

**RANCANG BANGUN SISTEM PENGENALAN WAJAH
UNTUK KEHADIRAN MENGGUNAKAN METODE
EIGENFACE BERBASIS ANDROID**

TUGAS AKHIR

Disusun untuk Memenuhi Syarat Kelulusan Program Strata I pada
Sekolah Tinggi Manajemen Informatika dan Komputer
(STMIK) Palangkaraya



OLEH

MUHAMMAD DADANG SETIAWAN
NIM C1555201095
PROGRAM STUDI TEKNIK INFORMATIKA

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
2019**

**RANCANG BANGUN SISTEM PENGENALAN WAJAH
UNTUK KEHADIRAN MENGGUNAKAN METODE
EIGENFACE BERBASIS ANDROID**

TUGAS AKHIR

Disusun untuk Memenuhi Syarat Kelulusan Program Strata I pada
Sekolah Tinggi Manajemen Informatika dan Komputer
(STMIK) Palangkaraya

OLEH

MUHAMMAD DADANG SETIAWAN
NIM C1555201095
PROGRAM STUDI TEKNIK INFORMATIKA

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
2019**

LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama Mahasiswa : MUHAMMAD DADANG SETIAWAN
N I M : C1555201095

Menyatakan bahwa Tugas Akhir dengan judul :

**RANCANG BANGUN SISTEM PENGENALAN WAJAH UNTUK
KEHADIRAN MAHASISWA MENGGUNAKAN METODE EIGENFACE
BERBASIS ANDROID**

adalah hasil karya saya dan bukan merupakan duplikasi sebagian atau seluruhnya dari karya orang lain, kecuali bagian yang sumber informasi dicantumkan.

Pernyataan ini dibuat dengan sebenar-benarnya secara sadar dan bertanggung jawab dan saya bersedia menerima sanksi pembatalan Tugas Akhir apabila terbukti melakukan duplikasi terhadap Tugas Akhir atau karya ilmiah lain yang sudah ada.

Palangka Raya, 19 Juli 2019

g Membuat Pernyataan,




MUHAMMAD DADANG SETIAWAN

PERSETUJUAN

RANCANG BANGUN SISTEM PENGENALAN WAJAH UNTUK KEHADIRAN MAHASISWA MENGGUNAKAN METODE *EIGENFACE* BERBASIS ANDROID

Tugas Akhir ini telah disetujui untuk diujikan
pada Tanggal 19 Juli 2019

Pembimbing I,

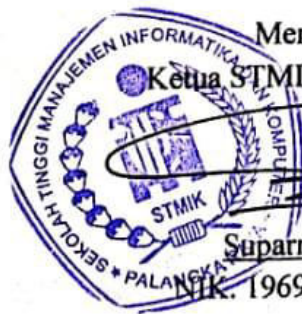


Sulistyowati, S.Kom., M.Cs
NIK. 198212162007002

Pembimbing II,




Herkules, S.Kom., M.Cs
NIK. 198510042010106



Mengetahui

Ketua STMIK Palangkaraya,



Suparno, S.Kom

NIK. 196901041995105

PENGESAHAN

RANCANG BANGUN SISTEM PENGENALAN WAJAH UNTUK KEHADIRAN MAHASISWA MENGGUNAKAN METODE *EIGENFACE* BERBASIS ANDROID

Tugas Akhir ini telah Diuji, Dinilai dan Disahkan
Oleh Tim Seminar pada Tanggal 19 Juli 2019

Tim Penguji Tugas Akhir :

1. Maura Widyarningsih, S.Kom., M.Cs
Ketua
2. Hotmian Sitohang, M.Kom
Sekretaris
3. Lili Rusdiana, M.Kom
Anggota
4. Sulistyowati, S.Kom., M.Cs
Anggota
5. Herkules, S.Kom., M.Cs
Anggota



.....

.....

.....

.....

.....

MOTTO DAN PERSEMBAHAN

*Barangsiapa bersungguh-sungguh
maka ia akan berhasil*

Tugas Akhir ini kupersembahkan untuk

- *Kedua orang tua :
Muhaimin dan mutmainah yang telah
memberiku dukungan dan motivasi
dalam segala hal serta memberikan
kasih sayang yang teramat besar yang
tak mungkin bisa ku balas dengan
apapun*
- *Saudara dan saudariku :
Muhammad Dani Efendi dan Syaza
Batrisyia*
- *Teman-temanku :
Kelas C Teknik Informatika 2015 untuk
semangat dan pendapat yang kalian
berikan*

ABSTRAK

Muhammad Dadang Setiawan, C1555201095, 2019. *Rancang Bangun Sistem Pengenalan Wajah Untuk Kehadiran Mahasiswa Menggunakan Metode Eigenface Berbasis Android*, Pembimbing I Sulistyowati, S.Kom., M.Cs., Pembimbing II Herkules, S.Kom., M.Cs.

Sistem kehadiran mahasiswa merupakan proses pencatatan kehadiran mahasiswa di kelas. Sistem kehadiran ini dapat menjadi bukti bahwa seorang mahasiswa telah berada di dalam kelas. Pencatatan kehadiran mahasiswa dapat dilakukan dengan berbagai cara. Pada Jurusan Teknik Informatika STMIK Palangkaraya, sistem kehadiran mahasiswa dilakukan secara manual yaitu melakukan tanda tangan pada kertas daftar kehadiran kelas. Hal ini menimbulkan celah kelemahan yaitu kemungkinan terjadinya kecurangan dalam pencatatan kehadiran.

Dalam tugas akhir ini dirancang sebuah sistem pengenalan wajah untuk kehadiran dengan mengimplementasikan pustaka (*library*) Android Face Recognition Eigenface dimana didalamnya terdapat algoritma dan pustaka-pustaka lainnya yang berhubungan dengan proses pengenalan wajah. Implementasi dari sistem ini diharapkan dapat mengatasi kemungkinan terjadinya kecurangan dalam pencatatan kehadiran.

Sistem pengenalan wajah berhasil diimplementasikan untuk kehadiran mahasiswa, hal itu ditunjukkan pada pengujian sistem mampu mengenali wajah mahasiswa yang terdaftar dan mengirim verifikasi kehadiran ke server untuk diolah menjadi rekapitulasi kehadiran. Hasil pengujian dari implementasi sistem menunjukkan hasil yang cukup memuaskan yaitu sistem memiliki tingkat akurasi 90%. Hal ini terlihat dari hasil pengujian dimana 9 dari 10 mahasiswa berhasil dikenali atau diverifikasi oleh sistem. Hasil pengujian kegunaan juga memiliki hasil yang cukup memuaskan. Hal ini terlihat dari hasil kuisioner pengujian yang memperoleh hasil diatas 4 dari skala 1 sampai 5. Dari penilaian ini dapat disimpulkan bahwa aplikasi ini cukup layak dan mudah untuk digunakan.

Kata kunci : Android, *Face Recognition*, Sistem Kehadiran Mahasiswa

ABSTRACT

Muhammad Dadang Setiawan, C1555201095, 2019. *Rancang Bangun Sistem Pengenalan Wajah Untuk Kehadiran Mahasiswa Menggunakan Metode Eigenface Berbasis Android*, Pembimbing I Sulistyowati, S.Kom., M.Cs., Pembimbing II Herkules, S.Kom., M.Cs.

Attendance system is the process of recording student attendance in class. Attendance system can be proof that a student has been in the class. Student attendance records can be done in various ways. In the STMIK Palangkaraya Informatics Engineering Department, the student attendance system is done manually, which is to sign on the class attendance list paper. This raises a weakness that is the possibility of fraud in recording attendance.

In this final project a face recognition system for attendance is designed by implementing an Android library Face Recognition Eigenface wherein there are other algorithms and libraries related to the face recognition process. The implementation of this system is expected to overcome the possibility of fraud in recording attendance.

face recognition system is successfully implemented for the presence of students, it is shown in the system testing that it is able to recognize the faces of registered students and send attendance verification to the server to be processed into a recapitulation of attendance. Test results from the system implementation show good results, the system has a 90% accuracy rate. This is seen from the results of the test where 9 out of 10 students were successfully identified or verified by the system. The results of usability testing also have good results. This can be seen from the results of the testing questionnaire which obtained results above 4 on a scale of 1 to 5.

Keywords : Android, Face Recognition, Attendance System.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah yang Maha Esa Karena atas Segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul **”RANCANG BANGUN SISTEM PENGENALAN WAJAH UNTUK KEHADIRAN MAHASISWA MENGGUNAKAN METODE *EIGENFACE* BERBASIS ANDROID”**.

Pengerjaan tugas akhir ini penulis lakukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana Komputer di Program Studi S-1 Jurusan Teknik Informatika STMIK Palangkaraya.

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan baik secara langsung maupun tidak langsung selama proses pengerjaan hingga selesai, antara lain :

1. Allah yang Maha Esa atas segala karunia dan rahmat-Nya yang telah diberikan selama ini.
2. Keluarga penulis, Ayah, Ibu, dan saudara kandung serta keluarga yang tidak dapat penulis tuliskan satu per satu yang telah memberikan dukungan moral, material, dan juga doa untuk penulis.
3. Suparno, M.Kom selaku Ketua Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya.
4. Sulistyowati, S.Kom., M.Cs. selaku dosen pembimbing I yang telah memberikan bimbingan dan arahan dalam pengerjaan tugas akhir ini.

5. Herkules, S.Kom., M.Cs. selaku dosen pembimbing II yang telah memberikan bimbingan dan arahan dalam pengerjaan tugas akhir ini.
6. Bapak/Ibu dosen Teknik Informatika yang telah banyak memberikan ilmu kepada penulis.
7. Teman-teman Mahasiswa Teknik Informatika STMIK Palangkaraya Angkatan 2015.

Penulis menyadari bahwa masih terdapat banyak kekurangan dalam tugas akhir ini. Oleh karena itu, penulis menerima dengan rendah hati kritik dan saran untuk pembelajaran dan perbaikan kedepannya. Semoga tugas akhir ini dapat memberikan manfaat yang sebaik-baiknya.

Palangkaraya, Juli 2019

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN	i
HALAMAN PENGESAHAN	ii
HALAMAN MOTTO DAN PERSEMBAHAN	iii
HALAMAN ABSTRAK	iv
HALAMAN ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv

BAB I PENDAHULUAN

A. Latar Belakang	1
B. Rumusan Masalah	3
C. Batasan Masalah	4
D. Tujuan	4
E. Manfaat	5
F. Metode Penelitian	5
1. Tahap Pengumpulan Data	5
a. Metode Kepustakaan	5
b. Metode Observasi	6
c. Metode Wawancara	6
2. Tahap Pengembangan Perangkat Lunak	6
a. <i>Communication</i>	6
b. <i>Quick Plan</i>	7
c. <i>Modelling Quick Design</i>	7
d. <i>Contraction of Prototype</i>	7
e. <i>Deployment Delivey and Feedback</i>	7
G. Sistematika Penulisan	8

BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait	10
B. Algortima Eigenface	12
C. Perhitungan Eigenface	13
1. Penyusunan Flatvector Matriks Citra	13
2. Hitung Nilai Tengah atau Mean	14
3. Hitung Selisih antara Training Image dengan Nilai Tengah .	14
4. Hitung Nilai Matriks Kovarian	15

5. Hitung Nilai Eigenvalue dan Eigenvector.....	15
D. Penggunaan Pustaka Android Face Recognition Eigenface	18
1. Penggunaan <i>OpenCV Library</i>	19
2. Penggunaan Algoritma Tahap Deteksi Wajah	22
3. Penggunaan Algoritma Tahap Preprocessing	23
4. Penggunaan Algoritma Tahap Ekstraksi Fitur	24
5. Penggunaan LIBSVM	25
6. Penggunaan Algoritma Tahap Klasifikasi	26
E. API	27
F. Restful API	27
G. JSON	29
H. UML	29
I. OOP	30

BAB III ANALISIS DAN PERANCANGAN SISTEM

A. Analisis	31
1. Analisis Permasalahan	31
2. Analisis Kebutuhan	31
a. Kebutuhan Fungsional	32
b. Kebutuhan Non-Fungsional	32
B. Perancangan	32
1. Perancangan Data	32
2. Perancangan Arsitektur Sistem	34
3. Perancangan Use Case	35
4. Perancangan Arsitektur Perangkat Lunak	36
5. Perancangan Diagram Aktivitas	37
6. Perancangan Diagram Kelas	39
7. Perancangan Antarmuka Pengguna	41
a. Rancangan Halaman Antarmuka Login	41
b. Rancangan Halaman Antarmuka Menu Utama	42
c. Rancangan Halaman Antarmuka Matakuliah	43
d. Rancangan Halaman Antarmuka Update Status Matakuliah.....	44
e. Rancangan Halaman Antarmuka Mahasiswa	45
f. Rancangan Halaman Antarmuka Tambah Data Wajah ...	46
g. Rancangan Halaman Antarmuka Web	47
8. Perancangan Proses	48
a. Rancangan Alir Proses Utama Perangkat Lunak	48
b. Rancangan Alir Proses Tambah Data Wajah Mahasiswa	49

c. Rancangan Alir Proses Mengaktifkan Status Matakuliah	50
d. Rancangan Alir Proses Verifikasi Kehadiran	51
e. Rancangan Alir Proses Menonaktifkan Status Matakuliah	53
f. Rancangan Alir Proses Pengenalan Wajah dengan Pustaka Android Face Recognition Eigenface	54

BAB IV IMPLEMENTASI DAN PENGUJIAN

A. Lingkungan Implementasi Perangkat Lunak	56
B. Implementasi Antarmuka Pengguna	56
1. Implementasi Halaman Antarmuka Login	57
2. Implementasi Halaman Antarmuka Menu Utama	58
3. Implementasi Halaman Antarmuka Matakuliah	59
4. Implementasi Halaman Antarmuka Update Status Matakuliah	60
5. Implementasi Halaman Antarmuka Menu Mahasiswa	61
6. Implementasi Halaman Antarmuka Tambah Data Wajah	62
7. Implementasi Halaman Antarmuka Latih Data Wajah	63
8. Implementasi Halaman Antarmuka Web	64
C. Listing Program	65
D. Lingkungan Pengujian	82
E. Pengujian Kegunaan	82
F. Hasil Pengujian Kegunaan	84
G. Pengujian Tingkat Akurasi	86
H. Pengujian Kapasitas Pengolahan Data Wajah	87

BAB V KESIMPULAN DAN SARAN

A. Kesimpulan	89
B. Saran	90

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel 1. Atribut Antarmuka Halaman Login	41
Tabel 2. Atribut Antarmuka Halaman Utama	42
Tabel 3. Atribut Antarmuka Halaman Matakuliah	43
Tabel 4. Atribut Antarmuka Halaman Update Status Matakuliah	44
Tabel 5. Atribut Antarmuka Halaman Menu Mahasiswa	45
Tabel 6. Atribut Antarmuka Halaman Tambah Data Wajah.....	46
Tabel 7. Lingkungan Implementasi Perangkat Keras dan Lunak	56
Tabel 8. Kode Program Koneksi Database Server.....	65
Tabel 9. Penjelasan Listing Program Koneksi Database Server	65
Tabel 10. Kode Listing Program Login Admin Server.....	66
Tabel 11. Penjelasan Listing Program Login Admin Server	66
Tabel 12. Kode Program Tambah Data Perkuliahan	67
Tabel 13. Penjelasan Listing Program Tambah Data Perkuliahan.....	67
Tabel 14. Kode Program Edit Data Perkuliahan	68
Tabel 15. Penjelasan Listing Program Edit Data Perkuliahan	68
Tabel 16. Kode Program Hapus Data Perkuliahan	68
Tabel 17. Penjelasan Listing Program Hapus Data Perkuliahan	69
Tabel 18. Kode Program Menampilkan Laporan Kehadiran	69
Tabel 19. Penjelasan Listing Program Menampilkan Laporan Kehadiran	69
Tabel 20. Kode Program Koneksi Database Client	70
Tabel 21. Penjelasan Listing Program Koneksi Database Client.....	70
Tabel 22. Kode Program API Login Client	70
Tabel 23. Penjelasan Listing Program API Login Client.....	71
Tabel 24. Kode Program API Verifikasi Kehadiran	72
Tabel 25. Penjelasan Listing Program API Verifikasi Kehadiran	73
Tabel 26. Kode Program API Menampilkan Matakuliah	73
Tabel 27. Penjelasan Listing Program API Menampilkan Matakuliah.....	74
Tabel 28. Kode Program Tambah Data Wajah	74
Tabel 29. Penjelasan Listing Program Tambah Data Wajah	76
Tabel 30. Kode Program Latih Data Wajah.....	76
Tabel 31. Penjelasan Listing Program Latih Data Wajah	78
Tabel 32. Kode Program Verifikasi Kehadiran.....	79
Tabel 33. Penjelasan Listing Program Verifikasi Kehadiran.....	81
Tabel 34. Lingkungan Pengujian Fungsionalitas Perangkat Lunak.....	82
Tabel 35. Kuisisioner Pengujian Kegunaan.....	83
Tabel 36. Peserta Pengujian Kegunaan	84
Tabel 37. Rekapitulasi Hasil Penilaian	85
Tabel 38. Hasil Pengujian Tingkat Akurasi	86
Tabel 39. Rincian Kapasitas Data Wajah Tersimpan.....	88

DAFTAR GAMBAR

Gambar 1. Citra Wajah 1	13
Gambar 2. Citra Wajah 2	13
Gambar 3. Libray Android Face Recognition(1)	18
Gambar 4. Libray Android Face Recognition(2)	19
Gambar 5. Proses Cropping Citra	20
Gambar 6. Proses Resize Citra.....	21
Gambar 7. Proses Grayscale Citra	21
Gambar 8. Proses Histogram Equalization Citra	22
Gambar 9. LIBSVM dalam Library Android Face Recognition	25
Gambar 10. Arsitektur LIBSVM.....	26
Gambar 11. Contoh Gambar Data Wajah Terambil dan Tersimpan.....	33
Gambar 12. Arsitektur Sistem Pengenalan Wajah untuk Kehadiran Mahasiswa ..	34
Gambar 13. Use Case Sistem Pengenalan Wajah untuk Kehadiran Mahasiswa ...	35
Gambar 14. Diagram Arsitektur Sistem.....	36
Gambar 15. Diagram Aktivitas Tambah Data Wajah	37
Gambar 16. Diagram Aktivitas Latih Data Wajah.....	38
Gambar 17. Diagram Aktivitas Mengaktifkan Matakuliah dan Verifikasi Kehadiran.....	38
Gambar 18. Diagram Aktivitas Menonaktifkan Status Matakuliah.....	39
Gambar 19. Diagram Kelas.....	40
Gambar 20. Rancangan Antarmuka Halaman Login	41
Gambar 21. Rancangan Antarmuka Halaman Menu Utama.....	42
Gambar 22. Rancangan Antarmuka Halaman Matakuliah	43
Gambar 23. Rancangan Antarmuka Halaman Update Status Matakuliah	44
Gambar 24. Rancangan Antarmuka Halaman Mahasiswa.....	45
Gambar 25. Rancangan Antarmuka Halaman Tambah Data Wajah.....	46
Gambar 26. Rancangan Antarmuka Halaman Web	47
Gambar 27. Diagram Alir Utama Perangkat Lunak.....	48
Gambar 28. Diagram Alir Tambah Data Wajah	50
Gambar 29. Diagram Alir Mengaktifkan Status Matakuliah	51
Gambar 30. Diagram Alir Verifikasi Kehadiran.....	52
Gambar 31. Diagram Alir Menonaktifkan Status Matakuliah	53
Gambar 32. Diagram Alir Pengenalan Wajah Menggunakan Pustaka Android Face Recognition Eigenface.....	55
Gambar 33. Implementasi Antarmuka Halaman Login	57
Gambar 34. Implementasi Antarmuka Halaman Menu Utama.....	58
Gambar 34. Implementasi Antarmuka Halaman Matakuliah	59
Gambar 36. Implementasi Antarmuka Halaman Update Status Matakuliah	60
Gambar 37. Implementasi Antarmuka Halaman Menu Mahasiswa	61
Gambar 38. Implementasi Antarmuka Halaman Tambah Data Wajah.....	62
Gambar 39. Implementasi Antarmuka Halaman Latih Data Wajah	63
Gambar 40. Implementasi Antarmuka Halaman Web	64
Gambar 41. Data Wajah 10 Mahasiswa dalam 1 Skenario Kelas.....	87

Gambar 42. Contoh Data Wajah Salah Satu Mahasiswa	88
---	----

DAFTAR LAMPIRAN

- Lampiran 1. Surat Persetujuan Sidang Tugas Akhir
- Lampiran 2. Kartu Konsultasi Tugas Akhir
- Lampiran 3. Daftar Hadir Peserta Pengujian
- Lampiran 4. Data Wajah Mahasiswa Peserta Pengujian
- Lampiran 5. Nilai Ujian Tugas Akhir

BAB I

PENDAHULUAN

A. Latar Belakang

Sistem kehadiran mahasiswa merupakan proses pencatatan kehadiran mahasiswa di kelas. Sistem kehadiran ini dapat menjadi bukti bahwa seorang mahasiswa telah berada di dalam kelas. Pencatatan kehadiran mahasiswa dapat dilakukan dengan berbagai cara. Pada Jurusan Teknik Informatika STMIK Palangkaraya, sistem kehadiran mahasiswa dilakukan secara manual yaitu melakukan tanda tangan pada kertas daftar kehadiran kelas. Hal ini menimbulkan celah kelemahan yaitu kemungkinan terjadinya kecurangan dalam pencatatan kehadiran. Kecurangan ini sering disebut titip absen. Titip absen merupakan kejadian di mana mahasiswa tercatat hadir pada kertas daftar kehadiran kelas tetapi kenyataannya mahasiswa tersebut tidak hadir di kelas. Dengan adanya celah kelemahan ini, mahasiswa dapat melakukan tanda tangan untuk mahasiswa lain. Oleh karena itu, diperlukan suatu sistem yang dapat digunakan untuk mengatasi kecurangan dalam pencatatan kehadiran mahasiswa di dalam kelas.

Perkembangan teknologi informasi di dunia semakin cepat, khususnya pada teknologi perangkat bergerak (*mobile*). Akses dan pengambilan informasi juga semakin mudah dan cepat diakses melalui penggunaan perangkat bergerak maupun web. Perkembangan ini juga mempengaruhi cara seseorang dalam melakukan proses verifikasi untuk masuk ke dalam sebuah sistem.

Verifikasi merupakan sebuah proses yang wajib ada dimana diperlukan sebuah kebenaran data/informasi misalnya verifikasi keaslian identitas. Proses verifikasi ini sudah banyak diterapkan seiring dengan perkembangan teknologi informasi oleh perusahaan, birokrasi, dan lain-lain. Proses verifikasi identitas yang sering digunakan adalah menggunakan alat deteksi sidik jari (*fingerprint*). Namun kelemahan dari penerapan ini adalah kurangnya fleksibilitas dari proses verifikasi ini. Fleksibilitas yang dimaksud adalah teknologi yang digunakan tidak bisa berpindah tempat dan diakses kapanpun. Oleh karena itu, tugas akhir ini akan mengimplementasikan sebuah aplikasi mobile sistem kehadiran berbasis Android. Implementasi pada aplikasi *mobile* diharapkan mampu menambah fleksibilitas dari sistem kehadiran yang sudah ada saat ini. Aplikasi kehadiran yang akan dibuat akan dibatasi ruang lingkup penggunaanya. Pengguna dari aplikasi ini adalah mahasiswa Teknik Informatika STMIK Palangkaraya.

Sistem kehadiran pada aplikasi ini mengimplementasikan metode pencocokan wajah (*face recognition*) ke dalam aplikasi *mobile*. Aplikasi ini diimplementasikan dengan menggunakan pustaka (*library*) Android *Face Recognition Eigenface* dimana didalamnya terdapat algoritma dan pustaka yang berhubungan dengan proses pengenalan wajah. Metode *eigenface* digunakan karna tingkat akurasiya cukup tinggi. Secara umum, Prosedur pencocokan wajah dibagi menjadi dua tahapan. Tahap pertama adalah deteksi wajah (*face detection*), yaitu tahap di mana aplikasi mencari wajah dalam gambar tangkapan kamera untuk memastikan adanya wajah seseorang yang tertangkap. Tahap

kedua adalah tahap identifikasi atau pengenalan wajah (*face recognition*). Pada tahap ini dilakukan perbandingan hasil ekstraksi fitur wajah yang terdeteksi dengan gambar wajah yang tersimpan dalam database. Perbandingan ini dilakukan supaya sistem mendapatkan hasil identifikasi identitas seseorang dari wajah yang terdeteksi. Pemberian identitas ini dilakukan dengan melakukan training terhadap aplikasi sebelumnya. Langkah training melibatkan pendeteksian wajah yang kemudian hasil deteksi ini diolah dengan menggunakan beberapa teknik pemrosesan gambar supaya dapat diterima dengan baik oleh aplikasi, setelah itu mengajarkan sistem siapa identitas wajah yang terdeteksi dengan menggunakan algoritma training yang juga telah disediakan oleh pustaka *face recognition* yang digunakan dalam tugas akhir ini.

Berdasarkan latar belakang di atas maka penulis ingin mengangkat judul sebagai tugas akhir yaitu “Rancang Bangun Sistem Pengenalan Wajah Untuk Kehadiran Mahasiswa Menggunakan Metode *Eigenface* Berbasis Android”.

B. Rumusan Masalah

Berdasarkan latar belakang diatas maka rumusan masalah yang diangkat dalam tugas akhir ini adalah bagaimana cara merancang dan membangun sistem pengenalan wajah untuk kehadiran mahasiswa menggunakan metode *eigenface* berbasis android?

C. Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Perangkat lunak dapat berjalan pada perangkat mobile dengan spesifikasi versi minimal Android 5.0 (Lollipop / API 21) berkapasitas RAM 2 GB dan resolusi kamera depan 2 MP.
2. Pengambilan dan pengolahan foto wajah dilakukan dalam aplikasi *client* atau smartphone yang dikhususkan untuk absensi.
3. Proses pengenalan foto wajah dilakukan secara realtime.
4. Aplikasi ini dibangun menggunakan Android Studio versi 3.3 dan pustaka (*library*) Android *Face Recognition Eigenface*.
5. Output dari sistem ini yaitu berupa rekapitulasi kehadiran mahasiswa yang dapat diolah oleh admin pada web yang sudah disediakan.
6. Dalam tugas akhir ini penulis hanya berfokus membahas bagaimana membangun aplikasi Sistem Pengenalan Wajah di Andoid.

D. Tujuan

Berdasarkan rumusan masalah diatas maka tujuan tugas akhir ini adalah merancang sistem pengenalan wajah untuk kehadiran mahasiswa menggunakan metode *eigenface* dengan menggunakan fasilitas dalam android.

E. Manfaat

Adapun manfaat dari tugas akhir ini, yaitu :

1. Bagi Pengguna

Memudahkan pengguna dalam melakukan absensi dan meminimalisir terjadinya kecurangan.

2. Bagi STMIK Palangkaraya

Memperbarui sistem kehadiran mahasiswa teknik informatika menjadi lebih modern dan terhindar dari kecurangan dengan menerapkan metode pengenalan wajah.

3. Bagi Penulis

Penulis dapat mengimplementasikan *library* pengenalan wajah kedalam sebuah sistem kehadiran mahasiswa berbasis android.

F. Metode Penelitian

1. Tahap Pengumpulan Data

Dalam melakukan penelitian untuk menyusun proposal tugas akhir ini penulis menggunakan beberapa tahapan atau metode penelitian :

a. Metode Kepustakaan

Metode pengumpulan data yang peneliti lakukan adalah metode kepustakaan, yaitu metode dalam pengumpulan data dengan cara membaca literatur dari buku-buku dengan masalah yang akan peneliti lakukan baik itu dari perpustakaan atau sumber lain. Seperti buku-buku,

jurnal ataupun artikel yang berkaitan dengan permasalahan yang dibahas peneliti.

b. Metode Observasi

Metode observasi, peneliti melakukan observasi langsung ke kelas untuk melakukan analisis kebutuhan yang akan digunakan untuk membangun sistem ini.

c. Metode Wawancara

Wawancara adalah suatu cara pengumpulan data yang digunakan untuk memperoleh informasi langsung dari sumbernya. Wawancara dilakukan dengan cara tanya jawab antara penulis dan narasumber.

2. Tahap Pengembangan Perangkat Lunak

Dalam penyusunan tugas akhir ini penulis menggunakan metode *Prototype* untuk model pengembangan perangkat lunaknya. Dibawah ini merupakan tahapan yang ada dalam metode *Prototype* :

a. Communication

Langkah ini merupakan analisis terhadap kebutuhan sistem atau menganalisa semua kebutuhan perangkat lunak. Tahapan ini digunakan untuk mengetahui ruang lingkup informasi, fungsi – fungsi yang dibutuhkan, kebutuhan kinerja yang ingin dihasilkan dan perancangan antarmuka pemakai sistem pada program yang dibuat.

b. Quick Plan

Perancangan desain perangkat lunak merupakan proses perancangan antarmuka dari hasil analisis kebutuhan yang telah selesai dikumpulkan secara lengkap. Desain dirancang dengan sedemikian rupa, sehingga membuat pengguna aplikasi tidak merasa sulit dalam penggunaannya.

c. Modeling Quick Design

Proses ini berfokus pada penyajian pembuatan *Prototype* perancangan sementara pada sistem pengenalan wajah untuk kehadiran mahasiswa.

d. Contraction of Prototype

Contraction of Prototype merupakan proses membuat kode. Jika telah selesai, maka pengujian harus langsung dilakukan untuk meminimalisir kesalahan-kesalahan dalam *coding*. Tahap ini apakah *Prototyping* yang sudah dibangun sudah selesai dengan yang diharapkan. Jika sudah sesuai, maka langkah selanjutnya akan diambil. Namun jika tidak, *prototyping* direvisi dengan mengulang langkah-langkah sebelumnya.

e. Deployment Delivery & Feedback

Tahap ini bisa dikatakan final dalam pembuatan sebuah software atau sistem. Proses ini meliputi beberapa kegiatan yaitu koreksi *error* pada *Prototype*, perbaikan terhadap unit sistem yang telah

diimplementasikan dan pengembangan pelayanan sistem, serta penambahan persyaratan-persyaratan baru. Pengoperasian program dilingkungannya dan melakukan pemeliharaan.

Dalam model *Prototype*, sistem yang sudah dibuat tidak bisa langsung dikatakan sempurna hal ini dikarenakan masih memiliki kemungkinan untuk perubahan dilakukan oleh pihak pengembang ataupun pengguna terhadap sistem yang sudah dirancang. Oleh karena itu jika terdapat perubahan terhadap sistem yang sudah dirancang maka proses perancangan harus di mulai dari awal yaitu dari proses *communication*. (Jumanto, 2018)

G. Sistematika Penulisan Laporan Tugas Akhir

Adapun sistematika penulisan tugas akhir ini terdiri dari beberapa bagian yang membahas dan menguraikan pokok permasalahan yang berbeda. Sebagai gambaran disini penulis menyertakan garis-garis besarnya yaitu :

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, dan manfaat dari pembuatan tugas akhir.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini berisi tentang analisis permasalahan, deskripsi umum sistem, spesifikasi kebutuhan perangkat lunak, lingkungan perancangan, perancangan arsitektur sistem, diagram kelas, dan struktur data

Bab IV Implementasi dan Pengujian

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Serta juga membahas hasil dari pengujian penilaian aplikasi oleh pengguna.

Bab V Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang telah dilakukan dan saran untuk pengembangan perangkat lunak selanjutnya.

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terkait

Pada tahun 2014 Siswo, Romi, dan Raya melakukan penelitian tentang penggunaan algoritma *eigenface* untuk pengenalan wajah. Penelitian tersebut menggunakan algoritma *eigenface*, dengan menggunakan citra yang dihasilkan melalui webcam dan menggunakan informasi mentah dari pixel citra yang kemudian direpresentasikan dalam metode *Principal Component Analysis* (PCA). Hasil dari pengujian sistem presensi tersebut didapat sensitivitas 100%, spesifisitas 55.55 %, dan akurasi 69.33 %.

Sedangkan pada tahun 2015 Rizki, Beni, dan Yulrio melakukan penelitian tentang implementasi pengenalan wajah dengan metode *eigenface* pada sistem absensi. Penelitian tersebut menggunakan algoritma Eigenface yang terdapat pada OpenCV. Penelitian tersebut menghasilkan sebuah sistem pengenalan wajah untuk absensi dengan tingkat kecocokan 88%. Faktor yang mempengaruhi tingkat kecocokan ialah pencahayaan, jarak, bentuk wajah, serta jumlah data yang tersedia.

Pada tahun yang sama Anita dan Afrilyan melakukan penelitian tentang implementasi pengenalan wajah menggunakan metode *eigenface* dengan bahasa pemrograman java. Pada penelitian tersebut metode untuk mengenali wajah yang digunakan yaitu eigenface, sedangkan untuk mendeteksi wajah digunakan metode

Viola Jones. Hasil pengujian untuk mendeteksi wajah dengan Viola Jones adalah akurasi ketika wajah dalam posisi frontal dengan webcam 100% dengan waktu deteksi kurang dari 1 detik. Pengujian sistem menggunakan metode *eigenface* diperoleh nilai akurasi sebesar 90%.

Selain itu pada tahun 2017 Surya, Gede, dan Suar melakukan penelitian tentang penerapan metode *eigenface* untuk identifikasi wajah berbasis android. Dalam penelitian tersebut metode *eigenface* digunakan untuk mengesktrak informasi yang relevan dari sebuah citra wajah, kemudian mengubahnya dalam satu set kode yang paling efisien. Tingkat keberhasilan dari uji coba identifikasi wajah sebesar 68% dan tingkat salah pengenalan sebesar 32%, dari total uji coba sebanyak 25 kali identifikasi. Dalam penelitian tersebut dikatakan bahwa faktor penting yang mempengaruhi tingkat keberhasilan identifikasi yaitu posisi wajah dan intensitas cahaya saat melakukan pendaftaran.

Pada tahun berikutnya Fahmi, Gede, dan Fitri melakukan penelitian tentang pengenalan wajah untuk sistem kehadiran menggunakan metode *eigenface* dan *euclidean distance*. Penelitian tersebut mengimplementasikan penggunaan dari metode *viola-jones*, *eigenface*, dan *euclidean distance*. Metode *viola-jones* digunakan untuk mendeteksi wajah. Metode *eigenface* digunakan untuk menghitung nilai *eigenvalue* pada setiap citra wajah. Sedangkan metode *euclidean distance* digunakan untuk klasifikasi. Tingkat akurasi pada hasil penelitian ini yaitu 84% dari total 30 kali percobaan.

Berdasarkan uraian diatas dapat disimpulkan bahwa metode *eigenface* sudah sering digunakan untuk pengenalan wajah. Tingkat akurasi metode tersebut memang berbeda-beda, namun masih bisa dikatakan cukup tinggi yaitu pada angka 80%. Selain itu 4 dari penelitian diatas dikatakan bahwa mereka membangun sistem pengenalan wajah menggunakan metode *eigenface* berbasis desktop. Sedangkan sisanya dibangun pada mobile android, namun hanya sebatas pada analisis pengenalan wajah saja tidak sampai untuk sebuah sistem kehadiran. Maka dari itu penulis ingin mengimplementasikan metode tersebut kedalam sebuah sistem pengenalan wajah untuk kehadiran berbasis android.

B. Algoritma Eigenface

Kata eigenface sebenarnya berasal dari bahasa Jerman “eigenwert” dimana “eigen” artinya karakteristik dan “wert” artinya nilai. *Eigenface* adalah salah satu algoritma pengenalan pola wajah yang berdasarkan pada *Principle Component Analysis* (PCA) yang dikembangkan di MIT.

Banyak penulis lebih menyukai istilah eigenimage. Teknik ini telah digunakan pada pengenalan tulisan tangan, pembacaan bibir, pengenalan suara dan pencitraan medis. Menurut layman (Al Fatta, 2009) *Eigenface* adalah sekumpulan *standardize face ingredient* yang diambil dari analisis statistik dari banyak gambar wajah.

Algoritma *eigenface* secara keseluruhan cukup sederhana. *Training image* direpresentasikan dalam sebuah *vector flat* (gabungan vektor) dan digabung

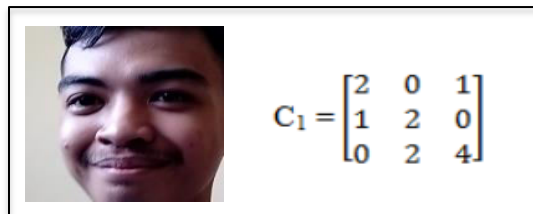
bersama-sama menjadi sebuah matriks tunggal. *Eigenface* dari masing-masing citra kemudian diekstraksi dan disimpan dalam file *temporary* atau *database*. *Test image* yang masuk didefinisikan juga nilai *eigenface*-nya dan dibandingkan dengan *eigenface* dari *image database* atau *file temporary*.

C. Perhitungan Eigenface

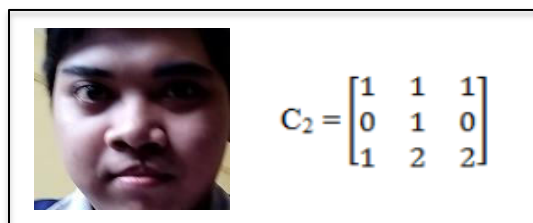
Algoritma pengenalan wajah dilakukan melalui beberapa tahapan, tahap pertama yaitu menyiapkan data dengan membuat suatu himpunan matriks yang ada di database, ambil nilai tengah atau mean, cari selisih antara training image dengan nilai tengah, hitung nilai matriks kovarian, menghitung eigenvalue dan eigenvector, tentukan nilai eigenface, dan terakhir adalah identifikasi. Untuk lebih jelasnya akan dijelaskan sebagai berikut.

1. Penyusunan Flatvector Matriks Citra

Langkah pertama adalah menyusun suatu himpunan S matriks yang terdiri dari seluruh *training image* ($\Gamma_1, \Gamma_2, \dots \Gamma_m$). Misalnya, *training image* terdapat dua data wajah seperti terlihat pada gambar 1 dan 2, yang masing-masing memiliki nilai matriks.



Gambar 1. Citra Wajah 1



Gambar 2. Citra Wajah 2

2. Hitung Nilai Tengah atau Mean (Ψ)

Dari himpunan matriks yang telah diperoleh, langkah selanjutnya adalah mencari nilai tengah atau mean. Jumlahkan nilai matrik wajah 1 dan wajah 2 kemudian bagi dengan jumlah data wajah yang ada di database.

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \dots\dots\dots(1)$$

Cari nilai tengah atau mean(Ψ).

$$\Psi = \frac{1}{2} \sum_{n=1}^2 \Gamma_n = \frac{1}{2} \left[\begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 2 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 2 \end{bmatrix} \right]$$

$$\Psi = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{bmatrix}$$

3. Hitung Selisih antara *Training Image* dengan Nilai Tengah

Dengan memakai nilai tengah citra di atas, langkah selanjutnya adalah mencari selisih (ϕ) antara *training image* (Γ) dengan nilai tengah (Ψ), dengan mengurangi *training image* (Γ) dengan nilai tengah (Ψ). Dengan rumus :

$$\phi_n = \Gamma_n - \Psi$$

$$\phi_1 = \Gamma_1 - \Psi = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 2 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\phi_2 = \Gamma_2 - \Psi = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 2 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

4. Hitung Nilai Matriks Kovarian

Nilai matriks kovarian (C) digunakan untuk menghitung *eigenvalue* (λ)

dan *eigenvector* (v).

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = AA^T \dots\dots\dots(3)$$

$$L = A^T A \quad L = \phi_m^T \phi_n$$

Hitung nilai matriks kovarian (C)

$L =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$L = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \rightarrow \text{matriks kovarian}$$

5. Hitung Nilai *Eigenvalue* dan *Eigenvector*

Langkah selanjutnya adalah menghitung nilai *eigenvalue* (λ) dan *eigenvector* (v) dari matriks kovarian (C).

$$C \times v_i = \lambda_i \times v_i \dots\dots\dots(4)$$

Cari nilai nilai *eigenvalue* (λ) dan *eigen-vector* (v).

$$\begin{aligned} L \times v &= \lambda \times v \\ L \times v &= \lambda I \times v \\ (L - \lambda I) &= 0 \text{ atau } (\lambda I - L) = 0 \end{aligned}$$

$$0 = \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

$$0 = \det \begin{bmatrix} \lambda - 2 & -1 & 0 \\ -1 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 3 \end{bmatrix}$$

Maka *eigenvalue* yang dihasilkan adalah $\lambda_1 = 3, \lambda_2 = 1, \lambda_3 = 3$

$$v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Eigenvector (v) dihasilkan dengan mensubstitusikan nilai *eigenvalue* (λ) kedalam persamaan $\lambda I - L v = 0$. *Eigenvector* dari masing-masing *eigenvalue* didapat berdasarkan masing-masing kolom *eigenvalue* dan kemudian dihipunkan menjadi satu matriks.

a. Untuk $\lambda_1 = 3$, maka :

$$\begin{bmatrix} \lambda - 2 & -1 & 0 \\ -1 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Dihasilkan *eigenvector* v_1 adalah $\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$,

b. Untuk $\lambda_2 = 1$, maka :

$$\begin{bmatrix} \lambda - 2 & -1 & 0 \\ -1 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Dihasilkan *eigenvector* v_2 adalah $\begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix}$,

c. Untuk $\lambda_3 = 3$, maka :

$$\begin{bmatrix} \lambda - 2 & -1 & 0 \\ -1 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Dihasilkan *eigenvector* v_3 adalah $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

Setelah didapat *eigenvector* v_1, v_2 , dan v_3 , maka *eigenvector* yang

dihasilkan dari matriks L adalah $\begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

d. Nilai *Eigenface*

Langkah selanjutnya, setelah *eigenvector* (v) diperoleh, maka nilai *eigenface* (μ) dapat dicari dengan:

$$\mu_l = \sum_{k=1}^M v_{lk} \phi_k \quad l = 1, \dots, M \quad \dots\dots\dots(5)$$

Cari nilai eigenface (μ) :

$$\mu_i = \sum_{k=1}^M v_{ik} \phi_k$$

$$\begin{aligned} \mu_1 &= v \times \phi_1 \\ &= \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

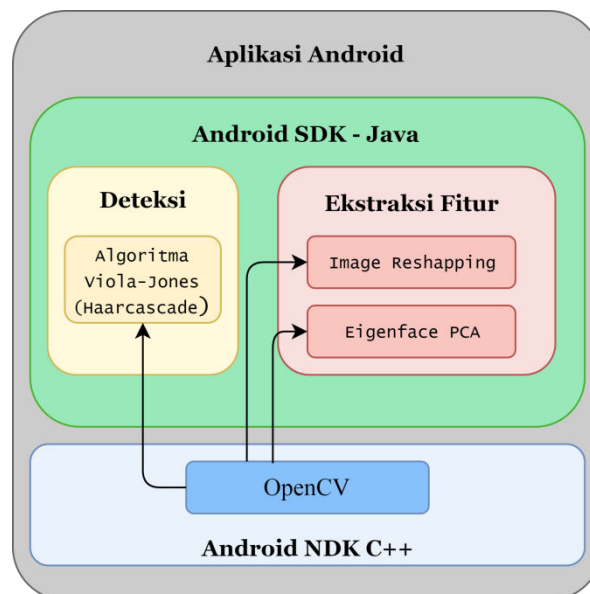
$$\mu_1 = \begin{bmatrix} 0 & -1 & 0 \\ -2 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} \mu_2 &= v \times \phi_2 \\ &= \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \end{aligned}$$

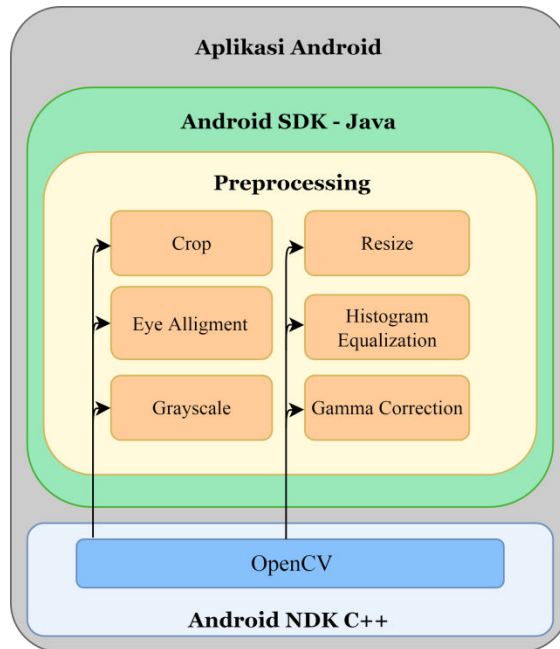
$$\mu_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

D. Penggunaan Pustaka Android Face Recognition Eigenface

Aplikasi Sistem Pengenalan Wajah untuk Kehadiran Mahasiswa ini menggunakan pustaka bantu dari *project open source* dari situs github.com bernama *Android Face Recognition Eigenface*. Hingga saat ini *library* ini terus dikembangkan dan bersifat *open source*. Pada Gambar 3 dan Gambar 4 dijabarkan arsitektur dari *library* ini.



Gambar 3. *Library Android Face Recognition Eigenface* (1)



Gambar 4. *Library Android Face Recognition Eigenface (2)*

Aplikasi yang dibangun memanfaatkan fitur dari *library* ini. Fitur yang digunakan antara lain fitur deteksi dengan menggunakan algoritma Viola-Jones, fitur *preprocessing* dengan menggunakan fitur *Crop*, *Eye Alignment*, *Grayscale*, dan *Resize*, ekstraksi fitur dengan menggunakan *Image Reshaping*, dan fitur klasifikasi dengan menggunakan algoritma *Eigenface PCA*.

1. Penggunaan *OpenCV Library*

OpenCV atau *Open Source Computer Vision Library* adalah sebuah *library* perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara *real-time*, *OpenCV* dibuat oleh Intel. Program ini bebas dan berada dalam naungan *open source* dari lisensi BSD. *Library* ini merupakan pustaka *cross-platform*. Program ini didedikasikan sebagian besar untuk pengolahan citra secara *realtime*.

Dalam pustaka *Android Face Recognition Eigenface*, library *OpenCV* diimplementasikan pada proses deteksi wajah, proses ekstraksi fitur, *preprocessing*, proses pengolahan foto wajah terdeteksi, serta fitur-fitur bantuan yang digunakan untuk pengolahan citra dinamis. Pada proses deteksi wajah digunakan algoritma Viola-Jones atau *Haarcascade*. Pada proses ekstraksi fitur akan digunakan algoritma *image reshaping*. Untuk tahap *preprocessing* akan digunakan beberapa tahapan antara lain *Standard Preprocessing*, *Brightness Correction* dan *Contrast Adjustment*. Algoritma dari tahap *Standard Preprocessing* yang digunakan yaitu *Grayscale*, *Crop*, dan *Eye Alignment*. Pada tahap *Brightness Correctness* dan *Contrast Adjustment* digunakan algoritma *Gamma Correction* dan *Histogram Equalization*.. Pada tahap *Standard Postprocessing* digunakan algoritma *Resize* untuk meningkatkan performa dari pengecilan gambar. Berikut contoh proses *Crop*, *Resize*, *Grayscale*, dan *Histogram Equalization*.

a. *Crop*

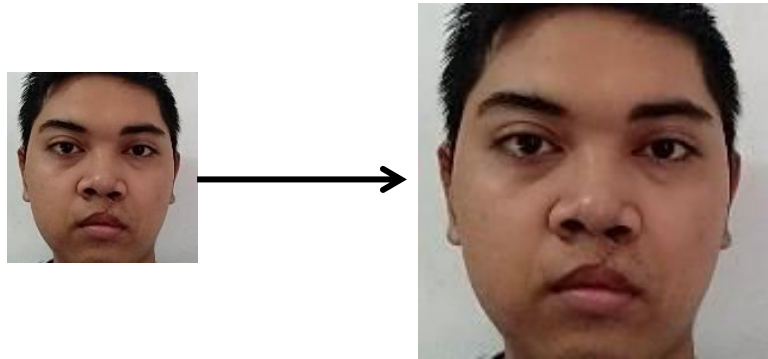
Pada proses ini citra dipotong sesuai dengan kebutuhan sistem. Citra awal dipotong menjadi ukuran 160 x 160 pixel.



Gambar 5. Proses *Cropping* Citra

b. *Resize*

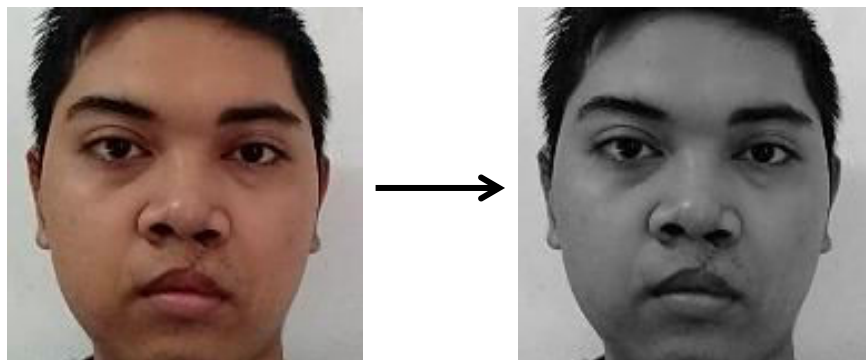
Pada proses ini citra yang sudah dipotong akan diperbesar kembali sesuai kebutuhan proses perhitungan nilai *eigenface*.



Gambar 6. Proses *Resize* Citra

c. *Grayscale*

Setelah proses *resize* maka citra lanjut ke proses *grayscale*. Proses ini berfungsi untuk mengubah warna citra menjadi abu-abu, hitam, dan putih.



Gambar 7. Proses *Grayscale* Citra

d. *Histogram Equalization*

Setelah proses grayscale maka citra lanjut ke proses *histogram equalization*. Proses ini berfungsi memperoleh penyebaran *histogram* yang merata sehingga setiap derajat keabuan memiliki jumlah piksel yang relatif sama. *Histogram equalization* juga diperlukan untuk mempermudah *library* Eigenface PCA menghitung nilai *eigenface*.



Gambar 8. Proses *Histogram Equalization* Citra

2. Penggunaan Algoritma Tahap Deteksi Wajah

Pada tahap pendeteksian wajah digunakan algoritma *Haarcascade* atau dikenal dengan Viola-Jones. Algoritma *Haarcascade* adalah metode pendeteksian objek yang efektif untuk digunakan. Algoritma ini diusulkan oleh Paul Viola dan Michael Jones pada tahun 2001 dalam penelitian mereka yang berjudul “Rapid Object Detection using a Boosted Cascade of Simple Features”. Algoritma ini merupakan sebuah pendekatan berbasis *machine learning* dimana fungsi *cascade* dilatih dari banyak gambar positif (gambar

wajah) dan gambar negatif (gambar tanpa wajah) yang kemudian hasilnya digunakan untuk mendeteksi objek pada gambar lainnya.

Dalam pustaka (*library*) *OpenCV*, algoritma ini sudah dilengkapi dengan banyak *pre-trained classifiers* untuk wajah, mata, senyuman, dan lain-lain. *Classifiers* ini disimpan dalam bentuk *file XML*. Untuk implementasi dari algoritma ini sudah diterapkan oleh pustaka (*library*) *Android Face Recognition Eigenface*, sehingga dalam pembuatan aplikasi tugas akhir ini hanya tinggal memakai saja.

3. Penggunaan Algoritma Tahap *Preprocessing*

Tahap *preprocessing* yang dilakukan dalam pustaka (*library*) *Android Face Recognition Eigenface* meliputi beberapa tahapan. Tahap-tahap yang dilakukan antara lain:

1. *Standard Preprocessing*
2. *Brightness Correction*
3. *Contours*
4. *Contrast Adjustment*
5. *Standard Postprocessing*

Pada tahap *Standard Preprocessing* dilakukan proses *Grayscale*, *Crop*, dan *Eye Alignment*. Proses *Grayscale* adalah proses mengubah warna gambar menjadi berwarna abu-abu. Proses *Crop* adalah proses untuk memotong gambar tepat pada area wajah, dan *Eye Alignment* adalah proses untuk mengubah posisi gambar supaya posisi mata sejajar secara horizontal.

Pada tahap *Brightness Correction* dilakukan proses koreksi kecerahan dari gambar yang terdeteksi oleh kamera. Tahap *Contours* diperlukan untuk menambah keakuratan untuk analisis bentuk dan pengenalan objek wajah. *Contours* merepresentasikan sebuah kurva yang menggabungkan semua titik-titik kontinyu yang memiliki warna atau intensitas yang sama. Pada tahap *Contrast Adjustment* dilakukan pengaturan kontras warna. Pada tahap *Standard Postprocessing* dilakukan proses *Resize*. Proses *Resize* bertujuan untuk memperkecil ukuran gambar wajah yang sudah diolah dari tahap 1 sampai 4. Pengubahan ukuran gambar wajah ini dimaksudkan untuk meningkatkan performa dari proses ekstraksi fitur.

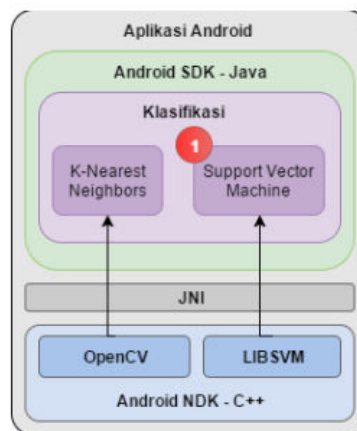
4. Penggunaan Algoritma Tahap Ekstraksi Fitur

Dalam *image processing*, ekstraksi fitur dimulai dari set awal data yang terukur. Ekstraksi fitur bertujuan membangun nilai-nilai turunan (fitur) hingga menjadi nilai yang informatif dan *nonredundant* (tidak berulang), memfasilitasi langkah pembelajaran dan generalisasi selanjutnya, dan dalam kasus tertentu dapat memberikan hasil interpretasi manusia yang lebih baik. Ekstraksi fitur biasanya berkaitan dengan dengan pengurangan dimensi dari data.

Pada implementasi tahap ekstraksi fitur dari aplikasi ini akan digunakan algoritma *Image Reshaping* yang telah tersedia di dalam pustaka (*library*) Android Face Recognition *Eigenface*. Algoritma ini akan mengubah dimensi

nilai matriks dari gambar wajah yang sudah diambil dan diolah pada tahap *preprocessing* menjadi matriks 1 dimensi saja karena gambar hanya berupa foto wajah. Hasil dari algoritma ini kemudian dipakai untuk tahap klasifikasi.

5. Penggunaan LIBSVM



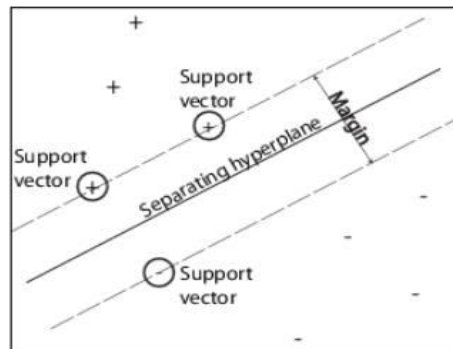
Gambar 9. LIBSVM dalam *Library Android Face Recognition*

LIBSVM merupakan sebuah *library* untuk *Support Vector Machines* (SVM). Pustaka ini telah dikembangkan sejak tahun 2000. Tujuan dari pembuatan *library* ini adalah untuk mempermudah pengguna untuk menerapkan SVM untuk aplikasi yang dibuat oleh pengguna tersebut. *Library* ini sudah banyak digunakan untuk berbagai macam *machine learning* dan studi kasus lainnya.

Dalam pustaka Android Face Recognition *Eigenface*, pustaka (*library*) LIBSVM digunakan pada saat proses klasifikasi data foto yang terkumpul. Penggunaan pustaka ini ditunjukkan pada nomor 1 Gambar 9. Tipe kernel yang dipakai menggunakan pengaturan awal (*default*) dari pustaka Android

Face Recognition *Eigenface*. Tipe *kernel* yang dipakai adalah *linear kernel*. Tipe ini digunakan karena gambar memiliki dimensi yang cukup untuk memberikan akurasi yang baik.

6. Penggunaan Algoritma Tahap Klasifikasi



Gambar 10. Arsitektur LIBSVM
(Sumber : www.github.com)

Support Vector Machine adalah model pembelajaran mesin yang umumnya digunakan untuk mengatasi masalah klasifikasi dan regresi. Klasifikasi dilakukan dengan cara menemukan *hyperplane* yang optimal (garis optimal untuk data 2 dimensi) yang memisahkan kelas data dengan margin maksimum seperti pada Gambar 10. Ketika nilai margin lebih besar umumnya hasil klasifikasi lebih baik generalisasi. Data pelatihan (*training data*) dengan menggunakan algoritma SVM menghasilkan vector pendukung (*support vector*). Vektor pendukung adalah data titik yang terletak paling dekat dengan *hyperplane* tersebut, dan fungsi keputusan SVM sepenuhnya bergantung pada nilai dari vector pendukung ini.

Pada tahap klasifikasi digunakan algoritma Support Vector Machine yang telah disiapkan dalam pustaka (*library*) Android *Face Recognition Eigenface*. Algoritma SVM yang digunakan menggunakan *kernel linear* karena fitur-fitur yang dihasilkan dari gambar foto wajah sederhana.

E. API

API atau *Application Programming Interface* adalah sekumpulan perintah, fungsi, serta *protocol* yang dapat digunakan oleh *programmer* saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi. API juga dapat menjelaskan cara sebuah tugas tertentu dilakukan. Sebuah API juga dapat digunakan untuk menspesifikasikan cara komponen aplikasi saling berinteraksi. Dengan bahasa yang lebih sederhana, API adalah fungsi-fungsi pemrograman yang disediakan oleh aplikasi atau layanan agar layanan tersebut bisa diintegrasikan dengan aplikasi yang dibuat. Di dalam pemrograman android, biasanya digunakan untuk mempermudah pertukaran data dari atau ke dalam server.

F. Restful API

RESTful API / REST API merupakan implementasi dari API (*Application Programming Interface*). REST (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi. Dimana tujuannya adalah untuk menjadikan sistem yang memiliki performa

yang baik, cepat dan mudah untuk di kembangkan (*scale*) terutama dalam pertukaran dan komunikasi data.

RESTful API memiliki 4 komponen penting di dalamnya diantaranya adalah

- URL Design

RESTful API diakses menggunakan protokol HTTP. Penamaan dan struktur URL yang konsisten akan menghasilkan API yang baik dan mudah untuk dimengerti developer. URL API biasa disebut endpoint dalam pemanggilannya.

- HTTP Verbs

Setiap request yang dilakukan terdapat metode yang dipakai agar server mengerti apa yang sedang di request client, diantaranya yang umum dipakai adalah :

- a. GET

GET adalah metode HTTP Request yang paling simpel, metode ini digunakan untuk membaca atau mendapatkan data dari sumber.

- b. POST

POST adalah metode HTTP Request yang digunakan untuk membuat data baru dengan menyisipkan data dalam body saat request dilakukan.

- c. PUT

PUT adalah metode HTTP Request yang biasanya digunakan untuk melakukan update data resource.

d. DELETE

DELETE adalah metode HTTP Request yang digunakan untuk menghapus suatu data pada resource.

- HTTP Response Code

HTTP response code adalah kode standarisasi dalam menginformasikan hasil request kepada client.

- Format Response

Setiap request yang dilakukan client akan menerima data response dari server, response tersebut biasanya berupa data XML ataupun JSON. Setelah mendapatkan data response tersebut barulah client bisa menggunakannya dengan cara memarsing data tersebut dan diolah sesuai kebutuhan.

G. JSON

JSON atau *JavaScript Object Notation* adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format JSON dibuat berdasarkan bagian dari bahasa pemrograman *JavaScript*.

H. *Unified Modeling Language*

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan

pemrograman berorientasi objek. Pada penelitian UML yang akan digunakan yaitu *Use Case*, *Class Diagram* dan *Activity Diagram*.

I. *Object Oriented Programming*

OOP (*Object Oriented Programming*) atau dalam bahasa indonesia dikenal dengan pemrograman berorientasikan objek (PBO) merupakan sebuah paradigma atau teknik pemrograman yang berorientasikan Objek. Pada OOP, Fungsi dan variabel dibungkus dalam sebuah objek atau class yang dapat saling berinteraksi, sehingga membentuk sebuah program.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

A. Analisis

Dalam melakukan analisis, penulis menggunakan pendekatan OOP (*Object Oriented Programming*) untuk melakukan perancangan aplikasi. Berikut tahapan yang dilakukan penulis dalam melakukan analisis.

1. Analisis Permasalahan

Pada Jurusan Teknik Informatika STMIK Palangkaraya, sistem kehadiran mahasiswa dilakukan secara manual yaitu melakukan tanda tangan pada kertas daftar kehadiran kelas. Hal ini menimbulkan celah kelemahan yaitu kemungkinan terjadinya kecurangan dalam pencatatan kehadiran. Kecurangan ini sering disebut titip absen. Titip absen merupakan kejadian di mana mahasiswa tercatat hadir pada kertas daftar kehadiran kelas tetapi kenyataannya mahasiswa tersebut tidak hadir di kelas. Dengan adanya celah kelemahan ini, mahasiswa dapat melakukan tanda tangan untuk mahasiswa lain. Oleh karena itu, pada tugas akhir ini bertujuan membangun sebuah sistem pengenalan wajah untuk kehadiran berbasis android dengan harapan untuk meminimalisir adanya kecurangan pada saat absensi.

2. Analisis Kebutuhan

Tahap analisis kebutuhan dibagi menjadi dua bagian, yaitu kebutuhan fungsional dan non-fungsional.

a. Kebutuhan Fungsional

Kebutuhan fungsional yang harus dipenuhi agar sistem dapat berjalan dengan baik, antara lain :

1. Mendaftarkan data wajah untuk proses pengenalan wajah.
2. Mengaktifkan status matakuliah untuk proses absensi.
3. Melakukan proses absensi melalui pengenalan wajah.

b. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang harus dipenuhi oleh sistem adalah kebutuhan performa, perangkat lunak akan berjalan dengan performa terbaik apabila dijalankan diatas atau sama dengan spesifikasi minimal yang sudah dijelaskan pada batasan masalah.

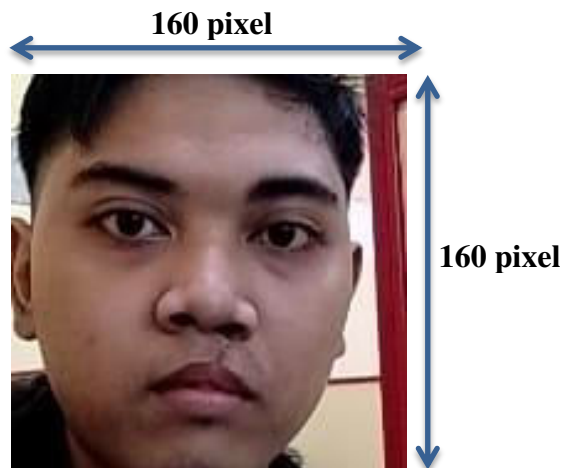
B. Perancangan

Perancangan dalam subbab ini membahas perancangan dari aplikasi tugas akhir bernama Sikemas Pengenalan Wajah. Subbab ini terdiri dari perancangan data, perancangan arsitektuk perangkat lunak, perancangan diagram aktivitas, perancangan diagram kelas, perancangan antarmuka pengguna dan perancangan proses.

1. Perancangan Data

Perancangan data dari aplikasi Sikemas Pengenalan Wajah dibuat untuk memberi penjelasan berkaitan dengan data yang dipakai dan dihasilkan oleh aplikasi ini. Perancangan data yang dilakukan meliputi penentuan jenis file yang digunakan, ekstensi file yang bisa dipakai, ukuran file, serta lokasi

penyimpanan dari file yang dipakai dan dihasilkan oleh aplikasi ini. Jenis data yang digunakan adalah data gambar atau *image*. Jenis data ini digunakan untuk menyimpan hasil dari pengambilan data wajah pada kasus pengguna mendaftarkan wajah. Jenis file ini juga digunakan untuk penyimpanan hasil unduhan data wajah. Semua data gambar atau *image* ini disimpan dengan ekstensi PNG. Ukuran dari data gambar atau *image* ini ditentukan untuk ukuran dimensi gambar 160 x 160 pixel. Gambar 11 merupakan contoh ukuran data *image* yang sudah dicrop pada bagian wajah dan memiliki ukuran dimensi 160 x 160 pixel sesuai dengan perancangan ukuran dimensi data *image*.

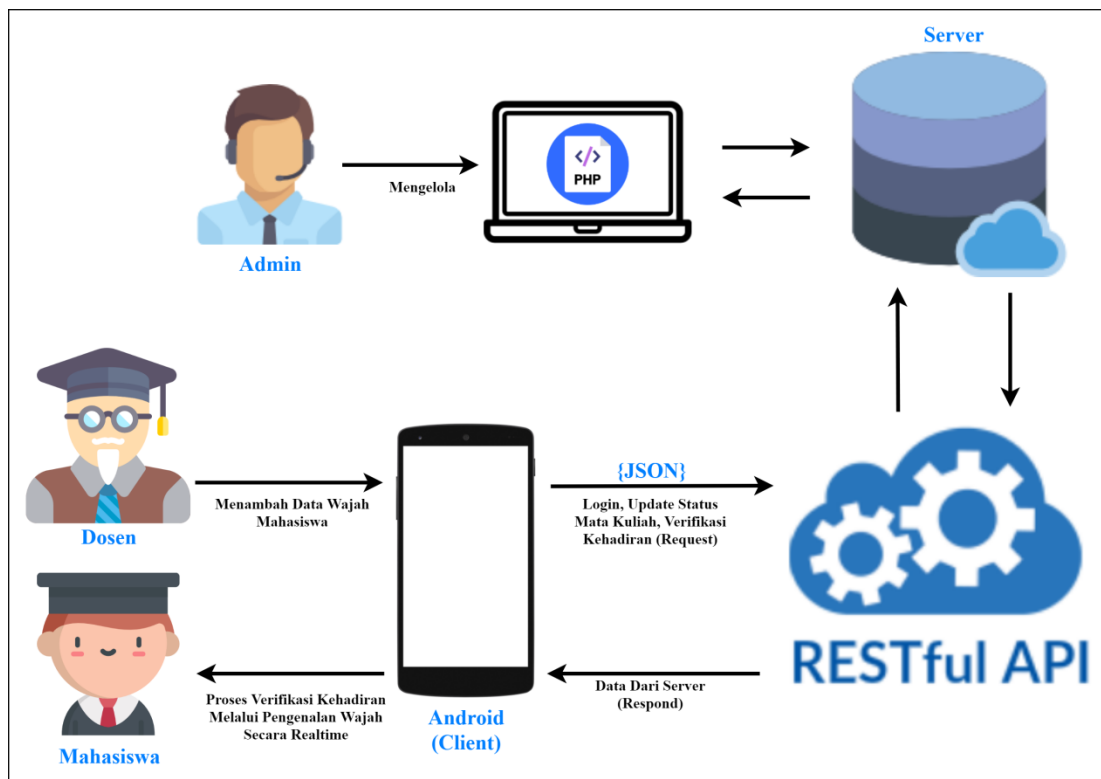


Gambar 11. Contoh Gambar Data Wajah Terambil dan Tersimpan

Penyimpanan untuk semua data dari aplikasi Sikemas Pengenalan Wajah akan disimpan di penyimpanan internal *smartphone*. Lokasi penyimpanan data Sikemas berada di *folder* facerecognition. Data *image* yang diambil pada saat pendaftaran wajah akan disimpan pada *subfolder* facerecognition bernama training. Data *image* hasil dari pelatihan (*training*)

data wajah dan file teks hasil dari klasifikasi algoritma SVM akan disimpan di *subfolder* dari facerecognition bernama data.

2. Perancangan Arsitektur Sistem



Gambar 12. Arsitektur Sistem Pengenalan Wajah Untuk Kehadiran Mahasiswa

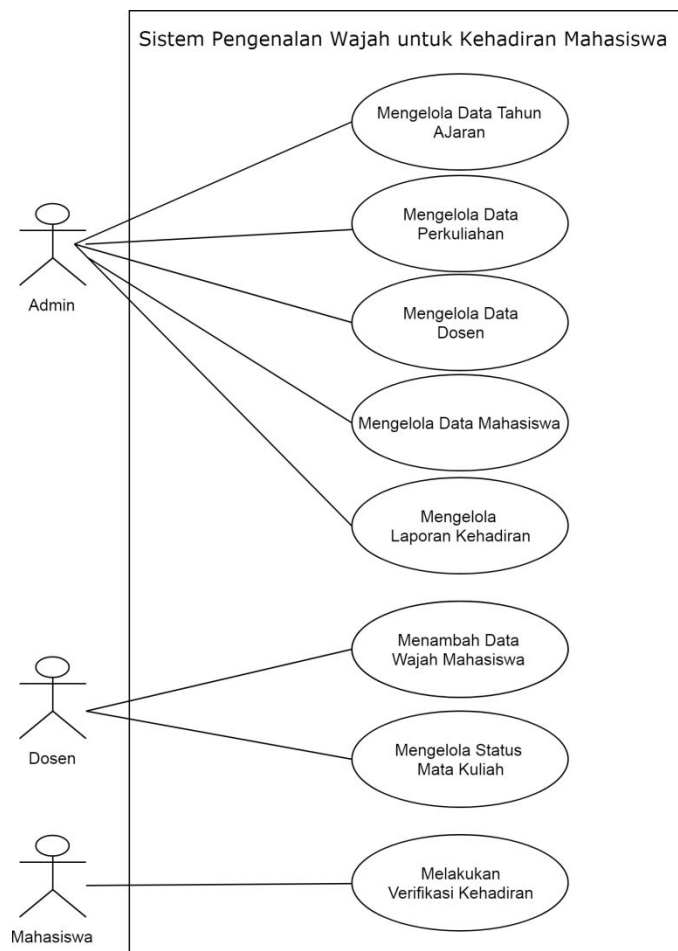
Gambar 12 merupakan arsitektur sistem yang akan dibangun. Pada gambar terdapat beberapa aktor dan fungsinya yang terhubung dengan sistem. Berikut penjelasan mengenai semua aktor dan fungsinya.

- a) Admin mengelola database melalui web berbasis php. Hal-hal yang dikelola meliputi kelola data dosen, kelola data mata kuliah, dan kelola laporan kehadiran.

- b) Dosen melakukan login, menambah data wajah mahasiswa, mengelola status mata kuliah, dan membuka fitur verifikasi kehadiran.
- c) Mahasiswa melakukan verifikasi kehadiran.

3. Perancangan *Use Case*

Use Case dirancang berdasarkan hasil dari perancangan arsitektur sistem. *Use case* dapat dilihat pada Gambar 13.

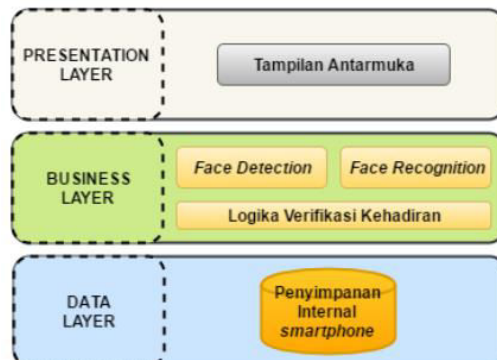


Gambar 13. *Use Case* Sistem Pengenalan Wajah Untuk Kehadiran Mahasiswa

4. Perancangan Arsitektur Perangkat Lunak

Arsitektur sistem untuk Tugas Akhir ini menggunakan arsitektur 3 lapis yang terdiri dari *Data Layer*, *Business Layer*, dan *Presentation Layer*. *Data Layer* berfungsi untuk menangani basis data aplikasi serta penyimpanan internal aplikasi, *Business Layer* berfungsi untuk menangani logika aplikasi serta proses-proses verifikasi kehadiran mahasiswa. Sedangkan *Presentation Layer* berfungsi menangani tampilan antarmuka pengguna. *Data Layer* pada aplikasi ini ditangani oleh penyimpanan internal smartphone yang diatur oleh *File system* dari sistem operasi Android. Gambar arsitektur aplikasi sistem dapat dilihat pada Gambar 14.

Pada lapisan *Business Layer*, digunakan RESTful API *web service* sebagai jembatan komunikasi atau pertukaran data antara sistem client dengan sistem server. Lapisan *Business Layer* melakukan pengolahan API ini supaya dapat dipakai oleh aplikasi. API yang diolah oleh aplikasi akan menggunakan tipe JSON. Pengolahan API ini dapat dilihat pada hasil semua implementasi kasus penggunaan.



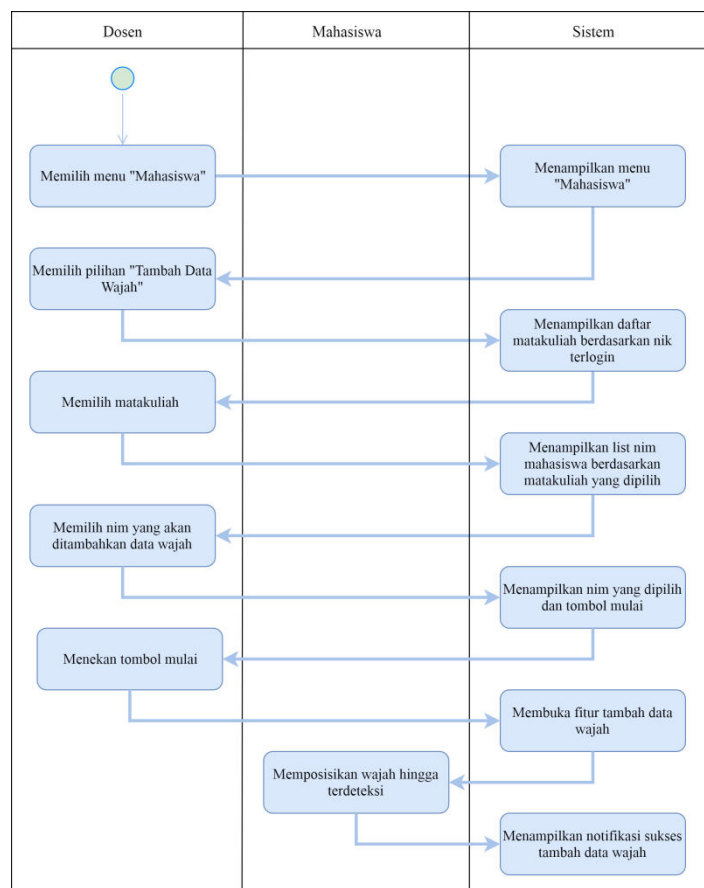
Gambar 14. Diagram Arsitektur Sistem

5. Perancangan Diagram Aktivitas

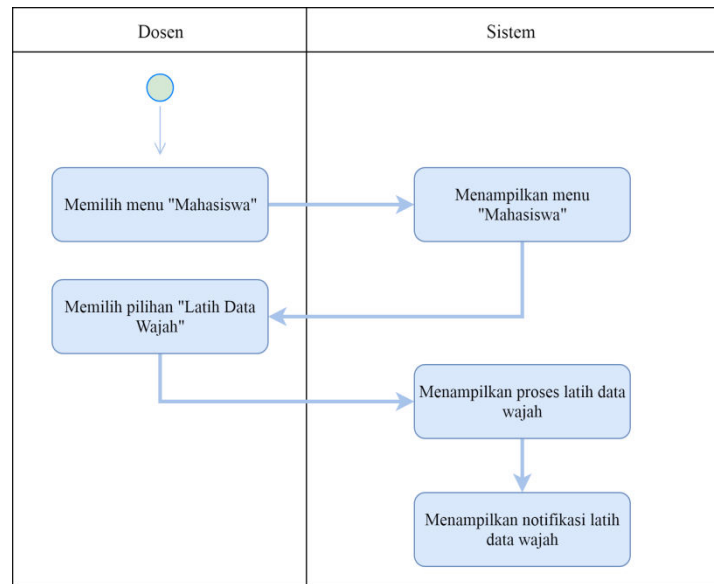
Diagram aktivitas dibagi menjadi 4 bagian, yaitu :

- Menambah data wajah.
- Melatih data wajah.
- Mengaktifkan status mata kuliah dan melakukan verifikasi kehadiran.
- Menonaktifkan status mata kuliah.

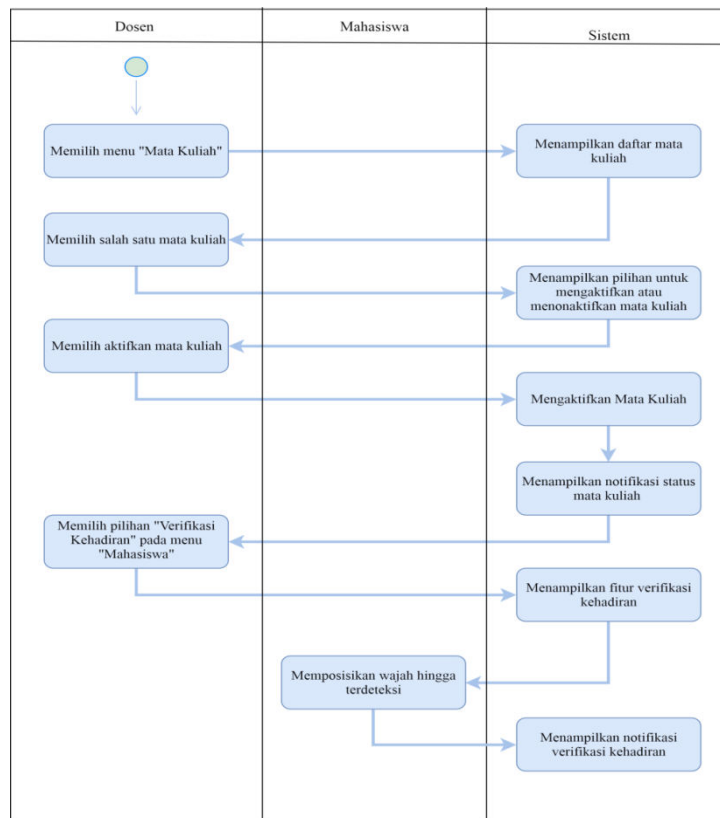
Masing-masing diagram dapat dilihat pada Gambar 15-18.



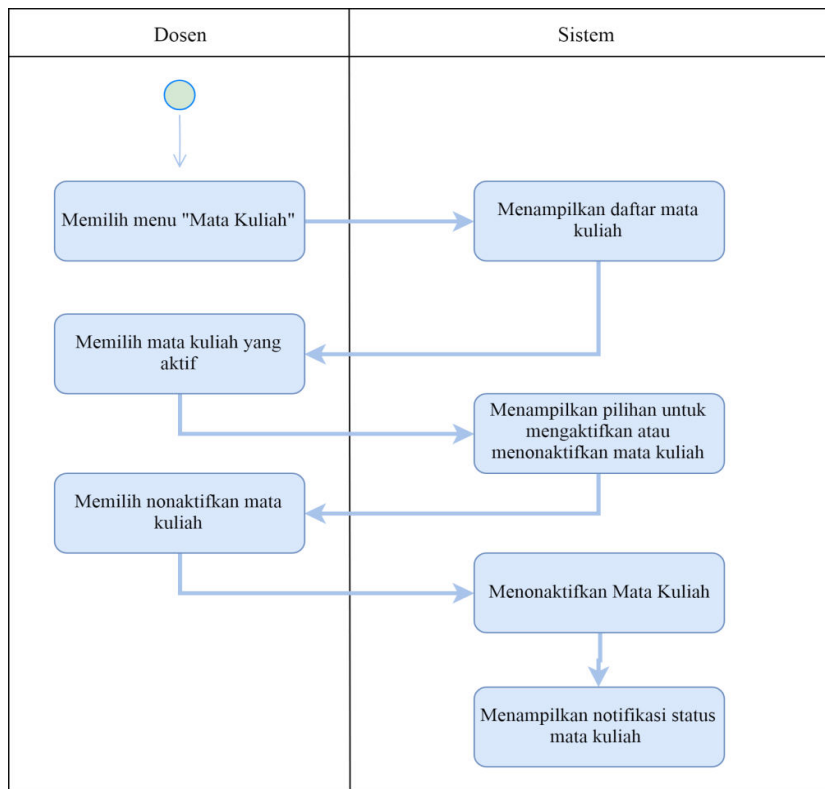
Gambar 15. Diagram Aktivitas Tambah Data Wajah



Gambar 16. Diagram Aktivitas Latih Data Wajah



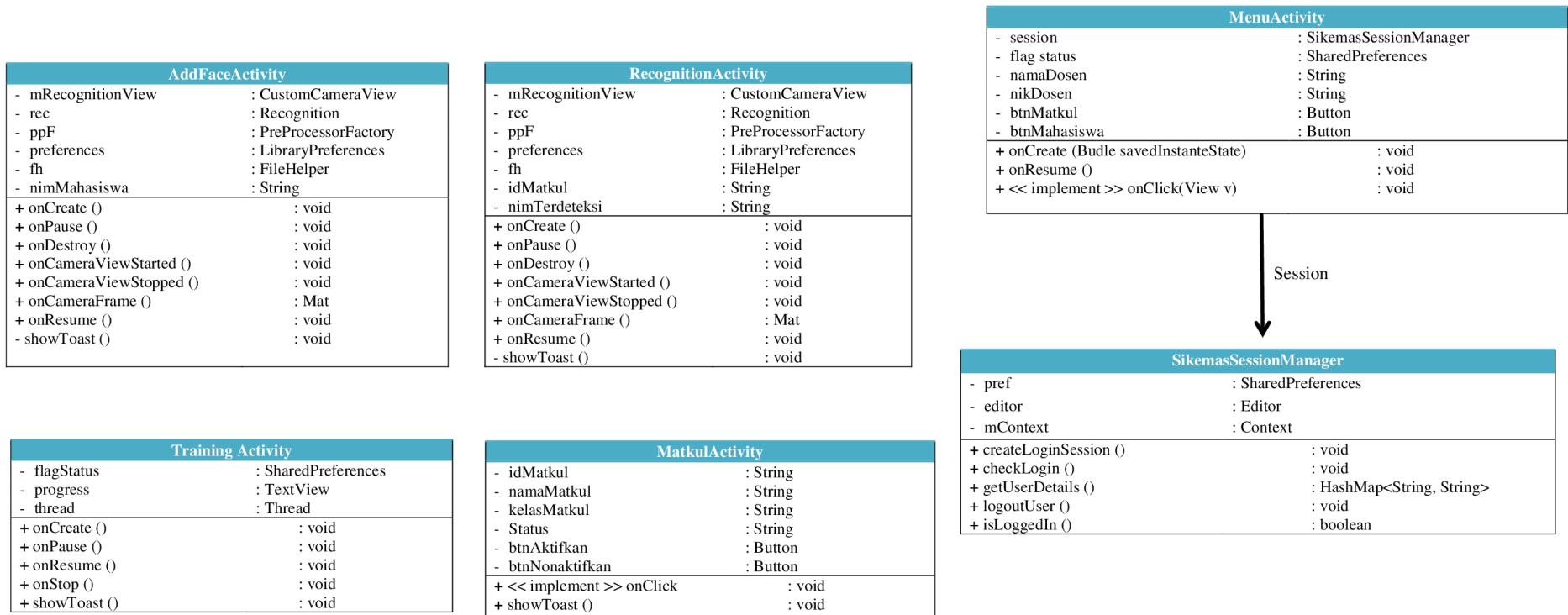
Gambar 17. Diagram Aktivitas Aktifkan Mata Kuliah dan Verifikasi Kehadiran



Gambar 18. Diagram Aktivitas Menonaktifkan Status Mata Kuliah

6. Perancangan Diagram Kelas

Rancangan diagram kelas dapat dilihat pada Gambar 19. Pada diagram kelas ini akan ditunjukkan kelas-kelas beserta beberapa atribut dan fungsi penting yang akan digunakan dalam aplikasi Sikemas Pengenalan Wajah



Gambar 19. Diagram Kelas

7. Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna merupakan hal yang penting dalam melakukan perancangan perangkat lunak. Antarmuka harus memiliki kemudahan dan tampilan yang menarik bagi penggunanya. Berikut rancangan antarmuka pengguna Sikemas Pengenalan Wajah.

a. Rancangan Antarmuka Halaman Login

Halaman ini akan ditampilkan ketika dosen pertama kali membuka aplikasi Sikemas Pengenalan Wajah. Rancangan dapat dilihat pada Gambar 20. Atribut antarmuka dapat dilihat pada Tabel 1.

Tabel 1. Atribut Antarmuka Halaman Login

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/Keluaran
1	<i>name_apk</i>	<i>TextView</i>	Menampilkan nama aplikasi	<i>String</i>
2	<i>editTextNim</i>	<i>EditText</i>	Menginput nik dosen yang akan login	<i>String</i>
3	<i>editTextPassword</i>	<i>EditText</i>	Menginput password dosen yang akan login	<i>String</i>
4	<i>btn_login</i>	<i>Button</i>	Tombol aksi login menuju halaman utama	<i>ButtonClick</i>



Gambar 20. Rancangan Antarmuka Halaman Login

b. Rancangan Antarmuka Halaman Menu Utama

Halaman ini akan ditampilkan ketika dosen sudah login ke aplikasi Sikemas Pengenalan Wajah. Rancangan dapat dilihat pada Gambar 21. Atribut antarmuka dapat dilihat pada Tabel 2.

Tabel 2. Atribut Antarmuka Halaman Utama

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/Keluaran
1	<i>tvNama</i>	<i>TextView</i>	Menampilkan nama dosen terlogin	<i>String</i>
2	<i>tvNik</i>	<i>TextView</i>	Menampilkan nik/nip dosen terlogin	<i>String</i>
3	<i>btn_matkul</i>	<i>Button</i>	Tombol aksi menuju halaman mata kuliah	<i>ButtonClick</i>
4	<i>btn_mahasiswa</i>	<i>Button</i>	Tombol aksi menuju halaman mahasiswa	<i>ButtonClick</i>



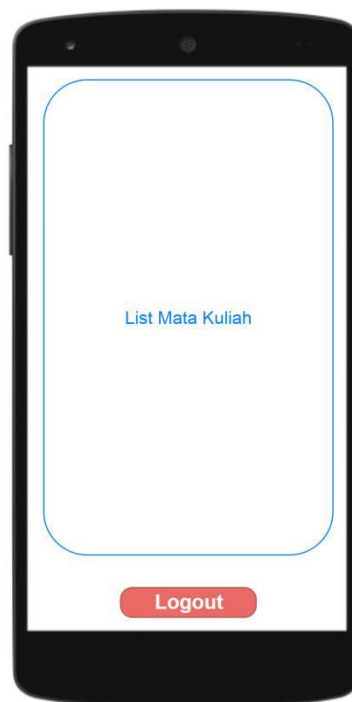
Gambar 21. Rancangan Antarmuka Halaman Menu Utama

c. Rancangan Antarmuka Halaman Matakuliah

Halaman ini ditampilkan ketika dosen mengklik menu mata kuliah dari menu utama. Halaman ini berisi daftar mata kuliah berdasarkan nip/nik dosen terlogin. Rancangan dapat dilihat pada Gambar 22. Atribut antarmuka dapat dilihat pada Tabel 3.

Tabel 3. Atribut Antarmuka Halaman Matakuliah

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>listMatkul</i>	<i>ListView</i>	Menampilkan daftar mata kuliah berdasarkan nip/nik dosen terlogin	<i>String</i>
2	<i>btn_logout</i>	<i>Button</i>	Tombol aksi untuk keluar dari sistem	<i>ButtonClick</i>



Gambar 22. Rancangan Antarmuka Halaman Matakuliah

d. Rancangan Antarmuka Halaman Update Status Mata Kuliah

Halaman ini ditampilkan ketika dosen mengklik salah satu mata kuliah dari daftar mata kuliah. Halaman ini fitur untuk mengaktifkan atau menonaktifkan status mata kuliah. Rancangan dapat dilihat pada Gambar 23. Atribut antarmuka dapat dilihat pada Tabel 4.

Tabel 4. Atribut Antarmuka Update Status Matakuliah

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>namaMatkul</i>	<i>TextView</i>	Menampilkan nama mata kuliah yang sedang diklik	<i>String</i>
2	<i>kelasMatkul</i>	<i>TextView</i>	Menampilkan kelas mata kuliah yang sedang diklik	<i>String</i>
3	<i>btn_aktif</i>	<i>Button</i>	Tombol aksi untuk mengaktifkan status mata kuliah	<i>ButtonClick</i>
4	<i>btn_nonaktif</i>	<i>Button</i>	Tombol aksi untuk menonaktifkan status mata kuliah	<i>ButtonClick</i>



Gambar 23. Rancangan Antarmuka Halaman Update Status Matakuliah

e. Rancangan Antarmuka Halaman Mahasiswa

Halaman ini ditampilkan ketika dosen mengklik menu mahasiswa dari menu utama. Rancangan dapat dilihat pada Gambar 24. Atribut antarmuka dapat dilihat pada Tabel 5.

Tabel 5. Atribut Antarmuka Menu Mahasiswa

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/Keluaran
1	<i>btn_tambah_data_wajah</i>	<i>Button</i>	Tombol aksi menuju halaman tambah data wajah	<i>ButtonClick</i>
2	<i>btn_latih_data_wajah</i>	<i>Button</i>	Tombol aksi menuju halaman latih data wajah	<i>ButtonClick</i>
3	<i>btn_verifikas_kehadiran</i>	<i>Button</i>	Tombol aksi menuju halaman verifikasi kehadiran	<i>ButtonClick</i>



Gambar 24. Rancangan Antarmuka Halaman Mahasiswa

f. Rancangan Antarmuka Halaman Tambah Data Wajah

Halaman ini ditampilkan ketika dosen mengklik menu tambah data wajah dari menu mahasiswa. Halaman ini digunakan untuk menambah data wajah mahasiswa. Rancangan dapat dilihat pada Gambar 25. Atribut antarmuka dapat dilihat pada Tabel 6.

Tabel 6. Atribut Antarmuka Tambah Data Wajah


No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>editTextNim</i>	<i>EditText</i>	Form untuk mengisi nim mahasiswa yang akan ditambahkan data wajah	<i>String</i>
2	<i>btn_start</i>	<i>Button</i>	Tombol aksi memulai fitur tambah data wajah	<i>ButtonClick</i>



Gambar 25. Rancangan Antarmuka Halaman Tambah Data Wajah

g. Rancangan Antarmuka Halaman Web

Halaman antarmuka web digunakan untuk memanajemen segala aspek yang berkaitan dengan sistem kehadiran mahasiswa. Halaman antarmuka web dikelola oleh admin. Rancangan halaman antarmuka web dapat dilihat pada gambar 26.

Manajemen Sistem Kehadiran Mahasiswa Pengenalan Wajah	
Tahun Ajaran <	
Dosen <	
Matakuliah <	
Perkuliahan <	
Peserta Kuliah <	
Mahasiswa <	
Laporan Kehadiran <	
Keluar	

Gambar 26. Rancangan Halaman Antarmuka Web

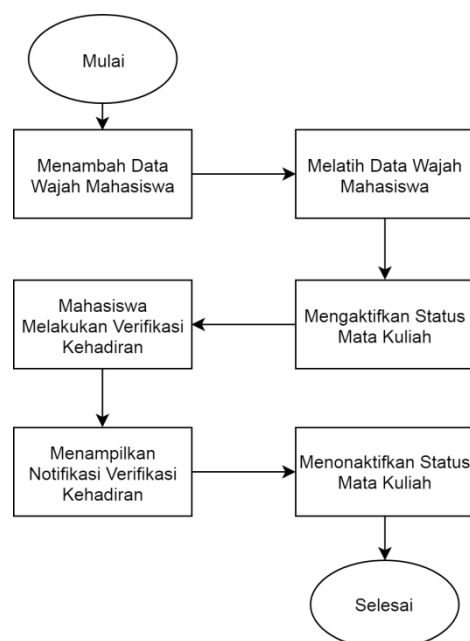
8. Perancangan Proses

Pada subbab ini akan dijelaskan tahap proses aplikasi Sikemas Pengenalan Wajah melalui diagram alir. Berikut adalah beberapa tahap proses yang akan dijelaskan.

a. Rancangan Alir Proses Utama Perangkat Lunak

Pada Gambar 27 dijelaskan alir proses perangkat lunak secara keseluruhan. Tahapan proses yang dilakukan antara lain :

- 1) Pada awal alir proses, dosen menambahkan data wajah mahasiswa ketika pertama kali masuk ke suatu kelas perkuliahan.
- 2) Dosen melatih data wajah mahasiswa.
- 3) Dosen mengaktifkan status mata kuliah.
- 4) Mahasiswa melakukan verifikasi kehadiran melalui smartphone dosen.
- 5) Dosen menonaktifkan status mata kuliah.



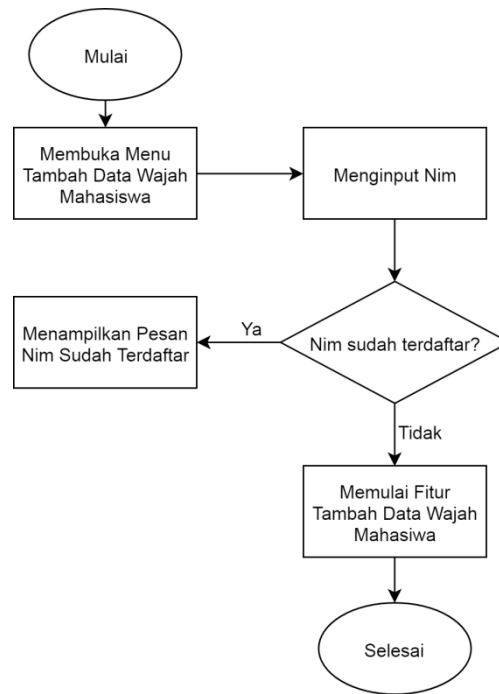
Gambar 27. Diagram Alir Utama Perangkat Lunak

Pada diagram alir terdapat 4 proses yang dapat didekomposisi yaitu proses menambah data wajah mahasiswa, mengaktifkan status mata kuliah, verifikasi kehadiran dan menonaktifkan status mata kuliah.

b. Rancangan Alir Proses Tambah Data Wajah Mahasiswa

Pada Gambar 28 dijelaskan alir proses tambah data wajah mahasiswa. Tahapan proses yang dilakukan antara lain :

- 1) Dosen memilih menu tambah data wajah dari menu mahasiswa.
- 2) Mahasiswa menambah data wajah dengan cara menginputkan nim pada form yang disediakan.
- 3) Mahasiswa menekan tombol mulai untuk memulai fitur penambahan data wajah.
- 4) Sistem akan mengecek apakah nim mahasiswa yang terinput sudah terdaftar atau belum, jika sudah maka akan muncul sebuah pesan bahwa nim sudah terdaftar.
- 5) Jika belum maka aplikasi akan memulai fitur menambah data wajah.
- 6) Mahasiswa harus memposisikan wajah ke kamera depan supaya sistem dapat mengambil data wajah.

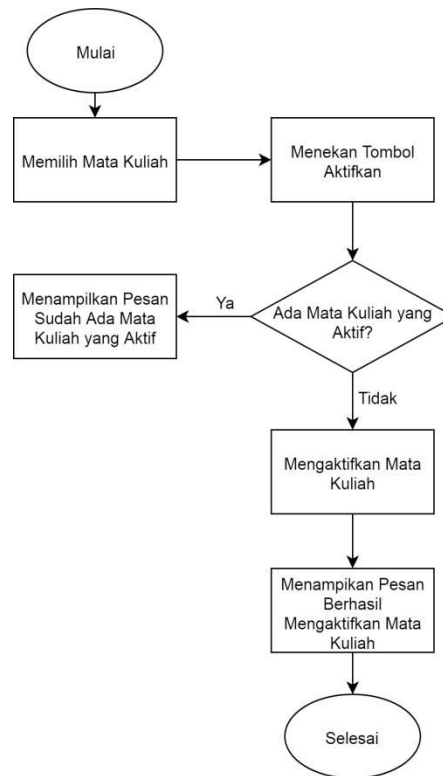


Gambar 28. Diagram Alir Tambah Data Wajah

c. Rancangan Alir Proses Mengaktifkan Status Mata Kuliah

Pada Gambar 29 dijelaskan alir proses mengaktifkan status mata kuliah. Tahapan proses yang dilakukan antara lain :

1. Dosen memilih mata kuliah yang ingin diaktifkan.
2. Dosen menekan tombol aktifkan.
3. Sistem akan mengecek apakah sudah ada mata kuliah yang aktif, jika sudah maka akan menampilkan pesan bahwa sudah ada mata kuliah yang aktif.
4. Jika belum maka sistem akan mengaktifkan status mata kuliah.
5. Sistem menampilkan pesan sukses mengaktifkan status mata kuliah.



Gambar 29. Diagram Alir Mengaktifkan Status Matakuliah

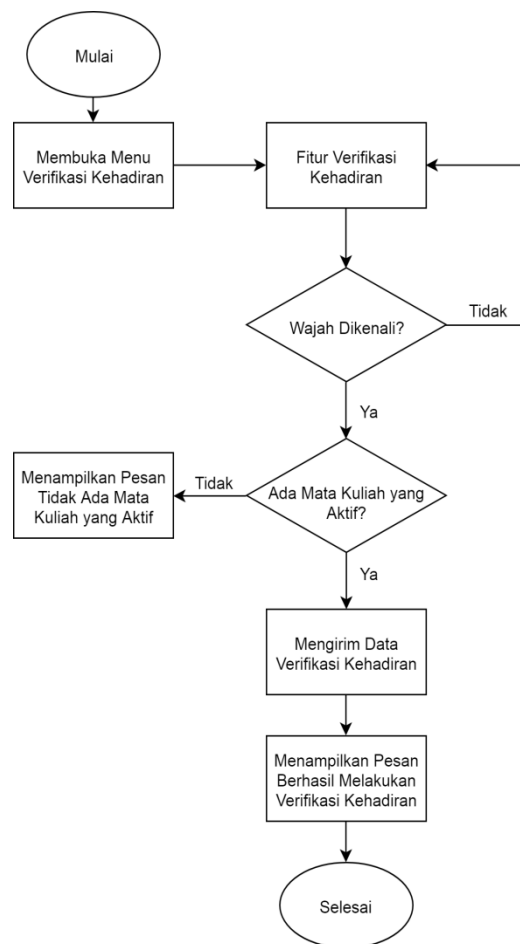
d. Rancangan Alir Proses Verifikasi Kehadiran

Pada Gambar 30 dijelaskan alir proses verifikasi kehadiran.

Tahapan proses yang dilakukan antara lain :

- 1) Dosen membuka menu verifikasi kehadiran.
- 2) Sistem akan memulai fitur verifikasi kehadiran.
- 3) Mahasiswa memposisikan wajah ke kamera.
- 4) Sistem akan mengecek apakah wajah dikenali atau tidak.
- 5) Jika iya maka sistem akan mengecek apakah ada status mata kuliah yang aktif.

- 6) Jika ada status mata kuliah yang aktif maka sistem akan mengirim data verifikasi kehadiran berdasarkan nim data wajah mahasiswa yang terdeteksi.
- 7) Jika tidak maka sistem akan menampilkan pesan bahwa tidak ada status mata kuliah yang aktif.



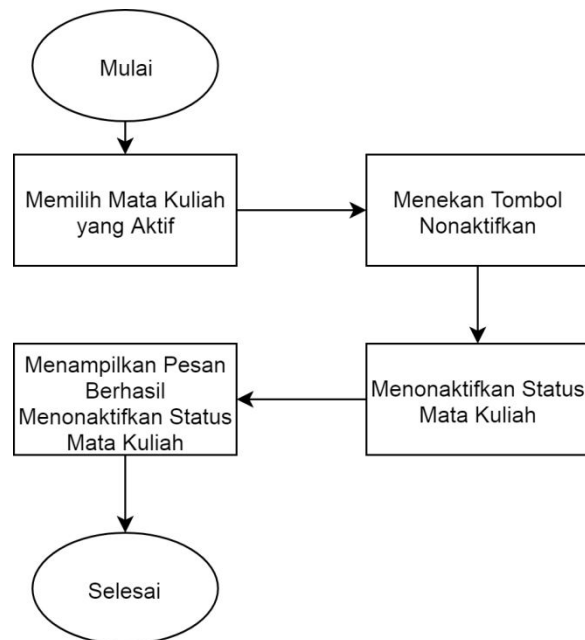
Gambar 30. Diagram Alir Verifikasi Kehadiran

e. Rancangan Alir Proses Menonaktifkan Status Matakuliah

Pada Gambar 31 dijelaskan alir proses verifikasi kehadiran.

Tahapan proses yang dilakukan antara lain :

- 1) Dosen memilih mata kuliah yang aktif.
- 2) Dosen menekan tombol nonaktifkan.
- 3) Sistem menonaktifkan status mata kuliah.
- 4) Sistem menampilkan pesan berhasil menonaktifkan status mata kuliah.



Gambar 31. Diagram Alir Menonaktifkan Status Matakuliah

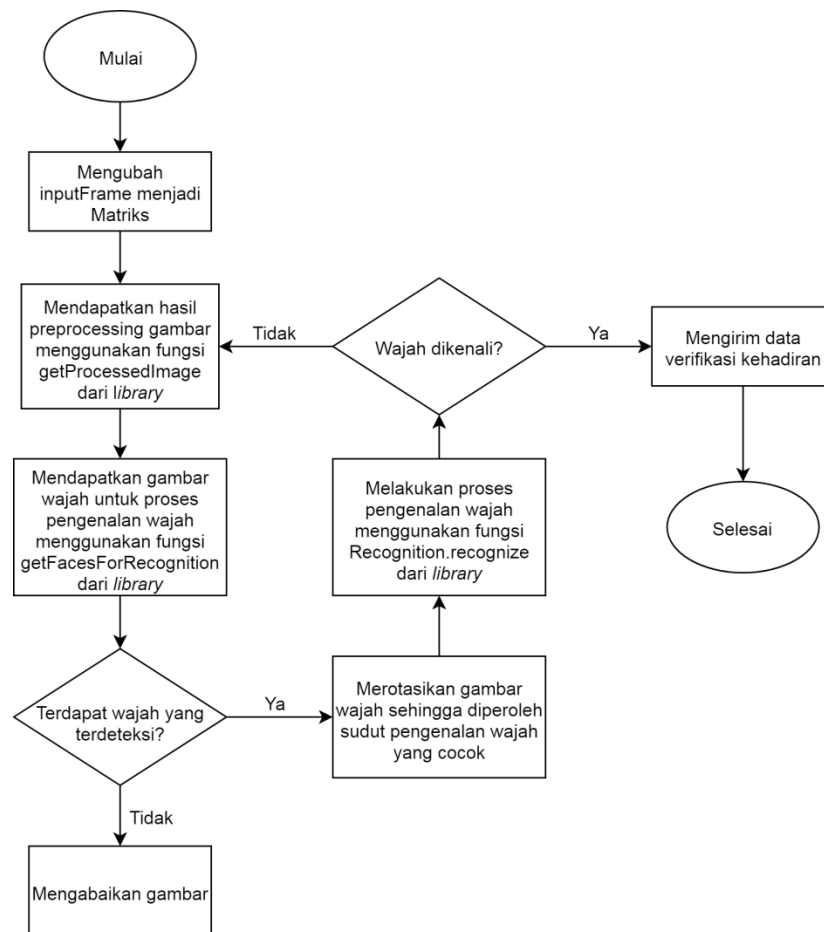
f. Rancangan Alir Proses Pengenalan Wajah dengan Pustaka Android *Face Recognition Eigenface*

Pada bagian ini dirancang alir proses pengenalan wajah dengan pustaka Android *Face Recognition Eigenface* pada aplikasi Sikemas. Proses ini digambarkan pada Gambar 32. Proses pengenalan wajah ini merupakan hasil dekomposisi dari proses utama verifikasi kehadiran.

Tahapan proses dari diagram alir Gambar 28 antara lain :

- 1) Kamera *smartphone* mendeteksi gambar wajah dalam bentuk `inputFrame` yang kemudian diubah menjadi matriks.
- 2) Aplikasi mendapatkan hasil preprocessing gambar menggunakan fungsi `getProcessedImage` dari pustaka (library) yang digunakan.
- 3) Aplikasi mendapatkan gambar wajah untuk proses pengenalan wajah menggunakan fungsi `getFacesForRecognition` dari pustaka (library) yang digunakan.
- 4) Aplikasi melakukan pengecekan apakah terdapat wajah yang terdeteksi.
- 5) Jika ada wajah yang terdeteksi maka gambar wajah akan dirotasikan hingga memperoleh sudut pengenalan wajah yang cocok.
- 6) Jika tidak ada wajah yang terdeteksi maka gambar akan diabaikan.

- 7) Dari tahap 5, kemudian aplikasi akan melakukan proses pengenalan wajah menggunakan fungsi `Recognition.recognize` dari pustaka (library) yang dipakai.
- 8) Jika ada wajah yang dikenali maka akan mengirim data verifikasi kehadiran.
- 9) Jika tidak ada maka akan kembali ke tahap 2.



Gambar 32. Diagram Alir Pengenalan Wajah Menggunakan Pustaka Android *Face Recognition Eigenface*

BAB IV

IMPLEMENTASI DAN PENGUJIAN

A. Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat keras serta perangkat lunak yang digunakan dalam tahap implementasi perangkat lunak ini dijelaskan pada Tabel 7.

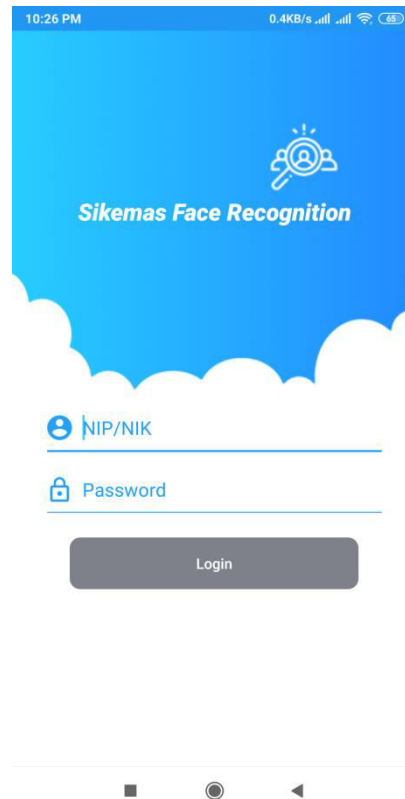
Tabel 7. Lingkungan Implementasi Perangkat Keras dan Lunak

Perangkat Keras	Komputer	Dell E6520
	Prosesor	Intel(R) Core(TM) i5-2540M CPU @ 2.60GHz (4 CPUs), ~2.6GHz
	Memori Primer	4GB
	Memori Sekunder	500GB
Perangkat Lunak	Sistem Operasi	Windows 7 Ultimate 64-bit
	Perangkat Lunak	Android Studio 3.3, Draw.io, Postman, Microsoft Word 2010

B. Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna aplikasi mobile Android ini menggunakan jenis file XML yang dibangun dan diimplementasikan dalam lingkungan kerja Android Studio. Penjelasan implementasi antarmuka aplikasi ditunjukkan melalui tampilan halaman XML yang sudah dirender oleh Android Studio berikut sesuai dengan rancangan antarmuka pada bagian bab III tugas akhir ini.

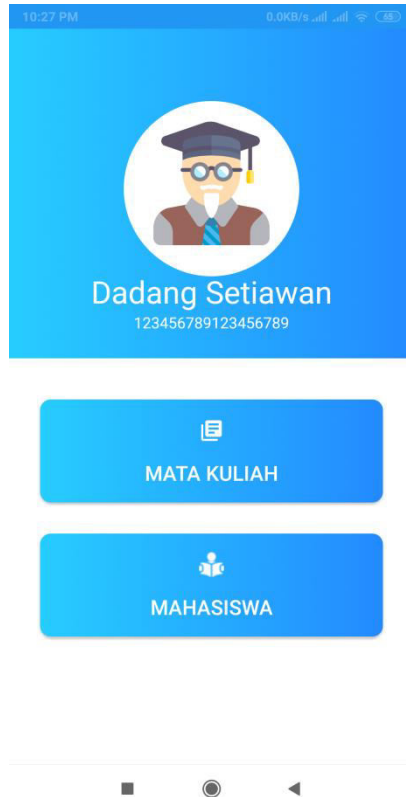
1. Implementasi Antarmuka Halaman Login



Gambar 33. Implementasi Antarmuka Halaman Login

Antarmuka halaman login pada Gambar 33 merupakan halaman yang pertama kali diakses oleh dosen ketika membuka aplikasi. Pada halaman ini terdapat dua form yaitu form NIP/NIK dan Password. Selain itu terdapat juga satu tombol Login. Form digunakan untuk mengisi identitas dosen yang dijadikan sebagai parameter login, sedangkan tombol digunakan untuk aksi verifikasi identitas agar dapat masuk ke sistem.

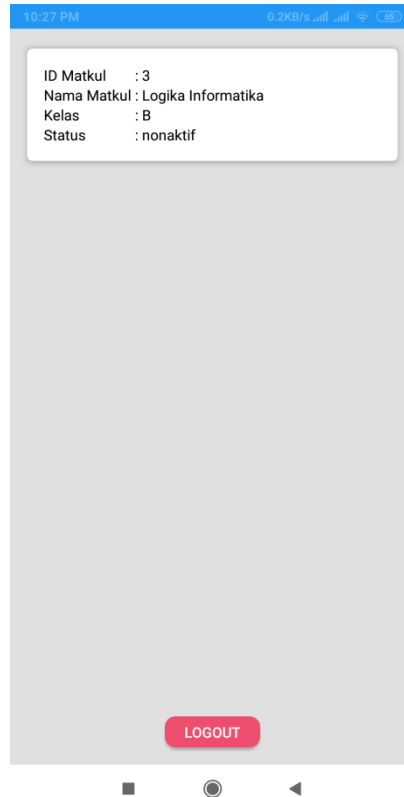
2. Implementasi Antarmuka Halaman Menu Utama



Gambar 34. Implementasi Antarmuka Halaman Menu Utama

Antarmuka Halaman Menu Utama pada Gambar 34 merupakan halaman yang akan ditampilkan ketika dosen berhasil masuk ke sistem. Halaman ini terdiri dari dua *textview*, satu *imageview*, dan dua tombol. *Textview* digunakan untuk memberikan informasi identitas nama dan nip/nik dosen terlogin. *Imageview* digunakan untuk menampilkan gambar ilustrasi seorang dosen yang dapat dilihat pada Gambar 30. Dua tombol digunakan untuk masuk ke menu mata kuliah dan menu mahasiswa.

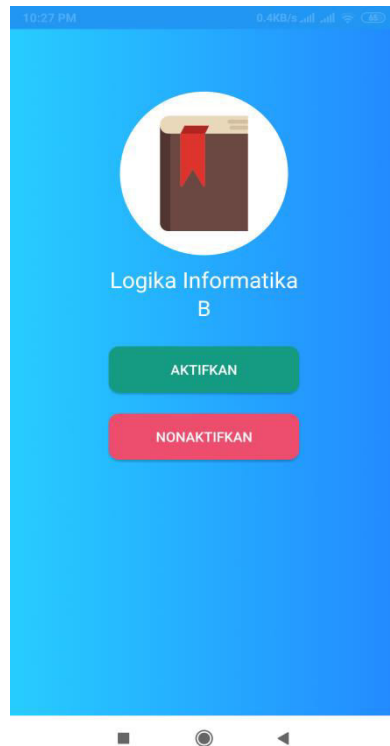
3. Implementasi Antarmuka Halaman Matakuliah



Gambar 35. Implementasi Halaman Antarmuka Matakuliah

Antarmuka halaman menu mata kuliah pada Gambar 35 merupakan halaman untuk menampilkan daftar mata kuliah berdasarkan nip/nik dosen yang terlogin. Pada halaman ini terdapat *recycler view* yang berfungsi untuk menampilkan daftar mata kuliah. *Recycler view* mengimplementasikan fungsi *onclick* yang bertujuan agar daftar mata kuliah dapat ditekan untuk menuju halaman *update* status mata kuliah. Selain itu terdapat juga tombol Logout yang berfungsi untuk keluar dari aplikasi ini.

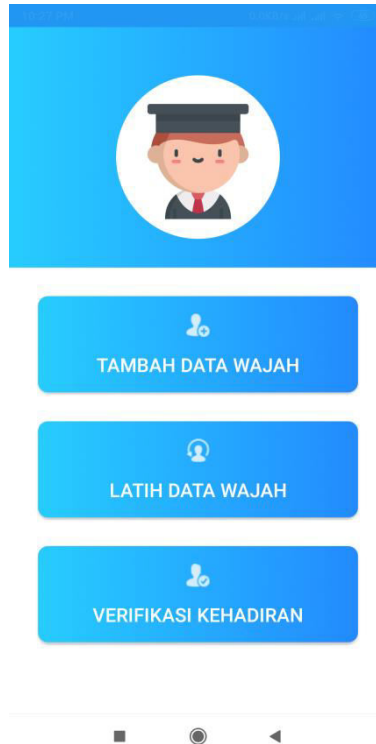
4. Implementasi Antarmuka Halaman Update Status Matakuliah



Gambar 36. Implementasi Antarmuka Halaman Update Status Matakuliah

Antarmuka halaman menu update status mata kuliah pada Gambar 36 merupakan halaman untuk mengaktifkan dan menonaktifkan status mata kuliah. Pada halaman ini terdapat satu *imageview*, dua *textview*, dan dua tombol. *Imageview* digunakan untuk menampilkan gambar ilustrasi sebuah buku. *Textview* digunakan untuk menampilkan informasi mata kuliah berupa nama dan kelas. Tombol digunakan untuk memperbarui status mata kuliah.

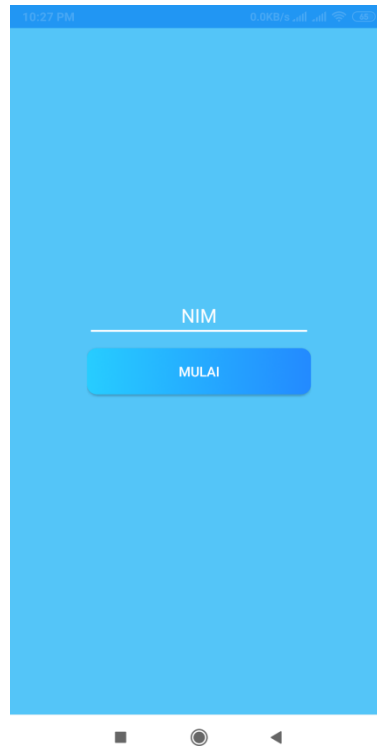
5. Implementasi Antarmuka Halaman Menu Mahasiswa



Gambar 37. Implementasi Antarmuka Halaman Menu Mahasiswa

Antarmuka halaman menu mahasiswa pada Gambar 37 merupakan halaman untuk mengelola mahasiswa. Pada halaman ini terdapat tiga tombol. Pertama tombol TAMBAH DATA WAJAH berfungsi untuk masuk ke menu tambah data wajah mahasiswa. Kedua tombol LATIH DATA WAJAH berfungsi masuk ke menu latih data wajah. Sedangkan tombol VERIFIKASI KEHADIRAN berfungsi untuk membuka fitur verifikasi kehadiran.

6. Implementasi Antarmuka Halaman Tambah Data Wajah



Gambar 38. Implementasi Antarmuka Halaman Tambah Data Wajah

Antarmuka halaman tambah data wajah mahasiswa pada Gambar 38 merupakan halaman untuk menambah data wajah mahasiswa. Pada halaman ini terdapat satu *form* dan tombol. *Form* digunakan untuk mengisi identitas mahasiswa yang akan ditambahkan data wajahnya, identitas tersebut berupa nim. Tombol digunakan untuk memulai fitur tambah data wajah.

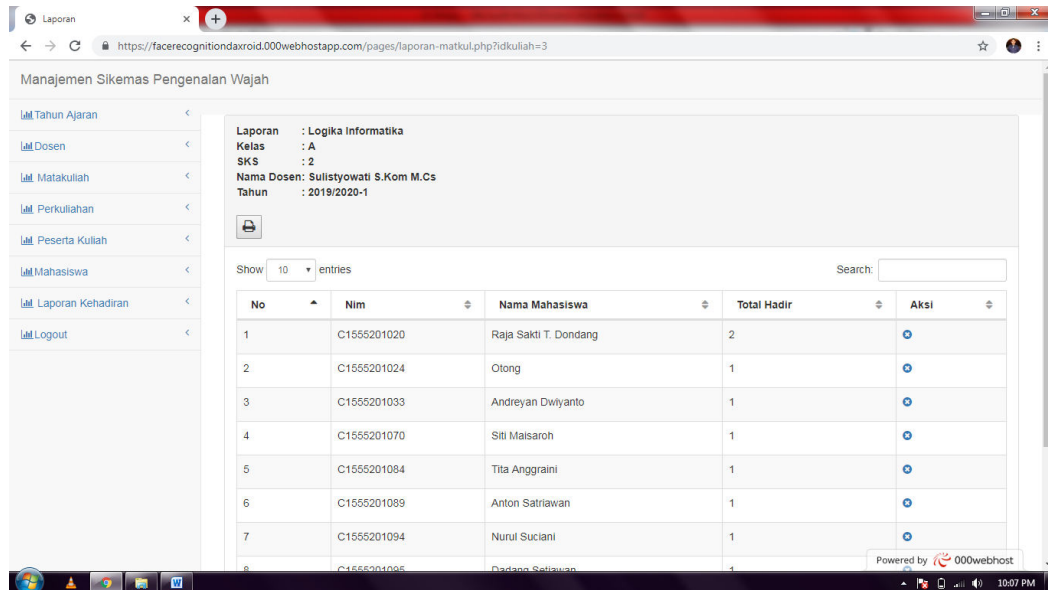
7. Implementasi Antarmuka Halaman Latih Data Wajah



Gambar 39. Implementasi Antarmuka Halaman Latih Data Wajah

Antarmuka halaman latih data wajah pada Gambar 39 merupakan halaman untuk melatih data wajah mahasiswa. Halaman ini hanya berisi *thread* yang berfungsi mengambil dan melatih data wajah mahasiswa dari *internal storage smartphone*.

8. Implementasi Antarmuka Halaman Web



Gambar 40. Implementasi Antarmuka Halaman Web

Antarmuka halaman web pada Gambar 40 merupakan halaman untuk admin mengelola segala sesuatu yang berhubungan dengan sistem kehadiran mahasiswa melalui pengenalan wajah. Hal-hal yang dikelola meliputi kelola mata kuliah, dosen, mahasiswa, tahun ajaran dan laporan.

C. Listing Program

Penulis menjelaskan listing program berdasarkan sisi client dan server.

Berikut penjelasan beberapa listing program yang digunakan untuk membangun sistem pengenalan wajah untuk kehadiran mahasiswa.

1. Koneksi Database Server

Tabel 8. Kode Program Koneksi Database Server

No. Baris	Kode Program
1.	<?php
2.	class Database {
3.	private \$_connection;
4.	private static \$_instance;
5.	private \$_host = "localhost";
6.	private \$_username = "id8042456_sikemas_wajah";
7.	private \$_password = "dadang";
8.	private \$_database = "id8042456_sikemas_wajah";
9.	}

Tabel 9. Penjelasan Listing Program Koneksi Database Server

No. Baris	Kegunaan
1	Tag pembuka php
2-4	<i>Class</i> untuk koneksi database
5	Nama host yang digunakan untuk koneksi
6	Nama username yang digunakan untuk koneksi
7	Password yang digunakan untuk koneksi
8	Nama database yang digunakan
9	Tag penutup php

2. Login Admin Server

Tabel 10. Kode Program Login Admin Server

No. Baris	Kode Program
1	function login(\$loginid,\$password){
2	if(!ctype_alpha(\$loginid) !ctype_alpha(\$password)){
3	echo "<script>alert('Either LoginId or Password is Missing')</script>";
4	else{\$db = Database::getInstance();
5	\$mysqli = \$db->getConnection();
6	\$query = "SELECT loginid, password FROM tbl_login where loginid=? and password=? ";
7	\$stmt= \$mysqli->prepare(\$query);
8	if(false=== \$stmt){trigger_error("Error in query: " . mysqli_connect_error(),E_USER_ERROR);} else{
9	\$stmt->bind_param('ss',\$loginid,\$password);
10	\$stmt->execute();
11	\$stmt -> bind_result(\$loginid,\$password);
12	\$rs=\$stmt->fetch();
13	if(!\$rs)
14	{ echo "<script>alert('Invalid Details')</script>";
15	header('location:login.php');}
16	else{ header('location:add-matkul.php');}}

Tabel 11. Penjelasan Listing Program Login Admin Server

No. Baris	Kegunaan
1	Fungsi login
2-3	Validasi field id dan password
4-5	Membuat koneksi ke database
6	Kueri yang digunakan untuk login
7-8	Mempersiapkan kueri untuk login
9-10	Mengirim data id dan password
11-13	Mengecek apakah id dan password sudah benar
14-15	Peringatan jika gagal dan meneruskan ke menu login
16	Jika benar maka diteruskan ke menu tambah matkul

3. Tambah Data Perkuliahan Pada Server

Tabel 12. Kode Program Tambah Data Perkuliahan

No. Baris	Kode Program
1	function tambah_kuliah(\$nik,\$idmatkul,\$matkul,\$kelas,\$tahun){
2	\$db = Database::getInstance();
3	\$mysqli = \$db->getConnection();
4	\$query = "select nik and idmatkul and kelas and tahun from kuliah where nik=? and idmatkul=? and kelas=? and tahun=?";
5	\$input="insert into kuliah (nik,idmatkul,matkul,kelas,status,tahun)values(?,?,?,',nonaktif,?)";
6	\$stmt=\$mysqli->prepare(\$query);
7	\$stmt->bind_param('ssss',\$nik,\$idmatkul,\$kelas,\$tahun);
8	\$stmt->execute();
9	\$stmt->store_result();
10	if(\$stmt->num_rows>0){
11	echo "<script>alert('Perkuliahan Sudah Ada')</script>";
12	echo "<script>window.location.href='add-kuliah.php'</script>";
13	} else {
14	\$stmt=\$mysqli->prepare(\$input);
15	\$stmt->bind_param('sssss',\$nik,\$idmatkul,\$matkul,\$kelas,\$tahun);
16	\$stmt->execute();
17	echo "<script>alert('Tambah Perkuliahan Berhasil')</script>";
18	echo "<script>window.location.href='view-kuliah.php'</script>";
19	}
20	}

Tabel 13. Penjelasan Listing Program Tambah Data Perkuliahan

No. Baris	Kegunaan
1	Fungsi menambah perkuliahan
2-3	Membuat koneksi ke database
4	Kueri untuk mengecek data nik, idmatkul, kelas, dan tahun
5	Kueri untuk mengirim data nik, idmatkul, kelas, status dan tahun
6-10	Mengeksekusi kueri untuk mengecek data
11-12	Jika data sudah ada maka menampilkan pesan error dan meneruskan ke menu tambah perkuliahan
13-18	Jika data belum ada maka akan menjalankan perintah input data dan menampilkan pesan berhasil

4. Edit Data Perkuliahan Pada Server

Tabel 14. Kode Program Edit Data Perkuliahan

No. Baris	Kode Program
1	function edit_kuliah(\$nik,\$idmatkul,\$matkul,\$kelas,\$idkuliah){
2	\$db = Database::getInstance();
3	\$mysqli = \$db->getConnection();
4	\$query = "update kuliah set nik=?, idmatkul=?, matkul=?, kelas=? where idkuliah=?";
5	\$stmt= \$mysqli->prepare(\$query);
6	\$stmt->bind_param('ssssi',\$nik, \$idmatkul,\$matkul,\$kelas,\$idkuliah);
7	\$stmt->execute();
8	echo "<script>alert('Perkuliahan Berhasil Diupdate')</script>";
9	echo "<script>>window.location.href='view-kuliah.php'</script>";
10	}

Tabel 15. Penjelasan Listing Program Edit Data Perkuliahan

No. Baris	Kegunaan
1	Fungsi untuk mengedit data perkuliahan
2-3	Membuat koneksi ke database
4	Kueri untuk mengupdate data perkuliahan
5-7	Mengeksekusi kueri
8	Menampilkan pesan berhasil
9	Meneruskan ke halaman <i>view</i> perkuliahan

5. Hapus Data Perkuliahan Pada Server

Tabel 16. Kode Program Hapus Data Perkuliahan

No. Baris	Kode Program
1	function del_kuliah(\$idkuliah){
2	\$db = Database::getInstance();
3	\$mysqli = \$db->getConnection();
4	\$query="delete from kuliah where idkuliah=?";
5	\$stmt= \$mysqli->prepare(\$query);
6	\$stmt->bind_param('s',\$idkuliah);
7	\$stmt->execute();

8	echo "<script>alert('Matakuliah Berhasil Dihapus')</script>";
9	echo "<script>window.location.href='view-kuliah.php'</script>";
10	}

Tabel 17. Penjelasan Listing Program Hapus Data Perkuliahan

No. Baris	Kegunaan
1	Fungsi untuk menghapus data perkuliahan
2-3	Membuat koneksi ke database
4	Kueri untuk menghapus data perkuliahan
5-7	Mengeksekusi kueri
8	Menampilkan pesan berhasil
9	Meneruskan ke halaman <i>view</i> perkuliahan

6. Menampilkan Laporan Kehadiran

Tabel 18. Kode Program Menampilkan Laporan Kehadiran

No. Baris	Kode Program
1	function LaporanMatkul(){
2	\$idkuliah=\$_GET['idkuliah'];
3	\$db = Database::getInstance();
4	\$mysqli = \$db->getConnection();
5	\$query = "select nim,namamhs, count(keterangan) as totalhadir from kehadiran where idkuliah='\$idkuliah' group by nim";
6	\$stmt= \$mysqli->query(\$query);
7	return \$stmt;

Tabel 19. Penjelasan Listing Program Menampilkan Laporan Kehadiran

No. Baris	Kegunaan
1	Fungsi untuk menampilkan laporan
2	Mendapatkan data idkuliah
3-4	Membuat koneksi ke database
5	Kueri untuk menghitung total hadir mahasiswa
6	Mengeksekusi kueri
7	Mengembalikan hasil kueri

7. Koneksi Database Client

Tabel 20. Kode Program Koneksi Database Client

No. Baris	Kode Program
1	<?php
2	define("DB_HOST", "localhost");
3	define("DB_USER", "id8042456_sikemas_wajah");
4	define("DB_PASSWORD", "dadang");
5	define("DB_DATABASE", "id8042456_sikemas_wajah");
6	?>

Tabel 21. Penjelasan Listing Program Koneksi Database Client

No. Baris	Kegunaan
1	Tag pembuka php
2	Mendefinisikan nama host
3	Mendefinisikan username
4	Mendefinisikan password
5	Mendefinisikan nama database
6	Tag penutup php

8. API Login Pada Client

Tabel 22. Kode Program API Login Client

No. Baris	Kode Program
1	<?php
2	require_once '/storage/ssd3/456/8042456/public_html/include/DB_Function.php';
3	\$db = new DB_Function();
4	// json response array
5	\$response = array("error" => FALSE);
6	if (isset(\$_POST['nik']) && isset(\$_POST['password'])) {
7	// menerima parameter POST (email dan password)
8	\$nik = \$_POST['nik'];
9	\$password = \$_POST['password'];
10	\$user = \$db->getUserByNikAndPassword(\$nik, \$password);
11	if (\$user != false) {

12	// user ditemukan
13	\$response["error"] = FALSE;
14	\$response["user"]["nik"] = \$user["nik"];
15	\$response["user"]["nama"] = \$user["nama"];
16	echo json_encode(\$response);
17	} else {
18	// user tidak ditemukan password/email salah
19	\$response["error"] = TRUE;
20	\$response["error_msg"] = "Login gagal. NIK/NIP/Password ada yang Salah";
21	echo json_encode(\$response);
22	}
23	} else {
24	\$response["error"] = TRUE;
25	\$response["error_msg"] = "Parameter (nip/nik atau password) ada yang kurang";
26	echo json_encode(\$response);
27	}
28	?>

Tabel 23. Penjelasan Listing Program API Login Client

No. Baris	Kegunaan
1	Tag pembuka php
2	Memanggil file Db_Function.php
3	Membuat fungsi baru
5-9	Menerima parameter nik dan password
10-11	Mengecek apakah ada user dengan nik dan password tersebut
13	Jika ada menampilkan respon error = FALSE
14	Mengirim respon data nik
15	Mengirim respon data nama
16	Membungkus data pesan dalam bentuk json
17-19	Jika data salah
20	Menampilkan pesan "Login gagal. NIK/NIP/Password ada yang Salah"
21	Membungkus data dalam bentuk json
23-24	Jika data ada yang kurang
26	Menampilkan pesan Parameter (nip/nik atau password) ada yang kurang"
26	Membungkus data pesan dalam bentuk json

9. API Verifikasi Kehadiran Pada Client

Tabel 24. Kode Program API Verifikasi Kehadiran

No. Baris	Kode Program
1	<?php
2	if(\$_SERVER['REQUEST_METHOD']=='POST') {
3	\$response = array();
4	date_default_timezone_set('Asia/Jakarta');
5	\$nim = \$_POST['nim'];
6	\$date = date('Y-m-d');
7	require_once('dbConnect.php');
8	\$sql = "SELECT * FROM kehadiran WHERE tanggal = '\$date' AND nim='\$nim' AND matkul = (SELECT matkul FROM jadwal WHERE status = 'aktif') AND kelas = (SELECT kelas FROM jadwal WHERE status='aktif') ";
9	\$check = mysqli_fetch_array(mysqli_query(\$con,\$sql));
10	if(isset(\$check)){
11	\$response["value"] = 0;
12	\$response["message"] = "Anda Sudah Melakukan Verifikasi Kehadiran";
13	echo json_encode(\$response);
14	} else {
15	\$sql = "INSERT INTO kehadiran (idmatkul,nim,namamhs,matkul,kelas,keterangan,tanggal,tahun) VALUES ((SELECT idmatkul FROM jadwal WHERE status = 'aktif'), '\$nim', (SELECT namamhs FROM mahasiswa WHERE nim = '\$nim'), (SELECT matkul FROM jadwal WHERE status = 'aktif'), (SELECT kelas FROM jadwal WHERE status = 'aktif'), 'hadir', '\$date', (select tahun from jadwal where status='aktif'))";
16	if(mysqli_query(\$con,\$sql)) {
17	\$response["value"] = 1;
18	\$response["message"] = "Berhasil Melakukan Verifikasi Kehadiran";
19	echo json_encode(\$response);
20	} else {
21	\$response["value"] = 0;
22	\$response["message"] = "Tidak ada Matkul yang AkiF";
23	echo json_encode(\$response);
24	}
25	}
26	mysqli_close(\$con);
27	}
28	?>

Tabel 25. Penjelasan Listing Program API Verifikasi Kehadiran

No. Baris	Kegunaan
1	Tag pembuka php
2-3	Mendeklarasikan metode post
4	Mengubah default waktu ke jakarta atau GMT +7
5	Menerima parameter nim
6	Set tanggal
7	Memanggil file dbConnect.php
8	Kueri untuk mengecek data mahasiswa sudah verifikasi atau belum
9-10	Mengeksekusi kueri
11-12	Jika sudah ada maka menampilkan pesan “Anda Sudah Melakukan Verifikasi Kehadiran”
13	Membungkus data pesan dalam bentuk json
14	Jika tidak
15	Kueri untuk menginput data kehadiran mahasiswa
16-18	Jika ada matakuliah yang aktif maka mengeksekusi kueri dan menampilkan pesan “Berhasil Melakukan Verifikasi Kehadiran”
20	Jika tidak
21-22	Menampilkan pesan “ Tidak ada Matkul yang Aktif “
23	Membungkus data pesan dalam bentuk json
26	Menutup koneksi

10. API Menampilkan Matakuliah Pada Client

Tabel 26. Kode Program API Menampilkan Matakuliah

No. Baris	Kode Program
1	<?php
2	require_once('dbConnect.php');
3	if(\$_SERVER['REQUEST_METHOD']=='GET') {
4	\$nik=\$_GET['nik'];
5	\$sql = "SELECT * FROM kuliah WHERE nik='\$nik' ORDER BY status ASC ";
6	\$res = mysqli_query(\$con,\$sql);
7	\$result = array();
8	while(\$row = mysqli_fetch_array(\$res)){
9	array_push(\$result, array(

	'idmatkul'=>\$row[0], 'nik'=>\$row[1], 'matkul'=>\$row[2], 'kelas'=>\$row[3], 'status'=>\$row[4], 'tahun'=>\$row[5]));
10	}
11	echo json_encode(array("value"=>1, "result"=>\$result));
12	mysqli_close(\$con);
13	}

Tabel 27. Penjelasan Listing Program API Menampilkan Matakuliah

No. Baris	Kegunaan
1	Tag pembuka php
2	Memanggil file dbConnect.php
3	Mendeklarasikan metode post
4	Mendapatkan parameter nim
5	Kueri untuk menampilkan matakuliah berdasarkan nik
6-8	Mengeksekusi kueri
9	Mendeskripsikan array hasil kueri
11	Membungkus data array hasil kueri dalam bentuk json

11. Tambah Data Wajah Pada Client

Tabel 28. Kode Program Tambah Data Wajah

No. Baris	Kode Program
1	public Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame inputFrame) {
2	Mat imgRgba = inputFrame.rgba();
3	Mat imgCopy = new Mat();
4	imgRgba.copyTo(imgCopy);
5	// Selfie / Mirror mode
6	if(front_camera){
7	Core.flip(imgRgba, imgRgba, 1);
8	}
9	long time = new Date().getTime();
10	if((method == MANUALLY) (method == TIME) && (lastTime + timerDiff < time)){
11	lastTime = time;
12	List<Mat> images = ppF.getCroppedImage(imgCopy);

13	if (images != null && images.size() == 1){
14	Mat img = images.get(0);
15	if(img != null){
16	Rect[] faces = ppF.getFacesForRecognition();
17	if((faces != null) && (faces.length == 1)){
18	faces = MatOperation.rotateFaces(imgRgba, faces, ppF.getAngleForRecognition());
19	if(((method == MANUALLY) && capturePressed) (method == TIME)){
20	MatName m = new MatName(name + "_" + total, img);
21	if (folder.equals("Test")) {
22	String wholeFolderPath = fh.TEST_PATH + name + "/" + subfolder;
23	new File(wholeFolderPath).mkdirs();
24	fh.saveMatToImage(m, wholeFolderPath + "/");
25	}
26	for(int i = 0; i<faces.length; i++){
27	MatOperation.drawRectangleAndLabelOnPreview(imgRgba, faces[i], String.valueOf(total), front_camera);
28	}
29	total++;
30	if(total >= numberOfPictures){
31	Intent intent = new Intent(getApplicationContext(), AddFaceActivity.class);
32	intent.putExtra("result_mesasge", "Berhasil menambahkan data wajah");
33	intent.putExtra("number_of_picture",total);
34	intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
35	startActivity(intent);
36	}
37	capturePressed = false;
38	} else {
39	for(int i = 0; i<faces.length; i++){
40	MatOperation.drawRectangleOnPreview(imgRgba, faces[i], front_camera);
41	}
42	}
43	}
44	}
45	}
46	}
47	return imgRgba;
48	}

Tabel 29. Penjelasan Listing Program Tambah Data Wajah

No. Baris	Kegunaan
1	Fungsi untuk mendeteksi wajah yang tertangkap pada frame kamera
2	Menampung rgba dari inputFrame kedalam bentuk matriks imgRgba
3	Menginstansiasi Matriks baru bernama img
4	Menyalin matriks imgRgba ke matriks img
5-8	Melakukan proses mencerminkan tampilan kameran depan dari smartphone.
12	Menampung hasil crop gambar ke dalam list bernama images
13	Mengecek jumlah wajah yang tertangkap oleh kamera
14-16	Mengambil wajah pada indeks ke 0 dari list image kemudian melakukan cek adanya wajah atau tidak, jika iya maka gambar akan terambil
17	Mengecek jika hanya terdapat 1 wajah saja yang terdeteksi oleh kamera
18	Melakukan rotasi terhadap gambar yang terambil.
20-29	Melakukan mekanisme penyimpanan file gambar yang terambil ke internal penyimpanan smartphone.
30-36	Menampilkan pesan “Berhasil menambah data wajah” jika data wajah sudah berhasil diambil sebanyak 20
40-48	Melakukan iterasi untuk penggambaran ulang area wajah yang terdeteksi

12. Latih Data Wajah Pada Client

Tabel 30. Kode Program Latih Data Wajah

No. Baris	Kode Program
1	<code>final Handler handler = new Handler(Looper.getMainLooper());</code>
2	<code>thread = new Thread(new Runnable() {</code>
3	<code>public void run() {</code>
4	<code>if(!Thread.currentThread().isInterrupted()){</code>
5	<code>PreProcessorFactory ppF = new PreProcessorFactory(getApplicationContext());</code>

6	PreferencesHelper preferencesHelper = new PreferencesHelper(getApplicationContext());
7	String algorithm = preferencesHelper.getClassificationMethod();
8	FileHelper fileHelper = new FileHelper();
9	fileHelper.createDataFolderIfNotExsiting();
10	final File[] persons = fileHelper.getTrainingList();
11	if (persons.length > 0) {
12	Recognition rec = RecognitionFactory.getRecognitionAlgorithm(getApplicationContext(), Recognition.TRAINING, algorithm);
13	for (File person : persons) {
14	if (person.isDirectory()){
15	File[] files = person.listFiles();
16	int counter = 1;
17	for (File file : files) {
18	if (FileHelper.isFileAnImage(file)){
19	Mat imgRgb = Imgcodecs.imread(file.getAbsolutePath());
20	Imgproc.cvtColor(imgRgb, imgRgb, Imgproc.COLOR_BGRA2RGBA);
21	Mat processedImage = new Mat();
22	imgRgb.copyTo(processedImage);
23	List<Mat> images = ppF.getProcessedImage(processedImage, PreProcessorFactory.PreprocessingMode.RECOGNITION);
24	if (images == null images.size() > 1) {
25	// More than 1 face detected --> cannot use this file for training
26	continue;
27	} else {
28	processedImage = images.get(0);
29	}
30	if (processedImage.empty()) {
31	continue;
32	}
33	// The last token is the name --> Folder name = Person name
34	String[] tokens = file.getParent().split("/");
35	final String name = tokens[tokens.length - 1];
36	
37	MatName m = new MatName("processedImage", processedImage);
38	fileHelper.saveMatToImage(m, FileHelper.DATA_PATH);
39	
40	rec.addImage(processedImage, name, false);
41	
42	fileHelper.saveCroppedImage(imgRgb, ppF, file, name, counter);
43	

44	// Update screen to show the progress
45	final int counterPost = counter;
46	final int filesLength = files.length;
47	progress.post(new Runnable() {
48	@Override
49	public void run() {
50	progress.append("Image " + counterPost + " of " + filesLength + " from " + name + " imported.\n");
51	}
52	});
53	
54	counter++;
55	}
56	}
57	}
58	}

Tabel 31. Penjelasan Listing Program Latih Data Wajah

No. Baris	Kegunaan
1	Menginstansiasi kelas handler untuk menjembatani komunikasi antara worker thread dengan main thread.
2	Menginstansiasi kelas thread untuk melakukan proses melatih data set wajah
5	Menginstansiasi kelas PreProcessorFactory (kelas dari library) untuk activity saat ini. Kelas PreProcessorFactory berfungsi sebagai kelas pembantu untuk pengambilan data-data yang akan diolah sebelum dan sesudah data diproses.
6	Menginstansiasi kelas PreferencesHelper (kelas dari library) untuk activity saat ini. Kelas PreferencesHelper berfungsi sebagai kelas pembantu untuk mendapatkan pengaturan-pengaturan dari metode yang akan dipakai.
7	Membuat variable algorithm bertipe data String untuk menampung nama metode klasifikasi yang dipakai.
10	Menginstansiasi kelas FileHelper (kelas dari library) untuk membantu dalam memanipulasi pembuatan file dan folder yang diperlukan dalam penggunaan library.
12	Membuat variable persons bertipe data Array of File untuk menampung semua data wajah mahasiswa yang disimpan di internal penyimpanan smartphone (file hasil dari pendaftaran/unduhan data wajah).

13-14	Mengambil semua lokasi dari data set pembandingan yang telah disediakan di dalam aplikasi.
15	Percabangan untuk memastikan terdapat data wajah mahasiswa dan data pembandingan yang tersimpan
16	Implementasi dari kelas Interface Recognition untuk mendapatkan algoritma recognition untuk training dari algoritma klasifikasi yang sudah ditentukan sebelumnya
18-19	Mendapatkan seluruh file data wajah pembandingan dari list file lokasi yang didapat dari baris kode 13 – 14.
20	Mengkonversi warna gambar dari BGRA ke RGBA.
21	Menginstansiasi kelas Mat baru bernama processedDataSetImage
22	Menyalin hasil konversi gambar pada baris program 20 ke processedDataSetImage
35	Menampung hasil dari pemrosesan gambar (pemrosesan gambar di dalam library) dari Matriks processedDataSetImage ke dalam sebuah list bernama images.
37	Memberi nama Matriks processedDataSetImage menjadi “processedImage”
38	Menyimpan matriks pada baris ke 37 menjadi gambar, dan menaruhnya sesuai lokasinya
40	Memproses data gambar (proses ekstraksi fitur di dalam library dengan metode yg sudah ditentukan)
45-50	Menampilkan proses training ke layar smartphone

13. Verifikasi Kehadiran Pada Client

Tabel 32. Penjelasan Listing Program Verifikasi Kehadiran

No. Baris	Kode Program
1	public Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame inputFrame) {
2	Mat imgRgba = inputFrame.rgba();
3	Mat img = new Mat();
4	imgRgba.copyTo(img);
5	List<Mat> images = ppF.getProcessedImage(img, PreProcessorFactory.PreprocessingMode.RECOGNITION);
6	Rect[] faces = ppF.getFacesForRecognition();
7	if(front_camera){
8	Core.flip(imgRgba,imgRgba,1);

9	}
10	if(images == null images.size() == 0 faces == null faces.length == 0 ! (images.size() == faces.length)){
11	return imgRgba;
12	} else {
13	faces = MatOperation.rotateFaces(imgRgba, faces, ppF.getAngleForRecognition());
14	for(int i = 0; i<faces.length; i++){
15	MatOperation.drawRectangleAndLabelOnPreview(imgRgba, faces[i], rec.recognize(images.get(i), ""), front_camera);
16	identitasTerdeteksi = rec.recognize(images.get(i), "");
17	byte[] bytes = identitasTerdeteksi.getBytes();
18	String nim = new String(bytes);
19	Retrofit retrofit = new Retrofit.Builder()
20	.baseUrl(URL)
21	.addConverterFactory(GsonConverterFactory.create())
22	.build();
23	RegisterAPI api = retrofit.create(RegisterAPI.class);
24	Call<Value> call = api.persensi(nim);
25	call.enqueue(new Callback<Value>() {
26	@Override
27	public void onResponse(Call<Value> call, Response<Value> response) {
28	String value = response.body().getValue();
29	String message = response.body().getMessage();
30	if (value.equals("1")) {
31	Toast.makeText(RecognitionActivity.this, message, Toast.LENGTH_SHORT).show();
32	} else {
33	Toast.makeText(RecognitionActivity.this, message, Toast.LENGTH_SHORT).show();
34	}
35	}
36	@Override
37	public void onFailure(Call<Value> call, Throwable t) {
38	Toast.makeText(RecognitionActivity.this, "Jaringan Error!", Toast.LENGTH_SHORT).show();
39	}
40	});
41	}
42	return imgRgba;
43	}
44	}

Tabel 33. Penjelasan Listing Program Verifikasi Kehadiran

No. Baris	Kegunaan
1	Fungsi untuk mengenali wajah yang tertangkap pada frame kamera
2	Menampung rgba dari inputFrame kedalam bentuk matriks imgRgba
3	Menginstansiasi Matriks baru bernama img
4	Menyalin matriks imgRgba ke matriks img
5	Melakukan proses pre processing terhadap matriks gambar yang terambil dari inputframe.
6	Mengambil area wajah untuk dikenali dalam bentuk kotak
7-9	Mencerminkan tampilan wajah yang terambil ketika menggunakan kamera depan smartphone.
10-11	Melakukan perubahan tampilan ketika tidak ada wajah yang terdeteksi.
13	Merotasikan wajah sehingga diperoleh sudut pengenalan wajah yang cocok.
14-15	Menggambar area persegi beserta label wajah untuk wajah yang terdeteksi.
16-18	Menerima parameter nim dari wajah yang terdeteksi
19-23	Membuat koneksi API untuk mengirim data
24-25	Mengirim data verifikasi menggunakan api.persensi
27-38	Menerima respon dari server
42	Mengulangi fungsi verifikasi dari awal

D. Lingkungan Pengujian

Pada proses pengujian perangkat lunak, dibutuhkan suatu lingkungan pengujian yang sesuai dengan standar kebutuhan. Spesifikasi lingkungan pengujian akan dijelaskan pada Tabel 34.

Tabel 34. Lingkungan Pengujian Fungsionalitas Perangkat Lunak

Spesifikasi	Deskripsi
Jenis Perangkat	<i>Smartphone</i>
Merek Perangkat	Xiaomi Redmi Note 5 Pro
Sistem Operasi	Android Oreo 8.1
Memori Internal	32 GB
RAM	4 GB

E. Pengujian Kegunaan

Pengujian kegunaan ini dilakukan untuk mengetahui tingkat kegunaan aplikasi terhadap pengguna. Kegunaan yang diuji meliputi kemudahan dalam melakukan verifikasi kehadiran dengan menggunakan pengenalan wajah.

Pengujian dilakukan dengan memberikan kesempatan pada pengguna untuk mencoba sendiri aplikasi yang telah dikembangkan dengan arahan dari pengembang aplikasi.

Pengujian dilakukan pada sebuah skenario simulasi kelas yang didalamnya terdapat 10 mahasiswa dan 1 dosen. Uji coba yang dilakukan adalah mahasiswa pada kelas tersebut melakukan verifikasi kehadiran mulai dari mendaftarkan data wajah hingga melakukan verifikasi kehadiran. Setelah pengujian selesai pengguna

diminta untuk mengisi kuisioner yang diberikan untuk mengetahui tanggapan pengguna. Rincian dari kuisioner dapat dilihat pada Tabel 35.

Tabel 35. Kuisioner Pengujian Kegunaan

No	Pernyataan	Penilaian				
		1	2	3	4	5
1	Menurut saya, aplikasi ini mudah digunakan					
2	Menurut saya, absensi kehadiran dengan menggunakan aplikasi ini lebih aman dari kecurangan daripada sistem manual (absensi di kertas)					
3	Menurut saya, lebih mudah dan efisien melakukan absensi dengan menggunakan aplikasi ini daripada sistem manual (presensi di kertas)					
4	Menurut saya, mudah menggunakan fitur absensi dengan pengenalan wajah					
5	Menurut saya, mudah memasukkan data wajah ke dalam aplikasi					
6	Menurut saya, hasil absensi dengan fitur pengenalan wajah memiliki hasil yang akurat					
7	Menurut saya, absensi dengan menggunakan fitur pengenalan wajah dalam aplikasi ini lebih aman dari kecurangan daripada sistem manual (absensi di kertas)					
8	Menurut saya, lebih mudah dan efisien melakukan absensi dengan menggunakan fitur pengenalan wajah dalam aplikasi ini daripada sistem manual (absensi di kertas)					

Keterangan :

1. Sangat tidak setuju
2. Tidak setuju
3. Netral
4. Setuju
5. Sangat Setuju

Daftar pengguna yang melakukan pengujian dapat dilihat pada Tabel 36.

Tabel 36. Peserta Pengujian Kegunaan

No	NIK/NIM	Nama Peserta	Kelas Skenario
1	198212162007112	Sulistiyowati S.Kom M.Cs	Kelas A
2	C1555201094	Nurul Suciani	Kelas A
3	C1555201089	Anton Satriawan	Kelas A
4	C1555201036	Puteri Diah P.E	Kelas A
5	C1555201070	Siti Maisarah	Kelas A
6	C1555201100	Rahmat Hidayat	Kelas A
7	C1555201020	Raja Sakti T. Dondang	Kelas A
8	C1555201024	Otong J.	Kelas A
9	C1555201084	Tita Anggraini	Kelas A
10	C1555201033	Andreyan Dwiyanto	Kelas A
11	C1555201104	Ahmad Rifani Ananda	Kelas A

Berikut ini adalah rincian pengujian kegunaan skenario kelas A:

- Nama Matakuliah : Logika Informatika
- Kelas : A
- Hari/Tanggal : Jumat, 5 Juli 2019

F. Hasil Pengujian Kegunaan

Hasil penilaian rekapitulasi dari masing-masing peserta dilakukan rekapitulasi untuk mendapatkan nilai rata-rata dari masing-masing pernyataan kuisioner. Penilaian kuisioner ini memiliki skala 1-5 dimana semakin kecil angka skala yang dipilih berarti semakin tidak setuju sedangkan semakin besar angka

skala yang dipilih berarti semakin setuju dengan pernyataan kuisioner. Rekapitulasi hasil kuisioner disajikan dalam bentuk rata-rata dari masing-masing pernyataan yang dapat dilihat pada Tabel 37.

Tabel 37. Rekapitulasi Hasil Penilaian

No	Pernyataan	Hasil Penilaian
1	Menurut saya, aplikasi ini mudah digunakan	4,1
2	Menurut saya, absensi kehadiran dengan menggunakan aplikasi ini lebih aman dari kecurangan daripada sistem manual (absensi di kertas)	4,1
3	Menurut saya, lebih mudah dan efisien melakukan absensi dengan menggunakan aplikasi ini daripada sistem manual (presensi di kertas)	4,2
4	Menurut saya, mudah menggunakan fitur absensi dengan pengenalan wajah	4,2
5	Menurut saya, mudah memasukkan data wajah ke dalam aplikasi	4,2
6	Menurut saya, hasil absensi dengan fitur pengenalan wajah memiliki hasil yang akurat	4,2
7	Menurut saya, absensi dengan menggunakan fitur pengenalan wajah dalam aplikasi ini lebih aman dari kecurangan daripada sistem manual (absensi di kertas)	4,1
8	Menurut saya, lebih mudah dan efisien melakukan absensi dengan menggunakan fitur pengenalan wajah dalam aplikasi ini daripada sistem manual (absensi di kertas)	4

Dari hasil rekapitulasi penilaian oleh 10 mahasiswa didapat nilai seperti pada tabel diatas. Menurut penulis hasil tersebut cukup memuaskan. Hasil tersebut merepresentasikan bahwa sistem yang penulis bangun cukup layak untuk digunakan.

G. Pengujian Tingkat Akurasi

Pengujian tingkat akurasi ini dilakukan untuk mengetahui seberapa tinggi tingkat akurasi pada proses pengenalan wajah. Hasil pengujian tingkat akurasi dapat dilihat pada Tabel 38.

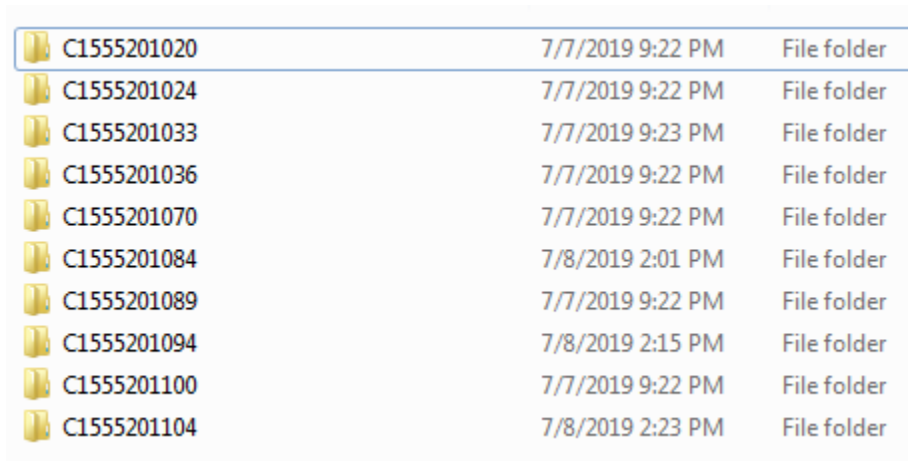
Tabel 38. Hasil Pengujian Tingkat Akurasi

No	Nim	Nama Peserta	Kelas Skenario	Akurat
1	C1555201094	Nurul Suciani	Kelas A	Ya
2	C1555201089	Anton Satriawan	Kelas A	Ya
3	C1555201036	Puteri Diah P.E	Kelas A	Ya
4	C1555201070	Siti Maisarah	Kelas A	Ya
5	C1555201100	Rahmat Hidayat	Kelas A	Tidak
6	C1555201020	Raja Sakti T. Dondang	Kelas A	Ya
7	C1555201024	Otong J.	Kelas A	Ya
8	C1555201084	Tita Anggraini	Kelas A	Ya
9	C1555201033	Andreyan Dwiyanto	Kelas A	Ya
10	C1555201104	Ahmad Rifani Ananda	Kelas A	Ya

Dari 10 kali percobaan mahasiswa mengirim verifikasi kehadiran menggunakan pengenalan wajah hasil yang diperoleh adalah 9 berhasil dan 1 gagal. Maka dari itu tingkat akurasi pengenalan wajah yaitu bernilai 90%. Menurut penulis kegagalan sistem untuk mengenali wajah adalah adanya sedikit kemiripan antara mahasiswa atas nama Rahmat Hidayat dan Anton Satriawan dan kurangnya intensitas cahaya saat melakukan pengenalan wajah. Pada saat Rahmat Hidayat melakukan verifikasi kehadiran, sistem mengenali wajahnya sebagai Anton Satriawan.

H. Pengujian Kapasitas Pengolahan Data Wajah

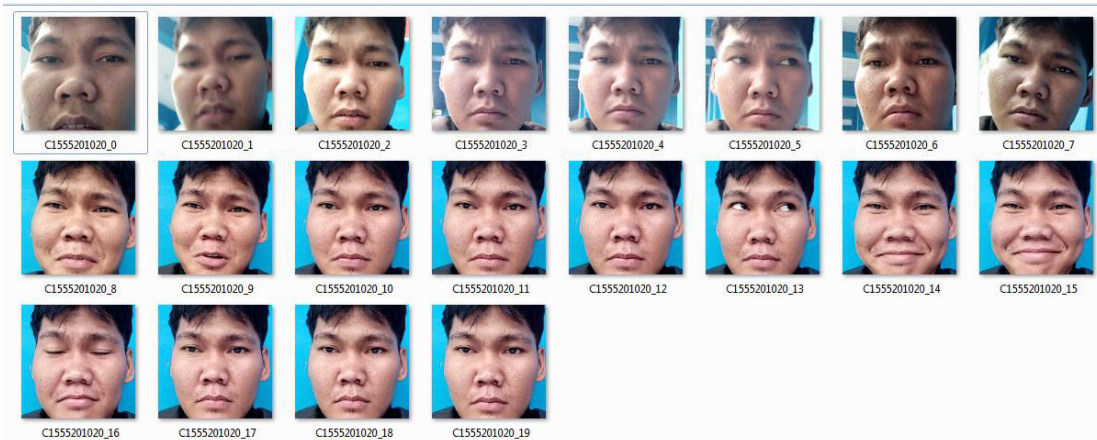
Pengujian kapasitas pengolahan data wajah merupakan pengujian terhadap kemampuan aplikasi untuk menampung dan mengolah seberapa banyak data wajah yang dapat diolah oleh perangkat *smartphone*. Pada Gambar 41 ditunjukkan contoh 1 folder data wajah dalam 1 kelas. Kelas ini memiliki jumlah mahasiswa 10 orang dimana tiap orang memiliki 20 foto wajah yang disimpan dan diolah. Pada Tabel 39 ditunjukkan rincian kapasitas yang dibutuhkan oleh tiap data wajah mahasiswa dalam 1 kelas.



C1555201020	7/7/2019 9:22 PM	File folder
C1555201024	7/7/2019 9:22 PM	File folder
C1555201033	7/7/2019 9:23 PM	File folder
C1555201036	7/7/2019 9:22 PM	File folder
C1555201070	7/7/2019 9:22 PM	File folder
C1555201084	7/8/2019 2:01 PM	File folder
C1555201089	7/7/2019 9:22 PM	File folder
C1555201094	7/8/2019 2:15 PM	File folder
C1555201100	7/7/2019 9:22 PM	File folder
C1555201104	7/8/2019 2:23 PM	File folder

Gambar 41. Data Wajah 10 Mahasiswa dalam 1 Skenario Kelas

Gambar diatas merupakan isi dalam 1 folder training yang didalamnya terdapat 10 data wajah mahasiswa, dimana masing-masing mahasiswa memiliki 20 data wajah. Contoh data wajah salah satu mahasiswa dapat dilihat pada Gambar 42.



Gambar 42. Contoh Data Wajah Salah Satu Mahasiswa

Tabel 39. Rincian Kapasitas Data Wajah Tersimpan

No	Nim	Nama Peserta	Kapasitas
1	C1555201094	Nurul Suciani	0,64 MB
2	C1555201089	Anton Satriawan	0,98 MB
3	C1555201036	Puteri Diah P.E	1,01 MB
4	C1555201070	Siti Maisarah	0,96 MB
5	C1555201100	Rahmat Hidayat	0,97 MB
6	C1555201020	Raja Sakti T. Dondang	1,10 MB
7	C1555201024	Otong J.	0,99 MB
8	C1555201084	Tita Anggraini	1,00 MB
9	C1555201033	Andreyan Dwiyanto	0,33 MB
10	C1555201104	Ahmad Rifani Ananda	1,10 MB
Total			9,08 MB

Berdasarkan rincian kapasitas data wajah tersimpan pada Tabel 39. aplikasi membutuhkan kapasitas memori kosong pada *smartphone* sebesar 9,08 MB (megabytes) untuk 1 kelas dengan total peserta sebanyak 10 mahasiswa.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Kesimpulan dari tugas akhir ini adalah :

1. Sistem dapat melakukan deteksi dan pengenalan wajah melalui android sehingga kemudian data kehadiran tersimpan di *cloud* database berupa hasil ekstraksi citra dengan perolehan id matakuliaa dan nim mahasiswa
2. Hasil ekstraksi citra akan tersimpan ke kehadiran.
3. Admin dapat mengelola data tahun ajaran, dosen, matakuliah, mahasiswa laporan kehadiran.
4. Hasil pengujian sistem pengenalan wajah yang sudah dibangun memiliki tingkat akurasi 90%. Hal itu ditunjukkan oleh 9 dari 10 mahasiswa berhasil diverifikasi atau dikenali.
5. Hasil pengujian kegunaan sangat memuaskan. Hal itu terlihat dari hasil rata-rata kuisisioner yang bernilai diatas 4.

B. Saran

1. Perlu dibuat suatu mekanisme khusus supaya data wajah mahasiswa yang tersimpan didalam penyimpanan *smartphone* dapat diunggah dan diunduh secara otomatis.
2. Perlu diimplementasikan algoritma yang lebih baik yang dapat mengatasi masalah deteksi wajah di ruang yang kurang mendapatkan pencahayaan yang optimal.

DAFTAR PUSTAKA

- Anita, T. Kurniawati, Afrilyan Ruli Dwi Rama. 2015. *Aplikasi Pengenalan Wajah Menggunakan Metode Eigenface Dengan Bahasa Pemrograman Java*. Institut Teknologi Adhi Tama Surabaya, Surabaya.
- Fahmi, Pasek, Fitri Bimantoro. 2018. *Pengenalan Wajah untuk Sistem Kehadiran Menggunakan Metode Eigenface dan Euclidean Distance*. Universitas Mataram, Nusa Tenggara Barat.
- Indra, Griha Tofik Isa, George Pri Hartawan. 2017. *Perancangan Aplikasi Koperasi Simpan Pinjam Berbasis Web (Studi Kasus Koperasi Mitra Setia)*. Universitas Muhammadiyah Sukabumi, Sukabumi.
- Jumanto. 2018. *Prototype Smart Home Menggunakan Arduino Uno R3 Berbasis Android*. STMIK Palangkaraya, Palangkaraya.
- Rizki, Beni, Yulrio Brianorman. 2015. *Implementasi Pengenalan Wajah dengan Metode Eigenface Pada Sistem Absensi*. Universitas Tanjungpura, Pontianak.
- Siswo, Romi, Raya Sagita. 2014. *Sistem Presensi Berbasis Algoritma Eigenface dengan Metode Principal Component Analysis*. Universitas Sultan Ageng Tirtayasa Cilegon, Cilegon.
- Surya, Darma, Kadek Suar Wibawa. 2017. *Aplikasi Identifikasi Wajah Berbasis Android*. Universitas Udayana, Bali.

LAMPIRAN



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA**

Jl. G. Obos No.114 Telp. 0536-3224593, 3225515 Fax. 0536-3225515 Palangkaraya
surel (email) : humas@stmikplk.ac.id – laman (website) : www.stmikplk.ac.id

SURAT TUGAS PENGUJI TUGAS AKHIR

No. 329 /STMIK-3.C.2/AK/VII/2019

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer
(STMIK) Palangkaraya menugaskan kepada nama- nama berikut :

1. Nama : **Maura Widyarningsih, S.Kom., M.Cs**
NIK : 197502232009002
Sebagai Ketua
2. Nama : **Hotmian Sitohang, M.Kom**
NIK : 198503282008002
Sebagai Sekretaris
3. Nama : **Lili Rusdiana, M.Kom**
NIK : 198707282011007
Sebagai Anggota
4. Nama : **Sulistiyowati, S.Kom., M.Cs**
NIK : 198212162007002
Sebagai Anggota
5. Nama : **Herkules, S.Kom, M.Cs**
NIK : 198510042010106
Sebagai Anggota

Tim Penguji Tugas Akhir Mahasiswa :

- Nama : **M. Dadang Setiawan**
NIM : C1555201095
Hari/ Tanggal Ujian : **Jumat, 19 Juli 2019**
Waktu : **15.00 WIB - Selesai**
Judul Tugas Akhir : **Rancang Bangun Sistem Pengenalan Wajah Untuk Kehadiran Mahasiswa Menggunakan Metode Eigenface Berbasis Android**

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 17 Juli 2019

Ketua Program Studi,



Hotmian Sitohang, M.Kom.
NIK. 198503282008002



SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA

Jl. G. Obos No.114 Telp. 0536-3225515 Fax. 0536-3236933 Palangkaraya
surel (email) : humas@stmikplk.ac.id – laman (website) : www.stmikplk.ac.id




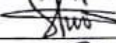






KARTU KEGIATAN KONSULTASI
TUGAS AKHIR

Nama Mahasiswa : Muhammad Dadang Setiawan
NIM : C155201090
Tanggal Persetujuan Judul : 27 Februari 2019
Judul Tugas Akhir : Rancang Bangun Sistem Pengedaran Obat Untuk Kehadiran Mahasiswa Menggunakan Metode E-learning Berbasis Android.

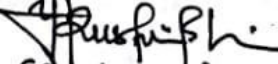
No.	Tanggal Konsultasi		Uraian	Tanda Tangan
	Terima	Kembali		
1	11/2/2019	11/2/2019	Pembahasan judul	Sti
2	22/2/2019	22/2/2019	Latar belakang + permasalahan	Sti
3	25/2/2019	25/2/2019	Kajian Relevan.	Sti
4	2/3/2019	2/3/2019	Acc. Seminar	Sti
		05/03	Perbaiki Batasan masalah dan judul	Sti
			Stop Seminar	
			Lengkapi Argitk sist, UML, Konsep OO	Sti
	27/02/2019		Latihkan min 5 mt.kul.	Sti
			kelanjutnya lanjut ke bab. IV	
			Bab III OK	
		02/03	Perbaiki Perancangan Class design	Sti
		03/03	Program tinggal diuji coba & selubun diujikan dgn mlis lanya.	Sti
			menyusun hasil hasil tugas diuraikan dicatat dan dilaporkan p/ hasil	
			Perbaiki penulisan tabel ngn.	
			Perbaiki kesimpulan + lengkapi daftar pustaka, abstrak daftar isi	Sti
			Lengkapi Daftar pustaka + Cek kesimpulan	Sti
			belum sesuai	
			Acc. Sidang.	Sti
		12/03	Pembahasan lisensi program	Sti
		15/03	Perbaiki kesimpulan dan paparan p-sta ke	Sti
		17/03	stop di uji	Sti

Daftar Peserta Simulasi Kelas
Pengujian Sistem Pengenalan Wajah untuk Kehadiran Mahasiswa

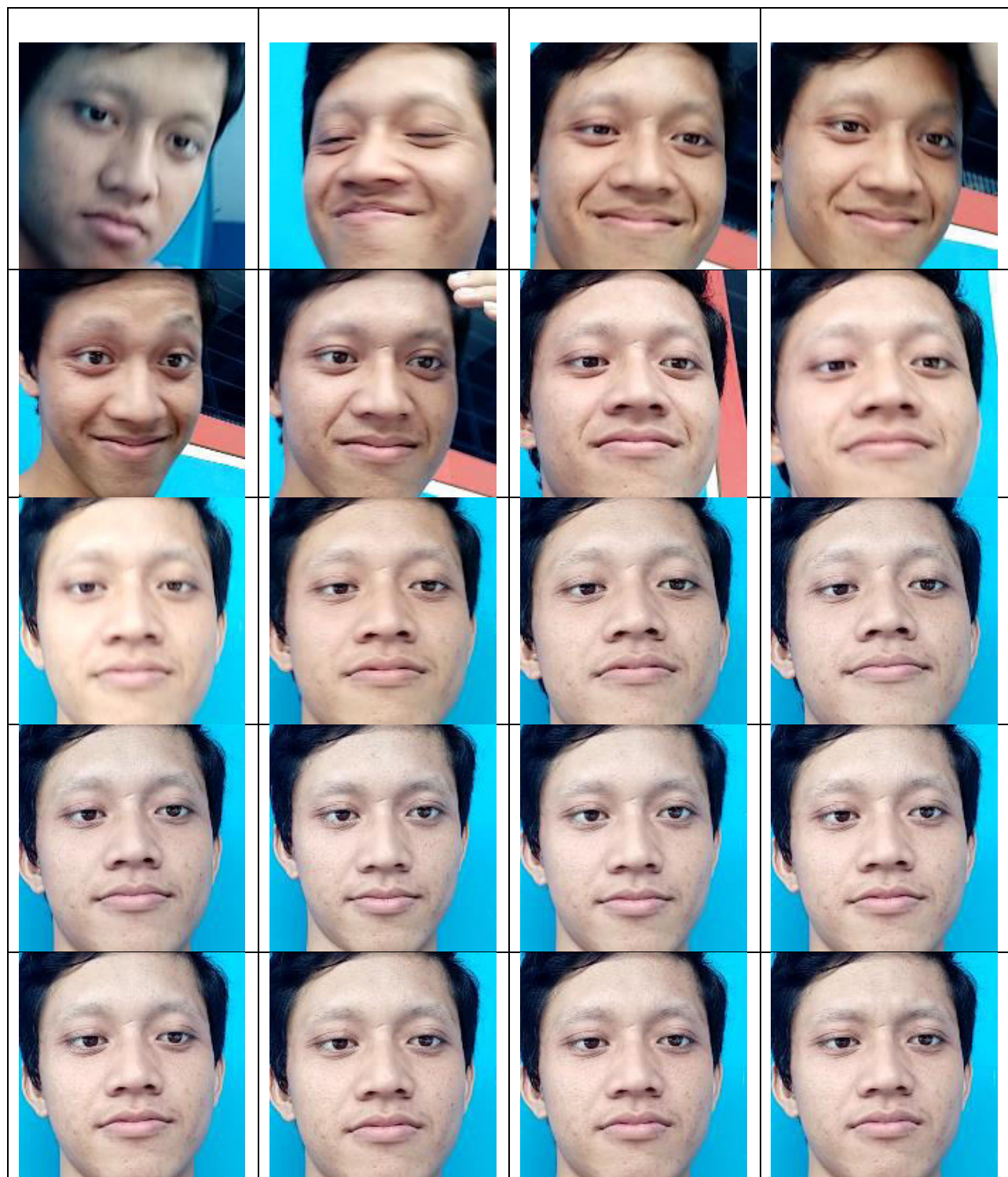
Matakuliah : Logika Informatika
Kelas : A
Hari/Tanggal : Jum'at, 05 Juli 2019

No	NIM	Nama Mahasiswa	TTD	Valid
1	C1555201094	Murul Sugian		✓
2	C1555201089	Anton Satriawan		✓
3	C1555201036	Putri Diah PE		✓
4	C1555201070	Siti Raisaroh		✓
5	C1555201100	Rahmat Hidayat		X
6	C1555201020	Raja Satri T. Dondang		✓
7	C1555201024	Orang-J		✓
8	C1555201084	Tita Angraini		✓
9	C1555201033	Andrian Dwiganto		✓
10	C1555201104	Ahmad Rifan Haris		✓
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				

Dosen Penguji


(Gusfiah)













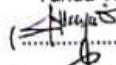
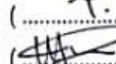
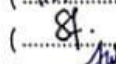
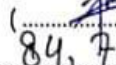
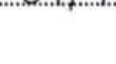
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA

Jl. G. Obos No.114 Telp.0536-3225515 Fax. 0536-3236933 Palangkaraya.
surel (email) : stmikplk@gmail.com – laman (website) : www.stmikplk.ac.id

BERITA ACARA
UJIAN TUGAS AKHIR

Periode (Bulan) :Juli.....Tahun ..2019

1. Hari/Tanggal Ujian : Jumat / 19 Juli 2019
2. Waktu (Jam) : 10.00 WIB sampai denganWIB
3. Nama Mahasiswa : Muhammad Dofang Setiawan
4. Nomor Induk Mahasiswa : C19020095
5. Program Studi : Teknik Informatika
6. Tahun Angkatan : 2015
7. Judul Tugas Akhir : Rancang Bangun Sistem Pengisian Ulang Untuk Kehadiran Mahasiswa Menggunakan Metode Eigenface Berbasis Android
8. Dosen Penguji :

Nama	Nilai	Tanda Tangan
1. Mauro W	=	()
2. Hotmian Siholang	=	()
3. Hil Rudianta	=	()
4. Sulistyowati	=	()
5. Herkues	=	()
9. Hasil Ujian : LULUS / TIDAK LULUS*) NILAI = 84,7.
Dengan Perbaikan/ Tanpa Perbaikan*)
10. Catatan Penting : 1. Lama Perbaikan : 7 hari
2. Jika lebih dari 1 (satu) bulan dikenakan sanksi berupa denda sebesar Rp. 600.000,- (Enam ratus ribu rupiah) per bulan dari tanggal ujian
3. Jika lebih dari 3 (tiga) bulan dari tanggal ujian maka hasil ujian dibatalkan dan wajib mengajukan judul dan pembimbing baru

Palangkaraya,18 - 7.....2019.

Mengetahui :
Ketua Program Studi Teknik Informatika,

Hotmian Siholang, M.Kom.
NIK. 198503282008002

Ketua Penguji,

Mauro Widyadarmasari, S.Kom, M.Cs.
NIK. 197501232009002

Tembusan:

1. Kabag AKMA
 2. Arsip Prodi
 3. Mahasiswa yang bersangkutan
- Dibawa saat konsultasi perbaikan dengan dosen penguji
) Coret yang tidak perlu