

**REKOGNISI WAJAH PADA SISTEM *SMART CLASS* UNTUK DETEKSI
KEHADIRAN MAHASISWA MENGGUNAKAN METODE VIOLA JONES
DAN *LOCAL BINARY PATTERNS HISTOGRAMS* (LBPH) BERBASIS
RASPBERRY PI**

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
mencapai gelar Sarjana Komputer

Disusun oleh:
Fitrahadi Surya Dharma
NIM: 135150307111036



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

REKOGNISI WAJAH PADA SISTEM *SMART CLASS* UNTUK DETEKSI KEHADIRAN MAHASISWA MENGGUNAKAN METODE VIOLA JONES DAN *LOCAL BINARY PATTERN HISTOGRAM* (LBPH) BERBASIS RASPBERRY PI

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh:
Fitrahadi Surya Dharma
NIM: 135150307111036

Skripsi ini telah diuji dan dinyatakan lulus pada
28 Desember 2018
Telah diperiksa dan disetujui oleh:

Pembimbing I

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

Pembimbing II

Rizal Maulana, S.T., M.T., M.Sc.
NIK. 2016078910091001

Mengetahui,
Ketua Jurusan Teknik Informatika



Tri Asto Kurniawan, S.T., M.T., Ph.D
NIP. 19710518 200312 1 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam 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 pustaka.

Apabila ternyata di dalam 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, 14 Desember 2018



Fitrahadi Surya Dharma

NIM:135150307111036

ABSTRAK

Fitrahadi Surya Dharma, Rekognisi Wajah Pada Sistem *Smart Class* Untuk Deteksi Kehadiran Mahasiswa Menggunakan Metode Viola Jones dan *Local Binary Patterns Histograms* (LBPH) Berbasis Raspberry

Pembimbing: Dr. Eng. Fitri Utaminingrum, S.T, M.T dan Rizal Maulana , S.T., M.T., M.Sc.

Rekognisi wajah merupakan salah satu teknik di dalam *computer vision* yang mampu mengenali wajah seseorang dari sebuah gambar. Penerapan rekognisi wajah ke dalam sistem presensi menjadi begitu penting mengingat masih ditemukannya kasus manipulasi data kehadiran oleh para mahasiswa di dalam sistem presensi yang menggunakan cara manual – pengisian tanda tangan pada lembar presensi. Kurang ketatnya pengawasan di dalam pengisian lembar presensi menjadi peristiwa yang rentan terhadap kasus pemanipulasian data kehadiran. Oleh karenanya di dalam penelitian ini mencoba untuk menghadirkan sebuah sistem presensi yang memanfaatkan gambar untuk mengetahui kehadiran mahasiswa. Caranya adalah dengan mengambil gambar menggunakan kamera yang di letakan di depan kelas, tepat di atas papan tulis menghadap ke arah mahasiswa. Dari gambar yang di ambil, sistem kemudian akan mendeteksi wajah mahasiswa dengan menggunakan metode Viola Jones dari OpenCV *library* yang dipadukan dengan deteksi piksel warna kulit YCbCr untuk menghindari deteksi palsu. Dan untuk pengenalan wajah mahasiswa akan dilakukan menggunakan metode *Local Binary Patterns Histograms* dari OpenCV *library*. Hasil akurasi yang diperoleh sistem menunjukkan tingkat akurasi pendeteksian sebesar 82,33% dan akurasi pengenalan sebesar 50,83% di waktu pagi, 61,11% di waktu siang, dan 58,89% di waktu malam. Rata-rata total waktu komputasi untuk deteksi satu mahasiswa adalah 0,293 detik, dua mahasiswa 0,297 detik, tiga mahasiswa 0,317 detik, empat mahasiswa 0,313 detik, lima mahasiswa 0,31 detik dan enam mahasiswa 0,307 detik. Sedangkan rata-rata total waktu komputasi pengenalan wajah untuk satu mahasiswa sebesar 2,17 detik, dua mahasiswa 2,58 detik, tiga mahasiswa 3,01 detik, empat mahasiswa 3,38 detik, lima mahasiswa 3,78 detik, dan enam mahasiswa 4,12 detik.

Kata kunci: rekognisi wajah, *local binary patterns histograms*, viola jones, *smart class*, sistem presensi

ABSTRACT

Fitrahadi Surya Dharma, Rekognisi Wajah Pada Sistem *Smart Class* Untuk Deteksi Kehadiran Mahasiswa Menggunakan Metode Viola Jones dan *Local Binary Patterns Histograms* (LBPH) Berbasis Raspberry

Pembimbing: Dr. Eng. Fitri Utaminingrum, S.T, M.T dan Rizal Maulana , S.T., M.T., M.Sc.

Facial recognition is one of the techniques in computer vision that is able to recognize a person's face from an image. The application of face recognition into the presence system is very important considering that there are still cases of attendance data manipulation by students in the presence system using manual - filling signatures on the attendance sheet. Lack of tight supervision in filling attendance sheets is an event that is vulnerable to cases of manipulating attendance data. Therefore in this study try to present a presence system that uses images to find out the presence of students. The trick is to take pictures using a camera that is placed in front of the class, just above the blackboard facing the student. From the images taken, the system will then detect the faces of students using the Viola Jones method of the OpenCV library combined with YCbCr skin color pixel detection to avoid false detection. And for face recognition students will be using the Local Binary Patterns Histograms method from the OpenCV library. Accuracy results obtained by the system showed the level of detection accuracy of 82.33% and recognition accuracy of 50.83% in the morning, 61.11% during the day, and 58.89% at night. The average total computing time for the detection of one student is 0.293 seconds, two students 0.297 seconds, three students 0.317 seconds, four students 0.313 seconds, five students 0.31 seconds and six students 0.307 seconds. While the average total face recognition computing time for one student is 2.17 seconds, two students 2.58 seconds, three students 3.01 seconds, four students 3.38 seconds, five students 3.78 seconds, and six students 4 .12 seconds.

Keywords: face recognition, local binary patterns histograms, viola jones, smart classes, presence system

DAFTAR ISI

PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS.....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
<i>ABSTRACT</i>	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.....	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Kajian Pustaka.....	5
2.2 Dasar Teori.....	7
2.2.1 <i>Smart Class</i>	7
2.2.2 Citra Digital.....	8
2.2.3 Citra RGB.....	9
2.2.4 Konversi Citra RGB ke YCbCr.....	9
2.2.5 Konversi Citra RGB ke <i>Grayscale</i>	10
2.2.6 Viola Jones.....	11
2.2.7 <i>Local Binary Patterns Histograms</i> (LBPH).....	14
2.2.8 Webcam.....	18

2.2.9 Raspberry Pi 3 Model B.....	18
2.2.10 Python Versi 3.....	20
2.2.11 OpenCV <i>library</i>	20
BAB 3 METODOLOGI PENELITIAN.....	21
3.1 Alur Metode Penelitian.....	21
3.2 Studi Literatur.....	22
3.3 Analisis Kebutuhan.....	22
3.3.1 Analisis Kebutuhan Perangkat Keras.....	22
3.3.2 Analisis Kebutuhan Perangkat Lunak.....	23
3.4 Perancangan Sistem.....	23
3.5 Implementasi Sistem.....	24
3.6 Pengujian dan Analisis.....	25
3.7 Kesimpulan dan Saran.....	25
BAB 4 REKAYASA KEBUTUHAN.....	26
4.1 Gambaran Umum Sistem.....	26
4.1.1 Tujuan.....	26
4.1.2 Karakteristik Pengguna.....	26
4.1.3 Lingkungan Operasi.....	26
4.1.4 Batasan Perancangan dan Implementasi.....	27
4.1.5 Asumsi dan Ketergantungan.....	27
4.2 Analisis Kebutuhan Sistem.....	28
4.2.1 Kebutuhan Fungsional.....	28
4.2.2 Kebutuhan Non-Fungsional.....	29
4.2.2.1 Kebutuhan Perangkat Keras.....	29
4.2.2.2 Kebutuhan Perangkat Lunak.....	29
BAB 5 PERANCANGAN DAN IMPLEMENTASI.....	31
5.1 Perangkat Keras.....	31
5.1.1 Perancangan Perangkat Keras.....	31
5.1.2 Implementasi Perangkat Keras.....	32

5.2 Perangkat Lunak.....	33
5.2.1 Perancangan Perangkat Lunak.....	33
5.2.1.1 Pengambilan Gambar.....	34
5.2.1.2 Pengisian Nama Dosen.....	35
5.2.1.3 Pendeteksian Wajah.....	36
5.2.1.4 Cek File Pengujian.....	37
5.2.1.5 <i>Resize</i> Area Deteksi.....	38
5.2.1.6 Deteksi Warna Kulit.....	39
5.2.1.7 Pengenalan Wajah.....	42
5.2.1.8 Pembacaan Data Sampel.....	43
5.2.2 Implementasi Perangkat Lunak.....	45
5.2.2.1 Pengambilan Gambar.....	45
5.2.2.2 Pengisian Nama Dosen.....	46
5.2.2.3 Pendeteksian Wajah.....	47
5.2.2.4 Cek File Pengujian.....	49
5.2.2.5 <i>Resize</i> Area Deteksi.....	49
5.2.2.6 Deteksi Warna Kulit.....	50
5.2.2.7 Pengenalan Wajah.....	50
5.2.2.8 Pembacaan Data Sampel.....	52
BAB 6 PENGUJIAN DAN ANALISIS.....	55
6.1 Pengujian Pengambilan Gambar.....	55
6.1.1 Tujuan Pengujian.....	55
6.1.2 Pelaksanaan Pengujian.....	55
6.1.3 Prosedur Pengujian.....	55
6.1.4 Hasil Pengujian.....	55
6.1.5 Analisis Hasil Pengujian.....	57
6.2 Pengujian <i>Input</i> Nama Dosen.....	57
6.2.1 Tujuan Pengujian.....	57
6.2.2 Pelaksanaan Pengujian.....	57

6.2.3	Prosedur Pengujian.....	57
6.2.4	Hasil Pengujian.....	58
6.2.5	Analisis Hasil Pengujian.....	58
6.3	Pengujian Akurasi Pendeteksian Wajah.....	59
6.3.1	Tujuan Pengujian.....	59
6.3.2	Pelaksanaan Pengujian.....	59
6.3.3	Prosedur Pengujian.....	59
6.3.4	Hasil Pengujian.....	60
6.3.5	Analisis Hasil Pengujian.....	62
6.4	Pengujian Akurasi Pengenalan Wajah.....	63
6.4.1	Tujuan Pengujian.....	63
6.4.2	Pelaksanaan Pengujian.....	63
6.4.3	Prosedur Pengujian.....	63
6.4.4	Hasil Pengujian.....	64
6.4.5	Analisis Hasil Pengujian.....	70
6.5	Pengujian Waktu Pendeteksian dan Pengenalan.....	76
6.5.1	Tujuan Pengujian.....	76
6.5.2	Pelaksanaan Pengujian.....	76
6.5.3	Prosedur Pengujian.....	76
6.5.4	Hasil Pengujian.....	76
6.5.5	Analisis Hasil Pengujian.....	78
BAB 7	PENUNTUP.....	80
7.1	Kesimpulan.....	80
7.2	Saran.....	80
	DAFTAR PUSTAKA.....	81
	LAMPIRAN A KODE PROGRAM.....	83

DAFTAR TABEL

Tabel 2.1 Spesifikasi Raspberry Pi 3 Model B.....	19
Tabel 5.1 Keterangan pin raspberry pi yang terhubung ke pushbutton.....	32
Tabel 5.2 Deteksi warna kulit.....	40
Tabel 5.3 Kode program pengambilan gambar.....	45
Tabel 5.4 Kode program pengisian nama dosen.....	46
Tabel 5.5 Kode program pendeteksian wajah.....	47
Tabel 5.6 Kode program cek file pengujian.....	49
Tabel 5.7 Kode program <i>resize</i> area deteksi.....	49
Tabel 5.8 Kode program deteksi warna kulit.....	50
Tabel 5.9 Kode program pengenalan wajah.....	50
Tabel 5.10 Kode program pembacaan data sampel.....	52
Tabel 6.1 Hasil pengujian pengambilan gambar.....	56
Tabel 6.2 Hasil pengujian pengisian nama dosen.....	58
Tabel 6.3 Hasil pengujian akurasi deteksi sistem.....	60
Tabel 6.4 Hasil pengujian akurasi rekognisi sistem pada pagi hari.....	64
Tabel 6.5 Hasil pengujian akurasi rekognisi sistem pada siang hari.....	66
Tabel 6.6 Hasil pengujian akurasi rekognisi sistem pada malam hari.....	68
Tabel 6.7 Pengujian waktu deteksi pada sistem presensi.....	77
Tabel 6.8 Pengujian waktu rekognisi pada sistem presensi.....	78

DAFTAR GAMBAR

Gambar 2.1 Beberapa perangkat keras di dalam <i>smart class</i>	7
Gambar 2.2 Contoh citra digital.....	8
Gambar 2.3 Dekomposisi citra RGB.....	9
Gambar 2.4 Konversi citra RGB ke citra <i>grayscale</i>	10
Gambar 2.5 (A) dan (B) merupakan fitur dua persegi panjang, (C) fitur tiga persegi panjang, dan (D) fitur empat persegi panjang.....	11
Gambar 2.6 Gambar integral.....	12
Gambar 2.7 Jumlah piksel di area persegi panjang D dapat diperoleh dengan cara $4+1-(2+3)$	12
Gambar 2.8 Proses pencarian wajah menggunakan fitur persegi panjang.....	13
Gambar 2.9 Proses pengklasifikasian <i>cascade</i>	14
Gambar 2.10 Pembagian wilayah pada gambar menjadi 8×8 blok.....	14
Gambar 2.11 Proses penentuan nilai biner pada area lokal.....	15
Gambar 2.12 Bentuk akhir histogram <i>local binary patterns</i>	15
Gambar 2.13 Beberapa bentuk operasi LBP dengan skala berbeda.....	16
Gambar 2.14 Makna dibalik pola biner yang dideteksi.....	16
Gambar 2.15 Pola LBP yang seragam menggunakan 8 titik sampling.....	17
Gambar 2.16 Logitech HD Webcam C310.....	18
Gambar 2.17 Raspberry Pi 3 Model B.....	19
Gambar 3.1 Tahap-tahap metodologi penelitian.....	21
Gambar 3.2 Gambaran sistem secara umum.....	24
Gambar 4.1 Ruang kelas tempat pelaksanaan penelitian.....	27
Gambar 5.1 Perancangan perangkat keras.....	31
Gambar 5.2 Foto alat.....	32
Gambar 5.3 Skenario penerapan sistem di dalam kelas (tampak samping).....	33
Gambar 5.4 Skenario penerapan sistem di dalam kelas (tampak depan).....	33

Gambar 5.5 <i>Flowchart</i> pengambilan gambar.....	34
Gambar 5.6 Lokasi penyimpanan gambar hasil tangkapan.....	34
Gambar 5.7 <i>Flowchart</i> pengisian nama dosen.....	35
Gambar 5.8 <i>Flowchart</i> pendeteksian wajah.....	36
Gambar 5.9 Lokasi penyimpanan gambar wajah yang terdeteksi.....	37
Gambar 5.10 <i>Flowchart</i> cek file pengujian.....	38
Gambar 5.11 <i>Flowchart</i> <i>resize</i> area deteksi.....	39
Gambar 5.12 Hasil <i>resize</i> area deteksi.....	39
Gambar 5.13 Ditemukan deteksi palsu dalam <i>face detection</i> OpenCV library.....	40
Gambar 5.14 <i>Flowchart</i> deteksi warna kulit.....	41
Gambar 5.15 <i>Flowchart</i> pengenalan wajah.....	42
Gambar 5.16 <i>Flowchart</i> pengenalan wajah (lanjutan).....	43
Gambar 5.17 Lokasi penyimpanan data sampel mahasiswa.....	44
Gambar 5.18 <i>Flowchart</i> pembacaan data sampel.....	44
Gambar 6.1 Grafik hasil pengenalan wajah di waktu pagi.....	70
Gambar 6.2 Grafik hasil pengenalan wajah di waktu siang.....	72
Gambar 6.3 Grafik hasil pengenalan wajah di waktu malam.....	74
Gambar 6.4 Grafik rata-rata waktu pendeteksian oleh sistem.....	77
Gambar 6.5 Grafik rata-rata waktu pengenalan oleh sistem.....	78



agar bisa mencegah kecurangan tanda tangan yang bisa saja terjadi sewaktu-waktu.

Berdasarkan teknologi rekognisi wajah yang ada di dalam *computer vision*, maka penerapan rekognisi wajah mahasiswa bisa dijadikan sebagai metode alternatif di dalam memantau kehadiran mahasiswa. Penerapan teknologi rekognisi wajah ini merupakan salah satu wujud dari penerapan teknologi *smart class*, dimana sarana pembelajaran di dalam kelas tersebut akan diintegrasikan dengan teknologi-teknologi pembelajaran seperti komputer, perangkat lunak khusus, teknologi *audience response system*, *networking*, dan kemampuan audio/visual (Block *et al.*, 2015).

Dalam menerapkan salah satu fitur dari fungsi sistem *smart class* ini, nantinya sistem dapat melakukan pendeteksian dan pengenalan wajah. Proses pendeteksian wajah dilakukan dengan menggunakan metode Viola Jones yang berfungsi untuk menemukan lokasi wajah mahasiswa dari gambar *scene* ruang kelas. Metode Viola Jones dipilih karena berdasarkan penelitian Balchoh *et al.* (2012) disebutkan bahwa metode Viola Jones merupakan metode yang paling efisien di dalam melakukan pendeteksian wajah. Sedangkan di dalam proses pengenalan wajah akan dilakukan dengan menggunakan metode *Local Binary Patterns Histograms* (LBPH). Metode *Local Binary Patterns Histograms* dipilih karena menurut Huang *et al.* (2011) metode *Local Binary Patterns Histograms* ini merupakan metode klasifikasi yang bisa memberikan toleransi terhadap perubahan cahaya monotonik di dalam piksel gambar. Dari proses pendeteksian dan pengklasifikasian wajah ini nantinya akan dihasilkan suatu label. Label tersebut akan digunakan untuk mengetahui nama mahasiswa. Dari label-label yang terkumpul nantinya akan diketahui nama-nama mahasiswa yang hadir di dalam jam perkuliahan tersebut. Sehingga dosen pengajar dapat mengetahui siapa saja yang hadir dan yang tidak hadir berdasarkan daftar nama mahasiswa yang terdeteksi.

1.2 Rumusan Masalah

1. Bagaimana nilai akurasi deteksi wajah menggunakan Viola Jones ?
2. Bagaimana nilai akurasi rekognisi wajah menggunakan *Local Binary Patterns Histograms* yang dilakukan pada pagi, siang dan malam hari ?
3. Berapa waktu komputasi untuk deteksi dan rekognisi ?

1.3 Tujuan

1. Mengetahui nilai akurasi deteksi wajah menggunakan Viola Jones.
2. Mengetahui nilai akurasi rekognisi wajah menggunakan *Local Binary Patterns Histograms* yang dilakukan pada pagi, siang, dan malam hari.
3. Mengetahui waktu komputasi untuk deteksi dan rekognisi.

1.4 Manfaat

1. Dengan adanya sistem presensi berdasarkan rekognisi wajah dapat membantu menyelesaikan kasus kecurangan tanda tangan oleh mahasiswa.
2. Membantu memotivasi mahasiswa untuk bisa hadir terus di setiap jam perkuliahan.

1.5 Batasan Masalah

1. Setiap mahasiswa diarahkan untuk menghadapkan wajahnya ke depan kamera.
2. Untuk membatasi jumlah data pengujian, maka jumlah mahasiswa yang diujikan di dalam kelas adalah sebanyak 6 orang.
3. Jarak kamera dari bangku terdepan 246 cm dengan ketinggian 212 cm.
4. Pencahayaan ruangan (diukur menggunakan sensor digital lux meter).
 1. Pagi : 180 lux s/d 183 lux.
 2. Siang : 118 lux s/d 136 lux.
 3. Malam : 114 lux s/d 122 lux.
5. Mahasiswa yang menggunakan kacamata diminta untuk melepaskan kacamatanya saat proses presensi berlangsung.

1.6 Sistematika Pembahasan

Bab I : Pendahuluan

Pada bagian ini menjelaskan mengapa permasalahan ini diangkat sehingga penelitian ini dianggap penting dilakukan. Dalam bab ini telah berisikan latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika penulisan laporan.

Bab II : Landasan Kepustakaan

Pada bab ini berisi uraian dan pembahasan mengenai konsep dan teori yang digunakan di dalam pelaksanaan penelitian ini.

Bab III : Metodologi Penelitian

Bab ini berisi sistematika yang digunakan selama melaksanakan penelitian. Setiap proses pengerjaannya akan diuraikan sehingga dapat memudahkan dalam menyelesaikan penelitian.

Bab IV : Rekayasa Kebutuhan

Pada bagian ini dijelaskan beberapa analisis kebutuhan yang diperlukan selama penelitian, baik dari sisi fungsionalitas, non-fungsionalitas, perangkat keras dan perangkat lunak.

Bab V : Perancangan dan Implementasi

Di bab ini menjelaskan proses perancangan dan implementasi dari sistem yang dibuat. Proses tersebut akan dijelaskan secara keseluruhan dari sisi perangkat lunak dan perangkat keras. Beberapa gambar juga disertakan sehingga dapat memvisualisasikan bentuk akhir dari sistem.

Bab VI : Pengujian dan Analisis

Di bab ini akan dilakukan pengujian dan analisis terhadap kinerja sistem yang telah berhasil dibuat.

Bab VII : Penutup

Di bagian ini akan diuraikan kesimpulan dari hasil penelitian yang sudah dilakukan. Beberapa saran akan disertakan sebagai masukan bagi penelitian selanjutnya dalam meningkatkan fungsionalitas dari sistem ini.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini akan membahas mengenai kajian pustaka dan dasar teori yang digunakan selama pengerjaan skripsi ini.

2.1 Kajian Pustaka

Pada penelitian sebelumnya yang berjudul “*Automatic Attendance System Using Face Recognition*” oleh Choudhary *et al.* (2016) telah melakukan penelitian mengenai pengisian presensi secara otomatis. Proses pengisian presensi tersebut diterapkannya dengan cara mengambil gambar *scene* ruangan kelas sebanyak dua kali, pertama ketika pelajaran dimulai dan yang kedua ketika pelajaran telah selesai. Hal itu dimaksudkan untuk memastikan bahwa setiap siswa tetap berada di ruangan kelasnya hingga akhir jam belajar selesai. Dari gambar yang diperoleh sistem, ia mencoba menerapkan proses perbaikan terhadap gambar supaya lebih mudah ketika masuk ke tahap proses pencocokkan. Selanjutnya, proses pendeteksian wajah diterapkannya menggunakan algoritme Viola Jones dan proses pengenalan wajah diterapkannya menggunakan algoritme PCA (*Principal Component Analysis*). Setelah melewati kedua tahap tersebut, ia mencoba membandingkan hasil pengolahan gambar di awal dan akhir pelajaran. Apabila siswa terlihat pada kedua waktu tersebut, maka siswa tersebut akan ditandai sebagai hadir di dalam basis data. Akan tetapi, apabila siswa yang ternyata hadir tetapi tidak dikenali oleh sistem, maka pengajar bisa mengubah status kehadirannya secara manual. Kelebihan yang ditawarkan di dalam penelitian ini adalah bahwa sistem presensinya tersebut dapat menghemat waktu dan usaha dari penggunaannya, terutama jika pengajar memiliki jumlah siswa yang cukup banyak. Selain itu, sistem presensi ini bisa diterapkan di tempat lain – seperti tempat kerja – untuk menggantikan sistem presensi yang telah ada. Kelemahan di dalam penelitian ini adalah tidak menyertakan secara rinci mengenai akurasi hasil sistem presensinya.

Penelitian serupa juga dilakukan oleh Balcoh *et al.* (2012) dalam penelitiannya yang berjudul “*Algorithm for Efficient Attendance Management: Face Recognition based Approach*”. Di dalam sistem presensi otomatisnya tersebut, ia mencoba menerapkan proses *histogram normalization*, *noise removal*, dan klasifikasi warna kulit untuk bisa menghindari *false detection* ketika menerapkan pendeteksian wajah. Penerapan klasifikasi warna kulit tersebut dilakukannya sebelum memasuki fase pendeteksian wajah. Secara keseluruhan, sistem presensinya menunjukkan hasil pendeteksian wajah untuk kategori wanita yang berjilbab sebesar 40%, orang yang berkumis sebesar 75%, dan orang yang tanpa menggunakan penutup kepala sebesar 95%. Sedangkan untuk pengenalan wajah, sistem presensinya menunjukkan hasil pengenalan wajah yang mampu

mengenali 2% dari wanita yang berjilbab, 63% orang yang berkumis, dan 85% orang yang tanpa menggunakan penutup kepala. Kelebihan yang ditawarkan di dalam sistem presensinya ini adalah adanya informasi tambahan mengenai informasi ruangan kelas yang digunakan, tanggal pembelajaran dan waktu pengambilan gambar. Sedangkan kekurangan yang ada di dalam penelitian ini adalah masih lemahnya akurasi di dalam pengenalan terhadap wanita yang menggunakan jilbab.

Untuk pemilihan algoritme rekognisi wajah, SudhaNarang *et al.* (2018) telah melakukan penelitian untuk membandingkan algoritme pengenalan yang paling akurat dan efisien. Dalam penelitiannya yang berjudul "*Comparison of Face Recognition Algorithms Using Opencv for Attendance System*", ia mengimplementasikan algoritme Eigenface, Fisherface, dan *Local Binary Patterns Histograms* (LBPH) di dalam sistem presensinya kemudian membandingkan mana yang memberikan hasil yang paling akurat dan efisien. Dengan menggunakan 50 sampel wajah untuk per orangnya, hasil yang didapatkan menunjukkan bahwa algoritme *Local Binary Patterns Histograms* (LBPH) mengungguli algoritme yang lain dengan tingkat *confidence* 2-5 dan *noise* yang minimum, sedangkan untuk algoritme fisherface didapatkan tingkat *confidence* sebesar 100-400, lalu eigenface sebesar 2000-3000 (nilai *confidence* semakin kecil semakin baik). Ia juga menyebutkan bahwa proses akurasi LBPH untuk memproses data secara *real-time* lebih baik dibandingkan algoritme lain dikarenakan tingkat kompleksitasnya yang lebih sedikit. Selain itu, perubahan dalam pencahayaan tidak menyebabkan masalah untuk algoritme *Local Binary Patterns Histograms* (LBPH) ini dalam merekognisi wajah siswa. Kekurangan di dalam penelitian ini adalah hanya menjelaskan cara kerja algoritme secara umum serta tidak menyertakan persamaan matematis untuk masing-masing algoritme eigenface, fisherface dan *Local Binary Patterns Histograms* (LBPH).

Dalam penelitian ini, metode pendeteksian wajah yang digunakan adalah metode Viola Jones, sedangkan untuk pengenalan wajah digunakan metode *Local Binary Patterns Histograms* (LBPH) yang mana kedua metode tersebut telah tersedia di dalam OpenCV *library*. Sebagai tambahan, untuk menghindari deteksi palsu ketika mendeteksi wajah di dalam gambar, digunakan deteksi piksel kulit untuk memastikan bahwa gambar wajah yang dideteksi oleh metode Viola Jones benar-benar merupakan gambar wajah. Hal ini disebabkan masih ditemukannya deteksi wajah palsu di dalam metode Viola Jones OpenCV *library*.

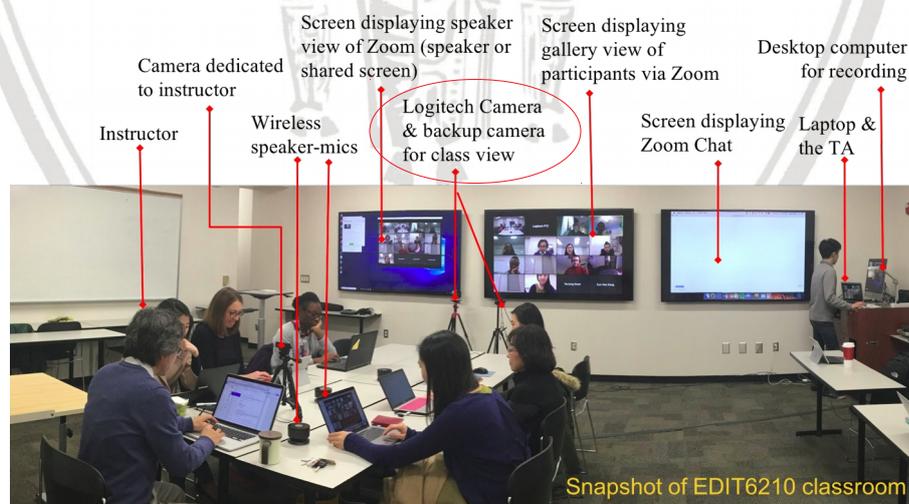
2.2 Dasar Teori

2.2.1 Smart Class

Smart Class merupakan sebuah revolusi di dalam sistem pembelajaran dengan memanfaatkan teknologi-teknologi pendukung selama proses belajar mengajar. Di dalam penerapannya, sistem *smart class* lebih menekankan kepada terciptanya suatu lingkungan belajar yang lebih kaya dan efektif antara guru dan siswa. Sehingga dengan pemanfaatan teknologi inovatif tersebut dapat membantu guru dalam menyampaikan ilmunya dan siswa dengan mudah dapat memahami apa yang disampaikan oleh guru (Menon, 2015).

Dari segi penerapannya, teknologi perangkat keras pada *smart class* terbagi menjadi tiga, yaitu penerapan perangkat keras untuk guru, siswa, dan infrastruktur kelas. Guru membutuhkan perangkat keras seperti PC atau laptop untuk bisa memasukkan nilai, data pribadi siswa, presentasi kelas, dan melakukan tugas-tugas administrasi yang diperlukan untuk kepentingan kelas. Sedangkan siswa membutuhkan perangkat keras seperti laptop atau tablet yang bisa digunakannya untuk mencari literatur secara *online* di internet. Sedangkan infrastruktur kelas, harus menyediakan titik akses wifi bagi para siswa dan sebuah *smart board* atau perangkat sejenis (seperti proyektor atau layar TV besar) sebagai media interaksi bagi para siswa agar lebih mudah memvisualisasikan informasi yang disampaikan oleh guru (Block *et al.*, 2015).

Dalam penelitian ini akan diterapkan perangkat keras untuk guru dalam mengotomasi presensi kelas berdasarkan teknologi rekognisi wajah dari *computer vision* menggunakan kamera Logitech C310. Sebagai ilustrasi, berikut beberapa tampilan perangkat keras yang ada di dalam *smart class* dapat dilihat pada Gambar 2.1



Gambar 2.1 Beberapa perangkat keras di dalam *smart class*

Sumber: (Xigui *et al.*, 2018)

2.2.2 Citra Digital

Citra digital merupakan sebuah fungsi dua dimensi $f(x,y)$ dengan nilai x dan y sebagai titik koordinat spasial pada gambar dan nilai f sebagai tingkat intensitas untuk titik lokasi (x,y) tertentu. Besarnya nilai intensitas tersebut (atau yang lebih kita kenal dengan istilah piksel) telah dibatasi pada tingkatan tertentu (Gonzales and Woods, 2002). Umumnya rentang nilainya berkisar antara 0 hingga 255 (Shapiro and Stockman, 2000). Berikut ini contoh citra digital beserta nilai intensitas pikselnya ditunjukkan pada Gambar 2.2



Gambar 2.2 Contoh citra digital

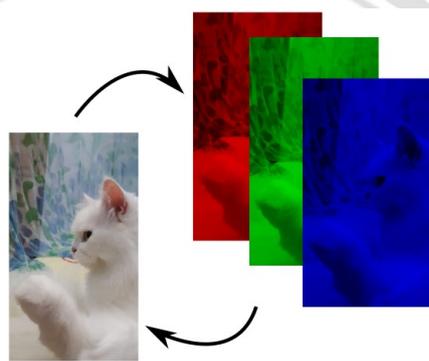
Dengan menerapkan sejumlah algoritme komputer, citra digital banyak digunakan untuk keperluan tertentu, seperti untuk mendapatkan gambar berkualitas tinggi atau mendapatkan karakteristik deskriptif dari suatu gambar. Hal ini dilakukan dikarenakan kemampuan komputer yang tidak sama dengan sistem visual manusia yang dapat beradaptasi dengan berbagai keadaan dan mampu menangkap makna dari suatu gambar secara langsung. Oleh karenanya, untuk bisa memanfaatkan citra digital untuk tujuan khusus, maka pengolahan terhadap citra digital perlu dilakukan lebih lanjut agar bisa memberikan hasil sesuai dengan apa yang diinginkan (Zhou, Wu and Zhang, 2010).

Berikut ini beberapa aplikasi pengolahan citra digital yang telah tersedia secara luas, di antaranya adalah pada bidang robotika/sistem cerdas, pencitraan medis, penginderaan jauh, fotografi, dan forensik (Zhou, Wu and Zhang, 2010). Penelitian ini akan menggunakan citra digital sebagai *input* dalam mengotomasi presensi kelas.

2.2.3 Citra RGB

Citra RGB merupakan sebuah model warna yang digunakan untuk merepresentasikan suatu warna pada citra digital. Warna RGB diusulkan oleh Thomas Young dan Herman Helmholtz pada awal abad pertengahan kesembilan belas yang lebih dikenal dengan sebutan penglihatan warna trikromatik. Model warna RGB didasari atas persepsi manusia dalam mengenali warna dan memiliki dimensi yang lebih tinggi daripada citra abu-abu, dimana dalam model ini, satu piksel citra digital akan terdiri dari tiga kanal warna yang berbeda, yaitu kanal merah, hijau, dan biru (Zhou, Wu *and* Zhang, 2010).

Berikut ini merupakan contoh bentuk dekomposisi dari citra RGB yang ditunjukkan oleh Gambar 2.3.



Gambar 2.3 Dekomposisi citra RGB

2.2.4 Konversi Citra RGB ke YCbCr

Warna YCbCr merupakan sebuah model warna yang terdiri dari parameter Y, Cb, dan Cr. Parameter Y digunakan untuk menyimpan nilai kecerahan warna (*luminance*) dari suatu piksel dan parameter Cb, Cr digunakan untuk menyimpan nilai warna *chrominance* nya (Kolkur *et al.*, 2016).

Warna YCbCr banyak digunakan pada proses klasifikasi warna piksel kulit manusia di dalam suatu gambar. Proses klasifikasi tersebut dilakukan dengan memperhatikan rentang nilai dari parameter Cb dan Cr nya. Jika nilai *chrominance* (Cb dan Cr) suatu piksel termasuk ke dalam rentang nilai tersebut, maka dapat dikatakan piksel tersebut merupakan piksel warna kulit manusia. Jika tidak maka piksel tersebut berarti bukan merupakan piksel warna kulit manusia (Patravali, Waykule *and* Katre, 2014).

Untuk dapat mengetahui ada atau tidaknya piksel warna kulit manusia, maka citra RGB pertama kali harus dikonversi ke dalam citra YCbCr. Proses konversi citra RGB ke dalam citra YCbCr dapat dilihat pada Persamaan 2.1 (OpenCV, 2018) . Sedangkan Persamaan 2.2 merupakan rentang nilai parameter

C_b dan C_r yang dijadikan sebagai acuan dalam mendeteksi piksel warna kulit manusia (Patravali, Waykule and Katre, 2014).

$$Y = (0.299 \times R) + (0.587 \times G) + (0.114 \times B)$$

$$C_b = (B - Y) \times 0.564 + 128 \quad (2.1)$$

$$C_r = (R - Y) \times 0.713 + 128$$

$$77 \leq C_b \leq 127 \text{ dan } 133 \leq C_r \leq 173 \quad (2.2)$$

Keterangan:

Y : *lumiance* (seberapa terang intensitas cahaya yang dihasilkan).

C_b : *chrominance* yang dihasilkan dari selisih warna biru dan nilai referensi.

C_r : *chrominance* yang dihasilkan dari selisih warna merah dan nilai referensi.

2.2.5 Konversi Citra RGB ke *Grayscale*

Warna *grayscale* merupakan sebuah model warna yang terdiri dari satu kanal piksel. Warna ini diambil dari penyederhanaan tiga kanal citra RGB yang dikonversi menjadi satu kanal piksel citra, dimana dalam kanal ini, warna piksel hitam sempurna dilambangkan dengan angka 0 dan warna putih sempurna dilambangkan dengan angka 255. Contoh citra RGB yang dikonversi menjadi citra *grayscale* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Konversi citra RGB ke citra *grayscale*

Konversi citra RGB ke *grayscale* banyak digunakan pada pengolahan citra digital karena dari tingkat kompleksitasnya yang lebih rendah dibandingkan citra RGB. Untuk bisa memperoleh citra *grayscale*, maka piksel pada citra RGB perlu

dikonversi ke dalam piksel citra *grayscale* dengan menggunakan Persamaan 2.3 (OpenCV, 2018) .

$$Grayscale = (0.299 \times R) + (0.587 \times G) + (0.114 \times B) \quad (2.3)$$

Keterangan:

R : nilai warna piksel merah

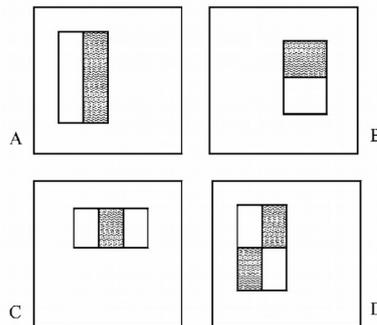
G : nilai warna piksel hijau

B : nilai warna piksel biru

2.2.6 Viola Jones

Metode Viola Jones merupakan sebuah *framework* yang berisikan beberapa algoritme yang digunakan untuk mendeteksi wajah seseorang di dalam gambar. Metode Viola Jones ini mampu memproses gambar dengan begitu cepat dengan tingkat pendeteksian wajah yang tinggi (Viola and Jones, 2004). Dari sekian banyak metode pendeteksian wajah yang ada, metode Viola Jones dianggap sebagai metode yang paling efisien dalam melakukan pendeteksian wajah seseorang di dalam sebuah gambar (Balcoh *et al.*, 2012).

Metode Viola Jones bekerja dengan memanfaatkan tiga jenis fitur persegi panjang: (1) Pertama fitur dua persegi panjang, (2) Kedua fitur tiga persegi panjang, dan (3) Ketiga fitur empat persegi panjang. yang masing-masingnya memiliki bentuk dan ukuran yang sama. Gambar 2.5 merupakan bentuk fitur yang digunakan pada metode Viola Jones ini. Dimana nilainya merupakan selisih antara jumlah piksel di area putih dikurangi jumlah piksel di area abu-abu.



Gambar 2.5 (A) dan (B) merupakan fitur dua persegi panjang, (C) fitur tiga persegi panjang, dan (D) fitur empat persegi panjang

Sumber: Viola and Jones (2004)

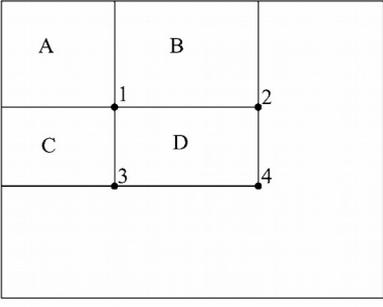
Fitur tersebut digunakan untuk mendeteksi adanya model wajah di dalam gambar. Caranya adalah dengan menggeser fitur persegi panjang tersebut di sepanjang titik lokasi gambar hingga ditemukan adanya wajah seseorang. Untuk mempercepat dan memudahkan proses itu, terlebih dahulu dilakukan proses representasi gambar menjadi gambar integral. Proses di dalam representasi tersebut dapat dilihat pada Gambar 2.6.

Original image	Integral image
1 1 1 1	1 2 3 4
1 1 1 1	2 4 6 8
1 1 1 1	3 6 9 10
1 1 1 1	4 8 12 16

Gambar 2.6 Gambar integral

Pada Gambar 2.6, nilai pada kotak merah merupakan penjumlahan dari $1+1+1+1+1+1 = 6$. Begitu pula jika kotak merah digeser terus. Blok berwarna kuning digunakan untuk menjumlahkan piksel yang bersangkutan dalam menentukan nilai integral pada titik (x,y) pada gambar.

Setelah berhasil merepresentasi gambar menjadi gambar integral, maka proses selanjutnya adalah mengukur perbandingan jumlah piksel pada masing-masing area persegi panjang. Untuk mengetahui jumlah piksel di sebuah area persegi panjang bisa dengan memperhatikan nilai integral pada titik acuan yang diinginkan, seperti yang ditunjukkan pada Gambar 2.7.



Gambar 2.7 Jumlah piksel di area persegi panjang D dapat diperoleh dengan cara $4+1-(2+3)$

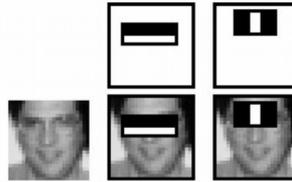
Sumber: Viola and Jones (2004)

Gambar 2.7 merupakan sebuah contoh dalam menghitung jumlah piksel di area persegi panjang. Jika menggunakan Gambar 2.6 sebagai dasar angka integralnya, maka akan diperoleh hasil sebagai berikut, $16 + 4 - (8 + 8) = 4$.

Setelah mendapatkan jumlah piksel pada masing-masing area persegi panjang, maka proses berikutnya adalah membandingkan nilai piksel tersebut.



Sebagai contoh, Gambar 2.8 menunjukkan proses penggunaan beberapa fitur yang digunakan dalam mendeteksi wajah, dengan asumsi bahwa jumlah piksel di area sekitar mata jauh lebih gelap dibanding jumlah piksel di area sekitar hidung dan pipi.

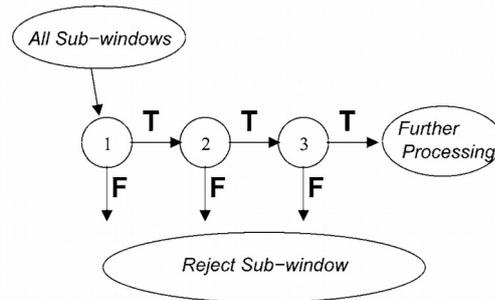


Gambar 2.8 Proses pencarian wajah menggunakan fitur persegi panjang

Sumber: Viola *and* Jones (2004)

Permasalahannya adalah setidaknya ada hampir sekitar 160.000 fitur persegi panjang berbeda yang tersedia dalam menemukan model wajah ini, sehingga apabila menggunakan keseluruhannya hanya akan menghabiskan banyak waktu. Metode Viola Jones telah menggunakan pendekatan lain di dalam menentukan fitur tersebut, yaitu dengan mengkombinasikan sejumlah pengklasifikasi lemah untuk menemukan fitur persegi panjang dengan tingkat kesalahan terendah pada sekumpulan gambar positif dan negatif wajah dengan bantuan algoritme pembelajaran AdaBoost. Sehingga dengan menerapkan AdaBoost ini dapat secara efektif mencari fitur yang bagus namun tetap memiliki variansi yang signifikan.

Selain itu, untuk bisa mengurangi waktu komputasi dalam menggunakan setiap fitur tersebut, maka diterapkanlah proses ketiga, yaitu *cascade*. *Cascade* merupakan proses pengklasifikasian fitur dengan mengujinya secara bertahap di setiap sub jendela pada gambar uji. Apabila dalam tahap pengklasifikasi pertama pada sub jendela gambar tersebut bernilai T (*True*), maka sub jendela gambar tersebut akan diuji dengan pengklasifikasi kedua dengan fitur pengklasifikasi yang lebih banyak. Begitu pula seterusnya hingga berhasil melewati tahap pengklasifikasi ketiga yang lebih kompleks. Jika sub jendela gambar berhasil melewati ketiga pengklasifikasi tadi, maka titik lokasi pada sub jendela gambar tersebut akan dianggap sebagai sebuah wajah. Akan tetapi, apabila dalam prosesnya ditemukan nilai F (*False*), maka seketika lokasi pada sub jendela gambar itu akan dilewati untuk proses pengklasifikasi *cascade* lebih lanjut dan dianggap sebagai bukan gambar wajah. Gambar 2.9 mengilustrasikan proses pengklasifikasian *cascade* dalam menemukan lokasi wajah pada gambar.



Gambar 2.9 Proses pengklasifikasian *cascade*

Sumber: Viola and Jones (2004)

2.2.7 Local Binary Patterns Histograms (LBPH)

Metode *Local Binary Patterns Histograms* merupakan salah satu metode yang dapat digunakan untuk mengekstraksi fitur yang paling penting dari gambar wajah untuk keperluan rekognisi. Metode ini pertama kali diperkenalkan oleh Ojala *et al.* pada tahun 1996 yang kemudian digunakan banyak pada bidang pengolahan citra dan visi komputer, khususnya pada aplikasi analisis citra wajah, seperti pendeteksian wajah, pengenalan wajah, dan analisis ekspresi wajah. Beberapa kelebihan yang ditawarkan di dalam metode ini adalah kesederhanaan komputasinya dan adanya toleransi terhadap perubahan cahaya monotonik di dalam suatu gambar (Huang *et al.*, 2011).

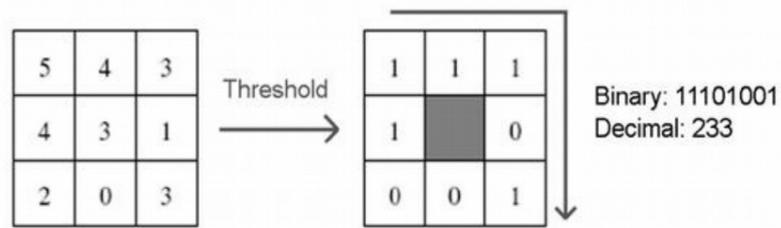
Untuk bisa mengenali wajah, pertama kali yang dilakukan metode ini adalah membagi wilayah pada gambar menjadi beberapa blok kecil, seperti yang ditunjukkan pada Gambar 2.10.



Gambar 2.10 Pembagian wilayah pada gambar menjadi 8x8 blok

Sumber: Rahim *et al.*, (2013)

Pada Gambar 2.10, wajah dibagi menjadi 8x8 wilayah, dimana di dalam setiap blok wilayah terdiri dari piksel-piksel. Dalam proses kedua LBP, piksel-piksel tersebut akan diolah menjadi angka biner dengan cara memperhatikan nilai intensitas piksel pada setiap 3x3 piksel tetangganya. Ilustrasi proses pengolahan piksel tersebut dapat dilihat pada Gambar 2.11.

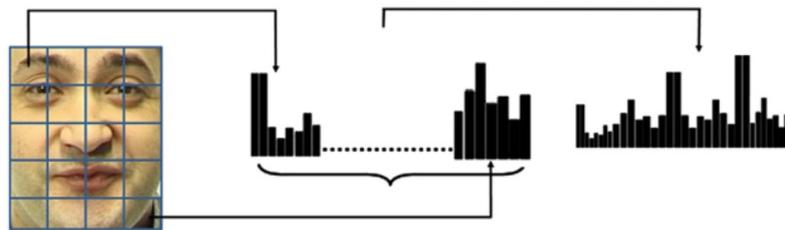


Gambar 2.11 Proses penentuan nilai biner pada area lokal

Sumber: Rahim *et al.*, (2013)

Pada Gambar 2.11, diketahui bahwa piksel tengahnya adalah bernilai 3. Dengan mengacu kepada angka tersebut, maka nilai setiap piksel tetangganya akan dibandingkan dengan piksel tengah tersebut. Apabila nilai piksel tetangga sama dengan atau lebih besar dari piksel tengah, maka dilabelkan dengan angka 1. Namun apabila sebaliknya, maka akan diberi label 0. Setelah memperoleh nilai biner untuk masing-masing tetangga, maka selanjutnya adalah menggabungkan angka biner tersebut menjadi satu *string* biner, dimulai dari piksel pojok kiri atas kemudian memutar searah jarum jam. Dan terakhir mengkonversikan nilai biner tersebut menjadi angka desimal.

Dalam proses LBP berikutnya, operasi *local binary patterns* tersebut akan diterapkan di setiap blok gambar. Dengan rentang nilai variasi antara 0 sampai 255, akan diperoleh sebuah histogram yang mewakili fitur di setiap blok gambar, dimana sumbu x menunjukkan variasi angka desimal (0 - 255), dan sumbu y menunjukkan frekuensi kemunculannya. Hasil akhir dari histogram yang terkumpul, akan dikumpulkan menjadi satu representasi histogram seperti yang ditunjukkan pada Gambar 2.12.

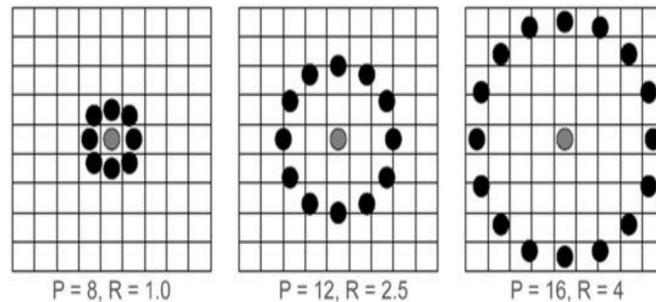


Gambar 2.12 Bentuk akhir histogram *local binary patterns*

Sumber: Norvig (2013)

Jika menerapkan operasi *local binary patterns* dengan ukuran tetangga 3x3 pada setiap blok gambar di Gambar 2.12 tentu akan sulit menemukan tekstur-tekstur penting pada gambar. Hal ini disebabkan karena ukuran operator yang terlalu kecil yang coba diterapkan pada gambar yang memiliki ukuran cukup besar. Oleh karena itu, operator LBP dikembangkan untuk bisa digunakan pada

lingkungan yang berbeda. Bentuk operator LBP yang dikembangkan tersebut ditunjukkan pada Gambar 2.13.

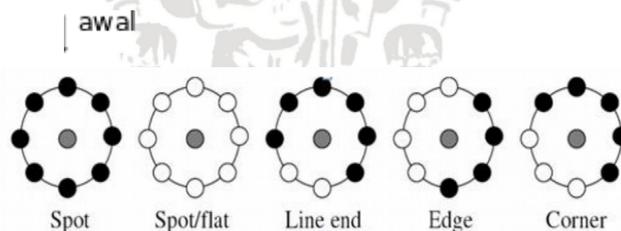


Gambar 2.13 Beberapa bentuk operasi LBP dengan skala berbeda

Sumber: Rahim *et al.*, (2013)

Pada Gambar 2.13, terlihat bahwa terdapat parameter P dan R yang bervariasi. P disitu merupakan jumlah titik sampling yang digunakan LBP, dan R adalah radius jaraknya dari piksel tengah. Secara *default*, operator dasar LBP ditunjukkan pada Gambar 2.13 paling kiri.

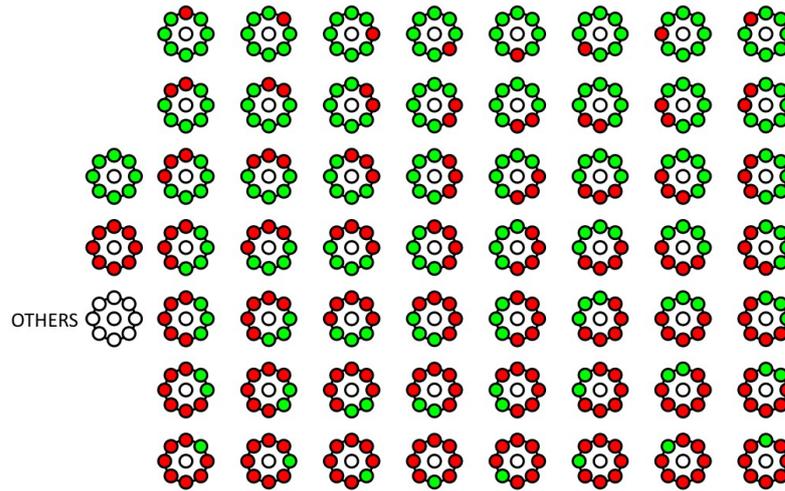
Selain itu, operator dasar LBP kembali dikembangkan untuk bisa hanya menemukan fitur penting saja dari pola biner yang terdeteksi. Pola biner dikatakan penting apabila memiliki pola seragam seperti yang ditunjukkan pada Gambar 2.14.



Gambar 2.14 Makna dibalik pola biner yang dideteksi

Sumber: Rahim *et al.*, (2013)

Pada Gambar 2.14, pola binernya disebut sebagai pola LBP yang seragam (*Uniform Pattern of LBP*) karena memiliki ciri khas berupa pola transisi biner 0 (hitam) ke 1 (putih) atau 1 (putih) ke 0 (hitam) yang tidak berlangsung lebih dari dua kali. Apabila transisi lebih dari dua kali, maka pola biner tersebut akan dikelompokkan ke dalam bin *others* (tidak seragam/tidak penting). Berikut contoh ilustrasi bentuk pola biner dengan menggunakan delapan titik sampling menggunakan pola LBP yang seragam ditunjukkan pada Gambar 2.15.



Gambar 2.15 Pola LBP yang seragam menggunakan 8 titik sampling

Sumber: Norvig (2013)

Pada Gambar 2.15, ada terdapat 59 fitur biner berbeda dengan menggunakan 8 titik sampling, dimana 1 diantaranya merupakan fitur *others*. Dibandingkan dengan yang tidak menggunakan pola seragam LBP, ada terdapat 256 fitur berbeda yang harus dikelompokkan. Penerapan pola seragam LBP tentu dapat menghemat waktu pemrosesan ketika mengelompokkan fitur LBP dibandingkan dengan yang tidak menggunakan pola seragam.

Akhirnya, setelah memperoleh representasi histogram dari gambar, maka proses akhir dari metode LBP adalah membandingkan histogram gambar uji dengan histogram gambar yang ada di dalam *database* dengan mengukur kesamaan (*distance*) antar histogram menggunakan persamaan 2.4.

$$x^2(S, M) = \sum_{j=1}^{k^2} \left(\sum_{i=1}^n \frac{(S_{i,j} - M_{i,j})^2}{(S_{i,j} + M_{i,j})} \right) \quad (2.4)$$

Keterangan:

S : histogram gambar uji

M : histogram gambar *database*

k^2 : total pembagian blok (baris \times kolom)

n : jumlah bin ($N \times (N-1) + 3$)

N : jumlah bit yang digunakan pada LBP

2.2.8 Webcamera

Webcamera merupakan sebuah perangkat keras berbentuk kamera mini yang dapat dihubungkan secara langsung ke perangkat komputer. Secara fungsional, webcamera digunakan sebagai media *stream* gambar secara *realtime*. Dari *streaming* tersebut, pengguna dapat mengambil gambar yang diinginkan kemudian menyimpannya ke media penyimpanan komputer atau mengirimkan *streaming*-nya ke media internet.

Webcamera banyak diterapkan dibidang-bidang spesifik, seperti *video monitoring*, *videocalling*, *input control devices*, dan *video security*. Pada penelitian ini, webcamera akan digunakan sebagai media pengambil gambar di ruangan kelas untuk keperluan sistem presensi. Jenis webcamera yang digunakan adalah Logitech HD Webcam C310 yang dapat dilihat pada Gambar 2.16.



Gambar 2.16 Logitech HD Webcam C310

Sumber: Logitech (2018)

2.2.9 Raspberry Pi 3 Model B

Raspberry Pi 3 Model B merupakan sebuah *single board computer* yang dikeluarkan *Raspberry foundation*. Model ini merupakan model pertama dari Raspberry pi generasi ketiga. Dimana sebelumnya produk terbarunya merupakan Raspberry Pi 2 Model B dan akhirnya pada bulan Februari 2016 digantikan oleh Raspberry Pi 3 Model B ini. Dalam *single board computer* ini telah tersedia spesifikasi yang cukup memadai di dalam melakukan pengolahan citra digital di dalam penelitian ini. Spesifikasi lengkapnya bisa dilihat pada Tabel 2.1. Sedangkan bentuk fisiknya dapat dilihat pada Gambar 2.17.

Tabel 2.1 Spesifikasi Raspberry Pi 3 Model B

CPU	Quad Core 1.2GHz Broadcom BCM2837 64bit
Memori	1GB RAM
WiFi	BCM43438 wireless LAN onboard
Bluetooth	Bluetooth Low Energy (BLE) onboard
Kartu jaringan internal	100 Base Ethernet
General purpose input / output	40-pin extended GPIO
USB	4 buah <i>port</i> USB 2
Audio dan video	Output stereo 4 kutub dan <i>port</i> composite video
HDMI	Full size HDMI
Port khusus	Port kamera CSI untuk menghubungkan ke Raspberry Pi Camera dan port display DSI untuk menghubungkan ke Raspberry Pi touchscreen display
Media penyimpanan	Micro SD untuk memuat sistem operasi dan menyimpan data
Sumber daya	Mikro USB sampai dengan 2,5 A
Sistem operasi yang didukung	NOOBS, Raspbian, dan lain-lain

Sumber: Raspberry Pi Foundation (2016)



Gambar 2.17 Raspberry Pi 3 Model B

Sumber: Raspberry Pi Foundation (2016)

2.2.10 Python Versi 3

Python merupakan bahasa pemrograman tingkat tinggi yang dapat digunakan untuk mengolah teks, angka, gambar, dan data file yang ada di dalam komputer. Selain itu, bahasa pemrograman ini juga memiliki dukungan perpustakaan yang cukup luas dan dapat digunakan pada sistem operasi yang berbeda. Dengan gaya penulisan dan dukungan perpustakaan yang cukup luas tersebut menjadikan python bahasa pemrograman pilihan di dalam melakukan pengolahan citra digital pada skripsi ini.

2.2.11 OpenCV *library*

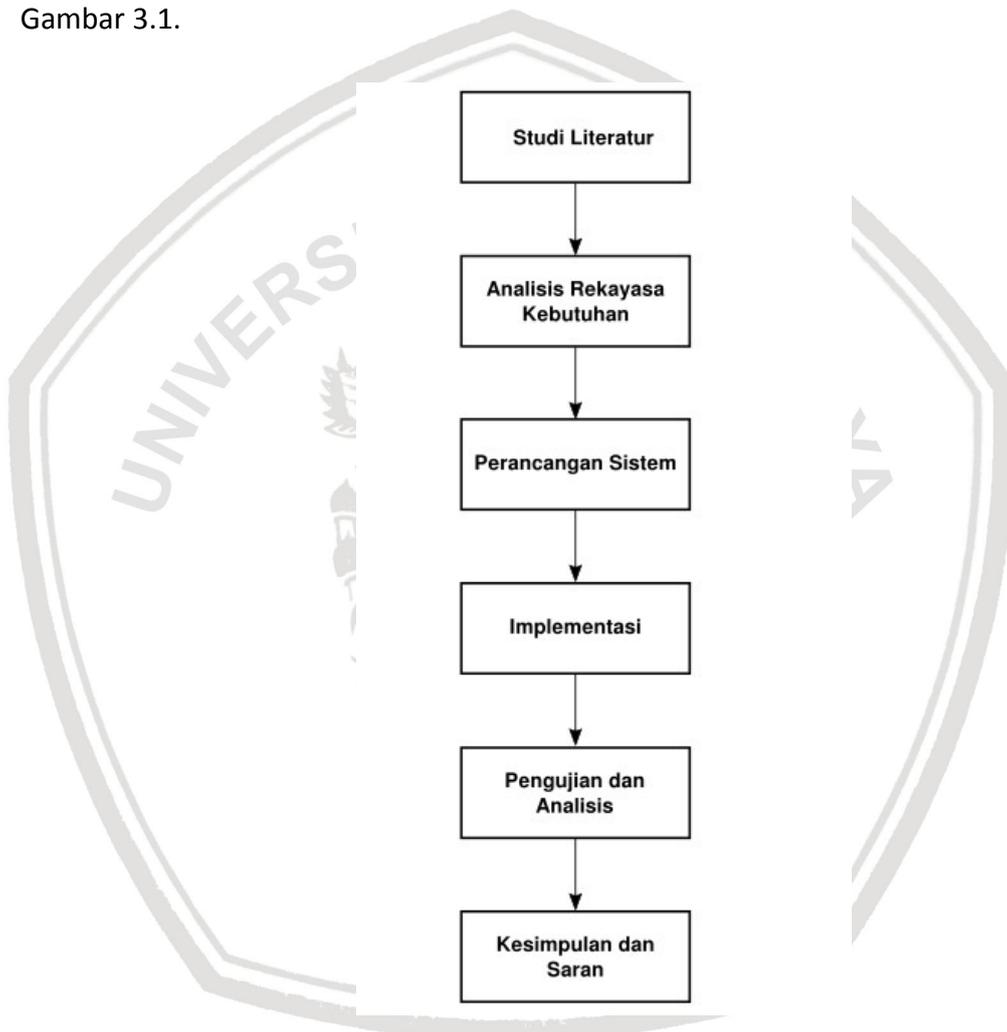
OpenCV merupakan perpustakaan perangkat lunak di dalam *computer vision* dan *machine learning* yang bersifat *open source*. OpenCV ini dibangun untuk menyediakan infrastruktur umum dalam membuat aplikasi *computer vision*. Di dalam perpustakaan ini telah disediakan lebih dari 2500 algoritme yang dioptimalkan. Di antaranya dapat digunakan untuk mendeteksi dan mengenali wajah seseorang, melakukan identifikasi terhadap objek, mengklasifikasikan perilaku manusia di dalam video, melacak objek yang bergerak, dan banyak lagi fungsi yang bisa diterapkan. OpenCV juga menyediakan beberapa pilihan bahasa pemrograman sebagai antarmuka, diantaranya adalah C++, Python, Java, dan MATLAB dan mendukung sistem operasi Windows, Linux, Android dan MacOS. Dengan beragam fasilitas yang disediakan menjadikan perpustakaan OpenCV ini banyak digunakan di dalam bisnis komersial, grup penelitian dan badan pemerintahan. (OpenCV team, 2000)

BAB 3 METODOLOGI PENELITIAN

Bab ini akan membahas mengenai alur metodologi yang digunakan untuk menyelesaikan permasalahan di dalam penelitian ini.

3.1 Alur Metode Penelitian

Tahapan-tahapan di dalam penelitian ini secara umum digambarkan pada Gambar 3.1.



Gambar 3.1 Tahap-tahap metodologi penelitian

3.2 Studi Literatur

Studi literatur merupakan tahapan awal di dalam pengerjaan penelitian ini. Dalam tahap studi literatur ini, akan dikaji penelitian-penelitian sebelumnya yang terkait dengan permasalahan pendeteksian wajah dan pengenalan wajah pada sistem presensi. Dengan demikian, akan memberikan masukan di dalam pengerjaan penelitian ini. Berikut studi literatur yang dilakukan diantaranya adalah

1. *“Algorithm for Efficient Attendance Management: Face Recognition based approach”* oleh Balcoh, N. K. et al. (2012)
2. *“Automatic Attendance Management System Using Face Recognition”* oleh Choudhary, A. et al. (2016)
3. *“Comparison of Face Recognition Algorithms Using Opencv for Attendance System”* oleh SudhaNarang et al. (2018)

Selain itu juga dilakukan pencarian dasar teori yang diperlukan dalam penyelesaian permasalahan di dalam penelitian ini. Diantaranya adalah mempelajari metode Viola Jones, metode *Local Binary Patterns Histograms* (LBPH), konversi warna RGB ke YCbCr, OpenCV *library*, dan Raspberry Pi 3 model B.

3.3 Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan kedua di dalam penyelesaian penelitian ini. Analisis kebutuhan ini perlu ditelaah dengan baik agar dapat mengidentifikasi seluruh kebutuhan yang diperlukan dalam menyelesaikan penelitian ini. Berdasarkan permasalahan yang diangkat, yaitu “*Rekognisi Wajah Pada Sistem Smart Class Untuk Deteksi Kehadiran Mahasiswa Menggunakan Metode Viola Jones Dan Local Binary Patterns Histograms (LBPH) Berbasis Raspberry Pi*”, maka sistem yang dibuat harus mampu menerima masukan berupa gambar, dapat melakukan pengolahan citra, mendeteksi fitur wajah, mengenali setiap wajah dan menampilkan nama-nama mahasiswa yang berhasil dikenali kepada pengguna beserta informasi terkait waktu perkuliahan yang sedang berlangsung. Oleh karenanya, berikut ini proses analisis kebutuhan sistem akan dibagi menjadi dua kategori sebagai berikut.

3.3.1 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras merupakan proses menganalisis kebutuhan sistem dari sisi perangkat keras agar bisa sesuai dengan apa yang menjadi kebutuhan sistem dalam penelitian ini. Dalam kasus di atas, perangkat keras yang dibutuhkan ialah.

1. Raspberry Pi 3 Model B
2. Webcam

3. Monitor
4. Keyboard
5. Kabel HDMI
6. Mouse
7. Pushbutton, dan
8. Kabel jumper

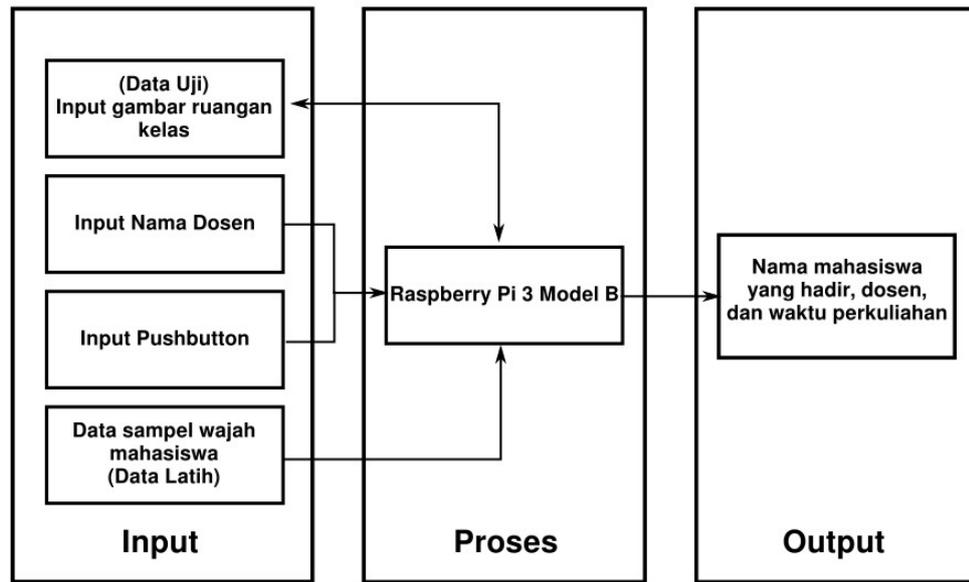
3.3.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak merupakan proses menganalisis perangkat lunak yang digunakan dalam sistem. Berikut daftar analisis perangkat lunak yang dibutuhkan dalam sistem ini.

1. Raspbian OS
2. Sampel wajah mahasiswa
3. Geany
4. Open CV *library*
5. Python versi 3
6. Numpy *library*
7. Python's OS module
8. Time *library*
9. Datetime *library*, dan
10. RPI.GPIO *library*

3.4 Perancangan Sistem

Perancangan sistem merupakan tahapan ketiga di dalam penelitian ini. Tujuan melakukan perancangan sistem adalah untuk mendapatkan gambaran umum terhadap sistem yang akan dibuat, baik dari sisi perangkat kerasnya dan perangkat lunak yang dibutuhkan. Berikut bentuk perancangan sistem yang digunakan dalam penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Gambaran sistem secara umum

Berdasarkan Gambar 3.2, cara kerja sistem akan dijelaskan sebagai berikut:

1. Pengguna mulai memberikan *input* melalui pushbutton untuk memulai proses pengambilan gambar.
2. Setelah itu, Raspberry Pi 3 Model B akan mulai mengambil gambar ruangan kelas.
3. Pengguna kemudian diminta memasukkan nama dosen pengajar menggunakan keyboard.
4. Gambar yang berhasil diambil kemudian diproses untuk pencarian wajah-wajah mahasiswa.
5. Wajah yang berhasil ditemukan akan diekstrak dari gambar dan disimpan ke dalam folder untuk dibandingkan dengan data latih.
6. Setiap kali wajah berhasil dikenali oleh sistem, maka nama mahasiswa yang mewakili wajah tersebut akan ditampilkan kepada pengguna.

3.5 Implementasi Sistem

Tahapan implementasi merupakan tahapan keempat di dalam penelitian ini. Pada tahap ini, hasil perancangan sistem akan diterapkan baik dari sisi perangkat keras dan perangkat lunak. Hal yang perlu diperhatikan adalah penempatan hasil implementasi perangkat keras sistem yang harus berada di depan kelas dan menghadap ke seluruh wajah mahasiswa.

3.6 Pengujian dan Analisis

Tahapan pengujian merupakan tahapan kelima dalam penelitian ini. Pada tahap ini, pengujian terhadap sistem akan dilakukan terkait fungsionalitas ketika melakukan pengambilan gambar, pengisian nama dosen, dan kemampuan akurasi sistem ketika mendeteksi dan mengenali wajah, serta total waktu komputasi yang dibutuhkan ketika menjalankan proses tersebut. Khusus pengujian akurasi pengenalan sistem akan diterapkan pada tiga waktu berbeda, yaitu pada waktu pagi, siang dan malam. Hal ini dimaksudkan untuk menguji kemampuan sistem dalam mengenali wajah dengan tingkat pencahayaan ruangan yang berbeda – dalam penelitian ini, tingkat pencahayaan pada pagi hari sebesar 180-183 lux, siang 118-136 lux, dan malam 114-122 lux. Selain itu, pengujian dilakukan dengan jumlah peserta yang bervariasi, dimulai dari 1 orang mahasiswa hingga 6 orang mahasiswa sekaligus di setiap pengambilan gambarnya. Dari masing-masing gambar yang diambil, akan dilakukan pergantian peserta sebanyak tiga kali (atau tiga kali pengambilan gambar dengan peserta yang berbeda di setiap pengujian jumlah mahasiswanya). Hal ini dilakukan untuk mengetahui rata-rata *error* yang dihasilkan oleh sistem ketika mengenali wajah mahasiswa dengan peserta yang berbeda dan dengan tingkat kepadatan mahasiswa yang semakin meningkat.

3.7 Kesimpulan dan Saran

Kesimpulan merupakan tahapan terakhir di dalam penelitian ini. Pada tahapan ini, akan dilakukan penarikan kesimpulan berdasarkan hasil dan analisis pengujian sistem. Selain itu juga akan diberikan masukan berupa saran yang akan digunakan untuk memperbaiki kesalahan-kesalahan yang ada dan/atau pertimbangan untuk pengembangan sistem lebih lanjut.

BAB 4 REKAYASA KEBUTUHAN

Bab ini akan membahas mengenai gambaran umum sistem, kebutuhan fungsional, dan non-fungsional sistem.

4.1 Gambaran Umum Sistem

Gambaran umum sistem akan memberikan penjelasan terkait tujuan pembuatan sistem, karakteristik pengguna, lingkungan operasi, batasan sistem dan asumsi dan ketergantungan di dalam sistem.

4.1.1 Tujuan

Tujuan sistem ini dibuat adalah untuk bisa mengotomasikan proses presensi kelas dengan cara mengambil gambar ruangan kelas yang dimana di dalamnya terdapat mahasiswa yang sedang mengikuti perkuliahan dosen.

4.1.2 Karakteristik Pengguna

Pengguna dari sistem ini adalah para mahasiswa dan dosen pengampu mata kuliah, baik laki-laki maupun perempuan.

4.1.3 Lingkungan Operasi

Sistem ini akan ditempatkan di dalam ruangan kelas dengan kondisi ruangan sebagai berikut.

1. Ruangan kelas yang digunakan adalah ruangan yang memiliki lebar: 786 cm, panjang: 782 cm dan tinggi: 308 cm. Gambar 4.1 merupakan lokasi ruangan yang digunakan dalam penelitian ini.
2. Setiap baris meja tulis kelas berbentuk seperti anak tangga. Semakin ke belakang, akan terlihat semakin tinggi.
3. Kamera yang terhubung dengan sistem akan ditempatkan di depan ruangan kelas, tepat berada di atas papan tulis putih, menempati tengah-tengah ruangan. Ketinggiannya sekitar: 212 cm dari dasar lantai.



Gambar 4.1 Ruang kelas tempat pelaksanaan penelitian

4.1.4 Batasan Perancangan dan Implementasi

Batasan perancangan dan implementasi dalam penelitian ini adalah.

1. Sistem ini menggunakan OpenCV *library* dalam menerapkan metode Viola Jones dan *Local Binary Patterns Histograms* (LBPH).
2. Hasil penerapan metode dapat dilihat melalui layar monitor.
3. Wajah orang asing yang belum dilatih ke dalam database tetap akan dicari dan dikenali oleh sistem berdasarkan hasil data latih yang paling mendekatinya.
4. Sistem mulai bekerja setelah pengguna menekan pushbutton.
5. Pushbutton dan Raspberry Pi 3 Model B terhubung dengan menggunakan kabel jumper.
6. Mouse dan Keyboard yang digunakan berbentuk *wireless*.

4.1.5 Asumsi dan Ketergantungan

Berikut asumsi dan ketergantungan dari sistem yang dibuat.

1. Sistem ini membutuhkan dua buah stopkontak untuk bisa menghidupkan perangkat monitor dan Raspberry Pi 3 Model B.
2. Pencahayaan ruangan perlu disesuaikan untuk bisa mendeteksi dan mengenali wajah oleh sistem.

4.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan proses menyelidiki segala kebutuhan yang diperlukan dalam membangun sistem yang diinginkan. Dalam penelitian ini, proses analisis kebutuhan sistem terbagi menjadi dua, yaitu analisis kebutuhan fungsional dan kebutuhan non-fungsional.

4.2.1 Kebutuhan Fungsional

Analisis kebutuhan fungsional di dalam sistem ini adalah.

1. Pengambilan gambar
Sistem dapat menangkap gambar yang ada di dalam ruangan kelas dengan menggunakan webcam. Webcam dengan panjang kabel 1,5m terhubung ke perangkat Raspberry Pi 3 Model B melalui *port* USB 2.0. Hasil tangkapan gambar kemudian disimpan ke dalam sistem untuk proses pendeteksian wajah.
2. Sistem dapat menerima *input string* untuk memasukkan nama dosen
Untuk memasukkan nama dosen ke dalam sistem, pengguna mesti mengetikkan nama dosen secara manual menggunakan keyboard yang terhubung secara *wireless* ke dalam perangkat Raspberry Pi 3 Model B.
3. Sistem dapat mendeteksi wajah mahasiswa
Untuk dapat mendeteksi wajah mahasiswa, sistem memerlukan OpenCV *library*. Dalam *library* tersebut telah disediakan algoritme Viola Jones yang siap dipakai. Sebagai tambahan, sistem ini menyertakan pengklasifikasi warna kulit manusia sebagai upaya menghindari *false detection* ketika menerapkan pendeteksian wajah, yaitu dengan mengkonversi warna piksel RGB ke dalam ruang warna YCbCr. Setiap piksel dari gambar YCbCr kemudian di-*threshold* dengan batas nilai $77 \leq Cb \leq 127$ dan $133 \leq Cr \leq 173$ sebagai warna piksel kulit manusia (Patravali, Waykule and Katre, 2014).
4. Sistem dapat mengenali wajah
Untuk bisa mengenali wajah mahasiswa, sistem memerlukan OpenCV *library*. Dalam *library* tersebut telah disediakan algoritme *Local Binary Patterns Histograms* (LBPH) yang siap dipakai. Sebagai syarat dalam menjalankan proses ini, jumlah wajah yang dideteksi tidak boleh bernilai nol. Itu disebabkan wajah yang terdeteksi akan digunakan sebagai pembandingan dengan wajah yang ada di dalam database.
5. Sistem dapat menampilkan waktu komputasi ketika melakukan proses pendeteksian dan pengenalan wajah
Untuk bisa menampilkan waktu komputasi, sistem memerlukan *time library*. *Library* ini menyediakan fungsi yang dapat mengetahui lamanya suatu proses berlangsung yang diukur dalam satuan detik. Caranya adalah

dengan menempatkan fungsi pembacaan waktu di awal dan di akhir proses , kemudian membandingkan waktu keduanya.

4.2.2 Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional dalam penelitian ini terbagi menjadi dua, yaitu kebutuhan perangkat keras dan perangkat lunak.

4.2.2.1 Kebutuhan Perangkat Keras

Perangkat keras yang perlu disediakan adalah

1. Raspberry Pi 3 Model B
Merupakan perangkat yang akan digunakan sebagai perangkat pengolah gambar digital. Karena secara spesifikasi, perangkat ini sudah memenuhi kebutuhan yang diperlukan di dalam penelitian ini, yaitu dapat diintegrasikan dengan webcam, pushbutton, mouse, keyboard dan bisa langsung terhubung dengan layar monitor melalui *port* HDMI.
2. Webcam
Merupakan komponen yang digunakan sebagai kamera dalam mengambil gambar di dalam ruangan kelas.
3. Monitor
Merupakan perangkat yang digunakan sebagai perangkat keluaran yang akan menampilkan daftar nama mahasiswa yang hadir di dalam kelas.
4. Keyboard, Kabel HDMI, dan Mouse
Merupakan minimal komponen dalam mempersiapkan Raspberry Pi 3 Model B ini untuk nantinya bisa digunakan dalam menjalankan fungsional sistem.
5. Pushbutton dan kabel jumper
Merupakan komponen yang digunakan untuk men-*trigger* proses dimulainya pengambilan gambar dengan menghubungkannya ke pin GPIO pada Raspberry Pi 3 Model B.

4.2.2.2 Kebutuhan Perangkat Lunak

Beberapa perangkat lunak yang perlu dipersiapkan, di antaranya adalah

1. Raspbian OS
Merupakan sistem operasi yang digunakan pada perangkat Raspberry Pi 3 Model B.

2. Sampel wajah
Merupakan file gambar wajah mahasiswa yang digunakan untuk proses rekognisi. Jumlah sampel yang dibutuhkan di dalam penelitian ini adalah paling tidak sebanyak 50 sampel wajah untuk satu orangnya.
3. Geany
Merupakan aplikasi IDE sederhana dan ringan yang digunakan dalam membuat program sistem.
4. Open CV *library*
Merupakan *library* yang digunakan di dalam pengolahan gambar pada penelitian ini karena telah menyediakan lebih dari 2500 algoritme yang siap pakai, dimana diantaranya dapat digunakan untuk mendeteksi wajah dan mengenali wajah.
5. Python versi 3
Merupakan bahasa pemrograman yang dipilih dalam membuat program sistem. Bahasa ini dipilih karena didukung oleh Opencv *library* dan beberapa *library* lainnya yang ada di dalam kebutuhan perangkat lunak sistem ini.
6. Numpy *library*
Merupakan *library* yang digunakan untuk mengkonversi list python ke dalam format numpy array. Karena sebagian besar pengolahan data di dalam OpenCV *library* menggunakan format numpy array. Salah satu diantaranya adalah proses *men-threshold* warna piksel kulit manusia pada ruang warna YCbCr.
7. Python's OS *module*
Merupakan *library* yang digunakan untuk berinteraksi dengan data sampel gambar wajah mahasiswa yang disimpan di dalam sistem untuk diproses sebagai data latih.
8. *Time library*
Digunakan sebagai penghitung waktu komputasi ketika menjalankan proses pendeteksian dan pengenalan wajah.
9. *Datetime library*
Digunakan untuk mendapatkan informasi waktu perkuliahan untuk keperluan sistem presensi.
10. RPI.GPIO *library*
Digunakan untuk berinteraksi dengan pin-pin Raspberry Pi 3 Model B yang dalam penelitian ini digunakan untuk *men-trigger* pengambilan gambar melalui penekanan tombol pushbutton.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini akan membahas mengenai perancangan dan implementasi sistem. Perancangan dan implementasi sistem di dalam penelitian ini dibagi menjadi dua, yaitu dari sisi perangkat keras dan perangkat lunak.

5.1 Perangkat Keras

5.1.1 Perancangan Perangkat Keras

Pada tahap perancangan perangkat keras, komponen seperti webcam, mouse dan keyboard dihubungkan langsung ke raspberry pi melalui *port* USB 2.0. Sedangkan untuk perangkat monitor dihubungkan melalui *port* HDMI, *power supply* di hubungkan ke *port* micro USB, dan pushbutton dihubungkan ke pin GPIO raspberry pi. Gambar 5.1 mengilustrasikan bentuk perancangan sistem di dalam penelitian ini.



Gambar 5.1 Perancangan perangkat keras

Pada Gambar 5.1, terlihat bahwa terdapat pin GPIO yang digunakan untuk mengaktifkan pushbutton. Untuk bisa menggunakan pin tersebut, maka perlu ditetapkan mode GPIO. Dalam penelitian ini, mode GPIO yang digunakan adalah mode GPIO.BOARD.

Berikut ini dijelaskan pin GPIO yang akan dihubungkan ke pin pushbutton yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Keterangan pin raspberry pi yang terhubung ke pushbutton

Pin Raspberry Pi 3 Model B	Pin Pushbutton
9	Pin kiri
11	Pin kanan

5.1.2 Implementasi Perangkat Keras

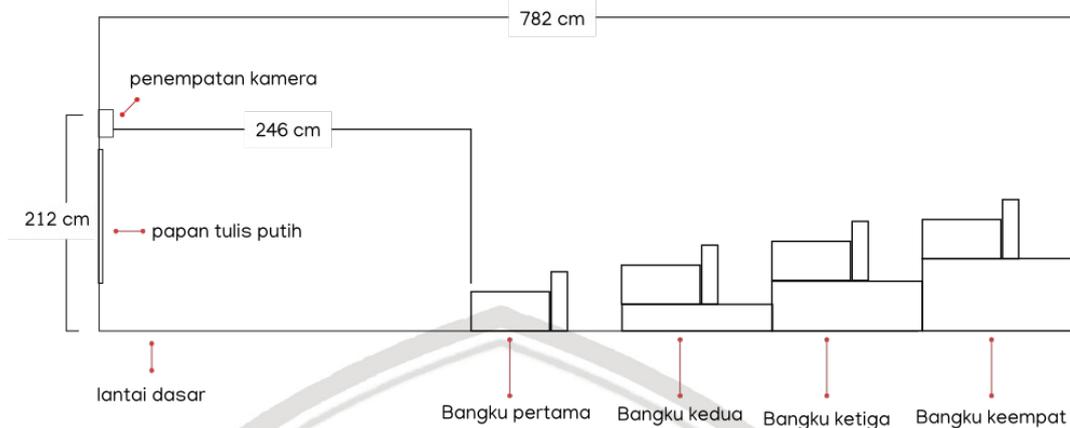
Berikut bentuk implementasi dari perangkat keras yang ditempatkan di dalam ruangan kelas dapat dilihat pada Gambar 5.2.



Gambar 5.2 Foto alat

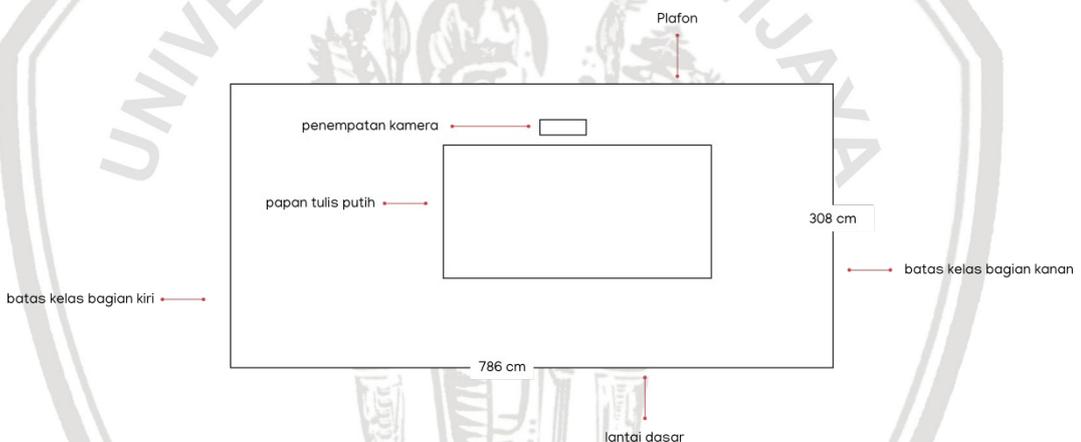
Pada Gambar 5.2, dapat dilihat bahwa raspberry pi dipasang di depan kelas dengan wadah yang ditempelkan di dinding agar dapat meng-*capture* seluruh wajah mahasiswa yang ada di dalam kelas. Selain itu, untuk dapat menunjukkan keseluruhan isi komponen sistem, maka monitor dan raspberry pi akhirnya ditempatkan dekat dengan perangkat kamera.

Berikut bentuk skenario penerapan sistem jika ditinjau dari samping dapat dilihat pada Gambar 5.3.



Gambar 5.3 Skenario penerapan sistem di dalam kelas (tampak samping)

Bentuk penerapan sistem jika ditinjau dari depan akan tampak seperti Gambar 5.4



Gambar 5.4 Skenario penerapan sistem di dalam kelas (tampak depan)

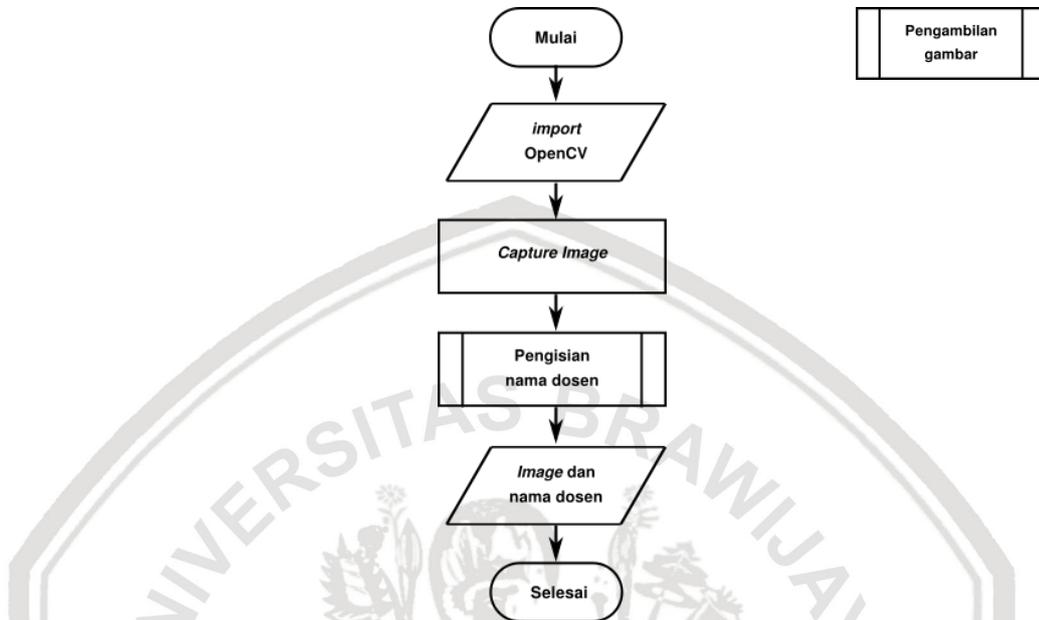
5.2 Perangkat Lunak

5.2.1 Perancangan Perangkat Lunak

Secara keseluruhan, proses-proses yang akan dilalui di dalam sistem presensi ini adalah pengambilan gambar, pengisian nama dosen, pendeteksian wajah, cek file pengujian, *resize* area deteksi, deteksi warna kulit, pengenalan wajah dan pembacaan data sampel yang diolah menggunakan bahasa pemrograman python.

5.2.1.1 Pengambilan Gambar

Pengambilan gambar merupakan proses meng-*capture* gambar ruangan di dalam kelas. Proses pengambilan gambar dalam penelitian ini ditunjukkan pada Gambar 5.5.



Gambar 5.5 *Flowchart* pengambilan gambar

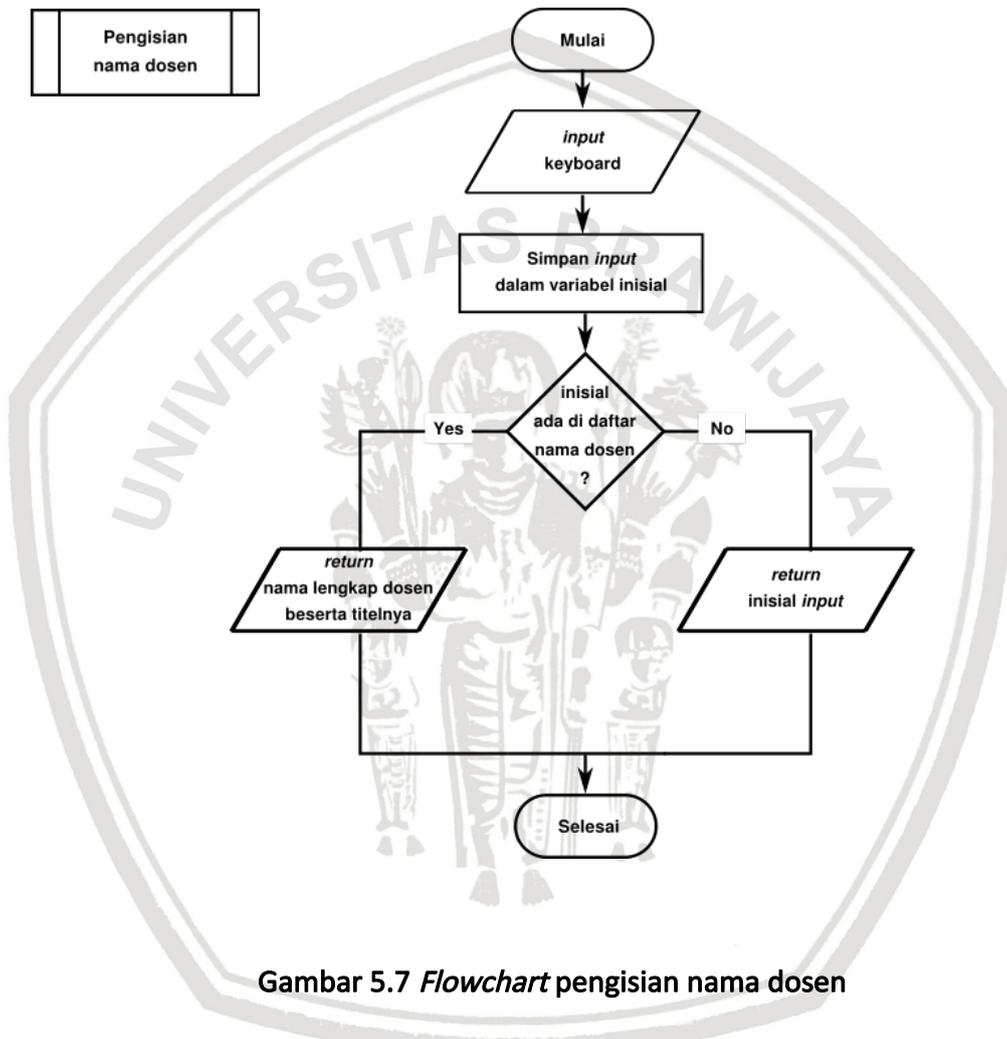
Pada Gambar 5.5, proses diawali dengan memasukkan OpenCV *library* ke dalam program untuk bisa melakukan pengambilan gambar. Kemudian proses pengambilan gambar dapat diterapkan. Gambar yang ditangkap akan disimpan di dalam folder 'test-data', seperti yang ditunjukkan pada Gambar 5.6, dimana file gambar tersebut akan diberi nama '1.png'. Setelah gambar berhasil ditangkap, maka akan dilakukan pemanggilan sub proses pengisian nama dosen yang akan mengembalikan nama dosen pengajar. Akhir proses dari *flowchart* ini akan mengembalikan gambar yang di *capture* dan nama dosen pengajar.



Gambar 5.6 Lokasi penyimpanan gambar hasil tangkapan

5.2.1.2 Pengisian Nama Dosen

Pengisian nama dosen merupakan proses memasukkan nama dosen ke dalam sistem melalui perangkat keyboard untuk informasi presensi kelas. Proses pengisian nama dosen hanya memerlukan inisial nama dosen sebagai *input*, selanjutnya sistem secara otomatis akan mengembalikan nama lengkap dosen yang paling mendekatinya beserta titel dosen tersebut. Berikut proses pengisian nama dosen ditunjukkan pada Gambar 5.7.

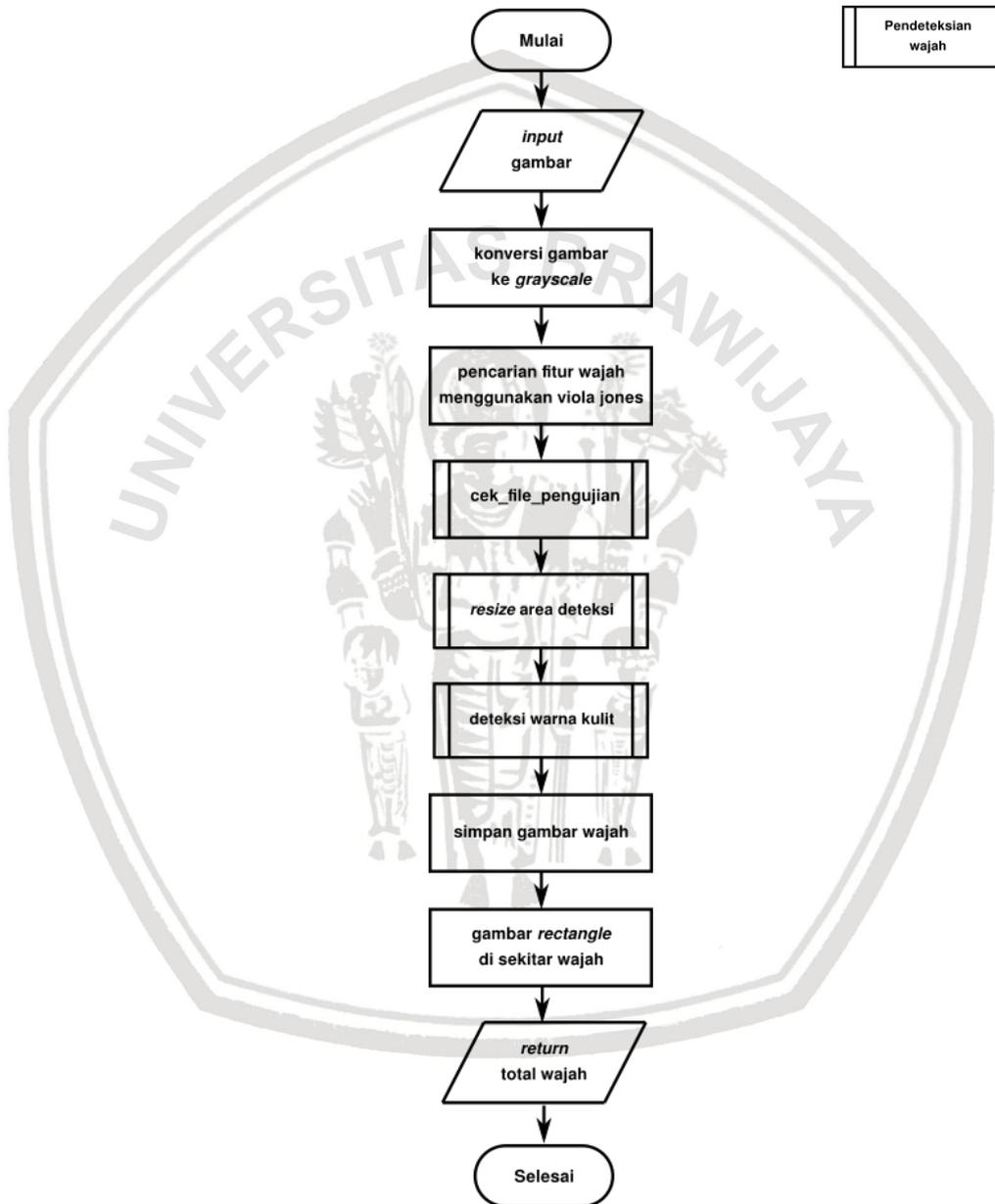


Pada Gambar 5.7, proses di awali dengan memasukkan nama dosen melalui perangkat keyboard. Kemudian nama dosen di simpan di dalam variabel *inisial* dan akan diuji apakah variabel *inisial* telah terdapat di dalam daftar nama dosen. Jika ada nama dosen yang menyerupai variabel *inisial*, maka nama dosen tersebut akan dikembalikan kepada proses pemanggil beserta titel dosen. Namun, apabila tidak ada nama dosen yang menyerupai variabel *inisial*, maka proses ini akan mengembalikan variabel *inisial* kepada proses pemanggilnya.



5.2.1.3 Pendeteksian Wajah

Pendeteksian wajah merupakan proses menemukan fitur-fitur yang berpotensi sebagai gambar wajah dalam sebuah gambar. Proses pendeteksian wajah dalam penelitian ini menggunakan metode Viola Jones dari OpenCV *library* yang memberikan keluaran berupa gambar wajah mahasiswa yang diekstraksi dari gambar dan memberikan *frame* berwarna biru di setiap lokasi wajah yang ditemukan. Berikut proses pendeteksian dalam penelitian ini dapat dilihat pada Gambar 5.8.



Gambar 5.8 *Flowchart* pendeteksian wajah

Pada Gambar 5.8, proses diawali dengan memberikan *input* gambar ruangan di dalam kelas. Gambar kemudian dikonversi ke dalam ruang warna *grayscale* agar proses pencarian wajah menggunakan OpenCV *library* dapat dijalankan. Setelah berhasil menjalankan proses pencarian wajah, akan dilakukan pengecekan file pengujian, ini dimaksudkan untuk memastikan bahwa semua file di dalam folder pengujian dalam keadaan kosong, dimana di dalam folder ini hanya akan berisi gambar wajah mahasiswa yang berhasil dideteksi untuk diproses lebih lanjut ke dalam proses pengenalan wajah.

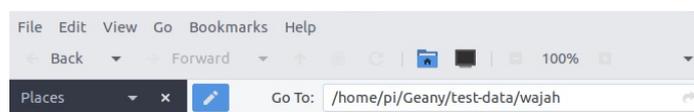
Setelah pengecekan file pengujian berhasil dijalankan, maka sub proses *resize* area deteksi akan dipanggil untuk melakukan pelebaran area deteksi wajah dari OpenCV *library* agar dapat mengambil gambar wajah mahasiswa secara keseluruhan. Kemudian wajah mahasiswa akan diuji apakah terdapat piksel kulit di dalamnya atau tidak sebelum disimpan di dalam folder pengujian. Jika ada terdapat piksel kulit, maka gambar wajah mahasiswa akan disimpan ke dalam folder pengujian. Namun, jika tidak terdapat piksel kulit, maka gambar wajah akan di *skip* dan tidak diproses lebih lanjut.

Setelah gambar wajah berhasil disimpan, maka area disekitar wajah akan diberi tanda berupa *frame* segi empat berwarna biru. Begitu seterusnya hingga apabila sudah tidak ada lagi wajah yang terdeteksi, maka proses akan diakhiri dengan mengembalikan total wajah yang berhasil dideteksi oleh sistem.

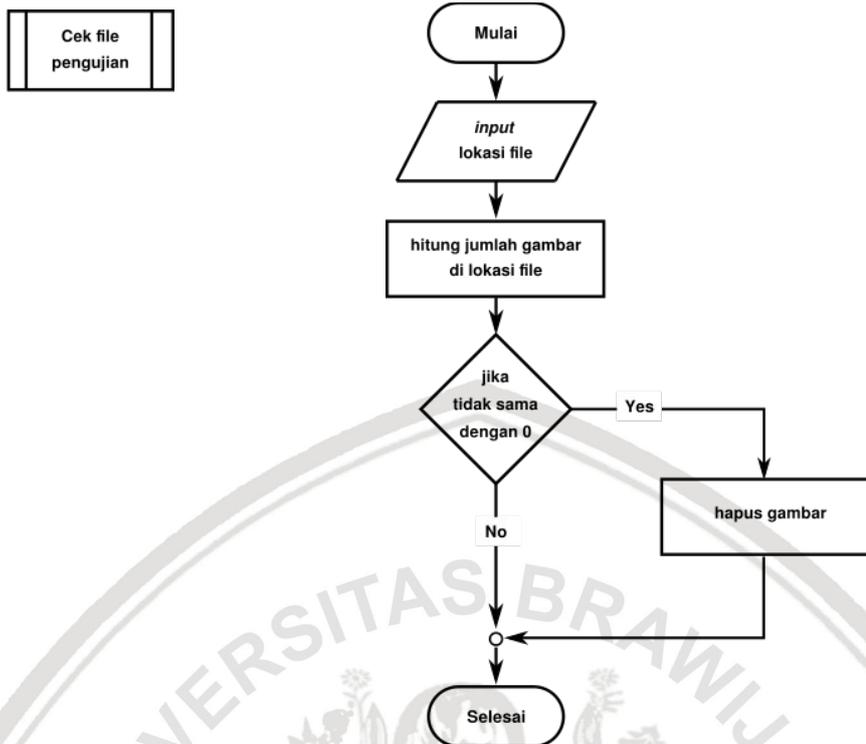
5.2.1.4 Cek File Pengujian

Cek file pengujian merupakan proses untuk mengecek file uji gambar wajah yang telah tersimpan karena menerapkan proses pendeteksian wajah sebelumnya. Proses ini diperlukan untuk memastikan bahwa ketika hendak menjalankan pendeteksian wajah yang baru, seluruh isi file di dalam folder pengujian dalam keadaan kosong, sehingga tidak menyatu dengan file uji gambar wajah sebelumnya. Karena setiap gambar wajah yang tersimpan di dalam folder pengujian akan dikenali satu per satu oleh sistem pada tahap pengenalan wajah.

Alur proses dalam cek file pengujian diawali dengan memberikan *input* berupa lokasi penyimpanan file untuk pengujian – dalam penelitian ini, lokasi file pengujian berada di `/home/pi/Geany/test-data/wajah`, seperti yang ditunjukkan pada Gambar 5.9. Setelah itu, proses akan berlanjut ke perhitungan jumlah file di dalam folder pengujian. Jika folder pengujian dalam keadaan kosong, maka proses akan berakhir. Namun, jika terdapat file di dalamnya, maka proses penghapusan file akan dijalankan. Alur proses cek file pengujian dapat dilihat pada Gambar 5.10.



Gambar 5.9 Lokasi penyimpanan gambar wajah yang terdeteksi

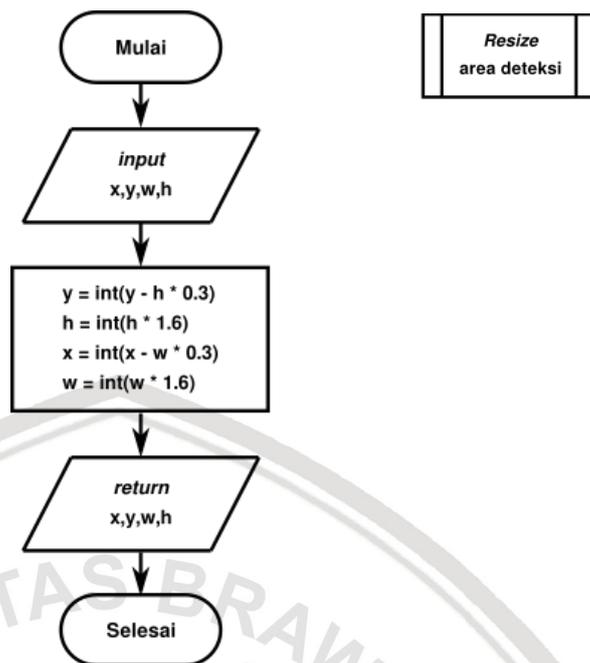


Gambar 5.10 Flowchart cek file pengujian

5.2.1.5 *Resize* Area Deteksi

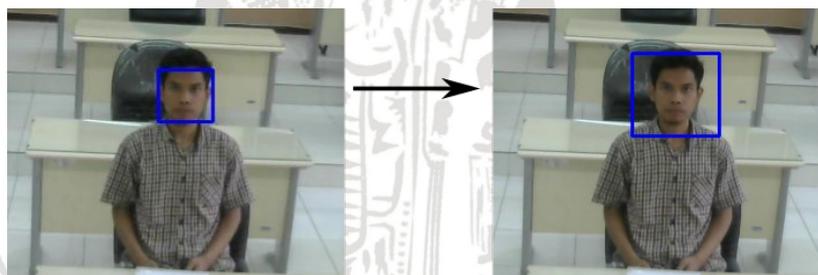
Proses *resize* area deteksi merupakan proses merubah ukuran area pendeteksian wajah dari OpenCV *library*. Proses ini diperlukan untuk memudahkan menangkap seluruh area wajah. Karena secara *default*, ukuran yang diterapkan di dalam OpenCV *library* tidak mencakup kesemua area yang diinginkan (atas rambut, kedua telinga hingga ke bawah dagu), sehingga ukuran pendeteksian tersebut perlu diperlebar. Alasan diterapkannya proses *resize* adalah untuk bisa mengambil keseluruhan area wajah mahasiswa yang terdeteksi dan menyimpannya ke dalam folder pengujian untuk di proses lebih lanjut di dalam pengenalan wajah.

Alur proses *resize* area deteksi diawali dengan memberikan *input* variabel x, y, w, h yang keempatnya merupakan titik koordinat gambar wajah di dalam gambar uji ruangan kelas – x adalah koordinat sumbu x , y adalah koordinat sumbu y , w adalah lebar dari titik x , dan h adalah tinggi dari titik y . Kemudian, keempat variabel dirubah dengan menggunakan perhitungan matematis sesuai dengan selera pengguna – dalam penelitian ini, angka yang digunakan adalah 0,3 dan 1,6, seperti yang ditunjukkan pada Gambar 5.11. Setelah itu, keempat variabel akan dikembalikan kepada program pemanggil. Secara keseluruhan, proses *resize* area dapat dilihat pada Gambar 5.11.



Gambar 5.11 *Flowchart resize area deteksi*

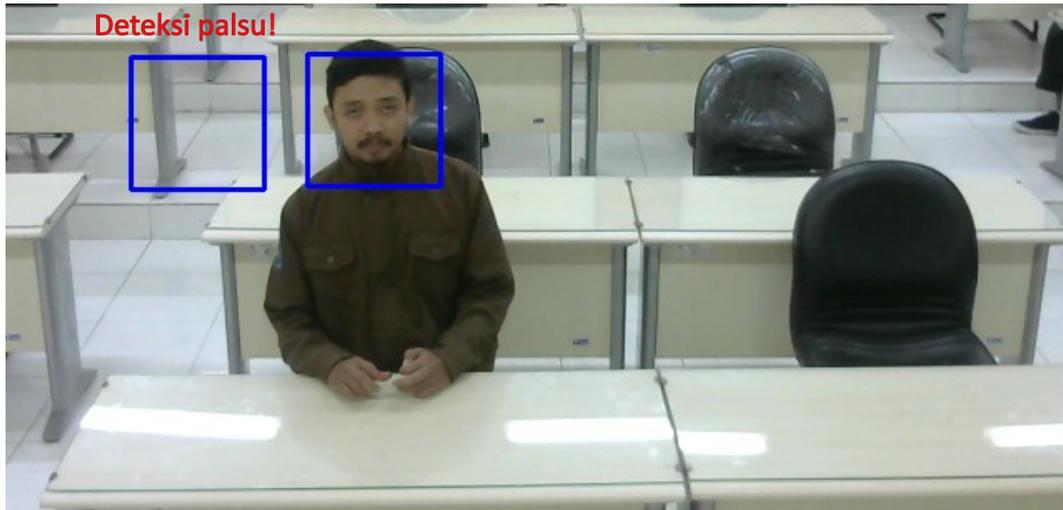
Hasil penerapan bilangan pada Gambar 5.11 adalah sebagaimana terlihat pada Gambar 5.12.



Gambar 5.12 Hasil *resize area deteksi*

5.2.1.6 Deteksi Warna Kulit

Proses deteksi warna kulit diterapkan untuk memastikan bahwa wajah yang terdeteksi setelah proses pendeteksian wajah merupakan gambar wajah. Hal itu disebabkan karena masih ditemukannya deteksi palsu ketika menampilkan lokasi fitur wajah di dalam gambar, yang mana lokasi yang dianggap sebagai wajah ternyata bukanlah gambar wajah. Contoh hasil deteksi yang tertangkap sebagai gambar wajah dapat dilihat pada Gambar 5.13.



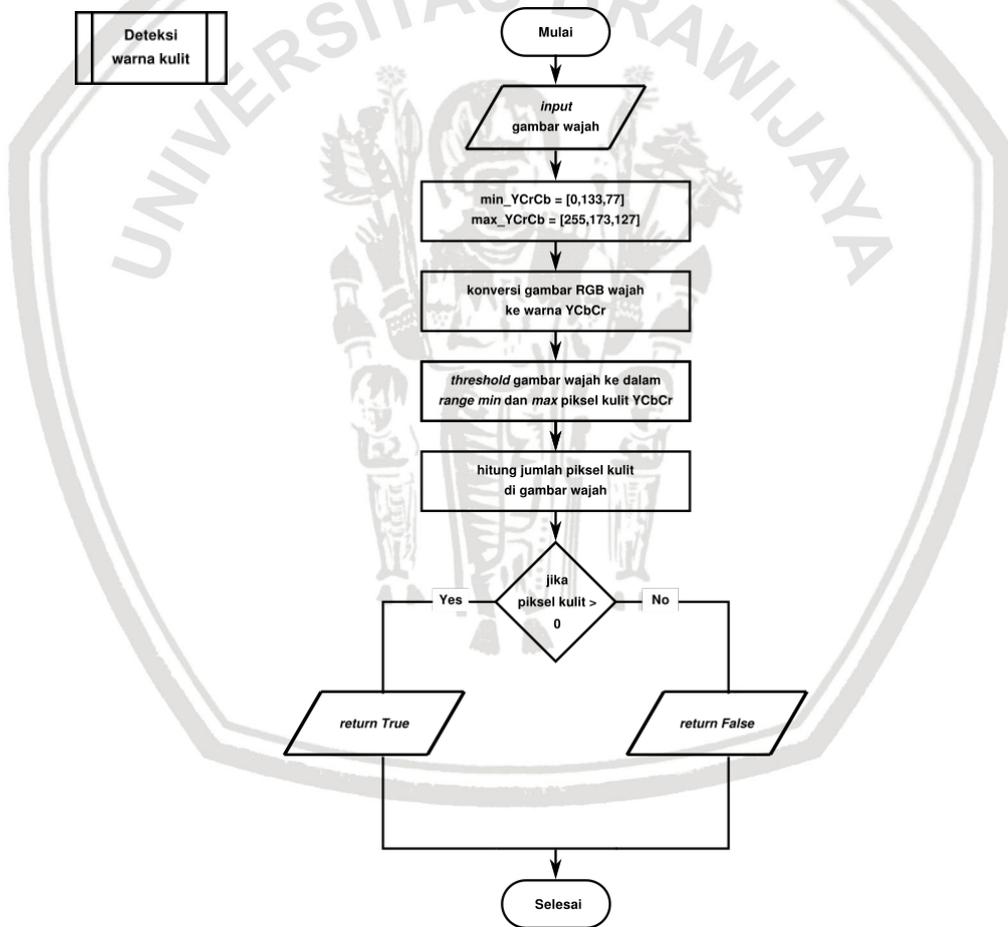
Gambar 5.13 Ditemukan deteksi palsu dalam *face detection OpenCV library*

Untuk meminimalisir deteksi palsu tersebut, diterapkanlah proses pendeteksian piksel warna kulit pada setiap area yang dianggap sebagai gambar wajah dengan cara mengkonversikannya ke ruang warna YCbCr dengan nilai *threshold* warna kulit berdasarkan penelitian Patravali, Waykule dan Katre (2014), yaitu $77 \leq Cb \leq 127$ dan $133 \leq Cr \leq 173$. Sebagai contoh dengan menggunakan kedua hasil deteksi di atas, di ambil beberapa sampel piksel pada masing-masing area yang terdeteksi untuk diproses menggunakan pendeteksian warna kulit yang dapat dilihat pada Tabel 5.2.

Tabel 5.2 Deteksi warna kulit

Gambar	Konversi RGB ke YCbCr	Status
	$R, G, B = [169, 175, 173]$ $Y = (0.299 \times R) + (0.587 \times G) + (0.114 \times B)$ $= 172$ $Cb = (B - Y) \times 0.564 + 128$ $= 128$ $Cr = (R - Y) \times 0.713 + 128$ $= 125$	Bukan piksel kulit
	$R, G, B = [127, 107, 80]$ $Y = (0.299 \times R) + (0.587 \times G) + (0.114 \times B)$ $= 109$ $Cb = (B - Y) \times 0.564 + 128$ $= 111$ $Cr = (R - Y) \times 0.713 + 128$ $= 140$	Piksel kulit

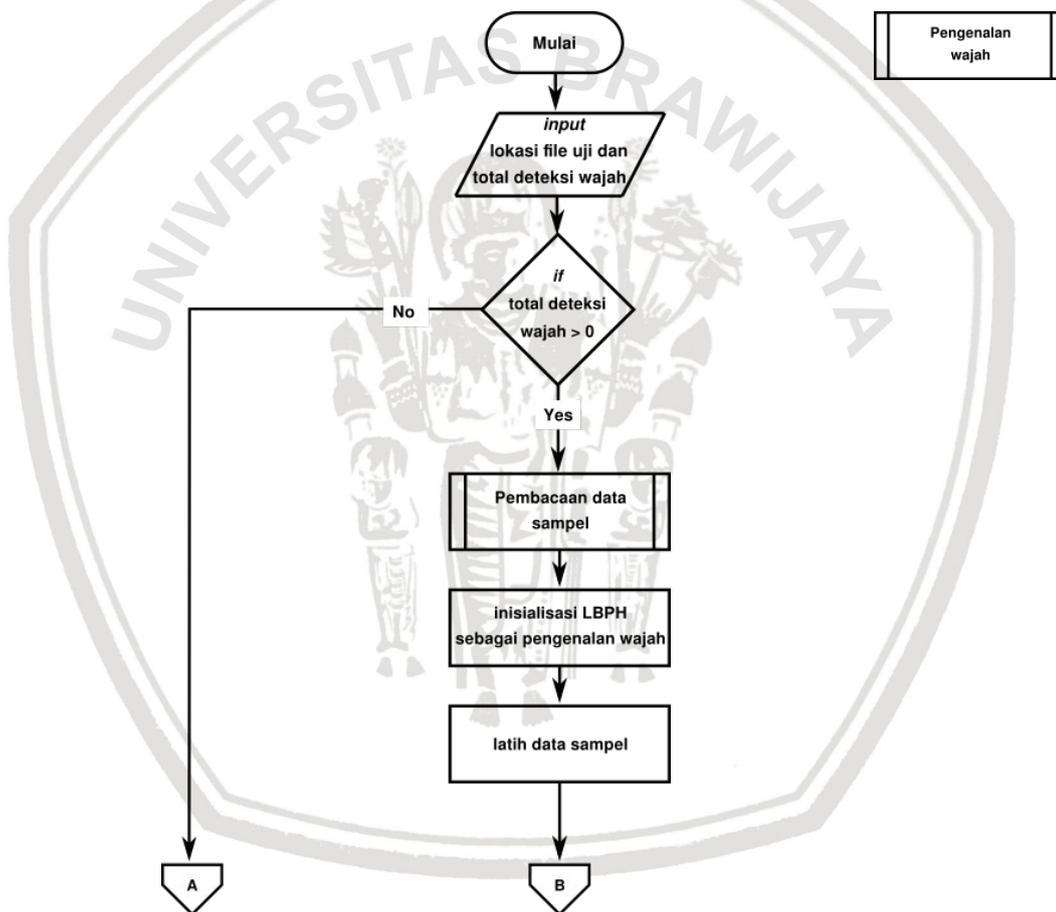
Proses pendeteksian piksel kulit diawali dengan memberikan *input* berupa gambar wajah yang dideteksi oleh OpenCV *library*. Kemudian menginisialisasi batas bawah dan batas atas dari *range* nilai piksel kulit dalam ruang warna YCbCr. Setelah itu, gambar wajah akan dikonversi ke dalam ruang warna YCbCr. Gambar wajah dengan format YCbCr tersebut akan di-*threshold* dengan batas bawah dan atas dari *range* piksel kulit YCbCr sebelumnya. Kemudian setelah di-*threshold*, program akan menghitung jumlah piksel kulit dalam area gambar wajah tersebut – piksel gambar wajah setelah mengalami *threshold* akan terdiri dari 2 model, 0 dan 255 (dalam OpenCV *library*, angka 255 merupakan angka 1 dalam gambar *binary*). Jika total piksel kulit (atau 255) lebih dari 0, maka gambar wajah tersebut benar merupakan gambar wajah mahasiswa dan program akan mengembalikan nilai *True*. Akan tetapi, jika tidak ditemukan piksel kulit di dalamnya (atau jumlah angka 255 tidak lebih dari 0), maka program akan mengembalikan nilai *False* yang menandakan bahwa gambar tersebut bukanlah gambar wajah mahasiswa. Secara keseluruhan, alur proses deteksi warna kulit dapat dilihat pada Gambar 5.14.



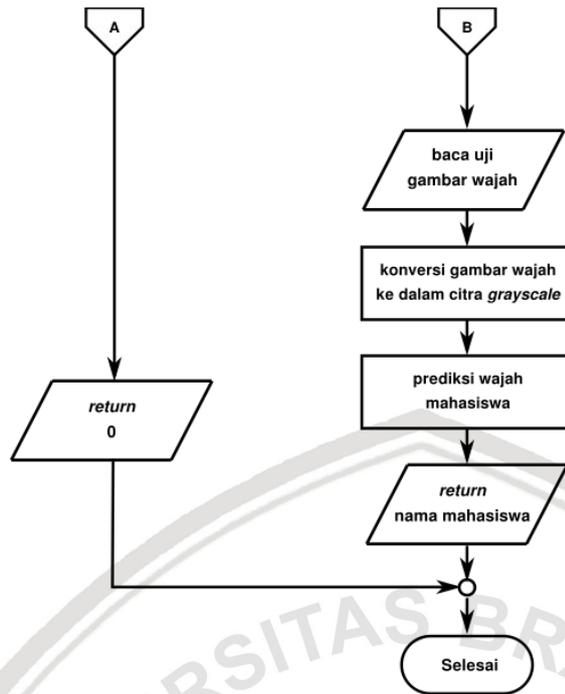
Gambar 5.14 Flowchart deteksi warna kulit

5.2.1.7 Pengenalan Wajah

Pengenalan wajah merupakan proses membandingkan wajah-wajah yang telah tersimpan sebelumnya di dalam proses pendeteksian wajah dengan data latih yang ada di dalam *database* dengan menggunakan metode *Local Binary Patterns Histograms* (LBPH) dari *OpenCV library*. Proses ini hanya akan dimulai apabila total wajah yang terdeteksi pada proses pendeteksian wajah sebelumnya tidak bernilai nol. Proses pengenalan wajah akan diawali dengan pembacaan data sampel ke dalam sistem kemudian memprediksi seluruh wajah yang diujikan. Kalkulasi hasil yang menunjukkan paling mendekati data latih, akan diberikan label berisi nama mahasiswa. Hasil akhir dari proses pengenalan wajah adalah list yang berisi nama-nama mahasiswa. Berikut ini proses pengenalan wajah di dalam penelitian ini dapat dilihat pada Gambar 5.15 dan 5.16.



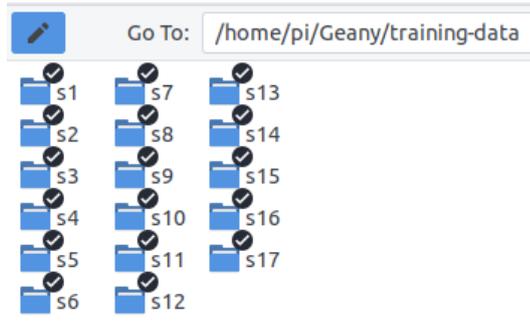
Gambar 5.15 Flowchart pengenalan wajah



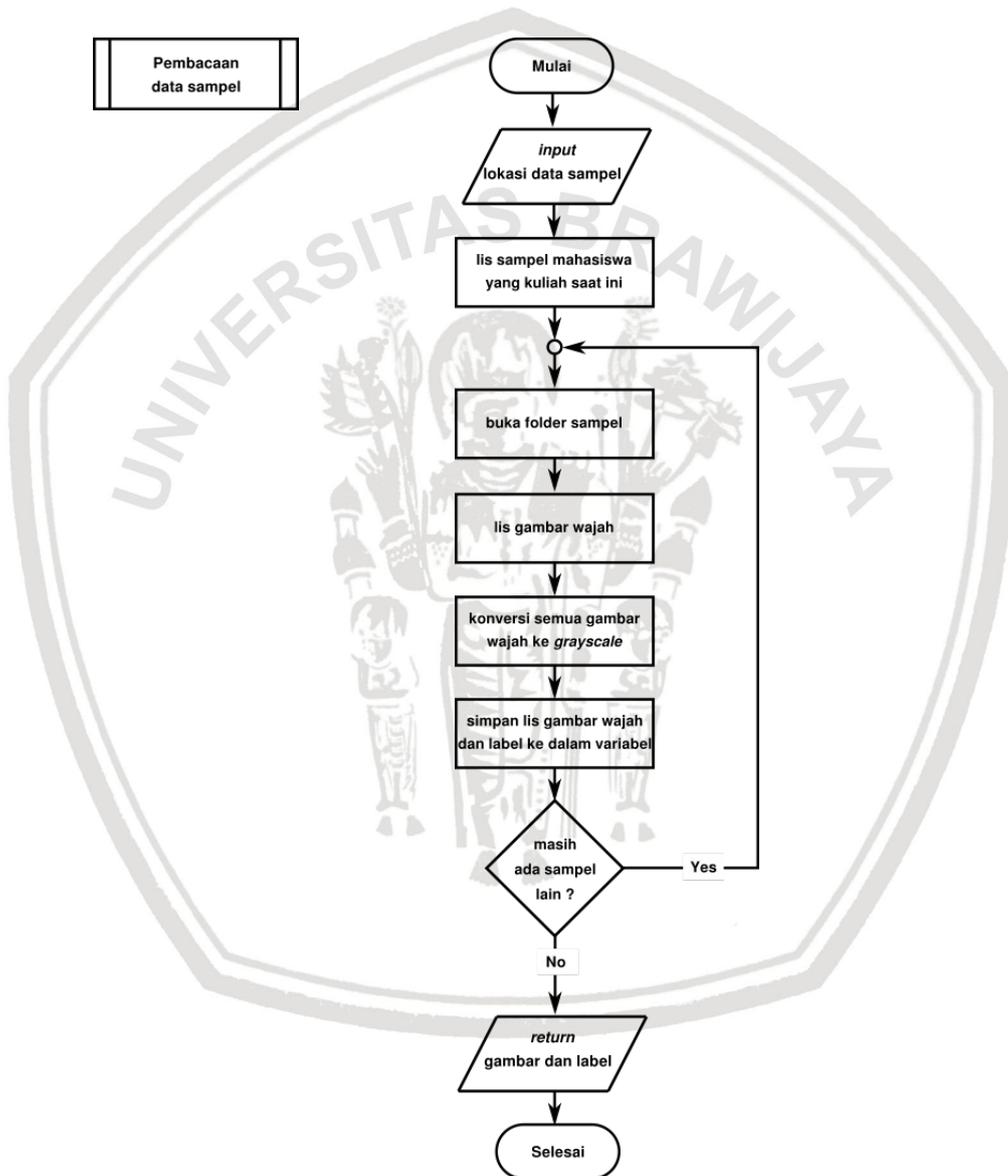
Gambar 5.16 Pengenalan wajah (lanjutan)

5.2.1.8 Pembacaan Data Sampel

Pembacaan data sampel merupakan proses membaca data sampel setiap mahasiswa untuk kemudian disimpan di dalam sistem presensi agar sistem nantinya dapat dilatih untuk memahami karakteristik setiap wajah mahasiswa. Proses pembacaan data sampel diawali dengan memberikan *input* berupa lokasi data sampel mahasiswa – dalam penelitian ini, lokasi data sampel berada pada `‘/home/pi/Geany/training-data/’`, seperti yang ditunjukkan pada Gambar 5.17, dimana dalam setiap folder sampel diberi label ‘s’ kemudian diikuti dengan nomor urut. Nomor urut ini akan digunakan untuk merujuk pada nama mahasiswa yang terdaftar mengikuti kelas perkuliahan tersebut. Kemudian program akan membuka folder sampel satu per satu dan membaca file sampel di dalamnya. Pembacaan gambar sampel akan dikonversi ke dalam ruang warna *grayscale* sebelum disimpan dalam lis sampel wajah mahasiswa, karena proses pengenalan wajah di *OpenCV library* mensyaratkan format gambar *grayscale* untuk bisa dijalankan. Kemudian proses pembacaan akan terus berlangsung hingga seluruh folder sampel mahasiswa telah terbaca oleh sistem. Akhir proses pembacaan data sampel akan memberikan keluaran berupa lis gambar wajah mahasiswa dan label sampel mahasiswa yang bersangkutan. Berikut proses pembacaan data sampel di dalam penelitian ini ditunjukkan pada Gambar 5.18.



Gambar 5.17 Lokasi penyimpanan data sampel mahasiswa



Gambar 5.18 Flowchart pembacaan data sampel

5.2.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak sistem terdiri dari implementasi pengambilan gambar, pengisian nama dosen, pendeteksian wajah, cek file pengujian, *resize* area deteksi, deteksi warna kulit, pengenalan wajah dan pembacaan data sampel.

5.2.2.1 Pengambilan Gambar

Bentuk implementasi pengambilan gambar dapat dilihat pada Tabel 5.3.

Tabel 5.3 Kode program pengambilan gambar

Baris	Kode Program
1	<code>def pengambilan_gambar(cap):</code>
2	<code> while True:</code>
3	<code> _, frame = cap.read()</code>
4	<code> # mengambil gambar</code>
5	<code> cv.imwrite('test-data/1.png', frame)</code>
6	<code> break</code>
7	
8	<code> # pembacaan gambar</code>
9	<code> img = cv.imread('test-data/1.png')</code>
10	<code> # pengisian nama dosen</code>
11	<code> dosen = pengisian_nama_dosen()</code>
12	
13	<code> return img, dosen</code>

Pada Tabel 5.3, kode program di baris ke-1 merupakan nama fungsi yang dipanggil ketika melakukan pengambilan gambar, dimana di dalam fungsi ini membutuhkan satu buah parameter berupa identifikasi webcam yang digunakan oleh sistem – dalam penelitian ini menggunakan webcam Logitech C310. Baris ke-2 merupakan pengulangan untuk membaca *streaming* gambar, dimana data pembacaan itu akan tersimpan di dalam variabel *frame*. Baris ke-5 merupakan program untuk menyimpan gambar ke dalam folder 'test-data' dan baris ke-6 merupakan program untuk keluar dari perulangan yang artinya data berhasil tersimpan ke dalam sistem.

Baris ke-9 pada Tabel 5.3 merupakan program untuk membaca gambar yang tadi berhasil disimpan. Kemudian sebagai tambahan, baris ke-11 digunakan untuk memanggil fungsi pengisian nama dosen yang akan mengembalikan nama dosen dan pada baris ke-13, fungsi pengambilan gambar ini mengembalikan dua buah variabel, gambar yang ditangkap dan nama dosen pengajar.

5.2.2.2 Pengisian Nama Dosen

Bentuk implementasi pengisian nama dosen dapat dilihat pada Tabel 5.4.

Tabel 5.4 Kode program pengisian nama dosen

Baris	Kode Program
1	<code>def pengisian_nama_dosen():</code>
2	
3	<code>daftar_nama_dosen = [</code>
4	<code>'Rizal Maulana, S.T., M.T., M.Sc.',</code>
5	<code>'Fitri Utamingrum, Dr. Eng., S.T, M.T',</code>
6	<code>'Dahnial Syauqy, S.T., M.T., M.Sc.',</code>
7	<code>'Gembong Edhi Setyawan, S.T, M.T',</code>
8	<code>'Adam Hendra Brata, S.Kom., M.T., M.Sc.',</code>
9	<code>'Mochammad Hannats Hanafi Ichsan, S.ST, M.T',</code>
10	<code>'Wijaya Kurniawan, S.T, M.T',</code>
11	<code>'Barlian Henryranu Prasetyo, S.T, M.T',</code>
12	<code>'Sabriansyah Rizqika Akbar, S.T, M.Eng',</code>
13	<code>'Agung Setia Budi, S.T, M.T',</code>
14	<code>'Eko Setiawan, S.T, M.T',</code>
15	<code>'Tibyani, S.T, M.T',</code>
16	<code>'Widhi Yahya, S.Kom, M.Sc.'</code>
17	<code>]</code>
18	
19	<code># memasukkan inisial nama dosen</code>
20	<code>inisial = input('Masukan nama dosen: ')</code>
21	
22	<code># pengecekan nama dosen</code>
23	<code>for nama_dosen in daftar_nama_dosen:</code>
24	<code>if inisial.capitalize() in nama_dosen:</code>
25	<code>return nama_dosen</code>
26	<code>else:</code>
27	<code>continue</code>
28	<code>return inisial</code>

Pada Tabel 5.4, kode program di baris ke-1 merupakan nama fungsi yang dipanggil ketika melakukan pengisian nama dosen. Baris ke-3 sampai ke-17 merupakan daftar nama dosen yang disimpan ke dalam sistem. Di baris ke-20 terdapat proses memasukkan nama dosen menggunakan keyboard yang kemudian disimpan ke dalam variabel *inisial*. Dari baris ke-23 sampai ke-28 digunakan untuk mengecek apakah nama dosen yang dimasukkan terdaftar di dalam daftar nama dosen. Jika terdaftar, maka di baris ke-25 digunakan untuk mengembalikan nama dosen yang terdaftar tersebut, akan tetapi jika tidak terdaftar, maka variabel *inisial* akan dikembalikan ke fungsi pemanggil.

5.2.2.3 Pendeteksian Wajah

Bentuk implementasi pendeteksian wajah dapat dilihat pada Tabel 5.5.

Tabel 5.5 Kode program pendeteksian wajah

Baris	Kode Program
1	<code>def pendeteksian_wajah(img):</code>
2	<code> # konversi gambar ke grayscale</code>
3	<code> gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)</code>
4	
5	<code> # pilih jenis pengklasifikasi cascade</code>
6	<code> face =</code>
	<code>cv.CascadeClassifier('XML/haarcascade_frontalface_default.xml')</code>
7	
8	<code> # untuk menampung jumlah wajah yang terdeteksi</code>
9	<code> total_wajah = 0</code>
10	
11	<code> faces = face.detectMultiScale(</code>
12	<code> gray,</code>
13	<code> scaleFactor=1.2,</code>
14	<code> minNeighbors=1,</code>
15	<code> minSize=(7, 7),</code>
16	<code> maxSize=(50, 50)</code>
17	<code>)</code>
18	
19	<code> if (len(faces) != 0):</code>
20	<code> # menghapus jika ada gambar wajah uji sebelumnya</code>
21	<code> cek_file_pengujian('test-data/wajah')</code>
22	
23	<code> n = 1</code>
24	<code> for x,y,w,h in faces:</code>
25	<code> # resize area deteksi</code>
26	<code> x,y,w,h = resize_area_deteksi(x,y,w,h)</code>
27	
28	<code> # area yang dianggap sebagai wajah</code>
29	<code> wajah = img[y:y+h, x:x+w]</code>
30	
31	<code> # uji gambar wajah</code>
32	<code> ada_piksel_kulit = deteksi_warna_kulit(wajah)</code>
33	
34	<code> if ada_piksel_kulit is True:</code>
35	<code> # proses menyimpan wajah ke dalam sistem</code>
36	<code> cv.imwrite('test-data/wajah/'+str(n)+'.png',</code>
	<code>wajah)</code>
37	<code> # proses menggambar frame disekitar wajah</code>
38	<code> cv.rectangle(img, (x,y), (x+w, y+h), (255, 0,</code>
	<code>0), 1)</code>
39	<code> n+=1</code>
40	<code> total_wajah += 1</code>
41	<code> else:</code>
42	<code> continue</code>
43	
44	<code> return total_wajah</code>

Pada Tabel 5.5, kode program di baris ke-1 merupakan nama fungsi yang dipanggil ketika melakukan pendeteksian wajah, dimana di dalam fungsi ini membutuhkan satu buah parameter berupa gambar yang diuji. Di baris ke-3 terdapat program untuk mengkonversi gambar uji yang semula merupakan citra

RGB kemudian dirubah menjadi citra *grayscale* sebagai syarat dalam menjalankan proses pendeteksian di OpenCV *library*. Di baris ke-6 digunakan untuk memilih jenis pengklasifikasi – di antaranya pengklasifikasi *cascade* untuk mendeteksi mata, wajah, seluruh tubuh, bagian atas tubuh, dan bagian bawah tubuh. Di sini kita memilih pengklasifikasi *cascade* untuk mendeteksi wajah, yaitu 'haarcascade_frontalface_default.xml'.

Di baris ke-9 terdapat variabel *total_wajah* yang digunakan untuk menghitung jumlah wajah yang terdeteksi di dalam proses ini. Di baris ke-11 sampai ke-17 merupakan inialisasi kriteria pendeteksian wajah yang diinginkan, diantaranya terdapat ukuran minimal deteksi, ukuran maksimal deteksi, dan lain-lain yang kemudian disimpan ke dalam variabel *faces*.

Di baris ke-19 merupakan pengujian terhadap jumlah wajah yang berhasil dideteksi. Jika jumlah deteksi wajah tidak sama dengan nol, maka proses berlanjut ke baris ke-21 yang akan memanggil fungsi *cek_file_pengujian*, dimana fungsi baris tersebut adalah memastikan lokasi file penyimpanan wajah dalam keadaan kosong tidak terisi oleh gambar apapun. Kemudian di baris ke-23 terdapat variabel *n* yang akan digunakan untuk menamai gambar wajah tersebut dan menyimpannya ke dalam sistem.

Di baris ke-24, terdapat pengulangan terhadap setiap lokasi wajah yang berhasil dideteksi. Masuk baris ke-26, variabel *x,y,w,h* dari area wajah akan di *resize* dengan memanggil fungsi *resize_area_deteksi*, dengan mengembalikan hasil *resize* yang lebih lebar dari awalnya. Kemudian di baris ke-29 terdapat variabel *wajah* yang menyimpan lokasi wajah dari gambar uji. Untuk memastikan bahwa wajah tersebut benar-benar gambar wajah, maka di baris ke-32 diterapkan deteksi warna kulit pada variabel *wajah* dengan memanggil fungsi *deteksi_warna_kulit*, dimana fungsi tersebut mengembalikan *True* jika benar terdapat piksel kulit, dan *False* jika tidak ditemukan piksel kulit.

Di baris ke-34 digunakan untuk menguji hasil pendeteksian piksel kulit pada daerah yang dianggap sebagai gambar wajah. Jika benar terdapat piksel kulit, maka proses berlanjut ke baris ke-36 yang digunakan untuk menyimpan gambar ke dalam sistem dengan nama 'n.png', dimana n tersebut telah diinisialisasi pada baris ke-23. Kemudian setelah menyimpan wajah, maka program akan membuat *frame* segi empat berwarna biru disekitar area wajah pada baris ke-38. Lalu di baris ke-39 program meng-*increment* variabel *n* untuk berikutnya digunakan untuk menamai gambar wajah yang berhasil diklasifikasi dan baris ke-40 digunakan untuk menambahkan total jumlah wajah yang terdeteksi.

Jika ternyata tidak ditemukan piksel kulit pada gambar yang dianggap sebagai wajah, maka baris ke-41 akan dieksekusi kemudian menguji gambar wajah uji berikutnya pada list *faces* jika ada. Terakhir pada baris ke-44 digunakan untuk mengembalikan total wajah yang berhasil dideteksi kepada fungsi pemanggil.

5.2.2.4 Cek File Pengujian

Bentuk implementasi cek file pengujian dapat dilihat pada Tabel 5.6.

Tabel 5.6 Kode program cek file pengujian

Baris	Kode Program
1	<code>def cek_file_pengujian(folder):</code>
2	<code> # mendata list nama-nama file dalam folder</code>
3	<code> subject_images_names = os.listdir(folder)</code>
4	
5	<code> # mengecek apakah folder memiliki file atau tidak</code>
6	<code> if len(os.listdir(folder)) is not 0:</code>
7	
8	<code> # jika ada, hapus semua file</code>
9	<code> for image_name in subject_images_names:</code>
10	<code> image_path = folder + '/' + image_name</code>
11	<code> os.remove(image_path)</code>

Pada Tabel 5.6, kode program di baris ke-1 merupakan nama fungsi yang dipanggil ketika melakukan cek file pengujian, dimana di dalam fungsi ini membutuhkan satu buah parameter berupa lokasi penyimpanan wajah dalam bentuk *string*. Di baris ke-3 digunakan untuk mendata lis nama-nama file yang ada di dalam folder. Proses berlanjut ke baris ke-6 untuk mengecek jumlah file di lokasi penyimpanan wajah apakah terisi oleh file atau tidak. Jika ada terdapat file, maka proses penghapusan data akan diterapkan dari baris ke-9 hingga baris ke-11.

5.2.2.5 Resize Area Deteksi

Bentuk implementasi *resize* area deteksi dapat dilihat pada Tabel 5.7.

Tabel 5.7 Kode program *resize* area deteksi

Baris	Kode Program
1	<code>def resize_area_deteksi(x,y,w,h):</code>
2	<code> y = int(y - h * 0.3)</code>
3	<code> h = int(h * 1.6)</code>
4	<code> x = int(x - w * 0.3)</code>
5	<code> w = int(w * 1.6)</code>
6	
7	<code> return x,y,w,h</code>

Pada Tabel 5.7, kode program di baris ke-1 merupakan nama fungsi yang dipanggil ketika melakukan *resize* area deteksi dari proses pendeteksian wajah OpenCV *library*, dimana di dalam fungsi ini membutuhkan empat buah parameter, yaitu *x,y,w,h*, dimana *x* dan *y* merupakan titik koordinat, *w* dan *h* merupakan lebar dan tinggi dari titik koordinat yang membentuk area deteksi. Di baris ke-2 hingga baris ke-5 adalah program pelebaran area deteksi dan diakhiri dengan pengembalian keempat variabel tersebut ke fungsi pemanggil pada baris ke-7.

5.2.2.6 Deteksi Warna Kulit

Bentuk implementasi deteksi warna kulit dapat dilihat pada Tabel 5.8.

Tabel 5.8 Kode program deteksi warna kulit

Baris	Kode Program
1	<code>def deteksi_warna_kulit(img):</code>
2	<code> min_YCrCb = np.array([0,133,77],np.uint8)</code>
3	<code> max_YCrCb = np.array([255,173,127],np.uint8)</code>
4	
5	<code> imageYCrCb = cv.cvtColor(img, cv.COLOR_BGR2YCR_CB)</code>
6	
7	<code> skinRegion = cv.inRange(imageYCrCb,min_YCrCb,max_YCrCb)</code>
8	
9	<code> num_one = (skinRegion == 255).sum()</code>
10	
11	<code> if num_one > 0:</code>
12	<code> return True</code>
13	<code> else:</code>
14	<code> return False</code>

Pada Tabel 5.8, kode program di baris ke-1 merupakan nama fungsi yang dipanggil ketika melakukan pendeteksian piksel warna kulit dari dalam gambar wajah, dimana di dalam fungsi ini membutuhkan satu buah parameter yaitu gambar yang berpotensi sebagai wajah. Pada baris ke-2 dan ke-3 merupakan inialisasi batas minimal dan maksimal dari piksel warna kulit yang diolah menggunakan numpy *library*.

Di baris ke-5 terdapat program untuk mengkonversi gambar wajah yang semula merupakan citra RGB kemudian dirubah menjadi citra YCbCr dan disimpan ke dalam variabel *imageYCrCb*. Masuk ke baris ke-7, terjadi proses *threshold* terhadap *imageYCrCb*, dimana setiap pikselnya akan diberi nilai 0 apabila tidak sesuai dengan batas minimal dan maksimal nilai pada baris ke-2 dan ke-3.

Di baris ke-9 terdapat program untuk menghitung jumlah piksel yang dianggap sebagai piksel warna kulit. Jika jumlah piksel warna kulit lebih dari nol maka gambar yang berpotensi sebagai wajah itu adalah benar merupakan gambar wajah dan fungsi akan mengembalikan nilai *True*. Namun, apabila tidak lebih dari nol, maka gambar tersebut bukanlah gambar wajah dan fungsi akan mengembalikan nilai *False*.

5.2.2.7 Pengenalan Wajah

Bentuk implementasi pengenalan wajah dapat dilihat pada Tabel 5.9.

Tabel 5.9 Kode program pengenalan wajah

Baris	Kode Program
1	<code>def pengenalan_wajah(subject_dir_path, jumlah_deteksi_wajah):</code>
2	<code> # jika tidak ada wajah yang dideteksi, return 0</code>



```

3     if jumlah_deteksi_wajah is 0:
4         return 0
5     else:
6         # memulai proses pembacaan data sampel
7         faces, labels = pembacaan_data_latih('training-data')
8
9         # membuat face recognizer LBPH
10        face_recognizer = cv.face.LBPHFaceRecognizer_create()
11
12        # melatih face recognizer dengan data sampel dan label
13        sebelumnya
14        face_recognizer.train(faces, np.array(labels))
15
16        # inialisasi daftar nama mahasiswa yang hadir
17        nama = []
18
19        # list semua file gambar
20        subject_images_names = os.listdir(subject_dir_path)
21
22        for image_name in subject_images_names:
23
24            # memuat lokasi gambar
25            # contoh = training-data/s1/1.png
26            image_path = subject_dir_path + "/" + image_name
27
28            # baca gambar uji
29            image = cv.imread(image_path,0)
30
31            # prediksi hasil gambar
32            label, _ = face_recognizer.predict(image)
33
34            # mengecek di dalam daftar nama mahasiswa
35            if subjects[label] in nama:
36                continue
37
38            # memasukkan nama mahasiswa ke dalam daftar hadir
39            nama.append(subjects[label])
40
41        return nama

```

Pada Tabel 5.9, kode program di baris ke-1 merupakan nama fungsi yang dipanggil ketika melakukan pengenalan wajah, dimana di dalam fungsi ini membutuhkan dua buah parameter, yaitu lokasi pengujian wajah dalam bentuk *string* dan jumlah wajah yang terdeteksi. Apabila jumlah wajah yang terdeteksi sama dengan nol, maka fungsi ini akan mengembalikan nilai 0 dan tidak akan diproses lebih lanjut. Namun, apabila jumlah wajah yang terdeteksi lebih dari nol, maka proses pengenalan wajah akan diteruskan ke baris ke-7, dimana terdapat proses pemanggilan fungsi pembacaan data sampel yang mengembalikan dua buah lis, lis pertama berupa gambar sampel dalam bentuk citra *grayscale* dan lis kedua berupa label yang disematkan pada gambar sampel tersebut.

Pada baris ke-10, terdapat proses inialisasi pengenalan wajah menggunakan metode *Local Binary Patterns Histograms* (LBPH) dari OpenCV *library*. Baris ke-11 kemudian digunakan untuk melatih sistem dengan data lis

gambar dan label yang diperoleh dari baris ke-7. Kemudian di baris ke-16 digunakan untuk menampung nama-nama mahasiswa yang hadir.

Proses masuk ke baris ke-19 dimana gambar uji akan di data dan dibandingkan satu per satu dengan data latih. Sebelum masuk ke tahap perbandingan, gambar uji akan dikonversi ke dalam citra *grayscale* seperti yang ditunjukkan pada baris ke-28. Kemudian gambar uji akan dibandingkan dengan data latih menggunakan metode *Local Binary Patterns Histograms* (LBPH) dari *OpenCV library* dan menghasilkan label yang berisi nomor indeks sampel mahasiswa seperti yang ditunjukkan pada baris ke-31.

Untuk menghindari pengenalan wajah dengan hasil yang sama, maka diterapkan proses pengecekan terhadap indeks sampel di dalam lis nama pada baris ke-34 dan 35. Jika terdapat indeks dengan nomor yang sama, maka indeks tersebut akan *diskip* dan proses pengenalan berlanjut ke gambar uji berikutnya jika ada. Namun, jika indeks tidak ada yang sama di dalam lis nama mahasiswa, maka indeks mahasiswa tersebut dimasukkan ke dalam lis nama. Terakhir, pada baris ke-40, fungsi pengenalan akan mengembalikan lis nama-nama mahasiswa yang berhasil dikenali oleh sistem.

5.2.2.8 Pembacaan Data Sampel

Bentuk implementasi pembacaan data sampel dapat dilihat pada Tabel 5.10.

Tabel 5.10 Kode program pembacaan data sampel

Baris	Kode Program
1	<code>def pembacaan_data_latih(data_folder_path):</code>
2	<code> # list semua folder sampel mahasiswa</code>
3	<code> dirs = os.listdir(data_folder_path)</code>
4	
5	<code> # menyimpan gambar sampel wajah</code>
6	<code> faces = []</code>
7	<code> # menyimpan nama mahasiswa</code>
8	<code> labels = []</code>
9	
10	<code> # daftar mahasiswa yang ikut kelas pagi</code>
11	<code> pagi = ['s1','s2','s3','s4','s5','s6','s7']</code>
12	<code> # daftar mahasiswa yang ikut kelas siang</code>
13	<code> siang = ['s3','s8','s9','s10','s11','s12','s13','s14']</code>
14	<code> # daftar mahasiswa yang ikut kelas malam</code>
15	<code> malam = ['s1','s2','s4','s15','s16','s17']</code>
16	
17	<code> # ambil nama-nama mahasiswa berdasarkan kelas saat ini</code>
18	<code> kelas = {'pagi':pagi, 'siang':siang, 'malam':malam}</code>
19	
20	<code> if 7 <= jam <= 11:</code>
21	<code> daftar_mahasiswa = kelas['pagi']</code>
22	<code> elif 13 <= jam <= 17:</code>
23	<code> daftar_mahasiswa = kelas['siang']</code>
24	<code> else:</code>
25	<code> daftar_mahasiswa = kelas['malam']</code>
26	
27	<code> # membuka semua folder sampel mahasiswa</code>

```

28     for dir_name in dirs:
29         if dir_name not in daftar_mahasiswa:
30             continue
31         # ekstrak label nomor dari nama folder
32         # format folder mahasiswa = slabel
33         # sehingga perlu menghapus huruf 's' dari nama folder
    untuk mendapatkan label mahasiswa
34         label = int(dir_name.replace("s", ""))
35
36         # memuat lokasi data sampel mahasiswa ke-i
37         # contoh = "training-data/s1"
38         subject_dir_path = data_folder_path + "/" + dir_name
39
40         # list semua file gambar
41         subject_images_names = os.listdir(subject_dir_path)
42
43         for image_name in subject_images_names:
44
45             # memuat lokasi gambar
46             # contoh = training-data/s1/1.png
47             image_path = subject_dir_path + "/" + image_name
48
49             # baca gambar
50             image = cv.imread(image_path,0)
51
52             # butuh lokasi wajah dalam bentuk grayscale
53             faces.append(image)
54
55             # masukan label untuk sampel wajah yang bersangkutan
56             labels.append(label)
57
58     return faces, labels

```

Pada Tabel 5.10, kode program di baris ke-1 merupakan nama fungsi yang dipanggil ketika melakukan pembacaan data sampel, dimana di dalam fungsi ini membutuhkan satu buah parameter yaitu lokasi penyimpanan data sampel. Masuk ke baris ke-3 program akan mendata seluruh folder yang berisi sampel wajah mahasiswa, dimana setiap folder khusus diisi oleh sampel wajah satu orang mahasiswa. Kemudian pada baris ke-6 dan ke-8 digunakan untuk mendata semua gambar sampel dan indeks folder asal gambar wajah itu dibaca. Format folder sampel mahasiswa yang diterapkan pada sistem adalah 's1', dimana s merupakan singkatan dari *subjects* dan angka 1 merujuk kepada indeks sampel nama mahasiswa.

Sebelum masuk ke dalam proses pembacaan, nama-nama mahasiswa akan dimasukkan ke dalam kelas pagi, siang, dan malam, sesuai dengan mahasiswa yang mengambil perkuliahan pada jam tersebut. Proses tersebut digunakan untuk membantu mempercepat proses pembacaan data sampel mahasiswa, yang mana proses tersebut ditunjukkan pada baris ke-10 sampai baris ke-25.

Proses pembacaan sampel baru dimulai pada baris ke-28. Pertama, proses pengecekan apakah folder sampel mahasiswa terdaftar di dalam perkuliahan saat itu. Jika tidak terdaftar maka akan dilewati untuk proses pembacaan. Namun, jika terdaftar, maka folder sampel akan dibuka dan seluruh file sampel wajah yang

ada di dalamnya akan dibaca. Proses itu berlangsung dari baris ke-28 hingga ke-58. Untuk menghapus huruf 's' dari nama folder sampel, maka di baris ke-34 terdapat proses pengambilan angka dari nama folder mahasiswa yang gambar wajahnya saat itu sedang dibaca untuk proses lis data sampel. Kemudian di baris ke-50, setiap sampel wajah akan dikonversi ke dalam citra *grayscale* lalu dimasukkan ke dalam lis *faces*, sedangkan label foldernya dimasukkan ke dalam lis *labels*. Setelah seluruh sampel folder mahasiswa berhasil dibaca, maka fungsi ini akan mengembalikan lis gambar sampel dalam bentuk *grayscale* dan lis label sampel.



BAB 6 PENGUJIAN DAN ANALISIS

Bab ini akan membahas mengenai hasil pengujian dan analisis terhadap performa sistem presensi di dalam penelitian ini.

6.1 Pengujian Pengambilan Gambar

6.1.1 Tujuan Pengujian

Pengujian ini berfungsi untuk menguji fungsionalitas sistem ketika melakukan pengambilan gambar di dalam ruangan kelas.

6.1.2 Pelaksanaan Pengujian

Pengujian dilaksanakan dengan menggunakan pushbutton dan webcam logitech HD Webcam C310 yang masing-masing terhubung ke *port* GPIO dan USB 2.0 Raspberry Pi 3 Model B.

6.1.3 Prosedur Pengujian

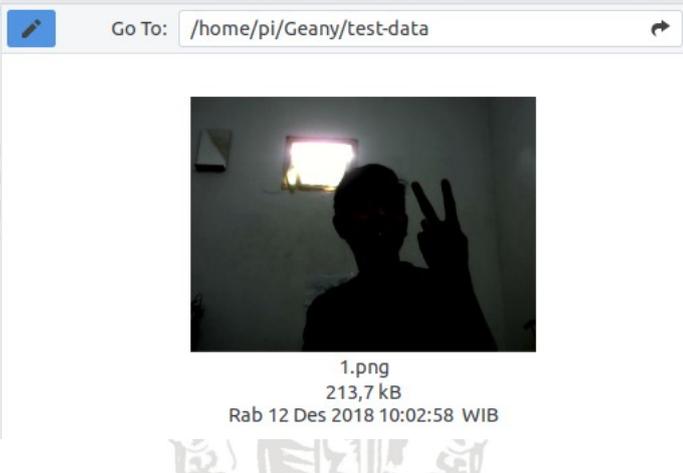
Berikut prosedur di dalam pengambilan gambar di dalam ruangan kelas:

1. Menghubungkan pushbutton, webcam, monitor, mouse, dan keyboard ke Perangkat Raspberry Pi 3 Model B.
2. Memberikan suplai *power* ke perangkat Raspberry Pi 3 Model B.
3. Membuka aplikasi Geany.
4. Menjalankan program “main.py”.
5. Kemudian pengguna menekan tombol pushbutton yang diletakkan di atas meja dosen untuk memulai pengambilan gambar.
6. Hasil pengambilan gambar dapat dilihat pada folder “test-data” pada perangkat Raspberry Pi 3 Model B.

6.1.4 Hasil Pengujian

Berikut hasil pengujian pengambilan gambar yang diterapkan di dalam ruangan kelas ditunjukkan pada Tabel 6.1.

Tabel 6.1 Hasil pengujian pengambilan gambar

Percobaan	Gambar	Status
1	 <p>Go To: /home/pi/Geany/test-data</p> <p>1.png 179,0 kB Rab 12 Des 2018 09:58:03 WIB</p>	Berhasil
2	 <p>Go To: /home/pi/Geany/test-data</p> <p>1.png 213,7 kB Rab 12 Des 2018 10:02:58 WIB</p>	Berhasil
3	 <p>Go To: /home/pi/Geany/test-data</p> <p>1.png 202,5 kB Rab 12 Des 2018 10:01:48 WIB</p>	Berhasil

6.1.5 Analisis Hasil Pengujian

Percobaan pengambilan gambar menggunakan pushbutton dan webcam logitech HD Webcam C310 pada penelitian ini berhasil diterapkan. Berhasilnya pengujian ini ditandai dengan munculnya file gambar pada folder “test-data” di perangkat Raspberry Pi 3 Model B dan perubahan informasi tanggal modifikasi file “1.png” di setiap kali pengambilan gambar.

6.2 Pengujian *Input* Nama Dosen

6.2.1 Tujuan Pengujian

Pengujian ini berfungsi untuk menguji fungsionalitas sistem ketika menerima masukan berupa teks untuk mengisi nama dosen pengajar pada sistem presensi.

6.2.2 Pelaksanaan Pengujian

Pengujian dilaksanakan dengan menggunakan keyboard yang dihubungkan dengan perangkat Raspberry Pi 3 Model B melalui *port* USB 2.0.

6.2.3 Prosedur Pengujian

Berikut prosedur di dalam pengisian nama dosen di dalam sistem presensi:

1. Menghubungkan pushbutton, webcam, monitor, mouse, dan keyboard ke Perangkat Raspberry Pi 3 Model B.
2. Memberikan suplai *power* ke perangkat Raspberry Pi 3 Model B.
3. Membuka aplikasi Geany.
4. Menjalankan program “main.py”.
5. Kemudian pengguna menekan tombol pushbutton yang diletakkan di atas meja dosen untuk memulai pengambilan gambar.
6. Selanjutnya akan muncul jendela terminal linux yang meminta pengguna untuk memasukkan nama dosen.
7. Pengguna memasukkan nama dosen menggunakan perangkat keyboard.
8. Hasil *input*-an nama dosen tersebut akan ditampilkan pada sistem informasi presensi di terminal linux.

6.2.4 Hasil Pengujian

Berikut hasil pengisian nama dosen ditunjukkan pada Tabel 6.2.

Tabel 6.2 Hasil pengujian pengisian nama dosen

Percobaan ke-	Teks <i>input</i>	Hasil
1	rizal	Masukan nama dosen: rizal [INFO:0] Initialize OpenCL runtime... ---- Memulai Presensi ---- Dosen : Rizal Maulana, S.T., M.T., M.Sc.
2	fitri	Masukan nama dosen: fitri [INFO:0] Initialize OpenCL runtime... ---- Memulai Presensi ---- Dosen : Fitri Utamingrum, Dr. Eng., S.T, M.T
3	syauqy	Masukan nama dosen: syauqy [INFO:0] Initialize OpenCL runtime... ---- Memulai Presensi ---- Dosen : Dahniel Syauqy, S.T., M.T., M.Sc.
4	adi	Masukan nama dosen: adi [INFO:0] Initialize OpenCL runtime... ---- Memulai Presensi ---- Dosen : adi
5	maudaw	Masukan nama dosen: maudaw [INFO:0] Initialize OpenCL runtime... ---- Memulai Presensi ---- Dosen : maudaw

6.2.5 Analisis Hasil Pengujian

Percobaan pengisian nama dosen menggunakan keyboard pada penelitian ini berhasil diterapkan. Berhasilnya pengujian ini ditandai dengan munculnya nama dosen pada sistem informasi presensi di terminal linux. Teks *input* yang terdaftar di dalam daftar nama dosen pada Tabel 5.4 ditampilkan dengan lengkap (nama dosen beserta titelnya), seperti yang ditunjukkan di dalam Tabel 6.2 pada percobaan ke-1,2, dan 3, sedangkan teks *input* yang tidak terdaftar pada daftar nama dosen tersebut akan ditampilkan apa adanya, seperti yang ditunjukkan di dalam Tabel 6.2, pada percobaan ke-4 dan ke-5.

6.3 Pengujian Akurasi Pendeteksian Wajah

6.3.1 Tujuan Pengujian

Pengujian ini berfungsi untuk menguji akurasi sistem ketika menerapkan pendeteksian wajah di dalam gambar uji.

6.3.2 Pelaksanaan Pengujian

Pengujian akan dilaksanakan pada ruangan dalam kelas.

6.3.3 Prosedur Pengujian

Berikut prosedur di dalam pengujian akurasi deteksi dan rekognisi wajah di dalam sistem presensi:

1. Menghubungkan pushbutton, webcam, monitor, mouse, dan keyboard ke Perangkat Raspberry Pi 3 Model B.
2. Memberikan suplai *power* ke perangkat Raspberry Pi 3 Model B.
3. Membuka aplikasi Geany.
4. Menjalankan program “main.py”.
5. Setiap mahasiswa diminta untuk menduduki kursi di dalam ruangan kelas. Mula-mula pengambilan gambar hanya menggunakan satu orang mahasiswa, kemudian ditambah satu lagi untuk mengetahui ketepatan fungsi pendeteksian wajah sistem. Hal itu dilakukan hingga jumlah mahasiswa yang duduk ketika pengambilan gambar adalah enam orang.
6. Kemudian pengguna menekan tombol pushbutton yang telah disediakan di atas meja dosen setiap kali ingin mengambil gambar.
7. Selanjutnya akan muncul jendela terminal linux yang meminta pengguna untuk memasukkan nama dosen.
8. Pengguna memasukkan nama dosen menggunakan perangkat keyboard.
9. Setelah itu, sistem akan menampilkan jendela yang memuat gambar hasil *capture* sebelumnya dengan hasil deteksi gambar wajah akan ditandai dengan kemunculan *frame* persegi berwarna biru di area sekitar wajah.
10. Menganalisis jumlah wajah yang terdeteksi dan menghitung persentase *error* di setiap pengujiannya. Untuk mengetahui persentase *error* dapat menggunakan persamaan 6.1.

$$\text{Persentase error} = \frac{\text{Jumlah wajah} - \text{Jumlah yang terdeteksi}}{\text{Jumlah wajah}} \times 100\% \quad (6.1)$$

11. Menghitung rata-rata persentase *error* di dalam pendeteksian wajah oleh sistem dengan menggunakan persamaan 6.2.

$$\text{Rata-rata persentase error} = \frac{\text{Jumlah persentase error}}{\text{Jumlah pengujian}} \quad (6.2)$$

12. Menghitung akurasi dengan menggunakan persamaan 6.3.

$$\text{Akurasi} = 100\% - \text{Rata-rata persentase error} \quad (6.3)$$

6.3.4 Hasil Pengujian

Berikut hasil pengujian pendeteksian wajah pada gambar yang diterapkan di dalam ruangan kelas yang ditunjukkan pada Tabel 6.3.

Tabel 6.3 Hasil pengujian akurasi deteksi sistem

Percobaan	Gambar	Jumlah wajah	Jumlah yang terdeteksi	Error
1		1	1	0%
2		2	2	0%
3		2	2	0%

4		3	1	66,67%
5		3	3	0%
6		4	4	0%
7		5	2	60%
8		5	5	0%

9		6	4	33,33%
10		6	5	16,67%
Rata-rata error				17,67%

6.3.5 Analisis Hasil Pengujian

Berdasarkan hasil pengujian deteksi wajah pada Tabel 6.3, maka rata-rata persentase *error*-nya adalah sebagai berikut:

$$\begin{aligned}
 \text{Rata-rata persentase error} &= \frac{\text{Jumlah persentase error}}{\text{Jumlah pengujian}} \\
 &= \frac{0 + 0 + 0 + 66,67 + 0 + 0 + 60 + 0 + 33,33 + 16,67}{10} \% \\
 &= \frac{176,67}{10} \% \\
 &= 17,67 \%
 \end{aligned}$$

Rata-rata persentase *error* yang diperoleh dari pengujian pendeteksian wajah oleh sistem tidak begitu besar, yakni sebesar 17,67%. Sehingga dengan diperolehnya nilai rata-rata persentase *error* tersebut, maka nilai akurasi pendeteksian di dalam sistem presensi ini dapat diperoleh dengan menggunakan persamaan 6.3.

$$\begin{aligned}
 \text{Akurasi} &= 100\% - \text{Rata-rata persentase error} \\
 &= 100\% - 17,67\% \\
 &= 82,33\%
 \end{aligned}$$

6.4 Pengujian Akurasi Pengenalan Wajah

6.4.1 Tujuan Pengujian

Pengujian ini berfungsi untuk menguji akurasi sistem ketika menerapkan pengenalan wajah di dalam gambar uji.

6.4.2 Pelaksanaan Pengujian

Pengujian akan dilaksanakan pada waktu pagi, siang, dan malam di dalam ruangan kelas.

6.4.3 Prosedur Pengujian

Berikut prosedur di dalam pengujian akurasi rekognisi wajah di dalam sistem presensi:

1. Pengujian akan dilaksanakan pada waktu pagi, siang dan malam.
2. Menghubungkan pushbutton, webcamera, monitor, mouse, dan keyboard ke Perangkat Raspberry Pi 3 Model B.
3. Memberikan suplai *power* ke perangkat Raspberry Pi 3 Model B.
4. Membuka aplikasi Geany.
5. Menjalankan program “main.py”.
6. Setiap mahasiswa diminta untuk menduduki kursi di dalam ruangan kelas. Mula-mula pengambilan gambar hanya menggunakan satu orang mahasiswa dan gambar diambil sebanyak tiga kali, kemudian ditambah satu lagi untuk mengetahui ketepatan fungsi pengenalan wajah sistem. Hal itu dilakukan hingga jumlah mahasiswa yang duduk ketika pengambilan gambar adalah enam orang.
7. Kemudian pengguna menekan tombol pushbutton yang telah disediakan di atas meja dosen setiap kali ingin mengambil gambar.
8. Selanjutnya akan muncul jendela terminal linux yang meminta pengguna untuk memasukkan nama dosen.
9. Pengguna memasukkan nama dosen menggunakan perangkat keyboard.
10. Setelah itu, sistem akan menampilkan terminal linux yang memuat nama-nama mahasiswa yang berhasil dikenali oleh sistem.
11. Menganalisis nama mahasiswa sebenarnya dengan nama mahasiswa yang ditampilkan dan menghitung persentase *error* di setiap pengujiannya. Untuk mengetahui persentase *error* dapat menggunakan persamaan 6.1.

12. Menghitung rata-rata persentase *error* di dalam pengenalan wajah oleh sistem dengan menggunakan persamaan 6.2.

13. Menghitung akurasi dengan menggunakan persamaan 6.3.

6.4.4 Hasil Pengujian

Berikut hasil pengujian pengenalan wajah yang diterapkan di waktu pagi ditunjukkan pada Tabel 6.4.

Tabel 6.4 Hasil pengujian akurasi rekognisi sistem pada pagi hari

Jumlah Mahasiswa	Nama Mahasiswa sebenarnya	Hasil Pengenalan Wajah	Screenshot	Benar	Error
1 orang	1. Latief	1. Latief	<pre> ----- Memulai Presensi ----- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T, M.T Tanggal : Tuesday, 01 January 2019 Waktu : 21:59 Presensi: Nama mahasiswa yang hadir: 1,Latief </pre>	1	0%
	1. Fitrahadi	1. Saleh	<pre> ----- Memulai Presensi ----- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T, M.T Tanggal : Tuesday, 01 January 2019 Waktu : 21:59 Presensi: Nama mahasiswa yang hadir: 1,Saleh </pre>	0	100%
	1. Ade	1. Ade	<pre> ----- Memulai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 20:51 Presensi: Nama mahasiswa yang hadir: 1,Ade </pre>	1	0%
2 orang	1. Latief 2. Ade	1. Latief 2. Ade	<pre> ----- Memulai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 20:53 Presensi: Nama mahasiswa yang hadir: 1,Latief 2,Ade </pre>	2	0%
	1. Latief 2. Doni	1. Latief 2. Doni	<pre> ----- Memulai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 20:54 Presensi: Nama mahasiswa yang hadir: 1,Latief 2,Doni </pre>	2	0%
	1. Doni 2. Fitrahadi	1. Doni 2. Ade	<pre> ----- Memulai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 20:55 Presensi: Nama mahasiswa yang hadir: 1,Doni 2,Ade </pre>	1	50%
3 orang	1. Fitrahadi 2. Kaesan 3. Gibrin	1. Saleh	<pre> ----- Memulai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 20:56 Presensi: Nama mahasiswa yang hadir: 1,Saleh </pre>	0	100%
	1. Latief 2. Ade 3. Fitrahadi	1. Latief 2. Saleh	<pre> ----- Memulai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 20:56 Presensi: Nama mahasiswa yang hadir: 1,Latief 2,Saleh </pre>	1	66,67%

	1. Latief 2. Doni 3. Fitrahadi	1. Latief 2. Saleh 3. Ade	----- Menuai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 20:59 Presensi: Nama mahasiswa yang hadir: 1.Latief 2.Saleh 3.Ade	1	66,67%
4 orang	1. Ade 2. Saleh 3. Kaesan 4. Gibrin	1. Latief 2. Ade	----- Menuai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 20:59 Presensi: Nama mahasiswa yang hadir: 1.Latief 2.Ade	1	75%
	1. Saleh 2. Latief 3. Kaesan 4. Gibrin	1. Saleh 2. Latief	----- Menuai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:00 Presensi: Nama mahasiswa yang hadir: 1.Saleh 2.Latief	2	50%
	1. Kaesan 2. Saleh 3. Gibrin 4. Doni	1. Kaesan 2. Saleh 3. Latief	----- Menuai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:01 Presensi: Nama mahasiswa yang hadir: 1.Kaes 2.Saleh 3.Latief	2	50%
5 orang	1. Ade 2. Saleh 3. Fitrahadi 4. Kaesan 5. Gibrin	1. Ade 2. Saleh 3. Doni	----- Menuai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:01 Presensi: Nama mahasiswa yang hadir: 1.Ade 2.Saleh 3.Doni	2	60%
	1. Saleh 2. Ade 3. Latief 4. Kaesan 5. Gibrin	1. Saleh 2. Ade 3. Latief	----- Menuai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:02 Presensi: Nama mahasiswa yang hadir: 1.Saleh 2.Ade 3.Latief	3	40%
	1. Saleh 2. Ade 3. Doni 4. Kaesan 5. Gibrin	1. Saleh 2. Ade	----- Menuai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:03 Presensi: Nama mahasiswa yang hadir: 1.Saleh 2.Ade	2	60%
	1. Ade 2. Fitrahadi 3. Latief 4. Saleh 5. Kaesan 6. Gibran	1. Ade 2. Doni 3. Latief	----- Menuai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:04 Presensi: Nama mahasiswa yang hadir: 1.Ade 2.Doni 3.Latief	2	66,67%
	1. Saleh 2. Latief 3. Doni 4. Ade 5. Kaesan 6. Gibrin	1. Saleh 2. Latief 3. Fitrahadi 4. Ade	----- Menuai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:05 Presensi: Nama mahasiswa yang hadir: 1.Saleh 2.Latief 3.Fitrahadi 4.Ade	3	50%
6 orang	1. Saleh 2. Latief 3. Ade 4. Fitrahadi	1. Saleh 2. Latief 3. Ade	----- Menuai Presensi ----- Dosen : Rizal Maulana, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:06 Presensi: Nama mahasiswa yang hadir: 1.Saleh 2.Latief 3.Ade	3	50%

	5. Kaesan 6. Gibrin				
Rata-rata error					49,17%

Berikut hasil pengujian pengenalan wajah yang diterapkan di waktu siang ditunjukkan pada Tabel 6.5.

Tabel 6.5 Hasil pengujian akurasi rekognisi sistem pada siang hari

Jumlah Mahasiswa	Nama Mahasiswa sebenarnya	Hasil Pengenalan Wajah	Screenshot	Benar	Error
1 orang	1. Ah yong	1. Joko	<pre> ----- Memulai Presensi ----- Dosen : Dahnilal Syaury, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:07 Presensi: Nama mahasiswa yang hadir: 1.Joko </pre>	0	100%
	1. Sulaiman	1. Ade	<pre> ----- Memulai Presensi ----- Dosen : Dahnilal Syaury, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:25 Presensi: Nama mahasiswa yang hadir: 1,Ade </pre>	0	100%
	1. Joko	1. Joko	<pre> ----- Memulai Presensi ----- Dosen : Dahnilal Syaury, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:26 Presensi: Nama mahasiswa yang hadir: 1.Joko </pre>	1	0%
2 orang	1. Joko 2. Mamat	1. Joko 2. Mamat	<pre> ----- Memulai Presensi ----- Dosen : Dahnilal Syaury, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:27 Presensi: Nama mahasiswa yang hadir: 1.Joko 2,Mamat </pre>	2	0%
	1. Joko 2. Ah yong	1. Joko 2. Deden	<pre> ----- Memulai Presensi ----- Dosen : Dahnilal Syaury, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:27 Presensi: Nama mahasiswa yang hadir: 1.Joko 2,Deden </pre>	1	50%
	1. Ah yong 2. Sulaiman	1. Ade 2. Sulaiman	<pre> ----- Memulai Presensi ----- Dosen : Dahnilal Syaury, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:28 Presensi: Nama mahasiswa yang hadir: 1,Ade 2,Sulaiman </pre>	1	50%
3 orang	1. Sulaiman 2. Nova 3. Mamat	1. Sulaiman 2. Joko 3. Mamat	<pre> ----- Memulai Presensi ----- Dosen : Dahnilal Syaury, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:28 Presensi: Nama mahasiswa yang hadir: 1,Sulaiman 2,Joko 3,Mamat </pre>	2	33,33%

	1. Mamat 2. Nova 3. Ah yong	1. Mamat 2. Nova 3. Joko	----- Memulai Presensi ----- Dosen : Dahnilal Syaughy, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:29 Presensi: Nama mahasiswa yang hadir: 1.Mamat 2.Nova 3.Joko	2	33,33%
	1. Mamat 2. Sulaiman 3. Joko	1. Mamat 2. Sulaiman 3. Joko	----- Memulai Presensi ----- Dosen : Dahnilal Syaughy, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:30 Presensi: Nama mahasiswa yang hadir: 1.Mamat 2.Sulaiman 3.Joko	3	0%
4 orang	1. Joko 2. Sulaiman 3. Mamat 4. Ah yong	1. Joko 2. Deden 3. Mamat	----- Memulai Presensi ----- Dosen : Dahnilal Syaughy, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:30 Presensi: Nama mahasiswa yang hadir: 1.Joko 2.Deden 3.Mamat	2	50%
	1. Joko 2. Sulaiman 3. Mamat 4. Nova	1. Joko 2. Sulaiman 3. Mamat	----- Memulai Presensi ----- Dosen : Dahnilal Syaughy, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:33 Presensi: Nama mahasiswa yang hadir: 1.Joko 2.Sulaiman 3.Mamat	3	25%
	1. Mamat 2. Joko 3. Nova 4. Sulaiman	1. Mamat 2. Joko 3. Nova	----- Memulai Presensi ----- Dosen : Dahnilal Syaughy, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:33 Presensi: Nama mahasiswa yang hadir: 1.Mamat 2.Joko 3.Nova	3	25%
5 orang	1. Joko 2. Sulaiman 3. Nova 4. Deden 5. Mamat	1. Joko 2. Sulaiman 3. Nova	----- Memulai Presensi ----- Dosen : Dahnilal Syaughy, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:34 Presensi: Nama mahasiswa yang hadir: 1.Joko 2.Sulaiman 3.Nova	3	40%
	1. Mamat 2. Joko 3. Ade 4. Sulaiman 5. Nova	1. Mamat 2. Joko 3. Ade 4. Sulaiman 5. Nova	----- Memulai Presensi ----- Dosen : Dahnilal Syaughy, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:35 Presensi: Nama mahasiswa yang hadir: 1.Mamat 2.Joko 3.Ade 4.Sulaiman 5.Nova	5	0%
	1. Sulaiman 2. Nova 3. Joko 4. Mamat 5. Nafiah	1. Sulaiman 2. Nova 3. Ade	----- Memulai Presensi ----- Dosen : Dahnilal Syaughy, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:36 Presensi: Nama mahasiswa yang hadir: 1.Sulaiman 2.Nova 3.Ade	2	60%
6 orang	1. Sulaiman 2. Mamat 3. Joko 4. Ade 5. Nova 6. Nafiah	1. Sulaiman 2. Mamat 3. Joko 4. Ade	----- Memulai Presensi ----- Dosen : Dahnilal Syaughy, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:36 Presensi: Nama mahasiswa yang hadir: 1.Sulaiman 2.Mamat 3.Joko 4.Ade	4	33,33%

	1. Sulaiman 2. Joko 3. Ade 4. Ah yong 5. Nova 6. Nafiah	1. Sulaiman 2. Joko 3. Ade 4. Mamat	---- Memulai Presensi ---- Dosen : Dahnila Syauqy, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:37 Presensi: Nama mahasiswa yang hadir: 1.Sulaiman 2.Joko 3.Ade 4.Mamat	3	50%
	1. Joko 2. Ade 3. Nova 4. Ah yong 5. Deden 6. Nafiah	1. Joko 2. Ade 3. Nova	---- Memulai Presensi ---- Dosen : Dahnila Syauqy, S.T., M.T., M.Sc. Tanggal : Tuesday, 01 January 2019 Waktu : 21:38 Presensi: Nama mahasiswa yang hadir: 1.Joko 2.Ade 3.Nova	3	50%
Rata-rata error					38,89%

Berikut hasil pengujian pengenalan wajah yang diterapkan di waktu malam ditunjukkan pada Tabel 6.6.

Tabel 6.6 Hasil pengujian akurasi rekognisi sistem pada malam hari

Jumlah Mahasiswa	Nama Mahasiswa sebenarnya	Hasil Pengenalan Wajah	Screenshot	Benar	Error
1 orang	1. Payadi	1. Payadi	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T, M.T Tanggal : Tuesday, 01 January 2019 Waktu : 21:40 Presensi: Nama mahasiswa yang hadir: 1.Payadi	1	0%
	1. Dimas	1. Doni	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T, M.T Tanggal : Tuesday, 01 January 2019 Waktu : 21:41 Presensi: Nama mahasiswa yang hadir: 1.Boni	0	100%
	1. Doni	1. Doni	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T, M.T Tanggal : Tuesday, 01 January 2019 Waktu : 21:42 Presensi: Nama mahasiswa yang hadir: 1.Boni	1	0%
2 orang	1. Doni 2. Payadi	1. Doni 2. Payadi	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T, M.T Tanggal : Tuesday, 01 January 2019 Waktu : 21:42 Presensi: Nama mahasiswa yang hadir: 1.Boni 2.Payadi	2	0%
	1. Doni 2. Fadlun	1. Doni 2. Fadlun	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T, M.T Tanggal : Tuesday, 01 January 2019 Waktu : 21:43 Presensi: Nama mahasiswa yang hadir: 1.Boni 2.Fadlun	2	0%
	1. Dimas 2. Fadlun	1. Doni 2. Latief	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T, M.T Tanggal : Tuesday, 01 January 2019 Waktu : 21:43 Presensi: Nama mahasiswa yang hadir: 1.Boni 2.Latief	0	100%

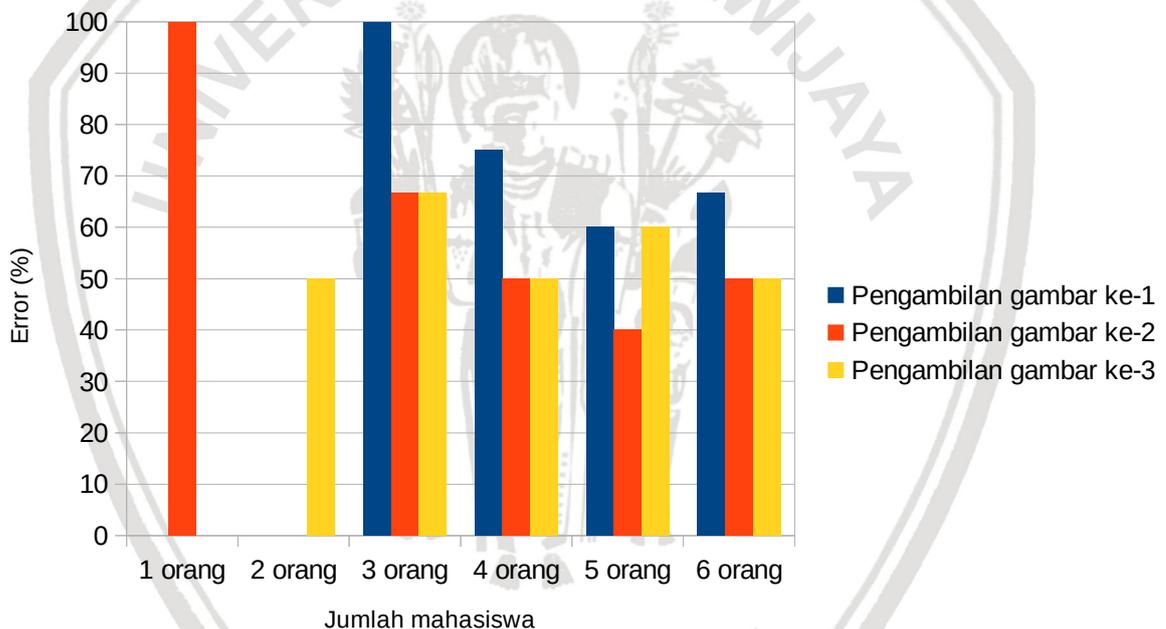


3 orang	1. Dimas 2. Payadi 3. Fitrahadi	1. Doni 2. Payadi 3. Fitrahadi	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T. M.T Tanggal : Tuesday, 01 January 2015 Waktu : 21:44 Presensi: Nama mahasiswa yang hadir: 1,Doni 2,Payadi 3,Fitrahadi	2	33,33%
	1. Fadlun 2. Payadi 3. Dimas	1. Fadlun 2. Latief	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T. M.T Tanggal : Tuesday, 01 January 2015 Waktu : 21:45 Presensi: Nama mahasiswa yang hadir: 1,Fadlun 2,Latief	1	66,67%
	1. Latief 2. Payadi 3. Dimas	1. Latief 2. Payadi	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T. M.T Tanggal : Tuesday, 01 January 2015 Waktu : 21:46 Presensi: Nama mahasiswa yang hadir: 1,Latief 2,Payadi	2	33,33%
4 orang	1. Payadi 2. Dimas 3. Doni 4. Latief	1. Payadi 2. Fadlun 3. Doni	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T. M.T Tanggal : Tuesday, 01 January 2015 Waktu : 21:47 Presensi: Nama mahasiswa yang hadir: 1,Payadi 2,Fadlun 3,Doni	2	50%
	1. Fadlun 2. Dimas 3. Payadi 4. Fitrahadi	1. Fadlun 2. Latief 3. Doni	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T. M.T Tanggal : Tuesday, 01 January 2015 Waktu : 21:48 Presensi: Nama mahasiswa yang hadir: 1,Fadlun 2,Latief 3,Doni	1	75%
	1. Latief 2. Payadi 3. Fadlun 4. Dimas	1. Latief 2. Payadi 3. Fadlun	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T. M.T Tanggal : Tuesday, 01 January 2015 Waktu : 21:49 Presensi: Nama mahasiswa yang hadir: 1,Latief 2,Payadi 3,Fadlun	3	25%
5 orang	1. Payadi 2. Doni 3. Latief 4. Fadlun 5. Dimas	1. Payadi 2. Doni 3. Latief	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T. M.T Tanggal : Tuesday, 01 January 2015 Waktu : 21:50 Presensi: Nama mahasiswa yang hadir: 1,Payadi 2,Doni 3,Latief	3	40%
	1. Fadlun 2. Latief 3. Payadi 4. Fitrahadi 5. Dimas	1. Fadlun 2. Latief 3. Payadi	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T. M.T Tanggal : Tuesday, 01 January 2015 Waktu : 21:51 Presensi: Nama mahasiswa yang hadir: 1,Fadlun 2,Latief 3,Payadi	3	40%
	1. Latief 2. Payadi 3. Fitrahadi 4. Fadlun 5. Dimas	1. Latief 2. Payadi	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T. M.T Tanggal : Tuesday, 01 January 2015 Waktu : 21:53 Presensi: Nama mahasiswa yang hadir: 1,Latief 2,Payadi	2	60%
6 orang	1. Fadlun 2. Doni 3. Latief 4. Payadi 5. Fitrahadi 6. Dimas	1. Fadlun 2. Doni 3. Latief 4. Payadi 5. Fitrahadi	---- Memulai Presensi ---- Dosen : Fitri Utaminingsrum, Dr. Eng., S.T. M.T Tanggal : Tuesday, 01 January 2015 Waktu : 21:54 Presensi: Nama mahasiswa yang hadir: 1,Fadlun 2,Doni 3,Latief 4,Payadi 5,Fitrahadi	5	16,67%

1. Latief 2. Fadlun 3. Payadi 4. Fitrahadi 5. Doni 6. Dimas	1. Latief 2. Fadlun 3. Payadi	<pre> ----- Memulai Presensi ----- Dosen : Fitri Utawiningrum, Dr. Eng., S.T., M.T Tanggal : Tuesday, 01 January 2019 Waktu : 21:55 Presensi: Nama mahasiswa yang hadir: 1.Latief 2.Fadlun 3.Payadi </pre>	3	50%
1. Latief 2. Fadlun 3. Payadi 4. Fitrahadi 5. Doni 6. Dimas	1. Latief 2. Fadlun 3. Payadi	<pre> ----- Memulai Presensi ----- Dosen : Fitri Utawiningrum, Dr. Eng., S.T., M.T Tanggal : Tuesday, 01 January 2019 Waktu : 21:57 Presensi: Nama mahasiswa yang hadir: 1.Latief 2.Fadlun 3.Payadi </pre>	3	50%
Rata-rata error				41,11%

6.4.5 Analisis Hasil Pengujian

Berdasarkan hasil pengujian pengenalan wajah pada Tabel 6.4, maka dapat diilustrasikan hasil pengujian di waktu pagi sebagaimana Gambar 6.1.



Gambar 6.1 Grafik hasil pengenalan wajah di waktu pagi

Pada Gambar 6.1, rata-rata *error* yang didapatkan dari masing-masing pengujian pengenalan wajah terhadap jumlah mahasiswa di dalam kelas adalah sebagai berikut.

$$\begin{aligned} \text{Rata-rata error untuk 1 orang} &= \frac{0 + 100 + 0}{3} \% \\ &= \frac{100}{3} \% \\ &= 33,33 \% \end{aligned}$$

$$\begin{aligned} \text{Rata-rata error untuk 2 orang} &= \frac{0 + 0 + 50}{3} \% \\ &= \frac{50}{3} \% \\ &= 16,67 \% \end{aligned}$$

$$\begin{aligned} \text{Rata-rata error untuk 3 orang} &= \frac{100 + 66,67 + 66,67}{3} \% \\ &= \frac{233,34}{3} \% \\ &= 77,78 \% \end{aligned}$$

$$\begin{aligned} \text{Rata-rata error untuk 4 orang} &= \frac{75 + 50 + 50}{3} \% \\ &= \frac{175}{3} \% \\ &= 58,33 \% \end{aligned}$$

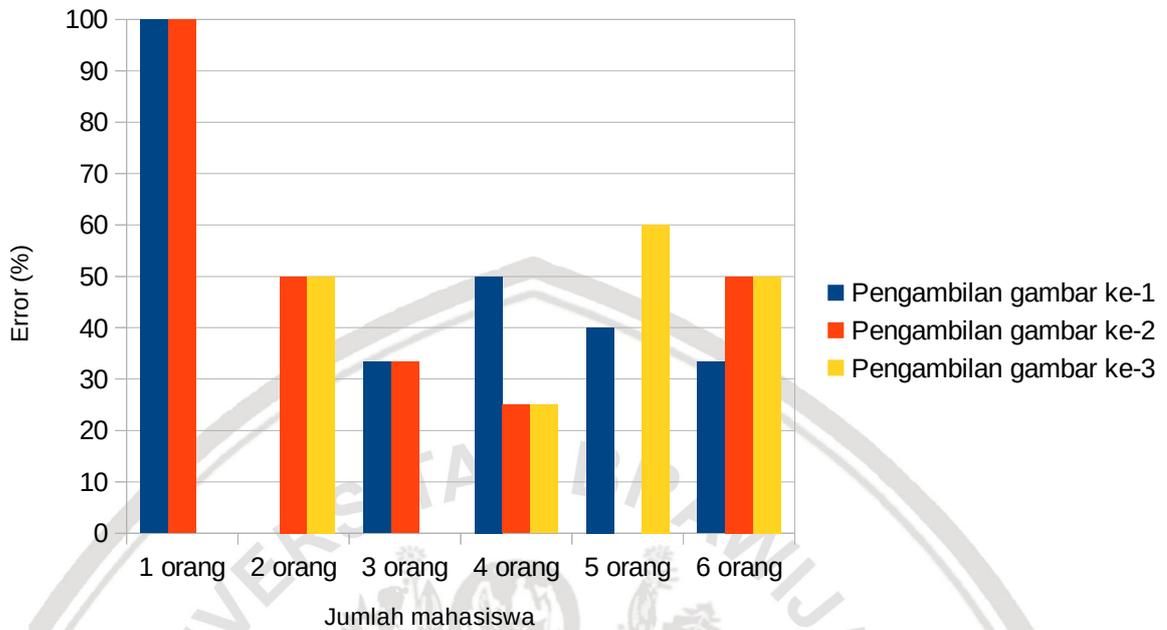
$$\begin{aligned} \text{Rata-rata error untuk 5 orang} &= \frac{60 + 40 + 60}{3} \% \\ &= \frac{160}{3} \% \\ &= 53,33 \% \end{aligned}$$

$$\begin{aligned} \text{Rata-rata error untuk 6 orang} &= \frac{66,67 + 50 + 50}{3} \% \\ &= \frac{166,67}{3} \% \\ &= 55,56 \% \end{aligned}$$

Sehingga dari hasil percobaan tersebut, didapatkan rata-rata persentase error pengenalan wajah di waktu pagi sebesar:

$$\begin{aligned} \text{Rata-rata persentase error di pagi hari} &= \frac{\text{Total rata-rata error tiap mahasiswa}}{\text{jumlah mahasiswa}} \% \\ &= \frac{33,33 + 16,67 + 77,78 + 66,67 + 53,33 + 55,56}{6} \% \\ &= \frac{303,34}{6} \% \\ &= 49,17 \% \end{aligned}$$

Berdasarkan hasil pengujian pengenalan wajah pada Tabel 6.5, maka hasil pengujian pengenalan wajah di waktu siang dapat diilustrasikan menggunakan Gambar 6.2.



Gambar 6.2 Grafik hasil pengenalan wajah di waktu siang

Pada Gambar 6.2, rata-rata *error* yang didapatkan dari masing-masing jumlah mahasiswa adalah sebagai berikut.

$$\begin{aligned}
 \text{Rata-rata error untuk 1 orang} &= \frac{100 + 100 + 0}{3} \% \\
 &= \frac{200}{3} \% \\
 &= 66,67\%
 \end{aligned}$$

$$\begin{aligned}
 \text{Rata-rata error untuk 2 orang} &= \frac{0 + 50 + 50}{3} \% \\
 &= \frac{100}{3} \% \\
 &= 33,33\%
 \end{aligned}$$

$$\begin{aligned}
 \text{Rata-rata error untuk 3 orang} &= \frac{33,33 + 33,33 + 0}{3} \% \\
 &= \frac{66,66}{3} \% \\
 &= 22,22\%
 \end{aligned}$$

$$\begin{aligned} \text{Rata-rata error untuk 4 orang} &= \frac{50 + 25 + 25}{3} \% \\ &= \frac{100}{3} \% \\ &= 33,33 \% \end{aligned}$$

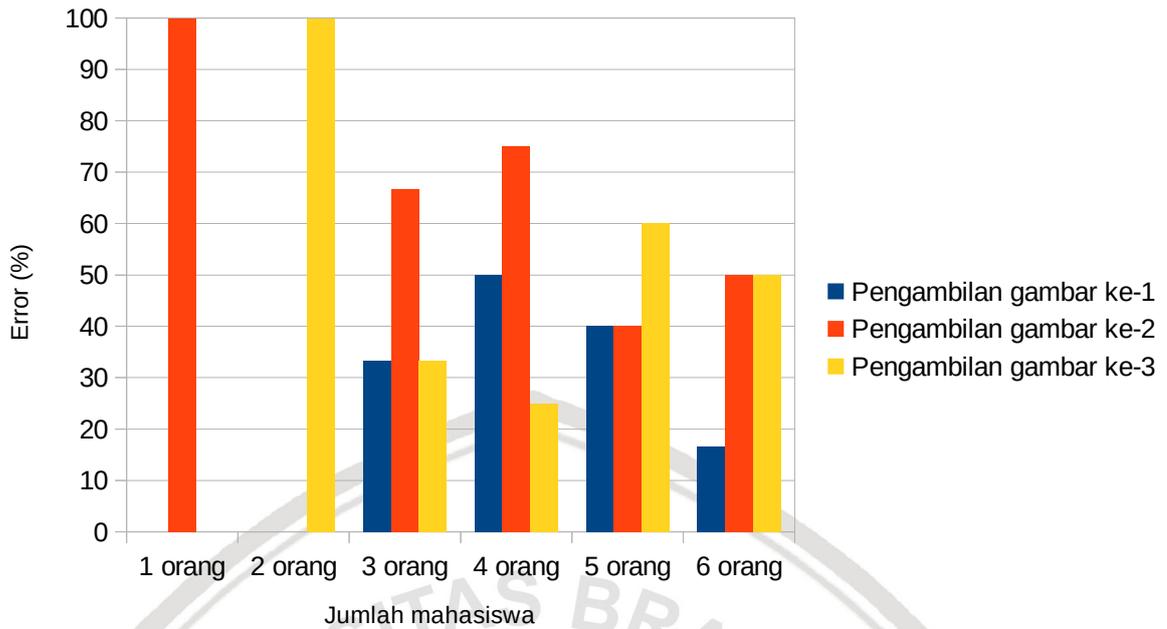
$$\begin{aligned} \text{Rata-rata error untuk 5 orang} &= \frac{40 + 0 + 60}{3} \% \\ &= \frac{100}{3} \% \\ &= 33,33 \% \end{aligned}$$

$$\begin{aligned} \text{Rata-rata error untuk 6 orang} &= \frac{33,33 + 50 + 50}{3} \% \\ &= \frac{133,33}{3} \% \\ &= 44,44 \% \end{aligned}$$

Sehingga dari hasil percobaan tersebut, didapatkan rata-rata persentase error pengenalan wajah di siang hari sebesar:

$$\begin{aligned} \text{Rata-rata persentase error di siang hari} &= \frac{\text{Total rata-rata error tiap mahasiswa}}{\text{jumlah mahasiswa}} \% \\ &= \frac{66,67+33,33+22,22+33,33+33,33+44,44}{6} \% \\ &= \frac{233,32}{6} \% \\ &= 38,89 \% \end{aligned}$$

Berdasarkan hasil pengujian pengenalan wajah pada Tabel 6.6, maka hasil pengenalan wajah di waktu malam dapat diilustrasikan melalui Gambar 6.3.



Gambar 6.3 Grafik hasil pengenalan wajah di waktu malam

Pada Gambar 6.3, rata-rata *error* yang didapatkan pada masing-masing percobaan adalah sebagai berikut.

$$\begin{aligned}
 \text{Rata-rata error untuk 1 orang} &= \frac{0 + 100 + 0}{3} \% \\
 &= \frac{100}{3} \% \\
 &= 33,33 \%
 \end{aligned}$$

$$\begin{aligned}
 \text{Rata-rata error untuk 2 orang} &= \frac{0 + 0 + 100}{3} \% \\
 &= \frac{100}{3} \% \\
 &= 33,33 \%
 \end{aligned}$$

$$\begin{aligned}
 \text{Rata-rata error untuk 3 orang} &= \frac{33,33 + 66,67 + 33,33}{3} \% \\
 &= \frac{133,33}{3} \% \\
 &= 44,44 \%
 \end{aligned}$$

$$\begin{aligned}
 \text{Rata-rata error untuk 4 orang} &= \frac{50 + 75 + 25}{3} \% \\
 &= \frac{150}{3} \% \\
 &= 50 \%
 \end{aligned}$$

$$\begin{aligned} \text{Rata-rata error untuk 5 orang} &= \frac{40 + 40 + 60}{3} \% \\ &= \frac{140}{3} \% \\ &= 46,67\% \end{aligned}$$

$$\begin{aligned} \text{Rata-rata error untuk 6 orang} &= \frac{16,67 + 50 + 50}{3} \% \\ &= \frac{116,67}{3} \% \\ &= 38,89\% \end{aligned}$$

Sehingga dari hasil percobaan tersebut, didapatkan rata-rata persentase *error* pengenalan wajah di malam hari sebesar:

$$\begin{aligned} \text{Rata-rata persentase error di malam hari} &= \frac{\text{Total rata-rata error tiap mahasiswa}}{\text{jumlah mahasiswa}} \% \\ &= \frac{33,33+33,33+44,44+50+46,67+38,89}{6} \% \\ &= \frac{246,66}{6} \% \\ &= 41,11\% \end{aligned}$$

Dari keseluruhan waktu pengujian yang diterapkan oleh sistem – baik pagi, siang, dan malam hari, diketahui bahwa rata-rata persentase *error* terbesar oleh sistem terjadi pada saat pagi hari, yaitu sebesar 49,17%. Sedangkan rata-rata persentase *error* pengenalan wajah terendah terjadi di siang hari, yaitu sebesar 38,89%.

Selain itu, dengan diketahuinya nilai rata-rata *error* pada pagi, siang, dan malam hari, maka besar nilai akurasi sistem dalam mengenali wajah-wajah mahasiswa adalah sebagai berikut.

$$\begin{aligned} \text{Akurasi di pagi hari} &= 100\% - \text{Rata-rata persentase errornya} \\ &= 100\% - 49,17\% \\ &= 50,83\% \end{aligned}$$

$$\begin{aligned} \text{Akurasi di siang hari} &= 100\% - \text{Rata-rata persentase errornya} \\ &= 100\% - 38,89\% \\ &= 61,11\% \end{aligned}$$

$$\begin{aligned} \text{Akurasi di malam hari} &= 100\% - \text{Rata-rata persentase errornya} \\ &= 100\% - 41,11\% \\ &= 58,89\% \end{aligned}$$

6.5 Pengujian Waktu Pendeteksian dan Pengenalan

6.5.1 Tujuan Pengujian

Pengujian waktu pendeteksian dan pengenalan wajah bertujuan untuk mengetahui rata-rata waktu yang dibutuhkan sistem untuk menjalankan fungsi pendeteksian dan pengenalan wajah selama proses presensi berlangsung.

6.5.2 Pelaksanaan Pengujian

Pelaksanaan pengujian waktu pendeteksian dan pengenalan wajah dalam sistem ini menggunakan *time library* dari pemrograman python. Baris program tersebut diletakkan di awal dan di akhir proses pendeteksian dan pengenalan wajah.

6.5.3 Prosedur Pengujian

Pengujian waktu pendeteksian dan pengenalan wajah dilakukan dengan prosedur sebagai berikut.

1. Menghubungkan pushbutton, webcam, monitor, mouse, dan keyboard ke Perangkat Raspberry Pi 3 Model B.
2. Memberikan suplai *power* ke perangkat Raspberry Pi 3 Model B.
3. Membuka aplikasi Geany.
4. Membuka program "main.py".
5. Mengimport *time library* ke dalam program "main.py".
6. Menambahkan baris program "start_time = time.time()" di awal fungsi pendeteksian dan "deteksi = round((time.time() - start_time),2)" di akhir proses pendeteksian. Begitu pula dengan proses pengenalan, "start_time = time.time()" di awal fungsi pengenalan dan "rekognisi = round((time.time() - start_time),2)" di akhir proses pengenalan.
7. Menjalankan program "main.py".
8. Setiap mahasiswa diminta untuk menduduki kursi di dalam ruangan kelas. Mula-mula pengambilan gambar hanya menggunakan satu orang mahasiswa dan gambar diambil sebanyak tiga kali, kemudian ditambah satu lagi untuk mengetahui ketepatan fungsi pengenalan wajah sistem. Hal itu dilakukan hingga jumlah mahasiswa yang duduk ketika pengambilan gambar adalah enam orang.
9. Kemudian pengguna menekan tombol pushbutton yang telah disediakan di atas meja dosen.

10. Selanjutnya akan muncul jendela terminal linux yang meminta pengguna untuk memasukkan nama dosen.
11. Pengguna memasukkan nama dosen menggunakan perangkat keyboard.
12. Pengguna mengamati waktu hasil deteksi dan rekognisi melalui terminal linux dan menghitung rata-rata waktu yang dibutuhkan oleh sistem presensi secara keseluruhan dalam menjalankan proses pendeteksian dan pengenalan wajah.

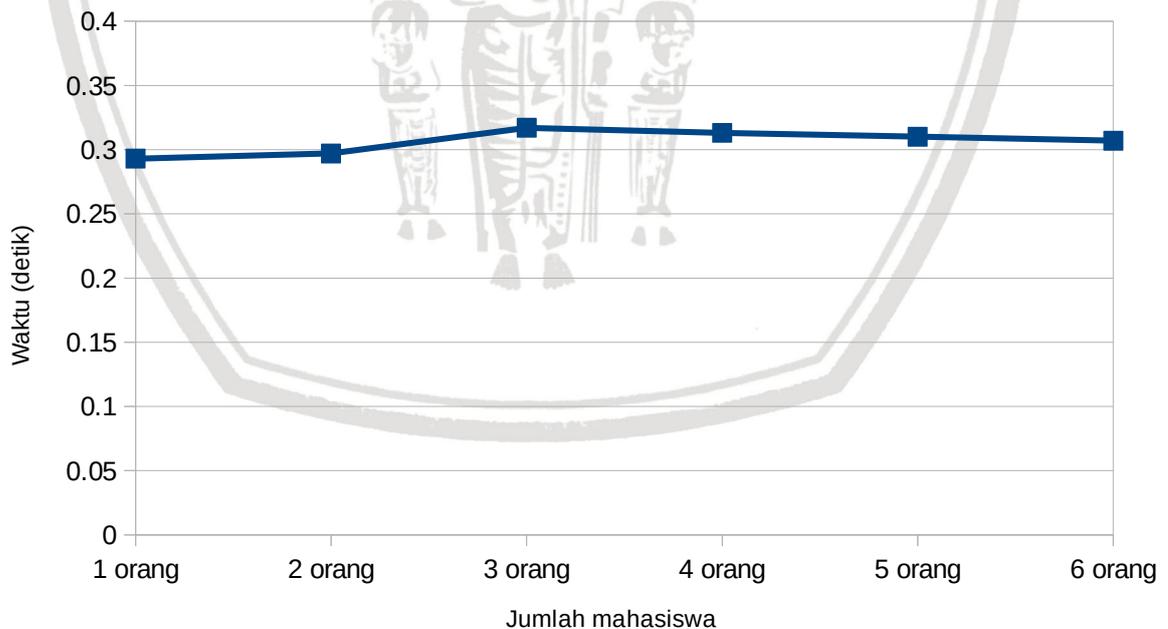
6.5.4 Hasil Pengujian

Berikut hasil pengujian waktu deteksi dan rekognisi di dalam sistem presensi ini masing-masing dapat dilihat pada Tabel 6.7 dan Tabel 6.8.

Tabel 6.7 Pengujian waktu deteksi pada sistem presensi

Percobaan	Waktu (detik)					
	1 orang	2 orang	3 orang	4 orang	5 orang	6 orang
1	0,29	0,3	0,32	0,31	0,31	0,3
2	0,29	0,29	0,32	0,31	0,32	0,31
3	0,3	0,3	0,31	0,32	0,3	0,31
Rata-rata	0,293	0,297	0,317	0,313	0,31	0,307

Jika menggunakan ilustrasi grafik, maka rata-rata waktu pendeteksian di dalam sistem akan terlihat seperti Gambar 6.4.



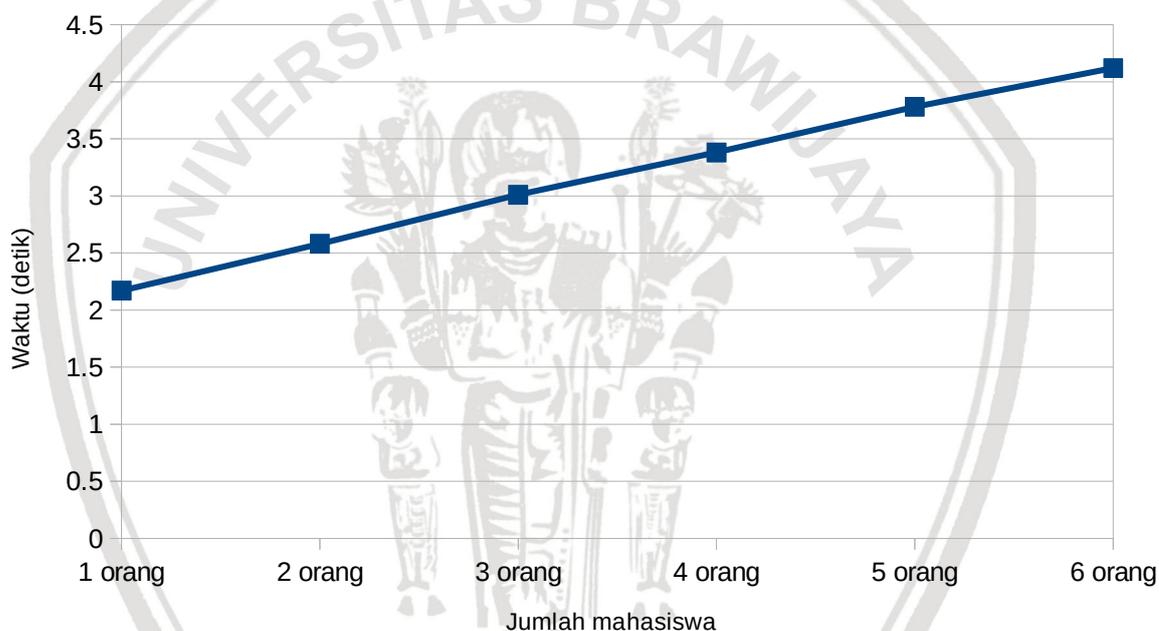
Gambar 6.4 Grafik rata-rata waktu pendeteksian oleh sistem

Dan untuk hasil pengujian rekognisi wajah pada sistem, dapat dilihat pada Tabel 6.8.

Tabel 6.8 Pengujian waktu rekognisi pada sistem presensi

Percobaan	Waktu (detik)					
	1 orang	2 orang	3 orang	4 orang	5 orang	6 orang
1	2,18	2,59	3,01	3,36	3,75	4,13
2	2,17	2,59	3,01	3,37	3,85	4,12
3	2,16	2,56	3,0	3,4	3,75	4,11
Rata-rata	2,17	2,58	3,01	3,38	3,78	4,12

Jika menggunakan ilustrasi grafik, maka rata-rata waktu pengenalan di dalam sistem akan terlihat seperti Gambar 6.5.



Gambar 6.5 Grafik rata-rata waktu pengenalan oleh sistem

6.5.5 Analisis Hasil Pengujian

Setelah mendapatkan waktu pendeteksian dan pengenalan wajah di dalam sistem presensi tersebut maka dapat diketahui bahwa waktu pendeteksian tidak mengalami perubahan signifikan terhadap perubahan sampel uji, yaitu ketika pendeteksian untuk 1 orang sebesar 0,293 detik, ketika 2 orang sebesar 0,297 detik, ketika 3 orang sebesar 0,317 detik, ketika 4 orang sebesar 0,313 detik, ketika 5 orang sebesar 0,31 detik dan ketika 6 orang sebesar 0,307 detik. Sedangkan waktu pengenalan mengalami perubahan terhadap perubahan

sampel uji, yaitu semakin bertambahnya jumlah uji maka semakin lama waktu yang dibutuhkan untuk mengenalinya, yaitu ketika pengenalan 1 orang dibutuhkan waktu sekitar 2,17 detik, ketika 2 orang sekitar 2,58 detik, ketika 3 orang sekitar 3,01 detik, ketika 4 orang sekitar 3,38 detik, ketika 5 orang sekitar 3,78 detik dan ketika 6 orang sekitar 4,12 detik.



BAB 7 PENUNTUP

Bab ini akan membahas mengenai kesimpulan dan saran terhadap hasil pengujian sistem presensi di dalam penelitian ini.

7.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut.

1. Nilai akurasi deteksi wajah di dalam sistem presensi ini adalah sebesar 82,33%. Dengan besar nilai akurasi pendeteksian tersebut, maka bisa dikatakan bahwa metode pendeteksian dalam sistem presensi ini bisa dijadikan pilihan di dalam mendeteksi wajah-wajah mahasiswa dari dalam sebuah gambar.
2. Nilai akurasi rekognisi wajah di dalam sistem presensi ini adalah sebesar 50,83% di waktu pagi, 61,11% di waktu siang, dan 58,89% di waktu malam. Dengan besar nilai akurasi pengenalan tersebut, maka metode *Local Binary Patterns Histograms* (LBPH) masih belum bisa dijadikan pilihan dalam mengenali wajah-wajah mahasiswa dari dalam gambar untuk keperluan sistem presensi.
3. Waktu komputasi yang dibutuhkan dalam pendeteksian wajah tidak mengalami perubahan signifikan terhadap perubahan sampel uji, yaitu ketika pendeteksian untuk 1 orang sebesar 0,293 detik, ketika 2 orang sebesar 0,297 detik, ketika 3 orang sebesar 0,317 detik, ketika 4 orang sebesar 0,313 detik, ketika 5 orang sebesar 0,31 detik dan ketika 6 orang sebesar 0,307 detik. Sedangkan waktu pengenalan mengalami perubahan terhadap perubahan sampel uji, yaitu semakin bertambahnya jumlah uji maka semakin lama waktu yang dibutuhkan untuk mengenalinya, yaitu ketika pengenalan 1 orang dibutuhkan waktu sekitar 2,17 detik, ketika 2 orang sekitar 2,58 detik, ketika 3 orang sekitar 3,01 detik, ketika 4 orang sekitar 3,38 detik, ketika 5 orang sekitar 3,78 detik dan ketika 6 orang sekitar 4,12 detik.

7.2 Saran

1. Melanjutkan pengerjaan sistem dengan menggunakan kamera yang memiliki tingkat resolusi yang lebih tinggi untuk mendapatkan hasil kerapatan piksel yang lebih baik dan memiliki lensa *wide-angle* untuk bisa menjangkau keseluruhan isi ruangan kelas.
2. Menggunakan metode pengenalan wajah yang lebih kuat.



- Patravali, S.D., Waykule, J. and Katre, A.D., 2014. Skin Segmentation Using YCBCR and RGB Color Models International Journal of Advanced Research in Skin Segmentation Using YCBCR and RGB Color Models. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(August).
- Rahim, A., Hossain, N., Wahid, T. and Azam, S., 2013. Face Recognition using Local Binary Patterns (LBP). *Global Journal of Computer Science and Technology Graphics & Vision*, [online] 13(4), pp.469–481. Available at: <<http://computerresearch.org/stpr/index.php/gjcst/article/viewArticle/1431>>.
- Raspberry Pi Foundation, 2016. *Raspberry Pi 3 Model B*. [online] Available at: <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>> [Accessed 30 Aug. 2018].
- Shapiro, L. and Stockman, G., 2000. *Computer Vision*. 1st ed. Pearson.
- Viola, P. and Jones, M.J., 2004. Robust Real-Time Face Detection. 57(2), pp.137–154.
- Xigui, Y., Elliot, K., Oh, J., Enid, T. and Thomas, P., 2018. *Creating a Practically Negotiated Learning Environment*. [online] Available at: <<http://ldtglobal.coe.uga.edu/design-document/>> [Accessed 30 Dec. 2018].
- Zhou, H., Wu, J. and Zhang, J., 2010. *Digital Image Processing: Part I*. Ventus Publishing ApS.

LAMPIRAN A KODE PROGRAM

Baris	Kode Program
1	import RPi.GPIO as GPIO
2	import time
3	import cv2 as cv
4	import os
5	import numpy as np
6	from datetime import datetime
7	
8	# daftar nama mahasiswa di dalam kelas x
9	subjects = ["",
10	"Fitrahadi", #1
11	"Doni", #2
12	"Ade", #3
13	"Latief", #4
14	"Saleh", #5
15	"Kaesan", #6
16	"Gibrin", #7
17	"Sulaiman", #8
18	"Ah yong", #9
19	"Joko", #10
20	"Nova", #11
21	"Mamat", #12
22	"Deden", #13
23	"Nafiah", #14
24	"Fadlun", #15
25	"Dimas", #16
26	"Payadi" #17
27]
28	
29	# proses penyediaan waktu presensi
30	now = datetime.now()
31	hari = now.day
32	jam = now.hour
33	menit = now.minute
34	
35	def main():
36	# set mode GPIO ke BOARD
37	GPIO.setmode(GPIO.BOARD)
38	
39	# inisialisasi pin Raspi
40	pinPushbutton = 11
41	GPIO.setup(pinPushbutton, GPIO.IN, pull_up_down=GPIO.PUD_UP)
42	
43	# menginisialisasi webcamera yg digunakan
44	cap = cv.VideoCapture(0)
45	print('Tekan pushbutton untuk memulai.')
46	# pengulangan forever
47	try:
48	while True:
49	# memulai stream gambar dengan webcamera
50	_, frame = cap.read()
51	
52	cv.imshow('a', frame)
53	# jika pushbutton ditekan
54	if GPIO.input(pinPushbutton) == False: