan mahasiswa.
4. Mengamati tampilan aplikasi sistem presensi yang berisi informasi berupa hasil citra yang ditangkap oleh *webcam*, kumpulan wajah yang terdeteksi dan hasil pengenalannya, serta waktu komputasinya.
5. Mencatat waktu komputasi sistem dalam melakukan pengenalan wajah mahasiswa untuk dihitung rata-ratanya.

6.4.3 Hasil Pengujian

Pengujian waktu komputasi pengenalan wajah dilakukan untuk mencari rata-rata waktu komputasi sistem dalam melakukan pengenalan wajah. Pada Gambar 6.7 dan 6.8 akan disajikan tampilan aplikasi saat pengujian waktu komputasi pengenalan wajah sedangkan pada Tabel 6.6 akan ditunjukkan tabel hasil dari pengujian waktu komputasi pengenalan wajah mahasiswa.



Sistem Presensi Mahasiswa

Tampilan Webcam

12:52:12

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Waktu Komputasi

Jumlah Wajah 3 subjek

Waktu Komputasi De 24 ms

Waktu Komputasi Pe 273 ms

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

Hasil Deteksi

Hasil Pengenalan

Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Renita L S	-	-	-	-	-

Gambar 6.7 Tampilan Aplikasi Sistem Presensi Saat Pengujian Waktu Komputasi Pengenalan Tiga Wajah Mahasiswa

Sistem Presensi Mahasiswa

Tampilan Webcam

12:55:14

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Waktu Komputasi

Jumlah Wajah 8 subjek

Waktu Komputasi De 31 ms

Waktu Komputasi Pe 743 ms

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

Hasil Deteksi

Hasil Pengenalan

Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S
Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S
Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S

Gambar 6.8 Tampilan Aplikasi Sistem Presensi Saat Pengujian Waktu Komputasi Pengenalan Delapan Wajah Mahasiswa



Tabel 6.6 Hasil Waktu Komputasi Pengenalan Wajah

Percobaan ke- n	Jumlah wajah (Aktual)	Waktu komputasi pengenalan wajah (ms)	Rata-rata waktu komputasi pengenalan untuk m -jumlah wajah (ms)
1		93	
2		87	
3	1	122	97.2
4		91	
5		93	
6		178	
7		176	
8	2	201	185
9		183	
10		187	
11		265	
12		294	
13	3	264	273.4
14		271	
15		273	
16		349	
17		282	
18	4	286	307.4
19		360	
20		260	
21		485	
22		454	
23	5	448	445.2
24		452	
25		387	



Tabel 6.6 Hasil Waktu Komputasi Pengenalan Wajah (lanjutan)

26		473	
27		469	
28	6	588	473.2
29		455	
30		381	
31		383	
32		442	
33	7	448	516
34		686	
35		621	
36		631	
37		654	
38	8	674	676
39		743	
40		678	

Dari hasil percobaan tersebut rata-rata waktu komputasi dalam pengenalan satu wajah adalah 97.2 ms, untuk dua wajah 185 ms, untuk tiga wajah 273.4 ms, untuk empat wajah 307.4 ms, untuk lima wajah 445.2 ms, untuk enam wajah 473.2 ms, untuk tujuh wajah 516 ms, dan untuk delapan wajah 676 ms. Sedangkan untuk rata-rata waktu komputasi dari seluruh percobaan pengenalan wajah yaitu 371.675 ms.

6.4.4 Analisis Pengujian

Dari 40 percobaan yang telah dilakukan, didapatkan rata-rata waktu komputasi dalam pendeteksian satu hingga delapan wajah berturut-turut yaitu 97.2 ms, 185 ms, 273.4 ms, 307.4 ms, 445.2 ms, 473.2 ms, 516 ms, dan 676 ms. Setiap penambahan satu wajah akan melambatkan waktu komputasi pengenalan wajah sebesar 28 milisekon hingga 160 milisekon sehingga perubahannya cukup signifikan. Hal yang dapat memengaruhi waktu komputasi dari pengenalan wajah yaitu banyaknya wajah yang diproses untuk pengenalannya, ukuran dari citra wajah yang diproses karena semakin besar ukuran maka semakin banyak juga piksel yang harus diproses, jumlah citra data latih juga dapat memengaruhi waktu komputasi karena setiap data latih akan diproses untuk dibandingkan kemiripannya dengan data uji.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan pengujian dari sistem yang telah dilakukan, maka dapat diambil kesimpulan dari pengujian tersebut guna menjawab rumusan masalah dari penelitian ini. Berikut ini merupakan kesimpulan yang dapat diambil yaitu:

1. Rata-rata akurasi sistem dalam pendeteksian wajah dari seluruh percobaan yaitu 88.88% karena dari 180 wajah yang ada hanya terdeteksi 160 saja. Untuk rata-rata akurasi sistem dalam pendeteksian satu hingga delapan wajah berturut-turut yaitu 100%, 100%, 100%, 85%, 96%, 83.33%, 80%, dan 90%. Rata-rata akurasi tertinggi dalam pendeteksian wajah didapat apabila wajah yang ada berjumlah satu sampai tiga dengan nilai 100%.
2. Rata-rata akurasi sistem dalam pengenalan wajah dari seluruh percobaan menggunakan nilai $k = 3, 5,$ dan 7 berturut-turut yaitu 78.125%, 74.375%, dan 68.125%. Rata-rata akurasi pengenalan wajah tertinggi didapat menggunakan nilai $k = 3$ yaitu 78.125%.
3. Rata-rata waktu komputasi sistem dalam pendeteksian wajah dari seluruh percobaan yaitu 26.2 ms. Untuk rata-rata waktu komputasi sistem dalam pendeteksian satu hingga delapan wajah berturut-turut yaitu 16.8 ms, 21.2 ms, 25.2 ms, 26.4 ms, 27.6 ms, 28.8 ms, 32.6 ms, dan 35.2 ms. Setiap penambahan satu wajah akan melambatkan waktu komputasi deteksi wajah sebesar 1 milisekon hingga 4 milisekon saja sehingga perubahannya tidak terlalu signifikan.
4. Rata-rata waktu komputasi sistem dalam pengenalan wajah dari seluruh percobaan yaitu 371.675 ms. Untuk rata-rata waktu komputasi sistem dalam pengenalan satu hingga delapan wajah berturut-turut yaitu 97.2 ms, 185 ms, 273.4 ms, 307.4 ms, 445.2 ms, 473.2 ms, 516 ms, dan 676 ms. Setiap penambahan satu wajah akan melambatkan waktu komputasi pengenalan wajah sebesar 28 milisekon hingga 160 milisekon sehingga perubahannya cukup signifikan.

7.2 Saran

Saran yang dapat diberikan dari peneliti dalam pengembangan sistem presensi selanjutnya agar bisa lebih baik lagi yaitu:

1. Ditambahkan metode untuk memastikan bahwa hasil pendeteksian dari *Haar Cascade Classifier* benar-benar merupakan wajah manusia dengan menambahkan deteksi pada kedua mata dan seleksi warna untuk mendeteksi kulit.
2. Data latih berupa citra hasil ekstraksi fitur *Local Binary Pattern* yang sudah dinormalisasi diperbanyak jumlahnya sehingga dapat melambatkan rata-rata akurasi sistem dalam melakukan pengenalan wajah.



DAFTAR REFERENSI

Amat, R., 2017. Implementasi Metode Local Binary Patterns Untuk Pengenalan Pola Huruf Hiragana Dan Katakana Pada Smartphone. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 15(2), pp. 162-172.

Dewi, C. & Supianto, A. A., 2015. *Pengolahan Citra Satelit Dengan Matlab*. Malang: UB Press.

Dharma, F. S., 2019. Rekognisi Wajah Pada Sistem Smart Class Untuk Deteksi Kehadiran Mahasiswa Menggunakan Metode Viola Jones dan Local Binary Patterns Histograms (LBPH) Berbasis Raspberry Pi. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(4), pp. 3538-3547.

Enterprise, J., 2018. *Aplikasi Face Detector dan Digital Imaging dengan Python*. Jakarta: PT Elex Media Komputindo.

Group, I., 2005. *Introduction to Database Management Systems*. New York: Tata McGraw-Hill Education.

Hassanien, A. E. & Oliva, D. A., 2017. *Advances in Soft Computing and Machine Learning in Image Processing*. New York: Springer.

Hipp, D. R., 2007. *About SQLite*. [Online] Available at: <https://www.sqlite.org/about.html> [Diakses 16 November 2019].

Huang, D. et al., 2011. Local Binary Patterns and Its Application to Facial Image Analysis: A Survey. *Jurnal IEEE*, 41(6), pp. 765-781.

JUD, 2017. *Mastering Python*. Jakarta: Jubilee Enterprise.

Kadir, A., 1999. *Konsep & Tuntunan Praktis Basis Data*. Yogyakarta: Penerbit Andi.

Ojala, T., Pietikainen, M. & Maenpaa, T., 2002. Multiresolution Gray-scale and Rotation Invariant Texture Classification With Local Binary Patterns. *IEEE Trans Pattern Anal Mach Intell*, Issue 24, pp. 971-987.

Puspaningrum, E. Y., 2018. Deteksi Wajah Dengan Boosted Cascade Classifier. *SCAN - Jurnal Teknologi Informasi dan Komunikasi*, XIII(3).

Raharjo, M. M., 2020. Sistem Deteksi Wajah Dan Sebuah Benda Menggunakan Algoritma Viola-Jones Berbasis Open CV.

Rouse, M., 2007. *Histogram*. [Online] Available at: <https://searchsoftwarequality.techtarget.com/definition/histogram> [Diakses 16 November 2019].

Rozaq, A., 2019. *Sistem Basis Data MYSQL Pada Konsep Jaringan Klien Server*. Banjarmasin: POLIBAN PRESS.

Shally, G. B., Cashman, T. J. & Vermaat, M. E., 2006. *Discovering Computers 2007: A Gateway to Information, Web Enhanced Complete*. s.l.:Course Technology.



Suprianto, D., Hasanah, R. N. & Santosa, P. B., 2013. Sistem Pengenalan Wajah Secara Real-Time dengan Adaboost, Eigenface PCA dan MySQL. *Jurnal EECCIS*, 7(2), pp. 179-180.

Susanto, A., 2018. A High Performance of Local Binary Pattern on Classify Javanese Character Classification. *Scientific Journals of Informatics*, 5(1).

Turiyanto, M. D., Purwanto, D. & Dikairono, R., 2014. Penerapan Teknik Pengenalan Wajah Berbasis Fitur Local Binary Pattern pada Robot Pengantar Makanan. pp. 1-6.



LAMPIRAN

A.1 Percobaan Aplikasi Sistem Presensi Dari Tengah

Tabel A.1.1 Tabel Percobaan Pertama Aplikasi Sistem Presensi Dari Tengah

No	Tampilan aplikasi sistem presensi	Wajah yang terdeteksi	Wajah yang dikenali dengan benar
1	<p>The screenshot shows the 'Sistem Presensi Mahasiswa' interface. On the left is a webcam feed showing a person. In the center is a table titled 'Daftar Presensi' with columns for 'No Urut', 'NIM', and 'Nama'. The table lists 8 students. On the right, there are fields for 'Waktu Terdeteksi' and 'Keterangan'. Below the table is a 'Hasil Deteksi' section showing a small image of the detected face and a 'Hasil Pengenalan' table. The 'Hasil Pengenalan' table has columns for 'Renita L S' and 'Tidak dikenal'. The 'Waktu Komputasi' section shows 'Jumlah Wajah: 1 subjek', 'Waktu Komputasi De: 18 ms', and 'Waktu Komputasi Pe: 93 ms'. There are 'Berhenti', 'Konfirmasi Presensi', and 'Reset Presensi' buttons.</p>	1	1
2	<p>The screenshot shows the 'Sistem Presensi Mahasiswa' interface. It is similar to the first one, but the 'Hasil Deteksi' section shows a different face. The 'Hasil Pengenalan' table shows 'Tidak dikenal' in the first column. The 'Waktu Komputasi' section shows 'Waktu Komputasi Pe: 87 ms'.</p>	1	1
3	<p>The screenshot shows the 'Sistem Presensi Mahasiswa' interface. The 'Hasil Deteksi' section shows a different face. The 'Hasil Pengenalan' table shows 'Leina A Z' in the first column. The 'Waktu Komputasi' section shows 'Waktu Komputasi De: 16 ms' and 'Waktu Komputasi Pe: 122 ms'.</p>	1	1



Sistem Presensi Mahasiswa

Tampilan Webcam:  12:51:24

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi: 

Waktu Komputasi: Jumlah Wajah: 1 subjek, Waktu Komputasi De: 16 ms, Waktu Komputasi Pe: 91 ms

Hasil Pengenalan: Renita L S

Kontrol: Berhenti, Konfirmasi Presensi, Reset Presensi

4

1

1

Sistem Presensi Mahasiswa

Tampilan Webcam:  12:51:25

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi: 

Waktu Komputasi: Jumlah Wajah: 1 subjek, Waktu Komputasi De: 16 ms, Waktu Komputasi Pe: 93 ms

Hasil Pengenalan: Rahma T P

Kontrol: Berhenti, Konfirmasi Presensi, Reset Presensi

5

1

1

Sistem Presensi Mahasiswa

Tampilan Webcam:  12:51:45

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi: 

Waktu Komputasi: Jumlah Wajah: 2 subjek, Waktu Komputasi De: 21 ms, Waktu Komputasi Pe: 178 ms

Hasil Pengenalan: Rahma T P, Renita L S

Kontrol: Berhenti, Konfirmasi Presensi, Reset Presensi

6

2

2

Sistem Presensi Mahasiswa

Tampilan Webcam:  12:51:48

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi: 

Waktu Komputasi: Jumlah Wajah: 2 subjek, Waktu Komputasi De: 20 ms, Waktu Komputasi Pe: 176 ms

Hasil Pengenalan: Rahma T P, Renita L S

Kontrol: Berhenti, Konfirmasi Presensi, Reset Presensi

7

2

2



Sistem Presensi Mahasiswa

Tampilan Webcam

12:51:49

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

--	--	--	--	--	--	--	--

Waktu Komputasi
Jumlah Wajah: 2 subjek
Waktu Komputasi De: 20 ms
Waktu Komputasi Pe: 201 ms

Hasil Pengenalan

Rahma T P	Renita L S	-	-	-	-	-	-
Rahma T P	Renita L S	-	-	-	-	-	-
Rahma T P	Renita L S	-	-	-	-	-	-

Kontrol: Berhenti

Konfirmasi Presensi | Reset Presensi

8

2

2

Sistem Presensi Mahasiswa

Tampilan Webcam

12:51:49

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

--	--	--	--	--	--	--	--

Waktu Komputasi
Jumlah Wajah: 2 subjek
Waktu Komputasi De: 23 ms
Waktu Komputasi Pe: 183 ms

Hasil Pengenalan

Rahma T P	Leina A Z	-	-	-	-	-	-
Rahma T P	Leina A Z	-	-	-	-	-	-
Rahma T P	Leina A Z	-	-	-	-	-	-

Kontrol: Berhenti

Konfirmasi Presensi | Reset Presensi

9

2

2

Sistem Presensi Mahasiswa

Tampilan Webcam

12:51:52

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

--	--	--	--	--	--	--	--

Waktu Komputasi
Jumlah Wajah: 2 subjek
Waktu Komputasi De: 22 ms
Waktu Komputasi Pe: 187 ms

Hasil Pengenalan

Rahma T P	Renita L S	-	-	-	-	-	-
Rahma T P	Renita L S	-	-	-	-	-	-
Rahma T P	Renita L S	-	-	-	-	-	-

Kontrol: Berhenti

Konfirmasi Presensi | Reset Presensi

10

2

2

Sistem Presensi Mahasiswa

Tampilan Webcam

12:52:07

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

--	--	--	--	--	--	--	--

Waktu Komputasi
Jumlah Wajah: 3 subjek
Waktu Komputasi De: 23 ms
Waktu Komputasi Pe: 265 ms

Hasil Pengenalan

Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Renita L S	-	-	-	-	-

Kontrol: Berhenti

Konfirmasi Presensi | Reset Presensi

11

3

3



12

Sistem Presensi Mahasiswa

Tampilan Webcam

12:52:07

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah 3 subjek
Waktu Komputasi De 29 ms
Waktu Komputasi Pe 274 ms

Hasil Pengenalan

Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Tidak dikenal	-	-	-	-	-
Rahma T P	Ilham S	Tidak dikenal	-	-	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

13

Sistem Presensi Mahasiswa

Tampilan Webcam

12:52:07

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah 3 subjek
Waktu Komputasi De 29 ms
Waktu Komputasi Pe 264 ms

Hasil Pengenalan

Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Tidak dikenal	-	-	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

14

Sistem Presensi Mahasiswa

Tampilan Webcam

12:52:08

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah 3 subjek
Waktu Komputasi De 23 ms
Waktu Komputasi Pe 271 ms

Hasil Pengenalan

Rahma T P	Ilham S	Tidak dikenal	-	-	-	-	-
Rahma T P	Ilham S	Tidak dikenal	-	-	-	-	-
Rahma T P	Ilham S	Tidak dikenal	-	-	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

15

Sistem Presensi Mahasiswa

Tampilan Webcam

12:52:12

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah 3 subjek
Waktu Komputasi De 24 ms
Waktu Komputasi Pe 273 ms

Hasil Pengenalan

Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Renita L S	-	-	-	-	-
Rahma T P	Ilham S	Renita L S	-	-	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi



16

Sistem Presensi Mahasiswa

Tampilan Webcam

12:52:32

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 4 subjek
Waktu Komputasi De: 24 ms
Waktu Komputasi Pe: 349 ms

Hasil Pengenalan

Ilham S	Rahma T P	Leina A Z	Renita L S	-	-	-	-
Leina A Z	Rahma T P	Leina A Z	Renita L S	-	-	-	-
Leina A Z	Rahma T P	Leina A Z	Tidak dikenal	-	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

17

Sistem Presensi Mahasiswa

Tampilan Webcam

12:52:36

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 3 subjek
Waktu Komputasi De: 24 ms
Waktu Komputasi Pe: 282 ms

Hasil Pengenalan

Ilham S	Rahma T P	Renita L S	-	-	-	-	-
Ilham S	Rahma T P	Renita L S	-	-	-	-	-
Ilham S	Rahma T P	Renita L S	-	-	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

18

Sistem Presensi Mahasiswa

Tampilan Webcam

12:52:36

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 3 subjek
Waktu Komputasi De: 34 ms
Waktu Komputasi Pe: 286 ms

Hasil Pengenalan

Ilham S	Rahma T P	Renita L S	-	-	-	-	-
Ilham S	Rahma T P	Renita L S	-	-	-	-	-
Ilham S	Rahma T P	Tidak dikenal	-	-	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

19

Sistem Presensi Mahasiswa

Tampilan Webcam

12:52:37

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 4 subjek
Waktu Komputasi De: 24 ms
Waktu Komputasi Pe: 360 ms

Hasil Pengenalan

Ilham S	Rahma T P	Rahma T P	Leina A Z	-	-	-	-
Ilham S	Rahma T P	Rahma T P	Leina A Z	-	-	-	-
Leina A Z	Rahma T P	Rahma T P	Leina A Z	-	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi



20

Sistem Presensi Mahasiswa

Tampilan Webcam

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

12:52:37

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 3 subjek
Waktu Komputasi De: 26 ms
Waktu Komputasi Pe: 260 ms

Hasil Pengenalan

Ilham S	Rahma T P	Renita L S	-	-	-	-
Ilham S	Rahma T P	Renita L S	-	-	-	-
Ilham S	Rahma T P	Renita L S	-	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

4

3

21

Sistem Presensi Mahasiswa

Tampilan Webcam

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

12:53:05

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 5 subjek
Waktu Komputasi De: 30 ms
Waktu Komputasi Pe: 485 ms

Hasil Pengenalan

Leina A Z	Ilham S	Rahma T P	Leina A Z	Renita L S	-	-
Leina A Z	Ilham S	Rahma T P	Leina A Z	Renita L S	-	-
Leina A Z	Ilham S	Rahma T P	Leina A Z	Renita L S	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

5

5

22

Sistem Presensi Mahasiswa

Tampilan Webcam

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

12:53:06

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 5 subjek
Waktu Komputasi De: 26 ms
Waktu Komputasi Pe: 454 ms

Hasil Pengenalan

Leina A Z	Ilham S	Rahma T P	Leina A Z	Renita L S	-	-
Ilham S	Ilham S	Rahma T P	Leina A Z	Renita L S	-	-
Leina A Z	Leina A Z	Rahma T P	Leina A Z	Renita L S	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

5

5

23

Sistem Presensi Mahasiswa

Tampilan Webcam

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

12:53:07

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 5 subjek
Waktu Komputasi De: 26 ms
Waktu Komputasi Pe: 448 ms

Hasil Pengenalan

Ilham S	Rahma T P	Ilham S	Leina A Z	Renita L S	-	-
Ilham S	Rahma T P	Ilham S	Leina A Z	Ilham S	-	-
Leina A Z	Rahma T P	Ilham S	Leina A Z	Ilham S	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

5

5



24

Sistem Presensi Mahasiswa

Tampilan Webcam

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

12:53:08

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 5 subjek
Waktu Komputasi De: 30 ms
Waktu Komputasi Pe: 452 ms

Hasil Pengenalan

Leina A Z	Rahma T P	Ilham S	Leina A Z	Leina A Z	-	-	-
Leina A Z	Rahma T P	Ilham S	Leina A Z	Leina A Z	-	-	-
Leina A Z	Rahma T P	Ilham S	Leina A Z	Leina A Z	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

25

Sistem Presensi Mahasiswa

Tampilan Webcam

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

12:53:10

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 4 subjek
Waktu Komputasi De: 26 ms
Waktu Komputasi Pe: 387 ms

Hasil Pengenalan

Leina A Z	Ilham S	Leina A Z	Rahma T P	-	-	-	-
Leina A Z	Ilham S	Leina A Z	Rahma T P	-	-	-	-
Leina A Z	Ilham S	Leina A Z	Rahma T P	-	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

26

Sistem Presensi Mahasiswa

Tampilan Webcam

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

12:53:41

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 5 subjek
Waktu Komputasi De: 30 ms
Waktu Komputasi Pe: 473 ms

Hasil Pengenalan

Leina A Z	Leina A Z	Leina A Z	Rahma T P	Rahma T P	-	-	-
Leina A Z	Leina A Z	Leina A Z	Rahma T P	Rahma T P	-	-	-
Leina A Z	Leina A Z	Leina A Z	Rahma T P	Rahma T P	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

27

Sistem Presensi Mahasiswa

Tampilan Webcam

Daftar Presensi

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

12:53:43

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 5 subjek
Waktu Komputasi De: 27 ms
Waktu Komputasi Pe: 469 ms

Hasil Pengenalan

Leina A Z	Ilham S	Leina A Z	Rahma T P	Rahma T P	-	-	-
Leina A Z	Ilham S	Leina A Z	Rahma T P	Rahma T P	-	-	-
Leina A Z	Ilham S	Leina A Z	Rahma T P	Rahma T P	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi



Sistem Presensi Mahasiswa

Tampilan Webcam:  12:53:48

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi: 

Waktu Komputasi: Jumlah Wajah: 6 subjek, Waktu Komputasi De: 33 ms, Waktu Komputasi Pe: 588 ms

Hasil Pengenalan: Leina A Z, Leina A Z, Ilham S, Rahma T P, Renita L S, -

Kontrol: Berhenti, Konfirmasi Presensi, Reset Presensi

28

6

6

Sistem Presensi Mahasiswa

Tampilan Webcam:  12:53:53

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi: 

Waktu Komputasi: Jumlah Wajah: 5 subjek, Waktu Komputasi De: 28 ms, Waktu Komputasi Pe: 455 ms

Hasil Pengenalan: Ilham S, Leina A Z, Leina A Z, Rahma T P, Renita L S, -

Kontrol: Berhenti, Konfirmasi Presensi, Reset Presensi

29

6

5

Sistem Presensi Mahasiswa

Tampilan Webcam:  12:53:55

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi: 

Waktu Komputasi: Jumlah Wajah: 4 subjek, Waktu Komputasi De: 26 ms, Waktu Komputasi Pe: 381 ms

Hasil Pengenalan: Ilham S, Leina A Z, Ilham S, Rahma T P, -

Kontrol: Berhenti, Konfirmasi Presensi, Reset Presensi

30

6

4

Sistem Presensi Mahasiswa

Tampilan Webcam:  12:54:07

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi: 

Waktu Komputasi: Jumlah Wajah: 4 subjek, Waktu Komputasi De: 28 ms, Waktu Komputasi Pe: 383 ms

Hasil Pengenalan: Gani K W, Ilham S, Rahma T P, Leina A Z, -

Kontrol: Berhenti, Konfirmasi Presensi, Reset Presensi

31

7

4



32

Sistem Presensi Mahasiswa

Tampilan Webcam

12:54:07

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 5 subjek
Waktu Komputasi De: 37 ms
Waktu Komputasi Pe: 442 ms

Hasil Pengenalan

Leina A Z	Ilham S	Rahma T P	Leina A Z	Renita L S	-	-	-
Ilham S	Ilham S	Rahma T P	Leina A Z	Renita L S	-	-	-
Ilham S	Ilham S	Rahma T P	Leina A Z	Leina A Z	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

33

Sistem Presensi Mahasiswa

Tampilan Webcam

12:54:08

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 5 subjek
Waktu Komputasi De: 28 ms
Waktu Komputasi Pe: 448 ms

Hasil Pengenalan

Leina A Z	Gani K W	Leina A Z	Rahma T P	Leina A Z	-	-	-
Leina A Z	Gani K W	Leina A Z	Rahma T P	Leina A Z	-	-	-
Leina A Z	Tidak dikenal	Leina A Z	Rahma T P	Leina A Z	-	-	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

34

Sistem Presensi Mahasiswa

Tampilan Webcam

12:54:22

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 7 subjek
Waktu Komputasi De: 35 ms
Waktu Komputasi Pe: 686 ms

Hasil Pengenalan

Leina A Z	Gani K W	Leina A Z	Leina A Z	Leina A Z	Renita L S	Rahma T P	-
Leina A Z	Gani K W	Leina A Z	Leina A Z	Leina A Z	Renita L S	Rahma T P	-
Leina A Z	Gani K W	Leina A Z	Leina A Z	Leina A Z	Renita L S	Rahma T P	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

35

Sistem Presensi Mahasiswa

Tampilan Webcam

12:54:23

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi
Jumlah Wajah: 7 subjek
Waktu Komputasi De: 35 ms
Waktu Komputasi Pe: 621 ms

Hasil Pengenalan

Ilham S	Gani K W	Leina A Z	Leina A Z	Rahma T P	Ilham S	Renita L S	-
Ilham S	Gani K W	Leina A Z	Leina A Z	Rahma T P	Ilham S	Renita L S	-
Ilham S	Gani K W	Leina A Z	Leina A Z	Rahma T P	Ilham S	Renita L S	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi



Sistem Presensi Mahasiswa

Tampilan Webcam:  12:54:55

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi: 

Waktu Komputasi: Jumlah Wajah 7 subjek, Waktu Komputasi De 32 ms, Waktu Komputasi Pe 631 ms

Hasil Pengenalan:

Ilham S	Vedric A S	Gani K W	Leina A Z	Ilham S	Rahma T P	Renita L S	-
Ilham S	Vedric A S	Gani K W	Leina A Z	Ilham S	Rahma T P	Renita L S	-
Ilham S	Vedric A S	Gani K W	Leina A Z	Ilham S	Rahma T P	Renita L S	-

Kontrol:

36

8

7

Sistem Presensi Mahasiswa

Tampilan Webcam:  12:55:04

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi: 

Waktu Komputasi: Jumlah Wajah 7 subjek, Waktu Komputasi De 30 ms, Waktu Komputasi Pe 654 ms

Hasil Pengenalan:

Gani K W	Igo V F	Ilham S	Leina A Z	Leina A Z	Rahma T P	Renita L S	-
Gani K W	Igo V F	Ilham S	Leina A Z	Leina A Z	Rahma T P	Renita L S	-
Gani K W	Igo V F	Ilham S	Leina A Z	Leina A Z	Rahma T P	Renita L S	-

Kontrol:

37

8

7

Sistem Presensi Mahasiswa

Tampilan Webcam:  12:55:05

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi: 

Waktu Komputasi: Jumlah Wajah 7 subjek, Waktu Komputasi De 35 ms, Waktu Komputasi Pe 674 ms

Hasil Pengenalan:

Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Renita L S	Rahma T P	-
Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Renita L S	Rahma T P	-
Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Renita L S	Rahma T P	-

Kontrol:

38

8

7

Sistem Presensi Mahasiswa

Tampilan Webcam:  12:55:14

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi: 

Waktu Komputasi: Jumlah Wajah 8 subjek, Waktu Komputasi De 31 ms, Waktu Komputasi Pe 743 ms

Hasil Pengenalan:

Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S
Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S
Vedric A S	Igo V F	Gani K W	Ilham S	Ilham S	Leina A Z	Rahma T P	Renita L S

Kontrol:

39

8

8



Sistem Presensi Mahasiswa

Tampilan Webcam

12:55:38

No Urut	NIM	Nama	Waktu Terdeteksi	Keterangan
1	165150300111009	Ilham S	-	-
2	165150301111002	Renita L S	-	-
3	165150307111003	Andika B	-	-
4	165150301111021	Leina A Z	-	-
5	165150307111004	Igo V F	-	-
6	165150307111010	Vedric A S	-	-
7	165150300111001	Rahma T P	-	-
8	165150707111009	Gani K W	-	-

Hasil Deteksi

Waktu Komputasi

Jumlah Wajah: 7 subjek

Waktu Komputasi De: 48 ms

Waktu Komputasi Pe: 678 ms

Hasil Pengenalan

Igo V F	Gani K W	Leina A Z	Ilham S	Leina A Z	Rahma T P	Rahma T P	-
Igo V F	Gani K W	Leina A Z	Ilham S	Leina A Z	Rahma T P	Rahma T P	-
Igo V F	Gani K W	Leina A Z	Ilham S	Leina A Z	Rahma T P	Rahma T P	-

Kontrol

Berhenti

Konfirmasi Presensi Reset Presensi

A.2 Kode Program Utama Keseluruhan

Tabel A.2.1 Kode Program Utama

Baris	Kode Program
1	"""
2	Author: Meidiana Adinda Prasanty
3	"""
4	# Program Aplikasi Sistem Presensi
5	
6	# import some PyQt5 modules
7	from PyQt5.QtWidgets import QApplication
8	from PyQt5.QtWidgets import QWidget
9	from PyQt5.QtGui import QImage
10	from PyQt5.QtGui import QPixmap
11	from PyQt5.QtCore import QTimer
12	
13	# impor modul untuk pemrosesan citra
14	import sys
15	import cv2
16	import numpy as np
17	import os
18	import operator
19	import copy
20	import sqlite3
21	import statistics
22	import time
23	



```

24 from gui import * # Mengimpor modul untuk tampilan
    aplikasi sistem presensi
25
26 ### PROSES UTAMA deklarasi variabel untuk pemrosesan citra
27 putih = ~np.zeros((128,128,3), dtype=np.uint8)
28
29 ### AWAL INISIALISASI UNTUK PROSES DETEKSI WAJAH ###
30 face_cascade =
    cv2.CascadeClassifier("file_referensi//haarcascade_frontal
    face_default.xml")
31 waktu_terdeteksi = 0
32
33 ### AWAL INISIALISASI UNTUK PROSES PENGENALAN WAJAH ###
34 roi_lbp = np.zeros((128, 128), np.uint8)
35 roi_gray = np.zeros((128, 128), np.uint8)
36 frame_lbp_norm = np.zeros((128,128), np.uint8)
37 id_mhs = 0
38 # Untuk pengenalan wajah juga (KNN (data_latih))
39 n_subjek = 8
40 n_sampel = 8
41 n_data_latih = n_subjek*n_sampel
42 sampel_lbp_normal = np.zeros((128,128,n_data_latih),
    np.uint8)
43 distance_euclidean = [[0, "-", "-", 0, 0] for _ in
    range(n_data_latih)]
44 hasil_pengenalan_k = [[ "-", "-", "-"] for _ in
    range(n_data_latih)]
45
46 # membaca data latih wajah LBP
47 index_wajah = 0
48 for i in range(n_subjek):
49     for j in range(n_sampel):
50         if(os.path.exists('data sampel/mhs_' +str(i)+'_' +str(j)+'_i
            bp.jpg')) == True:
51             sampel_lbp_normal[:, :, index_wajah] =
                cv2.imread('data sampel/mhs_' +str(i)+'_' +str(j)+'_lbp.jpg',
                    cv2.IMREAD_UNCHANGED) # membaca sampel bertipe grayscale
                    uint8
52         else:
53             sampel_lbp_normal[:, :, index_wajah] =
                ~np.zeros((128,128), dtype=np.uint8)

```



```

54     index_wajah += 1
55
56     # Inisialisasi objek database
57     database_sqlite = 'database/presensi.db'
58     try:
59         sql_con = sqlite3.connect(database_sqlite)
60         cursor = sql_con.cursor()
61
62         for i in range(0,8):
63             cursor.execute("SELECT * FROM mahasiswa WHERE id = ?",
64                             (i+1,))
65             hasil = cursor.fetchone()
66             nama = hasil[2]
67             nim = hasil[1]
68             id_subjek = hasil[0]
69             for j in range(0+(i*8), ((i+1)*8)):
70                 distance_euclidean[j][1] = nama
71                 distance_euclidean[j][2] = nim
72                 distance_euclidean[j][3] = id_subjek #id subjek
73                 distance_euclidean[j][4] = j%8 #id sampel
74             print("Koneksi SQLite berhasil!")
75         except sqlite3.Error as error:
76             print("Koneksi SQLite error!")
77     ### AKHIR INISIALISASI UNTUK PROSES PENGENALAN WAJAH ###
78
79     # Untuk waktu komputasi
80     t_sekarang = lambda: int(round(time.time() * 1000)) #
81     Membuat fungsi untuk mendapatkan waktu sekarang dalam
82     bentuk milisekon
83
84     # Untuk melakukan konfirmasi presensi
85     daftar_presensi = []
86     nama_pengenalan = []
87
88     class MainWindow(QWidget):
89         # class constructor
90         def __init__(self):
91             # call QWidget constructor

```



```

90     super().init_()
91     self.ui = Ui_Form()
92     self.ui.setupUi(self)
93
94     # create a timer
95     self.timer = QTimer()
96     self.timer.timeout.connect(self.viewCam) # set timer
97     timeout callback function
98     self.ui.control_bt.clicked.connect(self.controlTimer)
99     # set control_bt callback clicked function
100    self.ui.tombol_konfirmasi.clicked.connect(self.konfirmasiP
101    resensi)
102    self.ui.tombol_reset.clicked.connect(self.resetPresensi)
103
104    self.updateTabelPresensi()
105
106    def updateTabelPresensi(self):
107        global daftar_presensi
108        sql_con = sqlite3.connect(database_sqlite)
109        cursor = sql_con.cursor()
110        cursor.execute("SELECT * FROM mahasiswa")
111        daftar_presensi = cursor.fetchall()
112
113        for y in range(0,8):
114            self.ui.tableWidget.setItem(y, 0,
115            QtWidgets.QTableWidgetItem(str(daftar_presensi[y][0])) #
116            id
117            self.ui.tableWidget.setItem(y, 1,
118            QtWidgets.QTableWidgetItem(daftar_presensi[y][1])) # nim
119            self.ui.tableWidget.setItem(y, 2,
120            QtWidgets.QTableWidgetItem(daftar_presensi[y][2])) # nama
121            if str(daftar_presensi[y][3]) == 'None':
122                self.ui.tableWidget.setItem(y, 3,
123                QtWidgets.QTableWidgetItem('-')) # waktu terdeteksi
124            self.ui.tableWidget.setItem(y, 4,
125            QtWidgets.QTableWidgetItem('0')) # keterangan
126        else:
127            self.ui.tableWidget.setItem(y, 3,
128            QtWidgets.QTableWidgetItem(str(daftar_presensi[y][3])) #
129            waktu terdeteksi

```



```

119         self.ui.tableWidget.setItem(y, 4,
QtWidgets.QTableWidgetItem('Hadir')) # keterangan
120
121     def generate_QImage(self, frame):
122         return QImage(frame.data, frame.shape[1],
frame.shape[0], frame.shape[1]*frame.shape[2],
QImage.Format_RGB888).rgbSwapped()
123
124     def konfirmasiPresensi(self):
125         global nama_pengenalan, waktu_terdeteksi
126         if len(nama_pengenalan) != 0:
127
128             sql_con = sqlite3.connect(database_sqlite)
129             cursor = sql_con.cursor()
130             for i in range(len(nama_pengenalan)):
131                 if (nama_pengenalan[i][0] != '-') and
(nama_pengenalan[i][0] != 'Tidak dikenal'):
132                     sql = "UPDATE mahasiswa SET waktu_terdeteksi = ?
WHERE nama = ?"
133                     val = (waktu_terdeteksi, nama_pengenalan[i][0])
134                     cursor.execute(sql, val)
135                     sql_con.commit()
136                     self.updateTabelPresensi()
137
138     def resetPresensi(self):
139         sql_con = sqlite3.connect(database_sqlite)
140         cursor = sql_con.cursor()
141         cursor.execute("UPDATE mahasiswa SET waktu_terdeteksi
=NULL where waktu_terdeteksi is not NULL")
142         sql_con.commit()
143         self.updateTabelPresensi()
144         return
145
146         # view camera
147         def viewCam(self):
148             # clock
149             self.ui.lcd_jam.display(time.strftime("%H":"+" "%M":"+" "%S"))
150

```



```
151         ret, image = self.cap.read() # read image in_BGR
        format
152
153         frame_deteksi_bgr = []
154         global nama_pengenalan
155         nama_pengenalan = []
156         jumlah_wajah_terdeteksi = 0
157         waktu_komputasi_deteksi = t_sekarang()
158         waktu_komputasi_pengenalan = 0
159         image, frame_deteksi_bgr, nama_pengenalan,
        jumlah_wajah_terdeteksi, waktu_komputasi_deteksi,
        waktu_komputasi_pengenalan = proses_LBP(image,
        frame_deteksi_bgr, nama_pengenalan,
        jumlah_wajah_terdeteksi, waktu_komputasi_deteksi,
        waktu_komputasi_pengenalan)
160         self.ui.image_label.setPixmap(QPixmap.fromImage(self.g
        enerate_QImage(image)))
161
162         if(jumlah_wajah_terdeteksi>0):
163             # menampilkan citra hasil deteksi wajah
164             self.ui.detected_face_0.setPixmap(QPixmap.fromImage(self.g
        enerate_QImage(frame_deteksi_bgr[0])))
165             self.ui.detected_face_1.setPixmap(QPixmap.fromImage(self.g
        enerate_QImage(frame_deteksi_bgr[1])))
166             self.ui.detected_face_2.setPixmap(QPixmap.fromImage(self.g
        enerate_QImage(frame_deteksi_bgr[2])))
167             self.ui.detected_face_3.setPixmap(QPixmap.fromImage(self.g
        enerate_QImage(frame_deteksi_bgr[3])))
168             self.ui.detected_face_4.setPixmap(QPixmap.fromImage(self.g
        enerate_QImage(frame_deteksi_bgr[4])))
169             self.ui.detected_face_5.setPixmap(QPixmap.fromImage(self.g
        enerate_QImage(frame_deteksi_bgr[5])))
170             self.ui.detected_face_6.setPixmap(QPixmap.fromImage(self.g
        enerate_QImage(frame_deteksi_bgr[6])))
171             self.ui.detected_face_7.setPixmap(QPixmap.fromImage(self.g
        enerate_QImage(frame_deteksi_bgr[7])))
172
173         # menampilkan nama hasil pengenalan wajah
```



```

174 self.ui.nama_k_3_0.setText(nama_pengenalan[0][0])
175 self.ui.nama_k_5_0.setText(nama_pengenalan[0][1])
176 self.ui.nama_k_7_0.setText(nama_pengenalan[0][2])
177 self.ui.nama_k_3_1.setText(nama_pengenalan[1][0])
178 self.ui.nama_k_5_1.setText(nama_pengenalan[1][1])
179 self.ui.nama_k_7_1.setText(nama_pengenalan[1][2])
180 self.ui.nama_k_3_2.setText(nama_pengenalan[2][0])
181 self.ui.nama_k_5_2.setText(nama_pengenalan[2][1])
182 self.ui.nama_k_7_2.setText(nama_pengenalan[2][2])
183 self.ui.nama_k_3_3.setText(nama_pengenalan[3][0])
184 self.ui.nama_k_5_3.setText(nama_pengenalan[3][1])
185 self.ui.nama_k_7_3.setText(nama_pengenalan[3][2])
186 self.ui.nama_k_3_4.setText(nama_pengenalan[4][0])
187 self.ui.nama_k_5_4.setText(nama_pengenalan[4][1])
188 self.ui.nama_k_7_4.setText(nama_pengenalan[4][2])
189 self.ui.nama_k_3_5.setText(nama_pengenalan[5][0])
190 self.ui.nama_k_5_5.setText(nama_pengenalan[5][1])
191 self.ui.nama_k_7_5.setText(nama_pengenalan[5][2])
192 self.ui.nama_k_3_6.setText(nama_pengenalan[6][0])
193 self.ui.nama_k_5_6.setText(nama_pengenalan[6][1])
194 self.ui.nama_k_7_6.setText(nama_pengenalan[6][2])
195 self.ui.nama_k_3_7.setText(nama_pengenalan[7][0])
196 self.ui.nama_k_5_7.setText(nama_pengenalan[7][1])
197 self.ui.nama_k_7_7.setText(nama_pengenalan[7][2])
198
199 # menampilkan waktu komputasi
200
201 self.ui.n_wajah_terdeteksi.setText(str(jumlah_wajah_terdeteksi)+' subjek')
202 self.ui.t_deteksi.setText(str(waktu_komputasi_deteksi)+' ms')
203 self.ui.t_pengenalan.setText(str(waktu_komputasi_pengenalan)+' ms')
204
205 # start/stop timer
206 def controllTimer(self):
207     # if timer is stopped
208     if not self.timer.isActive():

```



```

208         # create video capture
209         self.cap = cv2.VideoCapture(0)
210         # start timer
211         self.timer.start(20)
212         # update control_bt text
213         self.ui.control_bt.setText("Berhenti")
214         # if timer is started
215         else:
216             # stop timer
217             self.timer.stop()
218             # release video capture
219             self.cap.release()
220             # update control_bt text
221             self.ui.control_bt.setText("Mulai")
222
223     # fungsi
224     def getPixelValue(img, tengah, y, x):
225         biner = 0
226         try:
227             if img[y][x] >= tengah:
228                 biner = 1
229             except:
230                 pass
231             return biner
232
233     def hitungPixelLBP(img, y, x):
234         '''
235         128 | 64 | 32
236         -----
237         1 | 0 | 16
238         -----
239         2 | 4 | 8
240         '''
241         tengah = img[y][x]
242         biner_ar = []
243         biner_ar.append(getPixelValue(img, tengah, y-1, x-1))
244         # atas kiri
245         biner_ar.append(getPixelValue(img, tengah, y-1, x)) #
246         # atas

```



```

244     biner_ar.append(getPixelValue(img, tengah, y-1, x+1))
        # atas kanan
245     biner_ar.append(getPixelValue(img, tengah, y, x+1))
        # kanan
246     biner_ar.append(getPixelValue(img, tengah, y+1, x+1))
        # bawah kanan
247     biner_ar.append(getPixelValue(img, tengah, y+1, x)) #
        bawah
248     biner_ar.append(getPixelValue(img, tengah, y+1, x-1))
        # bawah kiri
249     biner_ar.append(getPixelValue(img, tengah, y, x-1))
        # kiri
250
251     power_biner = [128, 64, 32, 16, 8, 4, 2, 1]
252     desimal = 0
253     for i in range(len(biner_ar)):
254         desimal += biner_ar[i] * power_biner[i]
255     return desimal
256
257 def ekstraksiFiturLBP(frame):
258     h = frame.shape[0]
259     w = frame.shape[1]
260     frame_lbp = ~np.zeros((h, w), np.uint8)
261     for y in range(0, h):
262         for x in range(0, w):
263             frame_lbp[y, x] = hitungPixelLBP(frame, y, x)
264     frame_lbp_norm = cv2.normalize(frame_lbp, 0, alpha=0,
265     beta=255, norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_8UC1)
266     return frame_lbp_norm
267
268 def proses_LBP(frame, frame_deteksi_bgr, nama_pengenalan,
269     n_wajah_terdeteksi, t_komputasi_deteksi,
270     t_komputasi_pengenalan):
271     global waktu_terdeteksi
272     roi_color = putih
273     #frame = cv2.flip(frame, 1)
274     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
275     faces = face_cascade.detectMultiScale(gray, 1.3, 5)
276     waktu_terdeteksi = time.strftime('%Y-%m-%d %H:%M:%S')
277     t_komputasi_deteksi = t_sekarang() - t_komputasi_deteksi

```



```
276
277     # Apabila ada wajah terdeteksi
278     n_wajah_terdeteksi = len(faces)
279     if n_wajah_terdeteksi>0:
280
281         t_komputasi_pengenalan = t_sekarang()
282         for f in faces:
283             (x,y,w,h) = f
284             frame =
285             cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,0),1)
286             roi_gray = gray[y:y+h, x:x+w]
287             roi_gray = cv2.resize(roi_gray, (128,128),
288                                 cv2.INTER_LINEAR)
289             roi_color = frame[y:y+h, x:x+w]
290             roi_color = cv2.resize(roi_color, (128,128),
291                                 cv2.INTER_LINEAR)
292
293             # PEMROSESAN LBP
294             roi_lbp_norm = ekstraksiFiturLBP(roi_gray)
295
296             # Perhitungan KNN (Klasifikasi KNN)
297             # KNN 8 sampel
298             for k in range(n_data_latih):
299                 a = roi_lbp_norm[:,:] # tetap
300                 b = sampel_lbp_normal[:,:,k]
301                 a_min_b = np.asarray(a-b)
302                 distance_euclidean[k][0] =
303                 np.sqrt(np.sum(a_min_b**2))
304
305             # Pengurutan keatas berdasarkan jarak Euclidean
306             sorted_distance_euclidean =
307             copy.copy(distance_euclidean)
308             sorted_distance_euclidean.sort(key =
309             operator.itemgetter(0))
310
311             nama_k_3 = 'Tidak dikenal'
312             nama_k_5 = 'Tidak dikenal'
313             nama_k_7 = 'Tidak dikenal'
314
315             try:
```



```

310         nama_k_3 =
311         statistics.mode([sorted_distance_euclidean[0][1],
312                         sorted_distance_euclidean[1][1],
313                         sorted_distance_euclidean[2][1]])
314         nama_k_5 =
315         statistics.mode([sorted_distance_euclidean[0][1],
316                         sorted_distance_euclidean[1][1],
317                         sorted_distance_euclidean[2][1],
318                         sorted_distance_euclidean[3][1],
319                         sorted_distance_euclidean[4][1]])
320         nama_k_7 =
321         statistics.mode([sorted_distance_euclidean[0][1],
322                         sorted_distance_euclidean[1][1],
323                         sorted_distance_euclidean[2][1],
324                         sorted_distance_euclidean[3][1],
325                         sorted_distance_euclidean[4][1],
326                         sorted_distance_euclidean[5][1],
327                         sorted_distance_euclidean[6][1]])
328     except:
329         pass
330     frame_deteksi_bgr.append(roi_color)
331     nama_pengenalan.append([nama_k_3, nama_k_5,
332                             nama_k_7])
333     t_komputasi_pengenalan = t_sekarang() -
334     t_komputasi_pengenalan
335     for i in range(n_subjek-len(frame_deteksi_bgr)):
336         frame_deteksi_bgr.append(putih)
337         nama_pengenalan.append(["-", "-", "-"])
338     return frame, frame_deteksi_bgr, nama_pengenalan,
339     n_wajah_terdeteksi, t_komputasi_deteksi,
340     t_komputasi_pengenalan
341     if __name__ == '__main__':
342         app = QApplication(sys.argv)
343         # create and show mainWindow
344         mainWindow = MainWindow()

```



```
344     mainWindow.show()  
345  
346     cursor.close()  
347     sql_con.close()  
348  
349     sys.exit(app.exec_())  
350
```