



TUGAS AKHIR - KI1502

**RANCANG BANGUN APLIKASI *MOBILE*  
ANDROID SISTEM KEHADIRAN MAHASISWA  
MELALUI PENCOCOKAN TANDA TANGAN  
STUDI KASUS JURUSAN TEKNIK INFORMATIKA  
ITS**

**NOVITA RETNO PUJI LESTARI  
NRP 5113100016**

**Dosen Pembimbing  
Sarwosri, S.Kom., MT.  
Dwi Sunaryono, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017**









**TUGAS AKHIR - KI1502**

**RANCANG BANGUN APLIKASI *MOBILE* ANDROID  
SISTEM KEHADIRAN MAHASISWA MELALUI  
PENCOCOKAN TANDA TANGAN STUDI KASUS  
JURUSAN TEKNIK INFORMATIKA ITS**

**NOVITA RETNO PUJI LESTARI  
NRP 5113100016**

**Dosen Pembimbing  
Sarwosri, S.Kom., MT.  
Dwi Sunaryono, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017**

*[Halaman ini sengaja dikosongkan]*



**UNDERGRADUATE THESES - KI1502**

**DEVELOPMENT OF STUDENT ATTENDANCE  
SYSTEM THROUGH SIGNATURE VERIFICATION IN  
ANDROID-MOBILE BASED APPLICATION FOR  
DEPARTMENT OF INFORMATICS ITS CASE STUDY**

**NOVITA RETNO PUJI LESTARI  
NRP 5113100016**

**Supervisors  
Sarwosri, S.Kom., MT.  
Dwi Sunaryono, S.Kom., M.Kom.**

**DEPARTMENT OF INFORMATICS  
Faculty of Information Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017**

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN

### RANCANG BANGUN APLIKASI *MOBILE* ANDROID SISTEM KEHADIRAN MAHASISWA MELALUI PENCOCOKAN TANDA TANGAN STUDI KASUS JURUSAN TEKNIK INFORMATIKA ITS TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Rekayasa Perangkat Lunak  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh  
**NOVITA RETNO PUJI LESTARI**  
**NRP: 5113 100 016**

Disetujui oleh Dosen Pembimbing Tugas Akhir

Sarwosri, S.Kom., MT  
(NIP 197608092001122001)

Dwi Sunaryono, S.Kom., M.Kom.  
(NIP 197205281997021001)



**SURABAYA**  
**JULI, 2017**

*[Halaman ini sengaja dikosongkan]*

**RANCANG BANGUN APLIKASI *MOBILE*  
ANDROID SISTEM KEHADIRAN MAHASISWA  
MELALUI PENCOCOKAN TANDA TANGAN  
STUDI KASUS JURUSAN TEKNIK  
INFORMATIKA ITS**

**Nama Mahasiswa : Novita Retno Puji Lestari**  
**NRP : 5113 100 016**  
**Jurusan : Teknik Informatika FTIf-ITS**  
**Dosen Pembimbing 1 : Sarwosri, S.Kom., MT.**  
**Dosen Pembimbing 2 : Dwi Sunaryono, S.Kom., M.Kom.**

**Abstrak**

*Sistem kehadiran mahasiswa merupakan proses pencatatan kehadiran mahasiswa di kelas. Dengan adanya sistem kehadiran ini maka dapat membuktikan bahwa mahasiswa tersebut telah berada di dalam kelas. Pencatatan kehadiran mahasiswa dapat dilakukan dengan berbagai cara. Di Jurusan Teknik Informatika ITS, sistem kehadiran mahasiswa dilakukan secara manual yaitu melakukan tanda tangan di kertas daftar kehadiran kelas. Hal ini menimbulkan celah kelemahan yaitu terjadi kecurangan dalam pencatatan kehadiran atau disebut titip absen. Titip absen merupakan kejadian di mana mahasiswa tercatat hadir di daftar hadir kertas tetapi kenyataannya mahasiswa tersebut tidak hadir di kelas. Oleh karena itu, diperlukan suatu sistem yang dapat digunakan untuk mengatasi kecurangan tersebut.*

*Tujuan dibangunnya aplikasi mobile ini sebagai tugas akhir yaitu membangun sistem pencatatan kehadiran online dengan pencocokan tanda tangan. Pencocokan tanda tangan dilakukan dengan cara mengambil fitur-fitur dari gambar tanda tangan kemudian menghitung jarak euclidean distance untuk mengetahui kecocokan tanda tangan.*

***Kata kunci: euclidean distanc, mobile application, normalisasi, pencocokan tanda tangan, sistem kehadiran***

# **DEVELOPMENT OF STUDENT ATTENDANCE SYSTEM THROUGH SIGNATURE VERIFICATION IN ANDROID-MOBILE BASED APPLICATION FOR DEPARTMENT OF INFORMATICS ITS CASE STUDY**

**Student Name** : Novita Retno Puji Lestari  
**NRP** : 5113 100 016  
**Major** : Informatics Department FTIf – ITS  
**Advisor I** : Sarwosri, S.Kom., MT.  
**Advisor II** : Dwi Sunaryono, S.Kom., M.Kom.

## **Abstract**

*Student attendance system is a process of registration of student attendance in class. This system can prove that the student is already present in class. This can be done in many ways. In Informatics ITS, the attendance system done manually which student needs to give their signature in presence paper. This way have a big hole in it, that in the way of make their friend fake the signature for them. So that the student who can be not present in class can be registred to attend the class. Therefore, there is a need of new system that can mitigate the deceitfulness in the old manual student attendance system in class.*

*The purpose of this mobile application built as the undergraduate thesis is to implement online student attendance system with signature verification. Signature verification can be done with extracting features of the signature images and then calculate euclidean distance to find out the matchness from each signature.*

**Keywords:** *attendance system, euclidean distance, mobile application, normalization, signature matching*

**[Halaman ini sengaja dikosongkan]**

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah rabbil'alam, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“RANCANG BANGUN APLIKASI *MOBILE ANDROID* SISTEM KEHADIRAN MAHASISWA MELALUI PENCOCOKAN TANDA TANGAN STUDI KASUS JURUSAN TEKNIK INFORMATIKA”**.

Pengerjaan tugas akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan tugas akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS.

Selesaiannya tugas akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak, sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Ibunda dan Ayahanda, yang selalu mendoakan penulis dan mendukung setiap pilihan yang penulis ambil.
3. Ibu Sarwosri, S.Kom., MT. selaku pembimbing I yang selalu memberikan memotivasi dan membimbing penulis selama pengerjaan tugas akhir.
4. Bapak Dwi Sunaryono, S.Kom., M.Kom. selaku pembimbing II yang selama ini telah memotivasi dan membimbing penulis selama pengerjaan tugas akhir.
5. Ibu Dr. Chastine Fatichah, S.Kom.,M.Kom. selaku dosen KCV yang telah membantu dan membimbing penulis selama pengerjaan tugas akhir.
6. Bapak Dr.Eng Darlis Herumurti, S.Kom.,M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Bapak Dr. Radityo

Anggoro, S.Kom.,M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah banyak memberikan ilmu kepada penulis.

7. Setiyo Adiwicaksono yang telah memberikan saran, ilmu dan motivasi kepada penulis
8. Teman-teman kelompok Tugas Akhir Sikemas (Ronald dan Bana) yang telah membantu dan berjuang bersama dalam mengerjakan tugas akhir ini.
9. Ilham Gurat A selaku pembimbing 3 yang telah membantu, membagi ilmu dan mengingatkan penulis untuk selalu ingat akan progress tugas akhir ini.
10. Teman-teman seperjuangan (Ayu, Devira, Lophita) yang telah menjadi keluarga selama penulis menimba ilmu di Teknik Informatika ITS.
11. Teman-teman Administrator Laboratorium RPL yang telah menjadi keluarga selama penulis menimba ilmu di Teknik Informatika ITS.
12. Teman-teman Pengurus Harian HMTC Optimasi ITS 2015 yang telah menjadi keluarga dan saling menyemangati selama pengerjaan tugas akhir ini.
13. Teman-teman Staff Departemen Dalam Negeri HMTC Optimasi 2015/2016 yang selalu mengingatkan penulis untuk menyelesaikan tugas akhir ini.
14. Teman-teman angkatan 2013 yang telah membantu, berbagi ilmu, menjaga kebersamaan, dan memberi motivasi kepada penulis, serta adik-adik angkatan 2014 dan 2015 yang membuat penulis untuk selalu belajar.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan, sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juli 2017  
Novita Retno Puji Lestari

## DAFTAR ISI

<b>LEMBAR PENGESAHAN.....</b>	<b>v</b>
<b>Abstrak.....</b>	<b>vii</b>
<b>Abstract.....</b>	<b>ix</b>
<b>KATA PENGANTAR.....</b>	<b>xi</b>
<b>DAFTAR ISI.....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xvii</b>
<b>DAFTAR TABEL.....</b>	<b>xix</b>
<b>DAFTAR KODE SUMBER .....</b>	<b>xxi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	3
1.5 Manfaat .....	3
1.6 Metodologi Pembuatan Tugas Akhir .....	3
1.7 Sistematika Penulisan Laporan Tugas Akhir .....	6
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>9</b>
2.1 Penelitian Terkait.....	9
2.2 Teknik Informatika ITS .....	10
2.3 Sistem Kehadiran.....	12
2.4 Tanda Tangan .....	13
2.5 Handwriting Recognition.....	13
2.6 Image Preprocessing .....	14
2.6.1 RGB to Grayscale Conversion.....	14

2.6.2	Cropping.....	15
2.7	Ekstraksi Fitur .....	16
2.7.1	Histogram Horisontal dan Vertikal Maksimal .....	16
2.7.2	Aspek Rasio .....	16
2.7.3	Area.....	17
2.7.4	Skewness .....	17
2.8	Normalisasi Skala.....	17
2.9	Euclidian Distance .....	18
2.10	Android .....	18
2.11	API .....	19
2.12	JSON .....	19
<b>BAB III ANALISIS DAN PERANCANGAN SISTEM ....</b>		<b>21</b>
3.1	Analisis.....	21
3.1.1	Analisis Permasalahan.....	21
3.1.2	Deskripsi Umum Sistem.....	22
3.1.3	Spesifikasi Kebutuhan Perangkat Lunak.....	23
3.2	Perancangan .....	35
3.2.1	Perancangan Arsitektur Perangkat Lunak .....	35
3.2.2	Perancangan Data.....	36
3.2.3	Perancangan Proses .....	36
3.2.4	Perancangan Antarmuka Pengguna.....	43
<b>BAB IV IMPLEMENTASI.....</b>		<b>53</b>
4.1	Lingkungan Implementasi Perangkat Lunak.....	53
4.2	Implementasi Antarmuka Pengguna .....	54
4.2.1	Implementasi Halaman Antarmuka Kelola Verifikasi Kehadiran dengan Tanda Tangan.....	54

4.2.2 Implementasi Halaman Antarmuka Kelola Dataset Tanda Tangan .....	54
4.2.3 Implementasi Halaman Antarmuka Tambah Dataset Tanda Tangan .....	55
4.2.4 Implementasi Halaman Antarmuka Verifikasi Kehadiran dengan Tanda Tangan .....	56
4.2.5 Implementasi Halaman Antarmuka Hasil Verifikasi Kehadiran.....	57
4.2.6 Implementasi Kasus Penggunaan .....	57
4.2.7 Implementasi Kasus Penggunaan Menambah Dataset Tanda Tangan .....	58
4.2.8 Implementasi Kasus Penggunaan Mengunggah Dataset Tanda Tangan.....	59
4.2.9 Implementasi Kasus Penggunaan Mengunduh Dataset Tanda Tangan.....	61
4.2.10 Implementasi Kasus Penggunaan Verifikasi Kehadiran dengan Tanda Tangan .....	64
<b>BAB V PENGUJIAN DAN EVALUASI .....</b>	<b>75</b>
5.1 Lingkungan Pengujian .....	75
5.2 Pengujian Fungsionalitas .....	76
5.2.1 Pengujian Fungsionalitas Menambah Dataset .....	76
5.2.2 Pengujian Fungsionalitas Melakukan Verifikasi Kehadiran.....	80
5.3 Pengujian Kebergunaan .....	82
5.3.1 Pengujian Kebergunaan Skenario Kelas A .....	87
5.3.2 Pengujian Kebergunaan Skenario Kelas B .....	88
5.3.3 Pengujian Kebergunaan Skenario Kelas C .....	88
5.3.4 Pengujian Kebergunaan Skenario Kelas D .....	89

5.3.5	Pengujian Kebergunaan Skenario Kelas E.....	90
5.4	Pengujian Waktu Eksekusi.....	92
5.5	Pengujian Keakuratan .....	95
5.6	Analisa Threshold untuk Euclidean Distance .....	96
5.6.1	Analisa Pengguna 1 .....	96
5.6.2	Analisa Pengguna 2.....	97
5.6.3	Analisa Pengguna 3 .....	97
5.6.4	Analisa Pengguna 4.....	98
5.6.5	Analisa Pengguna 5.....	98
5.7	Evaluasi Pengujian Fungsionalitas.....	99
5.8	Evaluasi Pengujian Kebergunaan.....	100
5.9	Evaluasi Pengujian Waktu Eksekusi .....	100
<b>BAB VI</b>	<b>KESIMPULAN DAN SARAN .....</b>	<b>101</b>
6.1	Kesimpulan .....	101
6.2	Saran.....	101
<b>DAFTAR PUSTAKA</b>	<b>.....</b>	<b>103</b>
<b>LAMPIRAN</b>	<b>.....</b>	<b>105</b>
<b>BIODATA PENULIS</b>	<b>.....</b>	<b>109</b>

## DAFTAR GAMBAR

Gambar 3.1 Arsitektur Sistem.....	22
Gambar 3.2 Diagram Kasus Penggunaan.....	25
Gambar 3.3 Diagram Aktivitas UC-01 .....	28
Gambar 3.4 Diagram Sekuens UC-01 .....	29
Gambar 3.5 Diagram Kolaborasi UC-01.....	30
Gambar 3.6 Diagram Aktivitas UC-02 .....	32
Gambar 3.7 Diagram Sekuens UC-02.....	33
Gambar 3.8 Diagram Kolaborasi UC-02.....	34
Gambar 3.9 Perancangan Arsitektur Sistem.....	35
Gambar 3.10 Contoh Data Tanda Tangan.....	36
Gambar 3.11 Diagram Alur Penggunaan Aplikasi Pencocokan Tanda Tangan.....	38
Gambar 3.12 Diagram Alur Kelola Data Tanda Tangan.....	39
Gambar 3.13 Diagram Blok Proses Aplikasi Pencocokan Tanda Tangan.....	40
Gambar 3.14 Gambar Pseudocode Normalisasi Skala (Min Max) .....	41
Gambar 3.15 Gambar Pseudocode Euclidean Distance .....	42
Gambar 3.16 Rancangan Antarmuka Halaman Kelola Verifikasi Kehadiran dengan Tanda Tangan.....	44
Gambar 3.17 Rancangan Antarmuka Halaman Kelola Dataset Tanda Tangan.....	46
Gambar 3.18 Rancangan Antarmuka Halaman Dialog Dataset Tanda Tangan.....	48
Gambar 3.19 Rancangan Antarmuka Halaman Dialog Verifikasi Kehadiran dengan Tanda Tangan.....	50
Gambar 3.20 Rancangan Antarmuka Halaman Hasil Verifikasi Kehadiran .....	50
Gambar 3.21 Diagram Kelas .....	52
Gambar 4.1 Implementasi Halaman Antarmuka Kelola Verifikasi Kehadiran dengan Tanda Tangan.....	55
Gambar 4.2 Implementasi Halaman Antarmuka Kelola Dataset Tanda Tangan.....	55

Gambar 4.3 Implementasi Halaman Antarmuka Tambah Dataset Tanda Tangan .....	56
Gambar 4.4 Implementasi Halaman Antarmuka Verifikasi Kehadiran dengan Tanda Tangan .....	56
Gambar 4.5 Implementasi Halaman Antarmuka Hasil Verifikasi Kehadiran .....	57
Gambar 5.1 Halaman Kelola Dataset Tanda Tangan .....	78
Gambar 5.2 Halaman Dialog Kanvas .....	78
Gambar 5.3 Tanda Tangan Penguji .....	78
Gambar 5.4 Alert Dialog Berhasil Simpan Dataset.....	78
Gambar 5.5 Tanda Tangan pada Kanvas .....	79
Gambar 5.6 Hapus Tanda Tangan pada Kanvas.....	79
Gambar 5.7 Menampilkan Informasi Jumlah Dataset .....	79
Gambar 5.8 Halaman Dialog Kanvas Tanda Tangan .....	81
Gambar 5.9 Tanda Tangan Verifikasi Kehadiran.....	81
Gambar 5.10 Halaman Hasil Verifikasi Tanda Tangan.....	82

## DAFTAR TABEL

Tabel 2.1 Kode Warna RGB .....	15
Tabel 3.1 Kebutuhan Fungsional .....	24
Tabel 3.2 Kasus Penggunaan .....	25
Tabel 3.3 Spesifikasi Kasus Penggunaan UC-01 .....	26
Tabel 3.4 Spesifikasi Kasus Penggunaan UC-02 .....	31
Tabel 3.5 Atribut Antarmuka Halaman Kelola Verifikasi Kehadiran dengan Tanda Tangan.....	43
Tabel 3.6 Atribut Antarmuka Halaman Kelola Dataset Tanda Tangan.....	45
Tabel 3.7 Atribut Antarmuka Halaman Dialog Tambah Dataset .....	47
Tabel 3.8 Atribut Antarmuka Halaman Dialog Verifikasi Kehadiran dengan Tanda Tangan.....	49
Tabel 3.9 Atribut Antarmuka Hasil Verifikasi Kehadiran .....	51
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak.....	53
Tabel 4.2 Penjelasan Kode Sumber 4-1 .....	59
Tabel 4.3 Penjelasan Kode Sumber 4-2 .....	60
Tabel 4.4 Penjelasan Kode Sumber 4-3 .....	61
Tabel 4.5 Penjelasan Kode Sumber 4-4 .....	62
Tabel 4.6 Penjelasan Kode Sumber 4-5 .....	63
Tabel 4.7 Penjelasan Kode Sumber 4-6 .....	64
Tabel 4.8 Penjelasan Kode Sumber 4-7 .....	66
Tabel 4.9 Penjelasan Kode Sumber 4-8 .....	67
Tabel 4.10 Penjelasan Kode Sumber 4-9 .....	68
Tabel 4.11 Penjelasan Kode Sumber 4-10 .....	69
Tabel 4.12 Penjelasan Kode Sumber 4-11 .....	69
Tabel 4.13 Penjelasan Kode Sumber 4-12 .....	70
Tabel 4.14 Penjelasan Kode Sumber 4-13 .....	72
Tabel 4.15 Penjelasan Kode Sumber 4-14 .....	72
Tabel 5.1 Lingkungan Pengujian Fungsionalitas Perangkat Lunak.....	75
Tabel 5.2 Skenario 1 Pengujian Fungsionalitas Menambah Dataset.....	76

Tabel 5.3 Skenario 1 Pengujian Fungsionalitas Melakukan Verifikasi Kehadiran .....	80
Tabel 5.4 Kuesioner Pengujian Kebergunaan .....	83
Tabel 5.5 Daftar Penguji Aplikasi .....	85
Tabel 5.6 Tabel Hasil Penilaian Kuesioner Kelas A .....	87
Tabel 5.7 Tabel Hasil Penilaian Kuesioner Kelas B.....	88
Tabel 5.8 Tabel Hasil Penilaian Kuesioner Kelas C.....	89
Tabel 5.9 Tabel Hasil Penilaian Kuesioner Kelas D .....	89
Tabel 5.10 Tabel Hasil Penilaian Kuesioner Kelas E.....	90
Tabel 5.11 Tabel Rekapitulasi Kuesioner.....	91
Tabel 5.12 Hasil Pengujian Waktu Eksekusi.....	93
Tabel 5.13 Tabel Pengujian Keakuratan.....	96
Tabel 5.14 Hasil Perhitungan Jarak Euclidean Pengguna 1 ...	97
Tabel 5.15 Hasil Perhitungan Jarak Euclidean Pengguna 2 ...	97
Tabel 5.16 Hasil Perhitungan Jarak Euclidean Pengguna 3 ...	98
Tabel 5.17 Hasil Perhitungan Jarak Euclidean Pengguna 4 ...	98
Tabel 5.18 Hasil Perhitungan Jarak Euclidean Pengguna 5 ...	99

## DAFTAR KODE SUMBER

Kode Sumber 4.1 Menyimpan Dataset Tanda Tangan.....	58
Kode Sumber 4.2 Mendapatkan Arraylist Dataset.....	59
Kode Sumber 4.3 Mengunggah Dataset Tanda Tangan.....	60
Kode Sumber 4.4 Mengunduh Dataset Tanda Tangan.....	62
Kode Sumber 4.5 Menyimpan Hasil Unduh Dataset .....	64
Kode Sumber 4.6 Melakukan Cropping Gambar Tanda Tangan .....	66
Kode Sumber 4.7 Melakukan Grayscale Gambar Tanda Tangan .....	67
Kode Sumber 4.8 Mencari Histogram Harisontal Maksimum .....	67
Kode Sumber 4.9 Mencari Aspek Rasio .....	68
Kode Sumber 4.10 Mencari Jumlah Pikel .....	68
Kode Sumber 4.11 Skewness .....	70
Kode Sumber 4.12 Melakukan Normalisasi Fitur.....	70
Kode Sumber 4.13 Mencari Euclidean Distance.....	71
Kode Sumber 4.14 Mengirimkan Status Verifikasi ke Web Service.....	73

*[Halaman ini sengaja dikosongkan]*

# BAB I

## PENDAHULUAN

Bab ini menjelaskan garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan, batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

### 1.1 Latar Belakang

Sistem kehadiran mahasiswa merupakan proses pencatatan kehadiran mahasiswa di kelas. Dengan adanya sistem kehadiran ini maka dapat membuktikan bahwa mahasiswa tersebut telah berada di dalam kelas. Pencatatan kehadiran mahasiswa dapat dilakukan dengan berbagai cara. Di Jurusan Teknik Informatika ITS, sistem kehadiran mahasiswa dilakukan secara manual yaitu melakukan tanda tangan di kertas daftar kehadiran kelas. Hal ini menimbulkan celah kelemahan yaitu terjadi kecurangan dalam pencatatan kehadiran atau disebut *titip absen*. *Titip absen* merupakan kejadian di mana mahasiswa tercatat hadir di daftar hadir kertas tetapi kenyataannya mahasiswa tersebut tidak hadir di kelas. Dengan menggunakan sistem kehadiran manual, mahasiswa lain 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*) dan internet. Hal ini diikuti dengan bertambahnya pengguna *smartphone*. Di Indonesia sendiri tercatat sebagai negara di Asia tenggara yang penduduk terbanyak menggunakan Android yaitu 41 juta pengguna atau 94%, sedangkan pengguna iOS hanya 2,8 juta pengguna atau 6% [1].

Akses dan pengambilan informasi juga semakin mudah dan cepat diakses melalui penggunaan perangkat bergerak dan web. Perkembangan ini juga mempengaruhi data sebagai validasi menggantikan kode sandi (*password*). Validasi merupakan sebuah

proses yang wajib ada dimana diperlukan sebuah kebenaran data/informasi, salah satunya yaitu dalam sistem kehadiran. Proses validasi sudah banyak diterapkan seiring dengan perkembangan teknologi informasi misalnya sistem kehadiran dengan menggunakan alat *fingerprint*, sistem kehadiran menggunakan deteksi suara, dan lain-lain. Namun kelemahan dari penerapan yang sudah ada adalah kurangnya fleksibilitas dari penerapan yang ada. 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 penggunaannya. Pengguna dari aplikasi ini adalah mahasiswa Teknik Informatika ITS. Proses pencatatan kehadiran pada aplikasi ini akan mengimplementasikan pencocokan tanda tangan ke dalam aplikasi *mobile*. Sistem kehadiran dengan pencocokan tanda tangan.

Hasil yang diharapkan dari pengerjaan tugas akhir ini adalah berupa aplikasi *mobile* yang menggantikan proses pencatatan kehadiran mahasiswa dari sistem manual melalui tanda tangan pada kertas daftar hadir kelas menjadi sistem *online* dengan metode pencocokan tanda tangan pada aplikasi *mobile*.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini, yaitu sebagai berikut:

1. Bagaimana cara membangun aplikasi sistem kehadiran mahasiswa melalui pencocokan tanda tangan?
2. Bagaimana cara menggunakan normalisasi skala dalam pencocokan gambar tanda tangan?
3. Bagaimana cara menggunakan *euclidean distance* untuk pencocokan gambar tanda tangan?

### 1.3 Batasan Masalah

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

1. Input dari perangkat lunak berupa gambar tanda tangan yang akan diuji kecocokannya dengan 5 gambar tanda tangan yang telah disimpan di dalam *database*
2. Dipastikan pemilik tanda tangan yang melakukan tanda tangan
3. Aplikasi *mobile* dapat berjalan pada perangkat *mobile* dengan versi minimal Android 4.3 (Jellybean / API 18) berkapasitas RAM 2GB.

### 1.4 Tujuan

Tugas akhir ini mempunyai beberapa tujuan, yaitu

1. Merancang dan membangun aplikasi sistem kehadiran mahasiswa melalui pencocokan tanda tangan.
2. Menerapkan normalisasi skala untuk dalam pencocokan tanda tangan.
3. Menerapkan *euclidean distance* dalam pencocokan tanda tangan.

### 1.5 Manfaat

Manfaat dari pembuatan tugas akhir ini adalah untuk memperbaharui sistem kehadiran mahasiswa dari sistem manual melalui tanda tangan pada kertas daftar hadir kelas menjadi sistem *online* dengan menggunakan metode pencocokan tanda tangan pada aplikasi Android. Dengan demikian, diperolehnya data kehadiran mahasiswa yang benar, jelas dan terperinci sehingga mempermudah pemantauan dari tata usaha jurusan, dosen maupun orang tua mahasiswa terhadap kehadiran mahasiswa.

### 1.6 Metodologi Pembuatan Tugas Akhir

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Penyusunan proposal tugas akhir.

Proposal Tugas Akhir ini berisi tentang perencanaan “Rancang Bangun Aplikasi *Mobile* Android Sistem Kehadiran Mahasiswa Melalui Pencocokan Tanda Tangan Studi Kasus Teknik Informatika ITS”.

Proposal Tugas Akhir ini terdiri dari deskripsi pendahuluan yang menjabarkan latar belakang dan rumusan masalah yang mendasari dibangunnya aplikasi ini, batasan masalah dalam pembangunan aplikasi ini, serta tujuan dan manfaat yang diharapkan dapat dicapai dengan dibangunnya aplikasi ini. Selain itu, pada proposal Tugas Akhir ini juga terdapat tinjauan pustaka yang menjelaskan teori-teori yang menjadi dasar pembuatan tugas akhir ini, Ringkasan isi tugas akhir yang menggambarkan secara umum aplikasi yang akan dibangun dan algoritma yang digunakan, serta bagian metodologi dari penyusunan proposal tugas akhir ini.

2. Studi literatur

Studi literatur yang dilakukan dalam pengerjaan Tugas Akhir ini adalah mengenai perancangan aplikasia Android untuk pencocokan tanda tangan kehadiran mahasiswa. Studi literatur diambil dari buku, internet, maupun materi mata kuliah yang berhubungan dengan metode yang digunakan. Sehingga, studi literatur ini dapat diterapkan pada perancangan aplikasi sistem kehadiran tanda tangan kehadiran mahasiswa dengan benar.

3. Analisis dan desain perangkat lunak

Analisis kebutuhan dan perancangan sistem dilakukan untuk merumuskan solusi yang tepat dalam pembuatan aplikasi serta kemungkinan yang dapat dilakukan untuk mengimplementasikan rancangan tersebut. Tahap desain meliputi arsitektur perangkat lunak yang digunakan, desain basis data, desain antarmuka, serta diagram-diagram yang mendukung pendeksripsian sistem aplikasi.

#### 4. Implementasi perangkat lunak

Aplikasi ini diimplementasikan dengan menggunakan kakas bantu:

1. Sistem operasi yang dipakai adalah Android dengan spesifikasi minimal Android 4.0 (Ice Cream Sandwich).
2. Bahasa pemrograman yang digunakan yaitu Java dan XML.
3. IDE yang digunakan yaitu Android Studio.
4. Library yang digunakan adalah OpenCV.

#### 5. Pengujian dan evaluasi

Pengujian dan evaluasi aplikasi perangkat lunak hasil dari Tugas Akhir ini akan diujicobakan pada mahasiswa Teknik Informatika ITS. Sebelum pengujian aplikasi kehadiran mahasiswa akan *diinstall* pada *smartphone* serta pengumpulan gambar tanda tangan mahasiswa sebagai data sampel terlebih dahulu ke dalam *database* aplikasi kehadiran mahasiswa. Data sampel ini digunakan untuk pencocokan gambar tanda tangan saat mahasiswa melakukan validasi kehadiran melalui pencocokan tanda tangan.

#### 6. Penyusunan buku tugas akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

##### 1. Pendahuluan

- a. Latar Belakang
- b. Rumusan Masalah
- c. Batasan Masalah
- d. Tujuan
- e. Manfaat
- f. Metodologi Pembuatan Tugas Akhir
- g. Sistematika Penulisan Laporan Tugas Akhir

2. Tinjauan Pustaka
3. Analisis dan Perancangan Sistem
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

## **1.7 Sistematika Penulisan Laporan Tugas Akhir**

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini.

### **Bab I Pendahuluan**

Bab yang berisi mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu metodologi yang digunakan dan sistematika penulisan laporan akhir juga merupakan bagian dari bab ini.

### **Bab II Tinjauan Pustaka**

Bab ini berisi penjelasan secara detail mengenai dasar-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**

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode sumber yang digunakan untuk proses implementasi.

**Bab V Pengujian dan Evaluasi**

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

**Bab VI Kesimpulan dan Saran**

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

*[Halaman ini sengaja dikosongkan]*

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini menjelaskan teori-teori yang berkaitan dengan pembangunan aplikasi *mobile* Android sistem kehadiran mahasiswa melalui pencocokan tanda tangan yang diajukan untuk tugas akhir ini. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap perangkat lunak yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

#### **2.1 Penelitian Terkait**

Terdapat penelitian terkait rancang bangun sistem informasi pencatatan kehadiran sebelumnya, yaitu Sistem Informasi Pencatatan Kehadiran Perkuliahan Dosen dan Mahasiswa dengan Mengimplementasikan Teknologi Pemindai Sidik Jari. Penelitian ini merupakan tugas akhir yang telah selesai dikerjakan oleh salah satu mahasiswa angkatan 2011 di Jurusan Teknik Informatika dengan memindai sidik jarinya pada alat pemindai sidik jari. Data sidik jari kemudian diolah dan disimpan pada *server* basis data. Sehingga kehadiran mahasiswa dan dosen dapat tersimpan dan terekap di dalam basis data. Pencatatan dengan menggunakan sidik jari memanfaatkan teknologi *biometric* dimana penggunaannya lebih efektif dan efisien serta menghindari munculnya peluang dalam memanipulasi data pencatatan kehadiran serta dengan dibangunnya sistem informasi pencatatan kehadiran perkuliahan mahasiswa dan dosen lebih memudahkan dalam proses pengelolaannya. Namun sistem pencatatan kehadiran menggunakan sidik jari dirasa kurang fleksibilitas dari penerapan yang ada. Fleksibilitas yang dimaksud adalah teknologi yang digunakan tidak bisa berpindah tempat dan diakses kapanpun serta harganya yang mahal.

Penelitian mengenai verifikasi tanda tangan pernah dilakukan oleh mahasiswa angkatan 2009 di Jurusan Teknik Informatika dengan judul, yaitu Verifikasi Tanda Tangan secara Offline dengan Menggunakan Distance Statistics. Tanda tangan merupakan salah

satu identitas yang unik yang dapat membedakan seseorang. Sehingga tanda tangan dapat digunakan dalam menentukan validasi dari sebuah dokumen fisik maupun non fisik (*scan*, foto atau gambar dari layar sentuh). Di dalam penelitian ini mengenai verifikasi tanda tangan secara *offline* yaitu dengan mengambil data tanda tangan melalui *scan* kemudian diolah dan dilakukan verifikasi kebenarannya.

Fenomena sekarang ini belum terdapat aplikasi kehadiran dengan menggunakan tanda tangan digital. Penggunaan tanda tangan digital telah dikembangkan oleh beberapa *startup* yaitu membuat aplikasi yang digunakan untuk menandatangani dokumen yang terdapat di komputer ataupun *smartphone*. Jika dulu harus menandatangani dokumen fisik, sekarang kamu cukup memasang aplikasi tanda tangan untuk melakukannya secara digital. Aplikasi tersebut antara lain SignEasy, SignDoc Mobile, Sign It!, HelloSign, CudaSign, dan Signature Maker [2]. Selain itu, tanda tangan juga dikembangkan oleh salah satu *startup*, yaitu PrivyID. PrivyID baru-baru ini mengembangkan sebuah aplikasi yang menyimpan identitas pengguna secara digital. PrivyID menyimpan data pribadi dari penggunanya berupa foto KTP, alamat email, nomor telepon, tanda tangan, rekening bank hingga informasi tempat bekerja dan riwayat pendidikan. Dalam pengembangannya, aplikasi ini akan memanfaatkan identitas digital untuk *login*, salah satunya yaitu *login* dengan menggunakan tanda tangan [3].

## 2.2 Teknik Informatika ITS

Jurusan Teknik Informatika yang awalnya tergabung dalam Fakultas Teknologi Industri (FTI). Mulai tahun 2002 menjadi bagian dari Fakultas Teknologi Informasi (FTIf), sesuai dengan Keputusan Menteri Pendidikan nasional RI Nomor: 109/O/2002, tanggal 2 Juli 2002 tentang Pendirian Fakultas Teknologi Informasi Pada Institut Teknologi Sepuluh Nopember.

Untuk peringkat akreditasi, Jurusan Teknik Informatika Program Studi S1 dinyatakan terakreditasi dengan nilai A berdasarkan Surat Keputusan Badan Akreditasi Nasional

Perguruan Tinggi (BAN-PT) Nomor 003/BAN-PT/Ak-X/S1/V/2006, tanggal 18 Mei 2006.

Saat ini, staf pengajar jurusan teknik informatika sebanyak 48 dosen, diantaranya 6 doktor dan 23 bergelar master. Jumlah karyawan teknik informatika sebanyak 20 orang. Sedangkan mahasiswa di jurusan teknik informatika hingga tahun 2017 ini sebanyak 745 mahasiswa dengan rincian sebanyak 179 mahasiswa angkatan 2016, 178 mahasiswa angkatan 2015, 167 mahasiswa angkatan 2014, 167 mahasiswa angkatan 2013, 27 mahasiswa angkatan 2012, 14 mahasiswa angkatan 2011, 9 mahasiswa angkatan 2010 dan 2 mahasiswa angkatan 2009.

Kurikulum untuk program pendidikan S1 yang sekarang digunakan di Jurusan Teknik informatika merupakan kurikulum baru yang telah diberlakukan mulai tahun akademik 2014/2015 hingga tahun akademik 2018/2019. Dalam kurikulum ini, total Satuan Kredit Semester (SKS) normal yang harus ditempuh oleh mahasiswa adalah 144 SKS. Beban perkuliahan sebesar 144 SKS ini dapat diselesaikan selama delapan semester. Keterangan lebih detil tentang kurikulum, dapat dilihat pada website [if.its.ac.id](http://if.its.ac.id). Total 144 SKS tersebut dikelompokkan menjadi 2 kelompok mata kuliah, yaitu: Mata Kuliah Wajib sejumlah 120 SKS (83%) dan Mata Kuliah Pilihan sejumlah 24 SKS (17%).

Jurusan Teknik Informatika menawarkan delapan bidang keahlian, yaitu komputasi cerdas dan visi (KCV); komputasi berbasis jaringan (KBJ); rekayasa perangkat lunak (RPL); algoritma dan pemrograman (AP); dasar dan terapan komputer (DTK); interaksi, grafika dan seni (IGS); manajemen informasi (MI) dan arsitektur dan jaringan komputer (AJK).

Untuk meningkatkan kenyamanan, mulai tahun 2008 Jurusan Teknik Informatika menempati gedung baru yang berjarak sekitar 500 meter ke arah timur dari gedung lama (berlokasi di dekat Grha Sepuluh Nopember ITS). Terdapat 13 ruang kelas yang semuanya dilengkapi dengan AC, LCD projector, dan *wireless network*. Teknik Informatika juga dilengkapi dengan perpustakaan atau ruang baca teknik informatika (RBTC). RBTC menyediakan

berbagai koleksi akademik, seperti buku, majalah, jurnal akademik, laporan kerja praktik, laporan tugas akhir dan tesis. Selain itu, terdapat 13 laboratorium/ *workshop* yang berfungsi untuk menunjang mahasiswa dalam mengembangkan keilmuannya.[4]

### **2.3 Sistem Kehadiran**

Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*systema*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat [5]. Sedangkan kehadiran menurut KBBI merupakan perihal hadir; adanya (seseorang, sekumpulan orang) pada suatu tempat [6]. Berdasarkan definisi sistem dan kehadiran, dapat disimpulkan bahwa Sistem kehadiran adalah sistem yang digunakan dalam pengambilan data guna mengetahui jumlah kehadiran pada suatu acara. Setiap kegiatan yang membutuhkan informasi mengenai peserta tentu akan melakukan konfirmasi jika telah hadir. Hal ini juga terjadi pada proses kegiatan perkuliahan mahasiswa. Kegunaan sistem kehadiran ini terjadi pada pihak mahasiswa dan pihak dosen maupun kampus. Sistem kehadiran dapat dilakukan dengan berbagai cara, Di Teknik Informatika ITS, Sistem kehadiran mahasiswa dilakukan secara manual yaitu melakukan tanda tangan di kertas daftar hadir mahasiswa. Salah satu kegunaan sistem kehadiran bagi mahasiswa adalah untuk syarat untuk mengikuti ujian mata kuliah, sedangkan bagi pihak dosen atau kampus, sistem kehadiran diperlukan sebagai salah satu bahan evaluasi kepada kepuasan mahasiswa terhadap suatu mata pelajaran dan pembuatan tolak ukur ke depan guna pemberian ilmu yang lebih baik.

## 2.4 Tanda Tangan

Tanda tangan atau bahasa Inggris yaitu *signature*. Kata *signature* berasal dari bahasa Latin: *signare* yang berarti "tanda". Nama lain dari tanda tangan yaitu paraf. Paraf merupakan tulisan tangan atau goresan tinta dari tangan. Paraf kadang memiliki gaya tulisan tertentu dari nama seseorang atau tanda identifikasi lainnya yang ditulis pada dokumen sebagai sebuah bukti dari identitas dan kemauan. Setiap orang memiliki gaya tulisan paraf yang berbeda-beda. Paraf berlaku sebagai segel [7].

Adapun Menurut Yahya harahap (2001: 544), beliau mengemukakan bahwa pengertian dari tanda tangan dan dokumen tertulis lainnya tidak mesti diatas kertas yang kemudian dapat menjadi bukti tertulis, tetapi hal itu hanyalah berlaku bagi negara yang menganut sistem pembuktian terbuka. Sehingga pada jaman sekarang, tanda tangan bisa dilakukan pada dokumen kertas, foto, layar sentuh dan perangkat lainnya.

Fungsi dari tanda tangan adalah untuk dapat memastikan identifikasi atau menentukan kebenaran ciri-ciri dari penandatanganan. Sekaligus pendatangan menjamin keberadaan dari apa yang telah ditandatangani.

## 2.5 Handwriting Recognition

Pengenalan tulisan tangan (*handwriting recognition*) adalah kemampuan komputer untuk menerima dan menafsirkan input tulisan tangan yang dapat dimengerti dari sumber seperti dokumen kertas, foto, layar sentuh dan perangkat lainnya. Dalam melakukan pengenalan tulisan dapat dilakukan dengan beberapa pendekatan. Pendekatan ini memberikan pengenalan kontrol yang lebih atas properti yang digunakan dalam identifikasi tanda tangan. Ekstraksi fitur/ properti yang dapat digunakan untuk mengidentifikasi tanda tangan antara lain:

- Aspek rasio
- Persen *piksel* di atas titik setengah horizontal
- Persen *piksel* ke kanan setengah titik vertikal

- Jumlah coretan
- Rata-rata jarak dari pusat gambar
- Apakah tercermin sumbu y
- Apakah tercermin x sumbu [8].

## 2.6 Image Preprocessing

Dalam ilmu pencitraan, *image preprocessing* atau pengolahan citra adalah pengolahan gambar menggunakan operasi matematika dengan menggunakan bentuk pemrosesan sinyal yang inputnya adalah gambar, serangkaian gambar, atau video, seperti foto atau bingkai video [9]. Tujuan dilakukan *image preprocessing* adalah untuk menghilangkan *noise* sehingga pengambilan informasi dari gambar tanda tangan dapat dilakukan dengan benar. *Image preprocessing* yang dilakukan di dalam tugas akhir adalah melakukan *RGB to grayscale conversion* dan melakukan *cropping* pada gambar tanda tangan.

### 2.6.1 RGB to Grayscale Conversion

*Grayscale* atau *greyscale* gambar digital adalah gambar dimana nilai setiap piksel adalah sampel tunggal, yaitu hanya membawa informasi intensitas. Gambar ini juga dikenal sebagai hitam-putih, dimana pada intensitas terlemah yaitu warna hitam, sedangkan pada intensitas terkuat yaitu warna putih [7]. Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = Red, Green dan Blue).

Sistem RGB digabungkan untuk memperoleh warna tertentu. Misalnya warna putih diperoleh dari hasil gabungan warna merah = 255, hijau = 255, dan biru = 255. Sedangkan warna hitam diperoleh dari hasil gabungan warna merah = 0, hijau = 0, dan biru = 0. Nilai dari setiap primer adalah 0 sampai 255. Pada Tabel 2.1 akan diperlihatkan beberapa hasil penggabungan kode warna RGB.

Di dalam gambar PNG tanda tangan pada tugas akhir ini menyimpan warna putih, hitam dan abu-abu. Sehingga setiap piksel jika *Colour* > 128 akan diubah menjadi 225 (putih),

sedangkan piksel dengan *Colour*  $\leq 128$  maka akan diubah menjadi 0 (hitam).

**Tabel 2.1 Kode Warna RGB**

<i>Colour</i>	<i>Red</i>	<i>Green</i>	<i>Blue</i>
<i>Black</i>	0	0	0
<i>Blue</i>	0	0	255
<i>Green</i>	0	255	0
<i>Cyan (Blue+Green)</i>	0	255	255
<i>Red</i>	255	0	0
<i>Magenta (Red+Blue)</i>	255	0	255
<i>Yellow (Red+Green)</i>	255	255	0
<i>White (Red+Green+Blue)</i>	255	255	255
<i>Gray</i>	128	128	128

### 2.6.2 Cropping

*Cropping* merupakan penghapusan bagian luar dari suatu gambar untuk meningkatkan framing, menekankan subjek atau mengubah aspek rasio [11]. *Cropping* bertujuan untuk mendapatkan image sesuai dengan ukuran tinggi dan lebar tandatangan saja. Hal ini dilakukan untuk menghilangkan perbedaan perhitungan rasio tinggi dan lebar tanda tangan akibat letak penulisan tandatangan.

Pada tugas akhir ini, *cropping* dilakukan dengan tujuan untuk penghitungan aspek rasio pada gambar tanda tangan. *Cropping* pada tugas akhir ini dilakukan dengan cara mencari *piksel* paling atas sebagai lebar maksimal, *piksel* paling bawah sebagai lebar minimal, *piksel* paling kiri sebagai panjang minimal dan *piksel* paling kanan sebagai panjang maksimal. Sehingga didapat panjang gambar baru yaitu panjang maksimal dikurangi

dengan panjang minimal dan lebar gambar baru yaitu lebar maksimal dikurangi lebar minimal.

## 2.7 Ekstraksi Fitur

Ekstraksi fitur adalah proses pengambilan ciri sebuah objek yang dapat menggambarkan karakteristik dari objek tersebut. Tujuan dari tahap ini adalah untuk mengekstrak fitur dari gambar tanda tangan di *database* yang akan dibandingkan dengan fitur gambar asli untuk tujuan verifikasi. Fitur yang digunakan di dalam tugas akhir ini adalah *histogram* horisontal maksimal, aspek rasio, *piksel* area, dan *skewness*.

### 2.7.1 Histogram Horisontal dan Vertikal Maksimal

Informasi penting mengenai konten citra dapat diketahui dengan membuat *histogram* citra. *Histogram* citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas *piksel* dari suatu atau bagian tertentu di dalam citra. Dari sebuah *histogram* dapat diketahui frekuensi kemunculan nisbi dari intensitas pada citra tersebut.

Di dalam mencari nilai *histogram* horisontal dan vertikal maksimal adalah dengan menghitung jumlah *piksel* hitam pada setiap baris/ kolom pada gambar tanda tangan. Nilai *histogram* horisontal dihitung dengan cara mencari nilai *piksel* pada setiap baris gambar tanda tangan. Kemudian baris yang memiliki jumlah maksimum *piksel* warna hitam merupakan *histogram* horisontal maksimum. Sedangkan nilai *histogram* vertikal dihitung dengan cari mencari nilai *piksel* pada setiap kolom gambar tanda tangan. Kemudian kolom yang memiliki jumlah maksimum *piksel* warna hitam merupakan *histogram* vertikal maksimal.

### 2.7.2 Aspek Rasio

Pada tahap ini akan dihitung rasio perbandingan antara tinggi dan lebar gambar tanda tangan. Rasio ini dapat dijadikan ciri karena tinggi dan lebar tanda tangan setiap orang tidak akan sama. Di dalam tugas akhir ini, aspek rasio dihitung dari tinggi dan lebar

tanda tangan yang sudah dilakukan *cropping* pada tahap *preprocessing*. Rumus untuk menghitung aspek rasio adalah lebar tanda tangan dibagi panjang tanda tangan.

### 2.7.3 Area

Pada tahap ini akan dihitung jumlah *piksel* hitam yang terdapat pada image hasil *preprocessing*. Hal ini bertujuan untuk mendapatkan informasi mengenai jumlah *piksel* dari suatu tanda tangan dan juga kepadatan dari tanda tangan tersebut.

### 2.7.4 Skewness

Dalam teori probabilitas dan statistik, *skewness* adalah ukuran asimetri distribusi probabilitas dari variabel random bernilai real tentang mean. Nilai *skewness* bisa positif atau negatif, atau bahkan tidak terdefinisi [12]. Miring negatif menunjukkan bahwa ekor di sisi kiri dari fungsi kepadatan probabilitas lebih panjang atau lebih gemuk daripada sisi kanan. Sebaliknya, condong positif menunjukkan bahwa ekor di sisi kanan lebih panjang atau lebih gemuk dari sisi kiri. Nilai nol berarti bahwa ekor di kedua sisi keseimbangan rata keluar keseluruhan.

## 2.8 Normalisasi Skala

Normalisasi adalah teknik untuk menstandarkan atau menyamakan rentang data sehingga tidak ada satu atribut yang terlalu dominan atas atribut yang lain. Pada tugas akhir ini, normalisasi dilakukan sekali, yaitu normalisasi *region of interest*.

Normalisasi skala adalah teknik penstandaran data pada suatu rentang tertentu, umumnya 0-1. Diketahui nilai maksimal dari data fitur sebagai  $X_{max}$  dan nilai minimal dari data fitur adalah  $X_{min}$ . Data yang akan dinormalisasi disimbolkan dengan  $X$ .

Nilai hasil normalisasi yang dilambangkan dengan huruf  $Y$  didapatkan dari rumus perhitungan normalisasi skala ditunjukkan pada persamaan (2-1).

$$Y = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2-1)$$

## 2.9 Euclidian Distance

Pengukuran kemiripan terhadap hasil ekstraksi fitur dapat dilakukan dengan menggunakan metode *euclidean distance*. Definisi dari *euclidean distance* adalah panjang dari garis lurus yang menghubungkan posisi dua buah objek. Secara umum diketahui bahwa jarak terpendek antara dua titik adalah garis lurus antara kedua titik tersebut seperti pada persamaan

$$d(p, q) = d(q, p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} \quad (2-2)$$

$$= \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (2-3)$$

Di dalam persamaan (2.2),  $p$  merupakan array fitur dari gambar tanda tangan di *database smartphone*, sedangkan  $q$  merupakan array fitur dari gambar saat verifikasi tanda tangan. Sedangkan  $i$  pada persamaan (2.3) merupakan fitur ke  $i$ , sedangkan  $n$  merupakan jumlah fitur yang dimiliki oleh gambar tanda tangan. Hasil pengukuran *euclidean distance* adalah  $d$ .

## 2.10 Android

Android adalah sistem operasi perangkat bergerak yang dikembangkan Google dan berbasis Linux yang dirancang dengan layar sentuh. Pertama kali, sistem operasi Android dikembangkan oleh Android, Inc hingga pada bulan Juli 2005 bergabung dengan Google. Android resmi dirilis pada tanggal 5 November 2007 dan setelah itu Android terus mengalami perkembangan hingga saat ini [13]. Dibandingkan dengan sistem operasi *smartphone* lainnya, Android sudah digunakan di Indonesia dengan pengguna sebanyak 93,6 persen, bahkan di tingkat asia tenggara, pengguna Android di

Indonesia merupakan pengguna Android paling banyak, yaitu sebesar 32,7 persen [2]. Hal ini merupakan salah satu hal yang menjadi pertimbangan penulis dalam membangun aplikasi mobile ini dengan sistem operasi Android.

## 2.11 API

API atau *Application Programming Interface* adalah sekumpulan perintah, fungsi, serta protokol 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, API biasanya digunakan untuk mempermudah pertukaran data dari atau ke dalam *server*[3].

## 2.12 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[4].

*[Halaman ini sengaja dikosongkan]*

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

Bab ini membahas analisis kebutuhan dan rancangan yang akan digunakan untuk membangun perangkat lunak yang diajukan sebagai tugas akhir.

#### **3.1 Analisis**

Tahap analisis dibagi menjadi beberapa bagian, antara lain analisis permasalahan, deskripsi umum sistem, dan spesifikasi kebutuhan perangkat lunak.

##### **3.1.1 Analisis Permasalahan**

Proses pencatatan kehadiran mahasiswa di Teknik Informatika untuk saat ini masih menggunakan cara manual yaitu menggunakan buku absen. Mahasiswa melakukan presensi kehadiran dengan cara tanda tangan di buku absen. Dalam melakukan presensi kehadiran sering kali terjadi kecurangan yaitu *titip absen*. *Titip absen* merupakan kejadian di mana mahasiswa tercatat hadir di daftar hadir kertas tetapi kenyataannya mahasiswa tersebut tidak hadir di kelas. Jika guru tidak mengawasi presensi kehadiran tidak memungkinkan kecurangan tersebut dapat terjadi. Pencatatan kehadiran mahasiswa yang aman dan efisien sangat diperlukan untuk mengatasi kecurangan presensi kehadiran tersebut. Sistem pencatatan kehadiran yang aman dan efisien sejauh ini menggunakan *fingerprint*, menggunakan sidik jari untuk memvalidasi kehadiran. Dengan *fingerprint*, presensi kehadiran lebih cepat dan terhindar dari kecurangan. Namun sistem ini membutuhkan biaya yang tidak murah dikarenakan harga alat yang lumayan mahal serta sistem ini tidak dapat dipindah-pindah sehingga setiap ruang kelas harus memiliki alat tersebut untuk proses pencatatan kehadiran mahasiswa.

Pada Tugas Akhir ini mengembangkan alternatif pada sistem pencatatan kehadiran yang aman, efisien dan menghabiskan biaya yang relatif lebih rendah dibanding dengan pencatatan kehadiran

*fingerprint*. Dengan memanfaatkan perkembangan teknologi informasi, khususnya teknologi perangkat bergerak dan internet, diimplementasikan sebuah aplikasi sistem kehadiran mahasiswa dengan pencocokan tanda tangan. Mahasiswa dapat melakukan tanda tangan kehadiran di perkuliahan pada *smartphone* masing-masing. Sistem pencocokan tanda tangan dilakukan dengan membandingkan gambar tanda tangan mahasiswa saat melakukan presensi dengan gambar tanda tangan sampel yang telah tersimpan di dalam *storage internal smartphone*. Dari tersebut dicari jarak *euclidean* antara masing-masing fitur pada gambar tanda tangan. Fitur yang digunakan adalah *histogram* horizontal maksimal, aspek rasio, *piksel* area dan *skewness*. Dalam menentukan fitur, dilakukan ekstraksi fitur dari gambar tanda tangan. Sebelum melakukan ekstraksi fitur dilakukan *preprocessing* terlebih dahulu untuk menghilangkan *noise*. *Preprocessing* yang digunakan adalah *RGB to grayscale conversion* dan *cropping*. Verifikasi kehadiran dengan tanda tangan berhasil jika tanda tangan yang dilakukan oleh mahasiswa di *smartphone* sesuai dengan gambar tanda tangan yang telah tersimpan di *storage internal smartphone*.

### 3.1.2 Deskripsi Umum Sistem

Arsitektur sistem aplikasi *mobile* ini digambarkan seperti pada Gambar 3.1. Perangkat lunak yang dibangun yaitu pencocokan gambar tanda tangan di *smartphone*. Hak akses yang dikerjakan dalam aplikasi ini hanya 1 hak akses, yaitu Mahasiswa. Mahasiswa dapat melakukan verifikasi kehadiran dengan menggunakan tanda tangan.



**Gambar 3.1** Arsitektur Sistem

Gambar 3.1 menunjukkan masukan, proses, dan keluaran pada fokus pengerjaan tugas akhir ini adalah sebagai berikut.

1. Tahap masukan ditunjukkan pada no 1 dan 4 yaitu mahasiswa memilih validasi kehadiran tanda tangan dan melakukan presensi kehadiran dengan tanda tangan pada kanvas *smartphone*. Selain itu masukan yang dilakukan oleh pengguna dalam aplikasi ini adalah mengelola data sampel tanda tangan. Mengelola data sampel antara lain menambah data sampel, mengunduh data sampel ke *internal storage smartphone* dan mengunggah data sampel ke *server*.
2. Pada tahap proses ditunjukkan pada nomor 2 dan 3, aplikasi *mobile* melakukan proses simpan data sampel, unggah data sampel tanda tangan dari *internal storage smartphone* dan unduh data sampel dari server ke *internal storage smartphone*, serta proses pencocokan tanda tangan.
3. Tahap keluaran ditunjukkan pada nomor 3 yaitu pengiriman status mahasiswa berhasil dalam melakukan presensi kehadiran.

### 3.1.3 Spesifikasi Kebutuhan Perangkat Lunak

Sesuai dengan cakupan perangkat lunak yang telah dijelaskan pada bagian deskripsi umum sistem, dibutuhkan adanya spesifikasi perangkat lunak agar dapat memberikan solusi dari permasalahan yang diberikan dan dapat bekerja dengan baik dalam mengakomodasi kebutuhan. Diharapkan dengan adanya spesifikasi ini dapat menyesuaikan kebutuhan-kebutuhan pengguna. Spesifikasi kebutuhan perangkat lunak adalah penjelasan mengenai kebutuhan sistem yang diinginkan pelanggan atau klien dalam bentuk tulisan. Spesifikasi kebutuhan perangkat lunak pada tugas akhir ini terdiri dari kebutuhan fungsional, kebutuhan non-fungsional, aktor, dan kasus penggunaan. Spesifikasi kebutuhan fungsional dapat dilihat pada Tabel 3.1, tabel kasus penggunaan dapat dilihat pada Tabel 3.2, sedangkan diagram kasus penggunaan yang ada di dalam perangkat lunak ini dapat dilihat pada Gambar 3.2.

### 3.1.3.1 Kebutuhan Fungsional

Tabel 3.1 Kebutuhan Fungsional

No	Kebutuhan Fungsional	Deskripsi
1	Menangani kelola proses presensi kehadiran mahasiswa dengan tanda tangan	Melakukan pendaftaran dataset tanda tangan dari mahasiswa, melakukan unggah dataset ke server, melakukan proses sinkronisasi dataset di database lokal <i>smartphone</i> dengan database server, serta melakukan proses verifikasi presensi kehadiran mahasiswa dengan tanda tangan.

### 3.1.3.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang harus dipenuhi oleh sistem sebagai berikut:

1. Kebutuhan Performa  
Perangkat lunak akan berjalan dengan performa terbaik jika dijalankan di atas spesifikasi minimal.
2. Kebutuhan Operasional  
Sistem hanya dapat dijalankan ketika sudah berada dalam posisi yang benar dalam GPS (posisi *smartphone* sesuai dengan posisi kelas pada perkuliahan aktif), Sistem hanya dapat berjalan pada perangkat bergerak dengan sistem operasi Android.

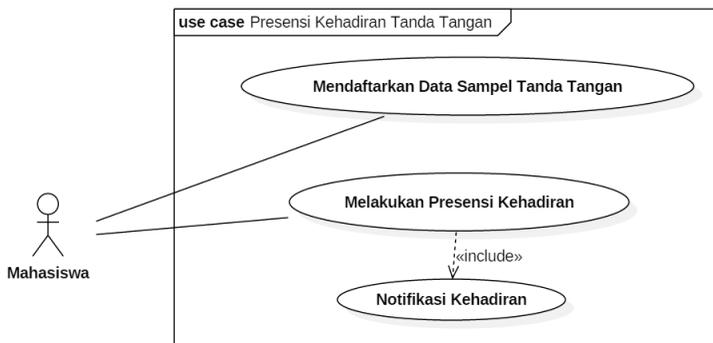
### 3.1.3.3 Aktor

Pengertian pengguna adalah pihak-pihak, baik manusia maupun sistem atau perangkat lain yang terlibat dan berinteraksi secara langsung dengan sistem. Pada perangkat lunak ini terdapat satu pengguna yaitu mahasiswa. Mahasiswa merupakan seseorang yang sedang menempuh pendidikan perguruan tinggi yang melakukan presensi kehadiran di kelas yang di ambil dalam perkuliahan yang sedang ditempuh.

### 3.1.3.4 Kasus Penggunaan

**Tabel 3.2 Kasus Penggunaan**

Kode Kasus Penggunaan	Nama	Aktor
UC-01	Mendaftarkan data sampel tanda tangan	Mahasiswa
UC-02	Melakukan presensi kehadiran	Mahasiswa



**Gambar 3.2 Diagram Kasus Penggunaan**

Berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem, dibuat kasus penggunaan sistem. Kasus penggunaan digambarkan dalam tabel penjelasan kasus penggunaan dan diagram kasus penggunaan. Tabel penjelasan kasus penggunaan dapat dilihat pada Tabel 3.2 dan diagram kasus penggunaan dapat dilihat pada Gambar 3.2.

#### 3.1.3.4.1 Mendaftarkan Dataset Tanda Tangan (UC-01)

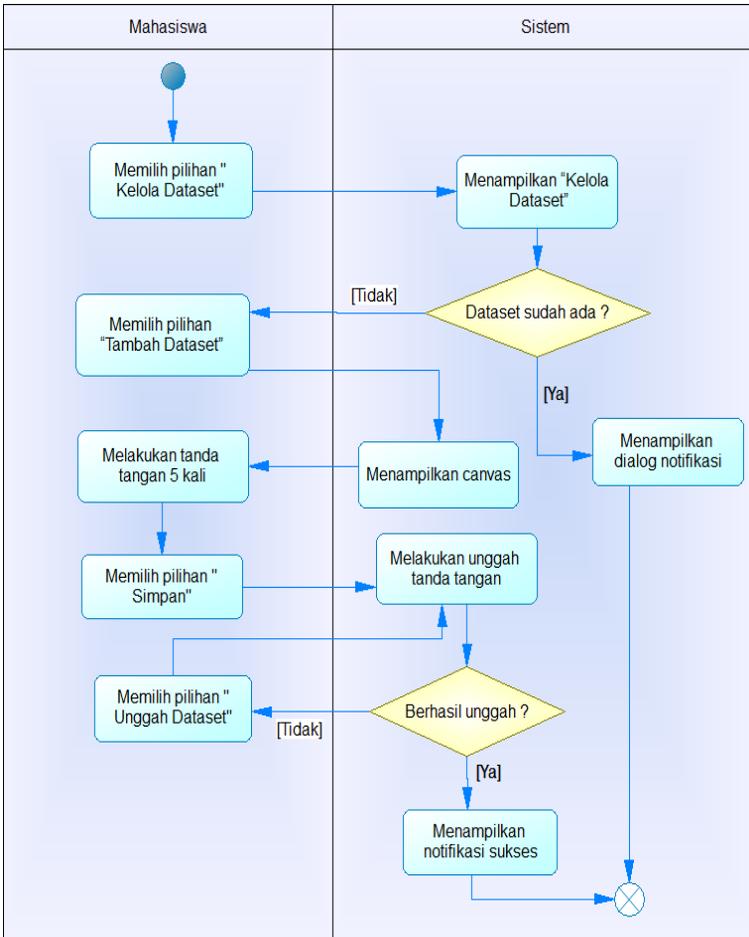
Kasus penggunaan nomor UC-01 ini diakses ketika mahasiswa melakukan pendaftaran akun untuk pertama kali ke dalam sistem kehadiran mahasiswa. Mahasiswa melakukan tanda tangan sebanyak 5 kali untuk mendapatkan data sampel yang akan

disimpan ke dalam *database* sebagai *dataset*. *Dataset* ini berguna sebagai data yang digunakan untuk pencocokan tanda tangan pada saat mahasiswa melakukan verifikasi kehadiran dengan tanda tangan. Spesifikasi, diagram aktivitas, sekuens dan kolaborasi kasus penggunaan ini dapat dilihat pada Tabel 3.3, Gambar 3.3, Gambar 3.4 dan Gambar 3.5.

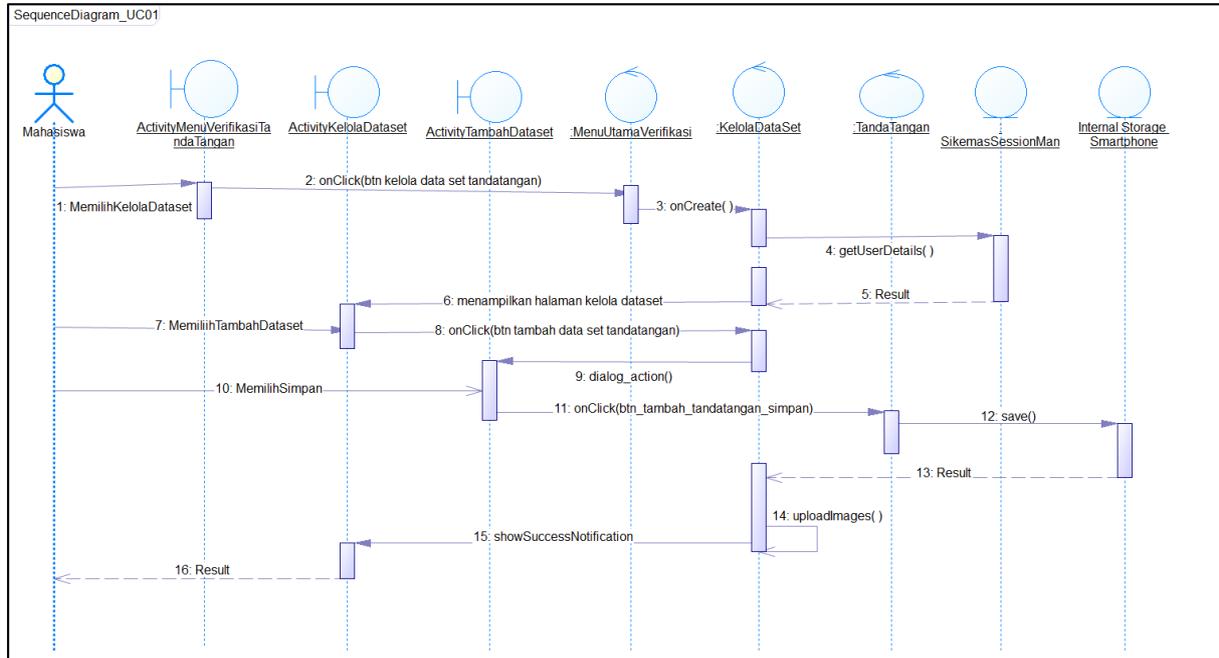
**Tabel 3.3 Spesifikasi Kasus Penggunaan UC-01**

<b>Kode Use Case</b>	UC-0001	
<b>Nama Use Case</b>	Mendaftarkan <i>dataset</i> tanda tangan mahasiswa	
<b>Aktor</b>	Mahasiswa	
<b>Deskripsi</b>	Mahasiswa melakukan pendaftaran akun dengan melakukan tanda tangan sebanyak 5 kali untuk memperoleh data sampel	
<b>Relasi</b>	-	
<b>Kondisi Awal</b>	Sistem belum memiliki data sampel tanda tangan mahasiswa	
<b>Kondisi Akhir</b>	Sistem sudah memiliki data sampel tanda tangan mahasiswa	
<b>Alur kejadian normal</b>	<b>Aktor</b>	<b>Sistem</b>
	1. Memilih pilihan “Kelola Dataset Tanda Tangan”	
		2. Menampilkan “Kelola Dataset Tanda Tangan”
	3. Memilih pilihan “Tambah Dataset Tanda Tangan”	
		4. Menampilkan canvas
5. Melakukan tanda tangan pada canvas sebanyak 5 kali		

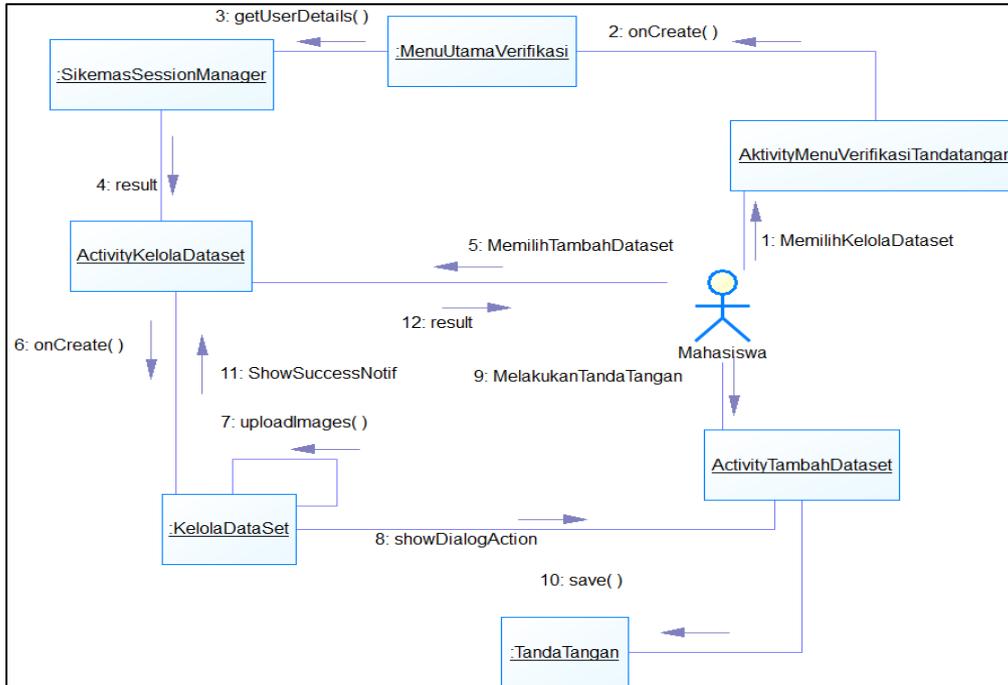
		5. Mengunggah data tanda tangan
		6. Menampilkan notifikasi sukses
<b>Alur kejadian alternatif</b>	<b>Aktor</b>	<b>Sistem</b>
	5.a. Memilih pilihan “Unggah Dataset”	
		5.a.2 Menampilkan notifikasi sukses



**Gambar 3.3 Diagram Aktivitas UC-01**



**Gambar 3.4 Diagram Sekuens UC-01**



**Gambar 3.5** Diagram Kolaborasi UC-01

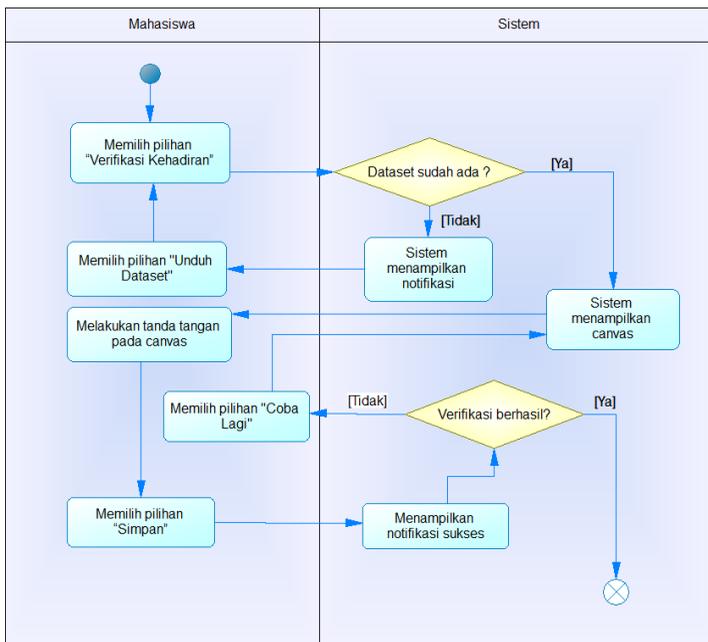
### 3.1.3.4.2 Melakukan Presensi Kehadiran (UC-02)

Kasus penggunaan nomor UC-02 ini diakses ketika mahasiswa melakukan presensi kehadiran di kelas. Mahasiswa melakukan tanda tangan di kanvas pada *smartphone* dengan menggunakan jari. Sistem akan melakukan pencocokan tanda tangan yang dilakukan oleh mahasiswa dengan dataset yang berada di *internal storage smartphone*. Spesifikasi, diagram aktivitas, sekuens dan kolaborasi kasus penggunaan ini dapat dilihat pada Tabel 3.4, Gambar 3.6, Gambar 3.7 dan Gambar 3.8

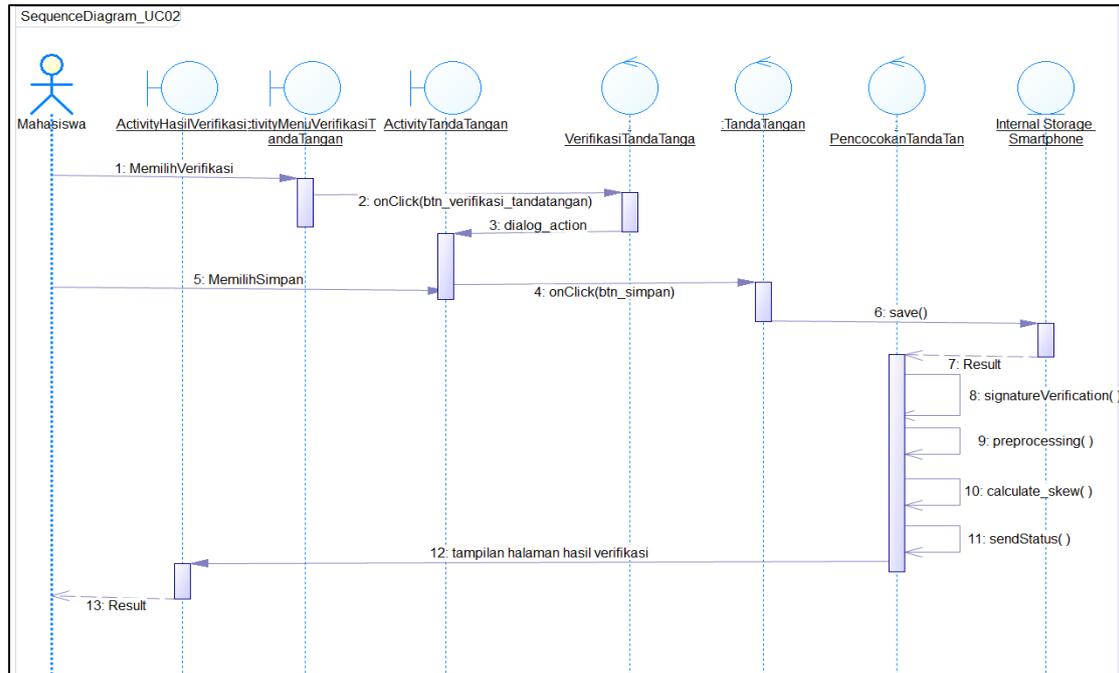
**Tabel 3.4 Spesifikasi Kasus Penggunaan UC-02**

<b>Kode Use Case</b>	UC-02	
<b>Nama Use Case</b>	Melakukan presensi kehadiran	
<b>Aktor</b>	Mahasiswa	
<b>Deskripsi</b>	Mahasiswa dapat melakukan presensi kehadiran di kelas dengan menggunakan tanda tangan pada <i>smartphone</i>	
<b>Relasi</b>	-	
<b>Kondisi Awal</b>	Mahasiswa belum melakukan presensi kehadiran	
<b>Kondisi Akhir</b>	Sistem menyimpan hasil presensi kehadiran dan mengirim ke server	
<b>Alur kejadian normal</b>	<b>Aktor</b>	<b>Sistem</b>
	1. Memilih pilihan “Verifikasi Kehadiran”	
		2. Sistem menampilkan canvas
	3. Melakukan tanda tangan pada canvas	
	4. Memilih pilihan “Simpan”	

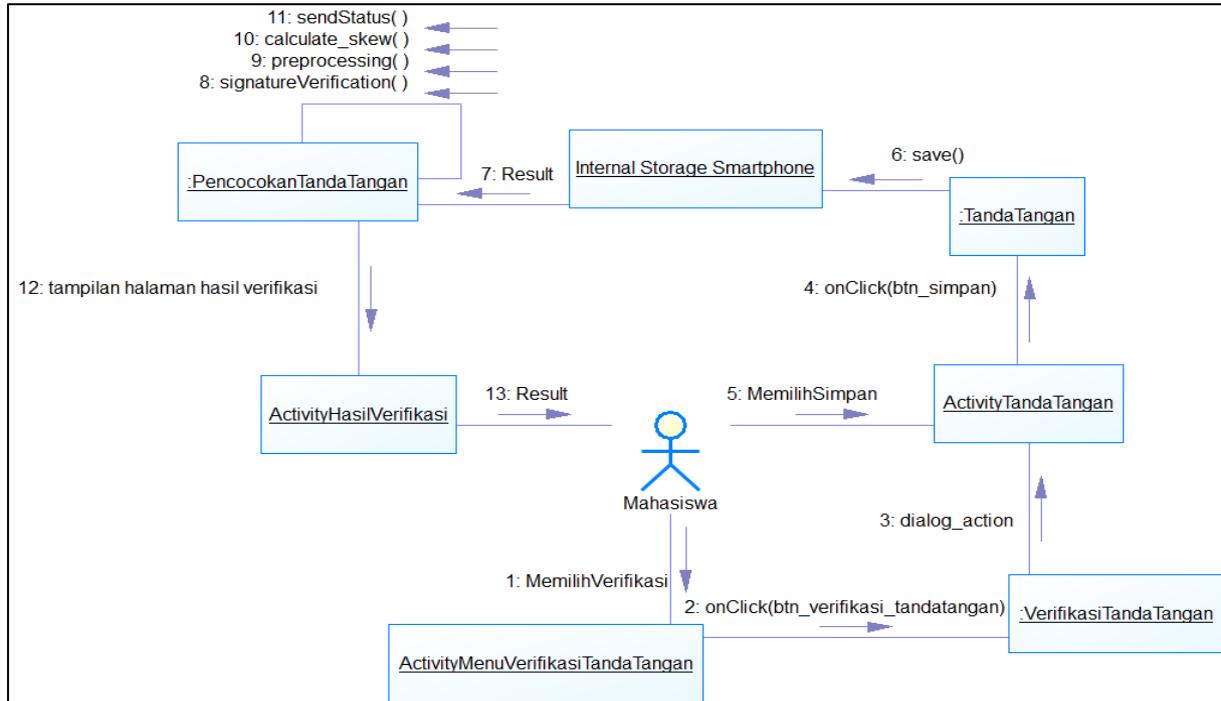
		5. Sistem menampilkan notifikasi
<b>Alur kejadian alternatif</b>	<b>Aktor</b>	<b>Sistem</b>
	1.a Memilih pilihan "Unduh Dataset" pada Kelola Dataset Tanda Tangan	
		1.a.1 Sistem menampilkan notifikasi sukses
	5.a. Memilih pilihan "Coba Lagi"	
		5.a.1 Sistem menampilkan canvas



**Gambar 3.6 Diagram Aktivitas UC-02**



**Gambar 3.7 Diagram Sekuens UC-02**



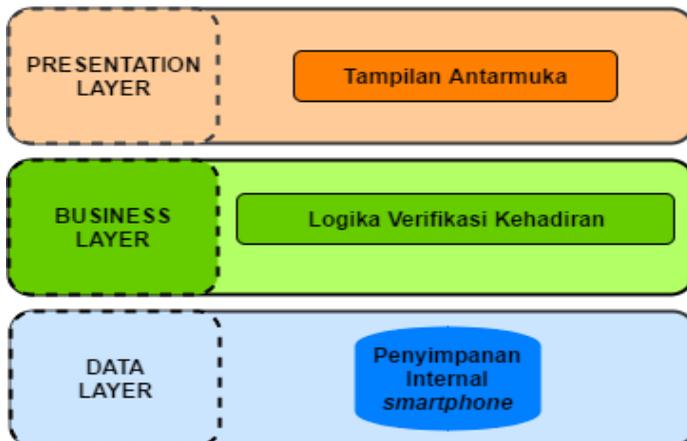
**Gambar 3.8 Diagram Kolaborasi UC-02**

## 3.2 Perancangan

Perancangan dalam subbab ini membahas perancangan dari aplikasi tugas akhir. Subbab ini terdiri dari perancangan arsitektur sistem, perancangan data, perancangan diagram kelas dan perancangan antarmuka pengguna. Perancangan diagram kelas seperti pada Gambar 3.21 Diagram Kelas

### 3.2.1 Perancangan Arsitektur Perangkat Lunak

Arsitektur perangkat lunak untuk tugas akhir ini menggunakan arsitektur 3 lapis yang terdiri dari *Data Layer*, *Bussiness Layer* dan *Presentation Layer*. *Data Layer* berfungsi untuk menangani basis data aplikasi, *Bussiness Layer* berfungsi untuk menangani logika aplikasi serta pencocokan tanda tangan. Sedangkan *Presentation Layer* berfungsi untuk menangani tampilan antarmuka pengguna. *Data Layer* aplikasi ini menggunakan *internal storage smartphone* untuk menyimpan data tanda tangan dari pengguna. Gambaran mengenai arsitektur aplikasi bisa dilihat pada Gambar 3.9.



Gambar 3.9 Perancangan Arsitektur Sistem

### 3.2.2 Perancangan Data

Perancangan basis data dilakukan untuk memastikan pengoperasian aplikasi berjalan dengan benar. Perancangan basis data yang akan digunakan untuk menyimpan data tanda tangan dari mahasiswa yang memiliki hak akses pada sistem. Data masukan (*input*) adalah *dataset* tanda tangan yang tersimpan di *internal storage smartphone* dan data tangan tangan yang dimasukkan ketika mahasiswa melakukan verifikasi kehadiran. Data tanda tangan berupa image dengan format PNG dengan ukuran panjang 432 piksel dan lebarnya 441 piksel. Gambar 3.10 merupakan contoh data tanda tangan.



**Gambar 3.10 Contoh Data Tanda Tangan**

### 3.2.3 Perancangan Proses

Gambar 3.11 Diagram Alur Penggunaan Aplikasi Pencocokan Tanda Tangan menggambarkan alur penggunaan aplikasi validasi kehadiran dengan menggunakan tanda tangan di dalam aplikasi mobile sistem kehadiran yang akan dibangun. Alur penggunaan aplikasi adalah sebagai berikut:

1. Mahasiswa masuk ke dalam fitur validasi kehadiran tanda tangan di dalam aplikasi mobile sistem kehadiran mahasiswa. Mahasiswa harus dalam kondisi ter-login terlebih dahulu.

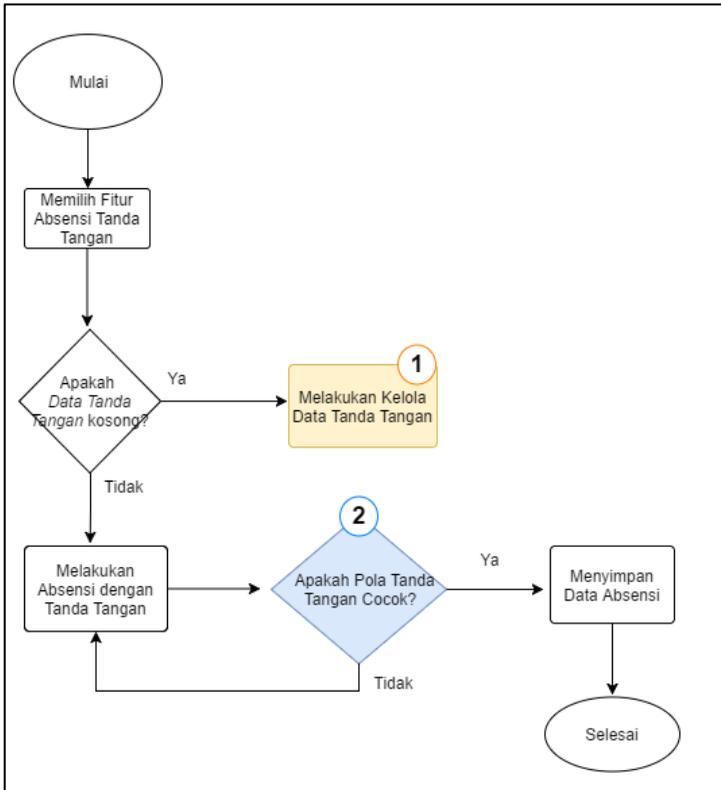
2. Pada saat melakukan validasi kehadiran dengan tanda tangan, aplikasi akan melakukan pengecekan jika *internal storage smartphone* kosong maka harus melakukan kelola data tanda tangan terlebih dahulu. Dimana alur kelola data tanda tangan akan dijelaskan pada Gambar 3.12.
3. Setelah proses kelola data tanda tangan selesai, maka dapat dilakukan presensi kehadiran dengan menuliskan tanda tangan di kanvas pada *smartphone* dengan menggunakan jari.
4. Proses selanjutnya yaitu pencocokan tanda tangan dengan cara membandingkan *dataset* gambar tanda tangan yang telah tersimpan di dalam *internal storage smartphone* dengan data gambar tanda tangan yang diperoleh pada saat melakukan presensi kehadiran. Proses pencocokan gambar tanda tangan dilakukan dengan menghitung jarak gambar *dataset* dengan gambar tanda tangan pada saat presensi yang akan dijelaskan pada Gambar 3.13.
5. Proses selanjutnya adalah hasil pengecekan, jika tanda tangan mahasiswa cocok dengan *dataset* tanda tangan di *internal storage smartphone* maka proses pencocokan gambar tanda tangan berhasil kemudian aplikasi mengirimkan data validasi kehadiran ke *database server* melalui REST API untuk disimpan. Ketika data validasi kehadiran berhasil disimpan kemudian aplikasi akan menampilkan notifikasi berhasil melakukan validasi kehadiran. Sebaliknya, jika gagal maka aplikasi akan menampilkan notifikasi gagal melakukan validasi kehadiran, kemudian mahasiswa melakukan validasi kehadiran dengan tanda tangan lagi.

Gambar 3.12 merupakan diagram alur kelola data tanda tangan. Kelola data tangan merupakan manajemen *dataset* tanda tangan mahasiswa yang digunakan sebagai pembandingan gambar tanda tangan saat presensi. Alur kelola data tanda tangan sebagai berikut:

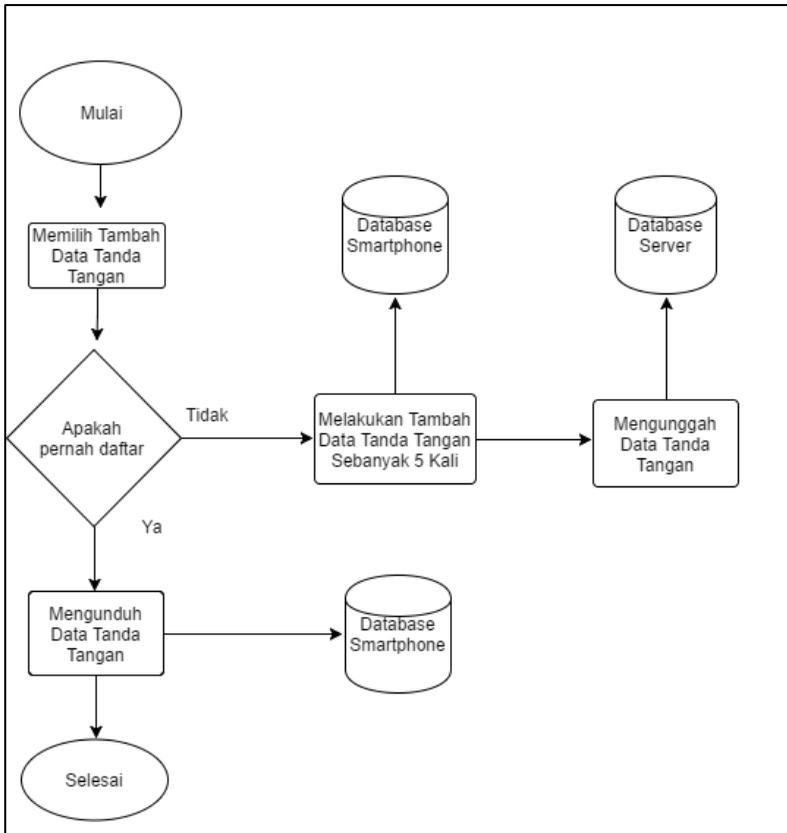
1. Pada saat masuk ke dalam kelola data tanda tangan maka memilih tambah data tanda tangan, maka aplikasi akan

melakukan pengecekan jika mahasiswa belum pernah melakukan daftar tanda tangan maka dapat melakukan tambah data tanda tangan sebanyak 5 kali dan mengunggah data tanda tangan ke *database server*.

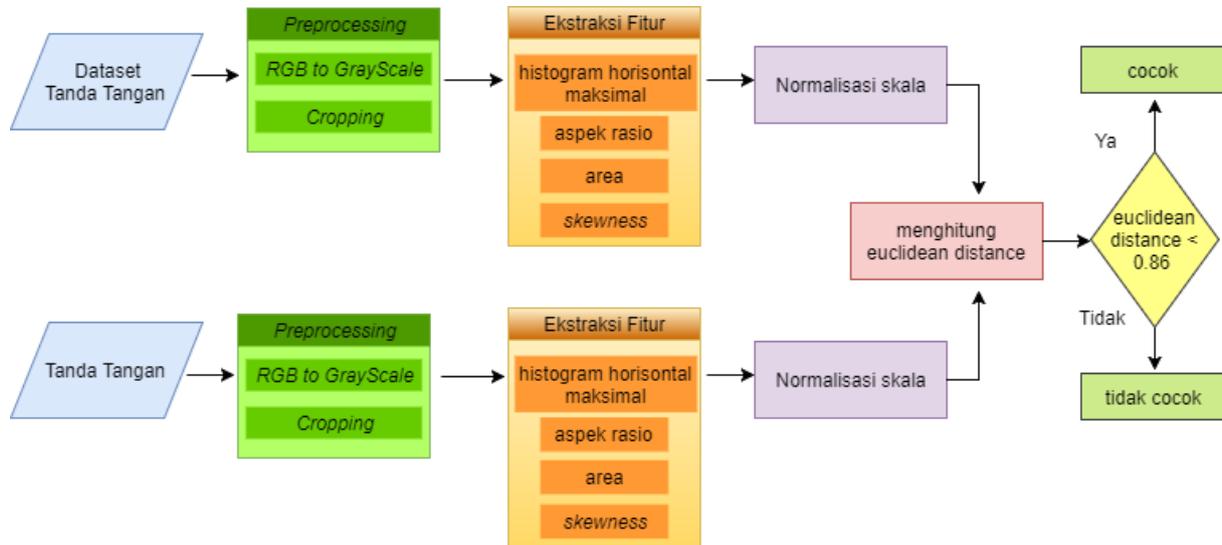
2. Jika telah mendaftarkan data tanda tangan sebelumnya, maka dilakukan proses unduh data tanda tangan yang sebelumnya telah disimpan di dalam *database server*.



**Gambar 3.11 Diagram Alur Penggunaan Aplikasi Pencocokan Tanda Tangan**



**Gambar 3.12 Diagram Alur Kelola Data Tanda Tangan**



**Gambar 3.13 Diagram Blok Proses Aplikasi Pencocokan Tanda Tangan**

Dalam sistem yang dikembangkan pada paper [14] dapat dilihat pada Gambar 3.13, gambar tanda tangan yang telah tersimpan di dalam database *smartphone* dilakukan *preprocessing* terlebih dahulu untuk mendapatkan informasi yang benar pada saat tahap ekstraksi fitur. *Preprocessing* yang dilakukan yaitu melakukan konversi RGB ke Grayscale dan melakukan *cropping* gambar tanda tangan. Kemudian dilakukan ekstraksi fitur untuk mendapatkan nilai fitur dari masing-masing tanda tangan. Fitur yang digunakan adalah *histogram* horisontal maksimal, aspek rasio, area (jumlah *piksel* pada gambar tanda tangan), dan *skewness* (kemiringan gambar tanda tangan). Setelah gambar tanda tangan di dalam *smartphone* mendapatkan fitur, maka gambar tanda tangan yang didapat pada saat melakukan verifikasi kehadiran diolah seperti gambar tanda tangan di dalam *database*. Sehingga masing-masing gambar tanda tangan memiliki fitur masing-masing. Setelah itu dilakukan normalisasi untuk mendapatkan nilai yang standar (rentang nilai sama). Dalam memperoleh hasil bahwa verifikasi kehadiran tanda tangan berhasil/ tidak dilakukan dengan menghitung jarak *euclidean* (*euclidean distance*) dari masing-masing tanda tangan di *database* dengan tanda tangan saat melakukan verifikasi kehadiran. Dari jarak *euclidean* tersebut dapat ditentukan tanda tangan tersebut cocok atau tidak.

```

1.  for i ← 0 to jumlah_dataset_tandatanganan
2.      for j ← 0 to total_fitur
3.          fitur_ttd_normalisasi ← ((fitur_ttd[i][j] - min(i))/max(i)-min(i))
4.          if (i=0)
5.              fitur_ttd_mhs_normalisasi ← ((fitur_ttd_mhs(i)-min(i))/max(i)-min(i))

```

**Gambar 3.14** Gambar *Pseudocode* Normalisasi Skala (Min Max)

Penjelasan dari *pseudocode* pada Gambar 3.14 yaitu *fitur\_ttd* adalah array fitur dataset tanda tangan, *fitur\_ttd\_mhs* adalah array fitur tanda tangan, *fitur\_ttd\_normalisasi* adalah array hasil

normalisasi fitur dataset tanda tangan, `fitur_ttd_mhs_normalisasi` adalah array hasil normalisasi fitur tanda tangan, `fitur_ttd` adalah array fitur data sampel tanda tangan, `min` adalah array untuk menyimpan nilai minimal fitur dan `max` adalah array untuk menyimpan nilai maksimal fitur. Normalisasi berfungsi untuk melakukan normalisasi nilai data sampel dan data tanda tangan agar memiliki nilai yang standar. Normalisasi yang digunakan dalam tugas akhir ini adalah normalisasi skala (min max) yaitu dengan masing-masing fitur dari data sampel dan data tanda tangan dikurangi nilai minimal dari keseluruhan di fitur tersebut kemudian dibagi dengan nilai maksimal dikurangi nilai minimal dari keseluruhan di fitur tersebut. Rumus normalisasi skala (min max) dapat dilihat pada persamaan (2-1).

```

1. // menghitung pengurangan fitur kedua gambar
2. for i ← 0 to jumlah_dataset
3.     fitur_a =fitur_ttd_mhs_normalisasi[i][0]-
        fitur_ttd_mhs_normalisasi(0)
4.     fitur_b =fitur_ttd_mhs_normalisasi[i][1]-
        fitur_ttd_mhs_normalisasi(1)
5.
6. // menghitung kali dan akar fitur kedua gambar
7.
8. for i ← 0 to jumlah_dataset
9.     f1= math.pow(fitur_a(i),2)
10.    f2= math.pow(fitur_b(i),2)
11.
12.    euclidean=math.sqrt(f1+f2)

```

**Gambar 3.15 Gambar Pseudocode Euclidean Distance**

Penjelasan dari pseudocode pada Gambar 3.15 yaitu `fitur_ttd_normalisasi` adalah array hasil normalisasi fitur dataset tanda tangan, `fitur_ttd_mhs_normalisasi` adalah array hasil normalisasi fitur tanda tangan, `fitur_n` adalah array untuk menyimpan fitur ke n, `fn` adalah array untuk menyimpan pangkat dua fitur ke-n dan `euclidean` adalah array untuk menyimpan jarak euclidean. *Euclidean distance* berfungsi untuk menghitung jarak

masing-masing fitur *dataset* dengan fitur tanda tangan pada saat verifikasi tanda tangan yaitu dengan mencari selisihnya kemudian hasil selisih dilakukan operasi pangkat dua. Hasil pangkat dua dari semua fitur dijumlah kemudian diakar dan didapat nilai *euclidean distance*. Rumus *euclidean distance* dapat dilihat pada persamaan (2.2) dan (2.3)

### 3.2.4 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna merupakan hal yang penting dalam melakukan perancangan perangkat lunak. Antarmuka pengguna yang berhubungan langsung dengan aktor harus memiliki kemudahan-kemudahan dan tampilan yang menarik bagi penggunanya. Aplikasi ini memiliki satu hak akses, yaitu mahasiswa. Pada hak akses mahasiswa, aplikasi ini memiliki beberapa antarmuka, yaitu halaman kelola *dataset* tanda tangan, halaman menambah *dataset*, halaman verifikasi kehadiran mahasiswa dan halaman hasil verifikasi kehadiran mahasiswa.

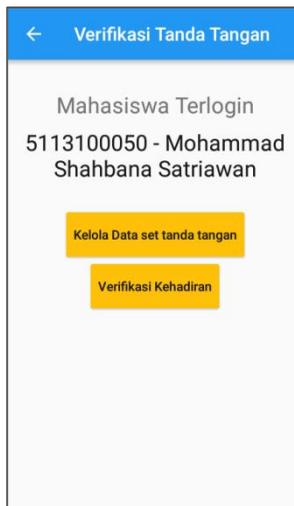
#### 3.2.4.1 Rancangan Halaman Antarmuka Kelola Verifikasi Kehadiran dengan Tanda Tangan

Halaman pada Gambar 3.16 digunakan oleh mahasiswa untuk mengelola verifikasi kehadiran mahasiswa dengan tanda tangan. Di dalam halaman ini akan tampil 2 *button* yaitu Kelola Dataset Tanda Tangan dan Verifikasi Kehadiran. Terdapat informasi NRP dan nama mahasiswa yang *terlogin* (masuk) ke dalam aplikasi. Spesifikasi atribut antarmuka untuk halaman kelola verifikasi kehadiran dapat dilihat pada Tabel 3.5.

**Tabel 3.5 Atribut Antarmuka Halaman Kelola Verifikasi Kehadiran dengan Tanda Tangan**

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>tv_user_detail</i>	<i>TextView</i>	Menampilkan NRP dan nama mahasiswa	<i>String</i>

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/Keluaran
			yang sedang login ke aplikasi	
2	<i>btn_kelola_data_set_tandatangan_view</i>	<i>EditText</i>	Tombol aksi untuk menuju ke halaman Kelola Dataset Tanda Tangan	<i>ButtonClick</i>
3	<i>btn_verifikasi_ttd_view</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Verifikasi Tanda Tangan	<i>ButtonClick</i>



**Gambar 3.16 Rancangan Antarmuka Halaman Kelola Verifikasi Kehadiran dengan Tanda Tangan**

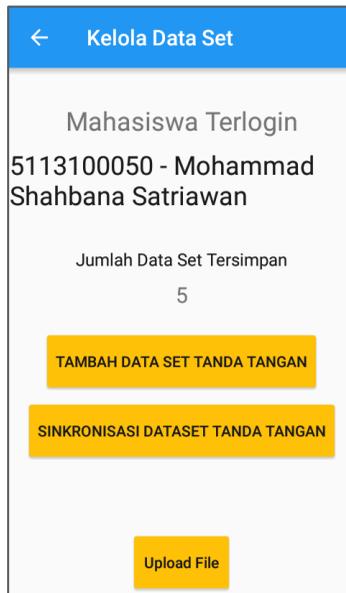
### 3.2.4.2 Rancangan Halaman Kelola Dataset Tanda Tangan

Halaman pada Gambar 3.17 akan ditampilkan pada sistem ketika mahasiswa memilih menu Kelola Dataset Tanda Tangan dan mahasiswa akan mengelola dataset tanda tangannya. Di dalam halaman ini mahasiswa dapat menambah *dataset* tanda tangan, melakukan unggah *dataset* tanda tangan ke *server* dan melakukan sinkronisasi *dataset* dari *database server* dengan *database* lokal *smartphone* mahasiswa. Spesifikasi atribut antarmuka untuk halaman kelola *dataset* tanda tangan dapat dilihat pada Tabel 3.6.

**Tabel 3.6 Atribut Antarmuka Halaman Kelola Dataset Tanda Tangan**

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>tv_user_detail</i>	<i>TextView</i>	Menampilkan NRP dan nama mahasiswa yang sedang login ke aplikasi	<i>String</i>
2	<i>tv_data_set</i>	<i>Button</i>	Menampilkan jumlah dataset yang berada di database lokal <i>smartphone</i>	<i>String</i>
3	<i>btn_tambah_data_set_tandatangan</i>	<i>Button</i>	Tombol aksi untuk menuju ke dialog tambah dataset tanda tangan	<i>ButtonClick</i>
4	<i>btn_sinkronisasi_tandatangan</i>	<i>Button</i>	Tombol aksi untuk melakukan sinkronisasi database lokal <i>smartphone</i>	<i>ButtonClick</i>

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/Keluaran
			dengan database server	
5	<i>btn_upload_file_ttd</i>	<i>Button</i>	Tombol aksi untuk melakukan unggah gambar tanda tangan di database lokal smartphone ke database server	<i>ButtonClick</i>



**Gambar 3.17 Rancangan Antarmuka Halaman Kelola Dataset Tanda Tangan**

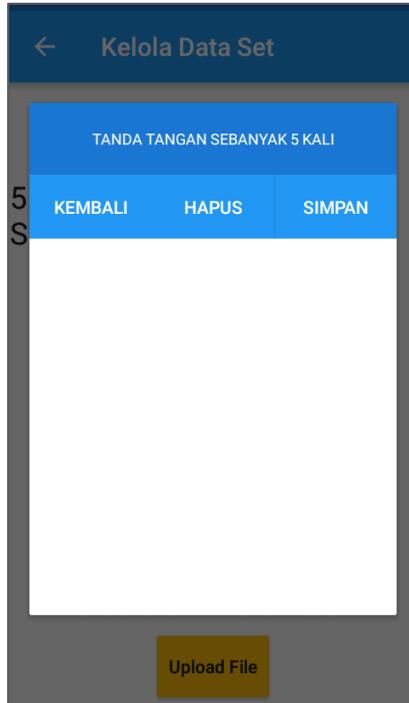
### 3.2.4.3 Rancangan Halaman Dialog Tambah Dataset Tanda Tangan

Halaman dialog pada Gambar 3.18 ditampilkan sistem ketika mahasiswa memilih tambah *dataset* tanda tangan. Halaman dialog ini menampilkan kanvas yang digunakan oleh mahasiswa untuk melakukan tanda tangan sebanyak 5 kali sebagai *dataset*. Spesifikasi atribut antarmuka untuk halaman dialog tambah *dataset* tanda tangan dapat dilihat pada Tabel 3.7.

**Tabel 3.7 Atribut Antarmuka Halaman Dialog Tambah Dataset**

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>tv_tambah_data_set_tandatangan</i>	<i>TextView</i>	Informasi mengenai halaman dialog tambah dataset	<i>String</i>
2	<i>btn_tambah_tandatangan_cancel</i>	<i>Button</i>	Tombol aksi untuk menutup halaman dialog tambah dataset	<i>ButtonClick</i>
3	<i>btn_tambah_tandatangan_clear</i>	<i>Button</i>	Tombol aksi untuk menghapus tanda tangan di kanvas halaman dialog	<i>ButtonClick</i>
4	<i>btn_tambah_tandatangan_simpan</i>	<i>Button</i>	Tombol aksi untuk menghapus tanda tangan di kanvas halaman dialog	<i>ButtonClick</i>
5	<i>buttonSimpan</i>	<i>Button</i>	Tombol aksi untuk menyimpan tanda tangan di	<i>ButtonClick</i>

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/Keluaran
			kanvas halaman dialog	



**Gambar 3.18 Rancangan Antarmuka Halaman Dialog Dataset Tanda Tangan**

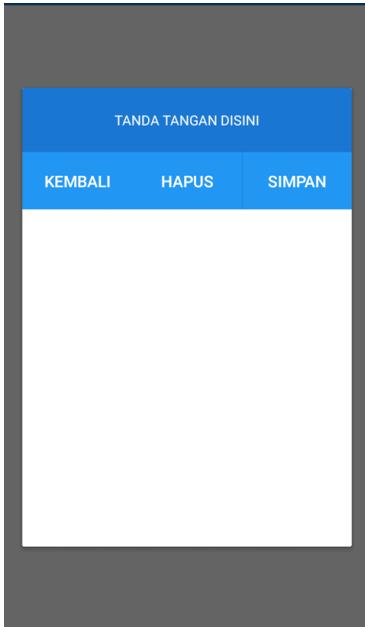
#### **3.2.4.4 Rancangan Halaman Antarmuka Halaman Dialog Verifikasi Kehadiran dengan Tanda Tangan**

Halaman dialog pada Gambar 3.19 ditampilkan sistem ketika mahasiswa melakukan verifikasi kehadiran dengan tanda tangan. Halaman dialog ini menampilkan kanvas yang digunakan mahasiswa untuk melakukan tanda tangan. Spesifikasi atribut

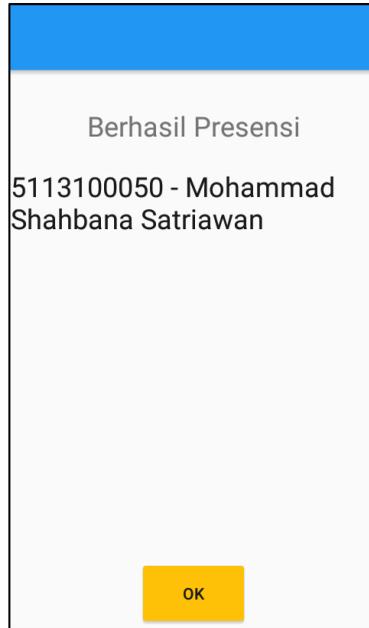
antarmuka untuk halaman dialog verifikasi kehadiran dengan tanda tangan dapat dilihat pada Tabel 3.8.

**Tabel 3.8 Atribut Antarmuka Halaman Dialog Verifikasi Kehadiran dengan Tanda Tangan**

<b>No.</b>	<b>Nama Atribut Antarmuka</b>	<b>Jenis Atribut</b>	<b>Kegunaan</b>	<b>Masukan/ Keluaran</b>
<b>1</b>	<i>tv_tambah_data_set_tandatangan</i>	<i>TextView</i>	Informasi mengenai halaman dialog verifikasi kehadiran dengan tanda tangan	<i>String</i>
<b>2</b>	<i>btn_tambah_tandatangan_cancel</i>	<i>Button</i>	Tombol aksi untuk menutup halaman dialog	<i>ButtonClick</i>
<b>3</b>	<i>btn_tambah_tandatangan_clear</i>	<i>Button</i>	Tombol aksi untuk menghapus tanda tangan di kanvas halaman dialog	<i>ButtonClick</i>
<b>4</b>	<i>btn_tambah_tandatangan_simpan</i>	<i>Button</i>	Tombol aksi untuk menghapus tanda tangan di kanvas halaman dialog	<i>ButtonClick</i>
<b>5</b>	<i>buttonSimpan</i>	<i>Button</i>	Tombol aksi untuk menyimpan dan mencocokkan tanda tangan di kanvas halaman dialog	<i>ButtonClick</i>



**Gambar 3.19 Rancangan Antarmuka Halaman Dialog Verifikasi Kehadiran dengan Tanda Tangan**



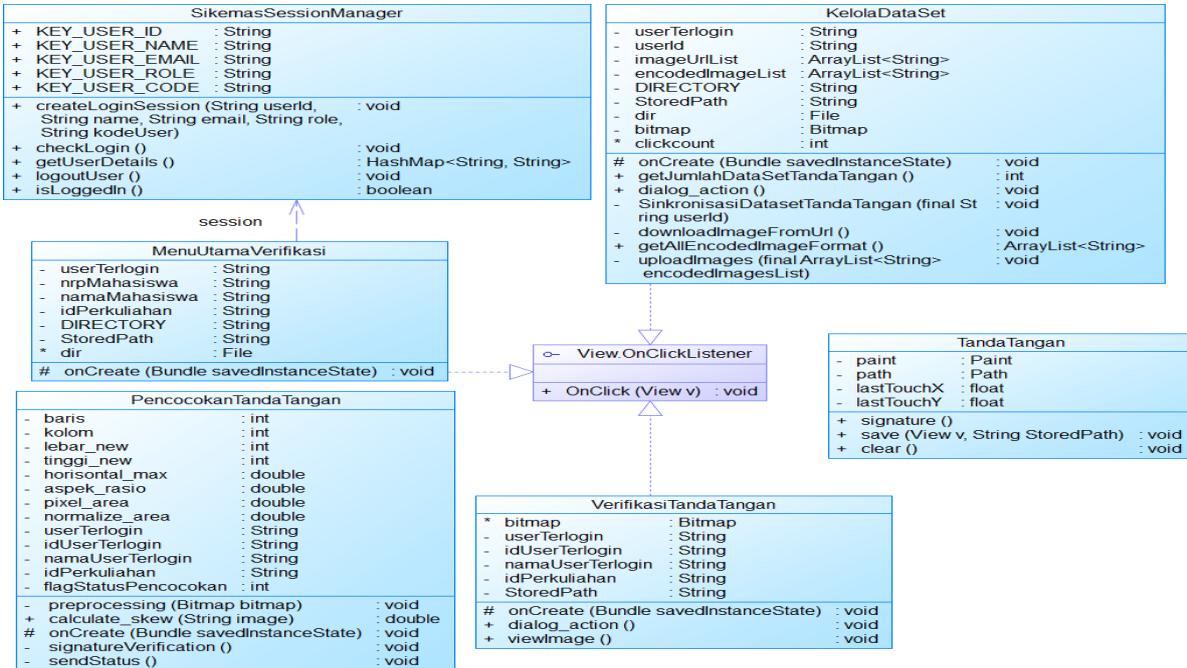
**Gambar 3.20 Rancangan Antarmuka Halaman Hasil Verifikasi Kehadiran**

#### **3.2.4.5 Rancangan Halaman Antarmuka Halaman Hasil Verifikasi Kehadiran**

Halaman pada Gambar 3.20 ditampilkan sistem ketika mahasiswa selesai melakukan verifikasi kehadiran dengan tanda tangan. Halaman ini menampilkan hasil verifikasi kehadiran berhasil atau gagal. Spesifikasi atribut antarmuka untuk halaman hasil verifikasi kehadiran dengan tanda tangan dapat dilihat pada Tabel 3.9.

**Tabel 3.9 Atribut Antarmuka Hasil Verifikasi Kehadiran**

<b>No.</b>	<b>Nama Atribut Antarmuka</b>	<b>Jenis Atribut</b>	<b>Kegunaan</b>	<b>Masukan/ Keluaran</b>
<b>1</b>	<i>tv_user_verifikasi_ttd</i>	<i>TextView</i>	Informasi mengenai NRP dan nama mahasiswa yang melakukan verifikasi kehadiran dengan tanda tangan	<i>String</i>
<b>2</b>	<i>btn_finish_verifikasi_ttd</i>	<i>Button</i>	Tombol aksi untuk masuk ke menu utama verifikasi kehadiran dengan tanda tangan	<i>ButtonClick</i>



Gambar 3.21 Diagram Kelas

## **BAB IV IMPLEMENTASI**

Bab ini membahas implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

Pada bagian implementasi ini juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam program tugas akhir ini dan disertai dengan kode sumber masing-masing fungsi utama.

### **4.1 Lingkungan Implementasi Perangkat Lunak**

Spesifikasi perangkat keras serta perangkat lunak yang digunakan dalam tahap implementasi perangkat lunak tugas akhir ini seperti dijelaskan pada Tabel 4.1.

**Tabel 4.1 Lingkungan Implementasi Perangkat Lunak**

Perangkat Keras	Komputer	ASUS A456U
	Prosesor	Intel® Core™ i7-6500U CPU @ 2.50GHz (2.59GHz)
	Memori Primer	8 GB
	Memori Sekunder	1 TB
Perangkat Lunak	Sistem Operasi	<i>Windows 10</i>

	Perangkat Lunak	Android Studio v2.3.1, Paint, Sybase PowerDesigner 16.5, Sniping Tools, StarUMLv2.8.0, Microsoft Word 2016
--	-----------------	---

## 4.2 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna berbasis perangkat bergerak ini menggunakan berkas XML yang dibangun pada lingkungan pengembangan Android Studio. Pada subbab ini akan dijelaskan dan ditampilkan tampilan halaman XML sesuai dengan rancangan antarmuka yang terdapat pada bab III.

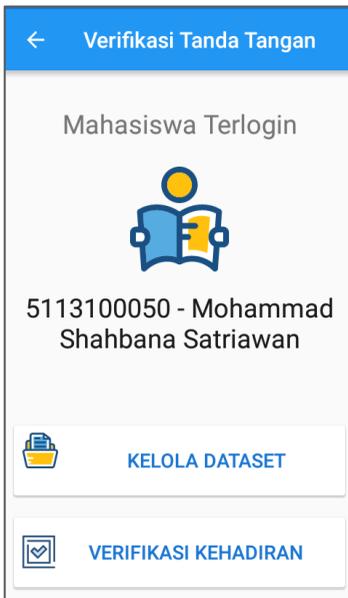
### 4.2.1 Implementasi Halaman Antarmuka Kelola Verifikasi Kehadiran dengan Tanda Tangan

Halaman antarmuka kelola verifikasi kehadiran pada Gambar 4.1 merupakan halaman utama yang digunakan oleh pengguna ketika akan melakukan verifikasi kehadiran dengan tanda tangan. Halaman ini dapat diakses setelah mahasiswa memilih verifikasi kehadiran dengan fitur tanda tangan. Halaman ini memiliki 2 fungsi utama yaitu tombol untuk kelola *dataset* dan tombol untuk melakukan verifikasi kehadiran. Di dalam halaman ini juga menyimpan informasi nama dan NRP mahasiswa yang *login* ke dalam aplikasi.

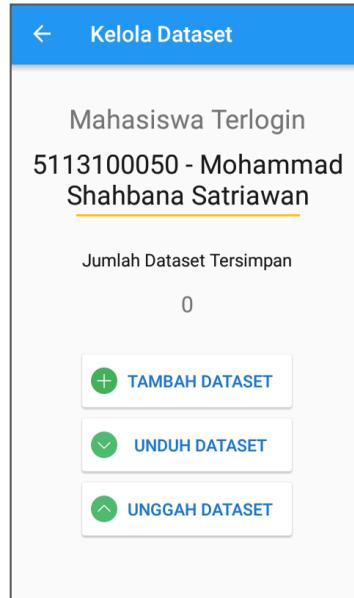
### 4.2.2 Implementasi Halaman Antarmuka Kelola Dataset Tanda Tangan

Halaman antarmuka kelola *dataset* tanda tangan pada Gambar 4.2 merupakan halaman yang akan ditampilkan pada setelah mahasiswa memilih tombol kelola *dataset* pada menu utama kelola verifikasi kehadiran dengan tanda tangan. halaman ini berfungsi untuk mengelola *dataset* tanda tangan. terdapat 3 tombol yaitu tombol tambah *dataset* untuk

menambah *dataset*, tombol unduh *dataset* untuk mengunduh *dataset* dari *server* ke *internal storage smartphone* dan tombol unggah untuk mengunggah *dataset* dari *internal storage smartphone* ke *server*.



**Gambar 4.1 Implementasi Halaman Antarmuka Kelola Verifikasi Kehadiran dengan Tanda Tangan**

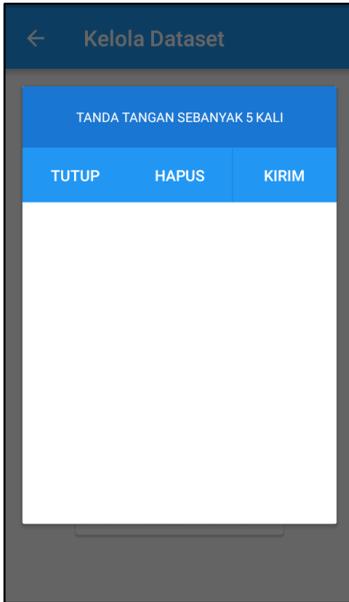


**Gambar 4.2 Implementasi Halaman Antarmuka Kelola Dataset Tanda Tangan**

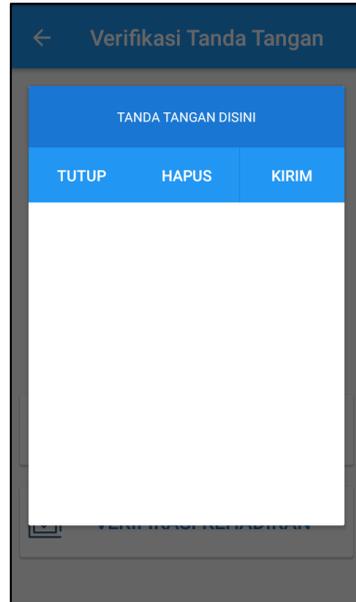
### 4.2.3 Implementasi Halaman Antarmuka Tambah Dataset Tanda Tangan

Halaman antarmuka tambah *dataset* tanda tangan pada Gambar 4.3 merupakan halaman yang akan ditampilkan sistem setelah mahasiswa memilih tambah *dataset*. Halaman ini adalah kanvas yang dapat digunakan untuk menulis gambar tanda tangan. Mahasiswa melakukan tanda tangan sebanyak 5 kali.

Pada saat simpan tanda tangan ke 5 maka secara otomatis *dataset* tanda tangan akan terunggah ke dalam *server*.



**Gambar 4.3 Implementasi Halaman Antarmuka Tambah Dataset Tanda Tangan**



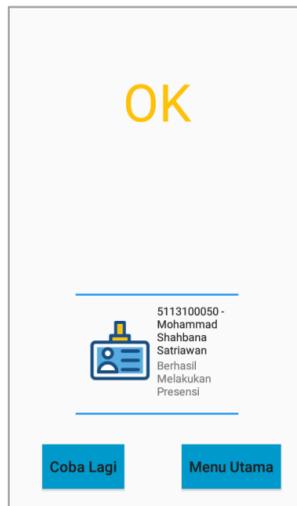
**Gambar 4.4 Implementasi Halaman Antarmuka Verifikasi Kehadiran dengan Tanda Tangan**

#### **4.2.4 Implementasi Halaman Antarmuka Verifikasi Kehadiran dengan Tanda Tangan**

Halaman antarmuka verifikasi kehadiran dengan tanda tangan pada Gambar 4.4 merupakan halaman yang akan ditampilkan pada sistem ketika mahasiswa memilih verifikasi kehadiran pada menu utama kelola verifikasi kehadiran dengan tanda tangan. Pada halaman ini berfungsi untuk melakukan verifikasi kehadiran dengan tanda tangan.

#### 4.2.5 Implementasi Halaman Antarmuka Hasil Verifikasi Kehadiran

Halaman antarmuka hasil verifikasi kehadiran pada Gambar 4.5 merupakan halaman yang akan ditampilkan setelah mahasiswa melakukan verifikasi kehadiran dengan tanda tangan. Halaman ini memberikan informasi mengenai verifikasi kehadiran gagal/ berhasil. Ketika hasil verifikasi gagal maka akan tertulis “GAGAL”, sedangkan ketika berhasil melakukan verifikasi kehadiran maka akan tertulis “OK” pada halaman ini. Halaman ini juga terdapat 2 tombol yaitu coba lagi digunakan untuk melakukan verifikasi kehadiran dengan tanda tangan dan menu utama yang digunakan untuk kembali ke halaman perkuliahan aktif mahasiswa.



**Gambar 4.5 Implementasi Halaman Antarmuka Hasil Verifikasi Kehadiran**

#### 4.2.6 Implementasi Kasus Penggunaan

Implementasi kasus penggunaan aplikasi *mobile* ini menggunakan bahasa pemrograman Java yang dibangun pada

lingkungan pengembangan Android Studio. Disisi lain pengembangan memanfaatkan *web service* untuk mengirimkan string request dan aplikasi akan menerima JSON *post response*. Pada subbab ini akan menjelaskan dan menampilkan kode sumber pada sisi aplikasi perangkat bergerak pada beberapa kasus penggunaan.

#### 4.2.7 Implementasi Kasus Penggunaan Menambah Dataset Tanda Tangan

Pada kasus penggunaan ini, dataset tanda tangan diambil dari kanvas dalam bentuk bitmap kemudian dikonversi ke gambar PNG. Kode sumber dapat dilihat pada Kode Sumber 4.1, sedangkan penjelasannya dapat dilihat pada Tabel 4.2

```

1. public void save(View v) {
2.     if (bitmap == null) {
3.         bitmap = Bitmap.createBitmap(mContent.getWidth(), mContent.getHeight(), Bitmap.Config.RGB_565); }
4.     try {
5.         String filename = userTerlogin+ "-" + clickcount+ ".png";
6.         if(!dir.exists())
7.             { dir.mkdirs(); }
8.             // file output binary
9.             File myFile = new File(dir.getAbsolutePath(), filename);
10.            myFile.createNewFile();
11.            FileOutputStream mFileOutStream = new FileOutputStream(myFile);
12.            v.draw(canvas);
13.            // Mengkonversi file output ke gambar .png
14.            bitmap.compress(Bitmap.CompressFormat.PNG, 90, mFileOutStream);
15.            mFileOutStream.flush();
16.            mFileOutStream.close();}
17.     ... }

```

**Kode Sumber 4.1 Menyimpan Dataset Tanda Tangan**

Tabel 4.2 Penjelasan Kode Sumber 4-1

No. Baris	Kegunaan
2-3	Membuat file bitmap dengan ukuran panjang kanvas dan lebar kanvas. Sebelum dibuat dipastikan terlebih dahulu nilai bitmap kosong
5	Membuat direktori dan membuat nama file gambar ketika disimpan ke <i>internal storage smartphone</i>
6-7	Mengecek jika direktori tidak ada maka akan membuat direktori
9-11	Membuat file dengan menulis byte gambar pada kanvas ke dalam file bentuk binary
13-15	Melakukan konversi file binary ke file PNG

#### 4.2.8 Implementasi Kasus Penggunaan Mengunggah Dataset Tanda Tangan

Pada kasus penggunaan ini, *dataset* tanda tangan dimasukkan ke dalam *arraylist* terlebih dahulu dalam bentuk *binary* ketika akan di unggah ke dalam *server*. Kode sumber utama dapat dilihat pada Kode Sumber 4.2, sedangkan penjelasannya dapat dilihat pada Tabel 4.3. Kode sumber mengunggah *dataset* seperti pada Kode Sumber 4.3, sedangkan penjelasannya dapat dilihat pada Tabel 4.4.

```

1. public ArrayList<String> getAllEncodedImageFormat() {
2. ... {
3. for (File image : files) {
4.     Bitmap bitmap = BitmapFactory.decodeFile(imagePath);
5.     ByteArrayOutputStream baos = new ByteArrayOutputStream();
6.     bitmap.compress(Bitmap.CompressFormat.PNG, 100, baos);
7.     byte[] imageBytes = baos.toByteArray();
8.     encodedImage.add(Base64.encodeToString(imageBytes, Base64
9.     .DEFAULT)); }
10. return encodedImage;} ...

```

#### Kode Sumber 4.2 Mendapatkan Arraylist Dataset

Tabel 4.3 Penjelasan Kode Sumber 4-2

No. Baris	Kegunaan
3	Melakukan perulangan sebanyak jumlah dataset
4-5	Mengubah dataset PNG menjadi file dalam bentuk binary
6-7	Mengubah binary file ke dalam bentuk string dan memasukkan ke dalam array
8	Mengirimkan array berisi string dataset

```

1. // mengunggah file ke server
2. private void uploadImages(final ArrayList<String> encodedImagesList) {
3.     progressDialog.setMessage("Mengunggah Dataset ke Server ...");
4.     progressDialog.show();
5.     for (int i = 0; i < encodedImagesList.size(); i++) {
6.         stringRequest = new StringRequest(Request.Method.POST, NetworkUtils.UPLOAD_DATASET_TANDATANGAN_SIKEMAS,
7.             new Response.Listener<String>() {
8.                 @Override
9.                 public void onResponse(String response) {
10.                    numberOfPhotos[0]++;
11.                    if (numberOfPhotos[0] == encodedImagesList.size()) {
12.                        progressDialog.dismiss();
13.                        Toast.makeText(KelolaDataSetTandaTangan.this, "Berhasil Kirim Dataset ke Server", Toast.LENGTH_SHORT).show();} }},
14.                    ... }) {
15.                @Override
16.                protected Map<String, String> getParams() throws AuthFailureError {
17.                    String image = encodedImagesList.get(index);
18.                    Map<String, String> params = new HashMap<>();
19.                    params.put("image", image);
20.                    params.put("image_name", userTerlogin + "-" + (index+1));
21.                    params.put("user_id", userId);
22.                    return params;}}};

```

Kode Sumber 4.3 Mengunggah Dataset Tanda Tangan

Tabel 4.4 Penjelasan Kode Sumber 4-3

No. Baris	Kegunaan
3-4	Menampilkan progress dialog ketika mengunggah dataset
9	Melakukan perulangan sebanyak jumlah dataset
6	Mengirimkan string request ke server
7	Mendapatkan respon dari server
13	Menampilkan informasi berhasil mengunggah dataset
18	Mengirimkan gambar berupa string
19	Mengirimkan nama gambar dataset tanda tangan
20	Mengirimkan NRP mahasiswa

#### 4.2.9 Implementasi Kasus Penggunaan Mengunduh Dataset Tanda Tangan

Pada kasus penggunaan ini, aplikasi mengirimkan *string request* ke *server* kemudian aplikasi menerima JSON Objek dari server. Kemudian dari JSON Objek diambil url gambar dan dari url gambar disimpan ke dalam *internal storage smartphone*. Kode sumber mengunduh *dataset* dapat dilihat pada Kode Sumber 4.4 dan Tabel 4.5 merupakan penjelasan dari kode programnya. Sedangkan kode sumber menyimpan hasil unduh *dataset* dapat dilihat pada Kode Sumber 4.5 dan Tabel 4.6

```

1. private void SinkronisasiDatasetTandaTangan(final String userId) {
2.     progressDialog.setMessage("Pengecekan Dataset Server... ");
3.     progressDialog.show();
4.     StringRequest stringRequest = new StringRequest(Request.Method.POST, NetworkUtils.SINKRONISASI_DATASET_TANDATANGAN_SIKEMAS,
5.     new Response.Listener<String>() {
6.         @Override
7.         public void onResponse(String response) {
8.         try {

```

```

8.   if (list.length() > 0) {
9.     for (int i = 0; i < list.length(); i++) {
10.    JSONObject urlImage = list.getJSONObject(i);
11.    String fotoUrl = urlImage.getString("foto").replace(" ",
"%20");
12.    imageUrlList.add(fotoUrl);
13.    progressDialog.dismiss();
14.    downloadImageFromUrl();
15.    Toast.makeText(KelolaDataSetTandaTangan.this, "Dataset d
itemukan", Toast.LENGTH_SHORT).show(); }
16.  else {
17.    progressDialog.dismiss();
18.    Toast.makeText(KelolaDataSetTandaTangan.this, "Dataset tid
ak ditemukan, "+ "Anda belum mendaftarkan Tanda Tangan Anda"
, Toast.LENGTH_SHORT).show(); }
19. } ... } },
20. new Response.ErrorListener() {
21.   @Override
22.   public void onErrorResponse(VolleyError volleyError)
23.   ...} {
24.   @Override
25.   protected Map<String, String> getParams() throws AuthFailur
eError {
26.   Map<String, String> params = new HashMap<>();
27.   params.put("nrp", userId);
28.   return params; } }; ... }

```

#### Kode Sumber 4.4 Mengunduh Dataset Tanda Tangan

Tabel 4.5 Penjelasan Kode Sumber 4-4

No. Baris	Kegunaan
2-3	Menampilkan progress dialog ketika mengunggah dataset
4	Mengirimkan string request ke server
5	Mendapatkan respon dari server
7	Mengecek jika panjang Array ditemukan maka terdapat dataset di server, jika tidak ditemukan maka dataset belum pernah di upload ke server

No. Baris	Kegunaan
11	Mendapatkan url dari dataset tanda tangan dengan mengganti spasi menjadi %20 dan menyimpan ke dalam Array
14	Masuk ke fungsi download dataset tanda tangan
18	Memberikan informasi jika dataset tidak ditemukan di server
27	Mengirimkan NRP dari mahasiswa yang akan diunduh datasetnya

Tabel 4.6 Penjelasan Kode Sumber 4-5

No. Baris	Kegunaan
3-4	Menampilkan progress dialog
	Menampilkan progress dialog ketika mengunggah dataset
5	Melakukan perulangan sebanyak Array yang menyimpan url dataset dari server
6	Membuat String yang digunakan untuk menyimpan nama dari gambar
12-13	Menyimpan path yang digunakan untuk menyimpan dataset beserta memberi nama pada gambar
14-15	Menyimpan dataset ke <i>internal storage smartphone</i> berdasarkan path yang telah ditentukan sebelumnya
18	Memberikan informasi berhasil melakukan unduh dataset

```

1. // Mengunduh dataset
2. private void downloadImageFromUrl() {
3.     progressDialog.setMessage("Unduh Dataset... ");
4.     progressDialog.show();
5.     for (int i = 0; i < imageUrlList.size(); i++) {
6.         InputStreamVolleyRequest inputStreamRequest = new Input
StreamVolleyRequest(Request.Method.GET, imageUrlList.get
(i),
7.         new Response.Listener<byte[]>() {

```

```

8.     @Override
9.     public void onResponse(byte[] response) {
10.    try {
11.        if (response != null) {
12.            String name = userTerlogin + "-" + nomor + ".png";
13.            String fullpath = wholeFolderPath + "/" + name;
14.            outputStream.write(response);
15.            outputStream.close();
16.            if ((index + 1) == imageUrlList.size()) {
17.                progressDialog.dismiss();
18.                Toast.makeText(getApplicationContext(), "Berhasil mel
akukan unduh dataset",
19.                    Toast.LENGTH_SHORT).show();}} ...} },
20.        new Response.ErrorListener() {
21.            ... }, null); ...}}

```

#### Kode Sumber 4.5 Menyimpan Hasil Unduh Dataset

### 4.2.10 Implementasi Kasus Penggunaan Verifikasi Kehadiran dengan Tanda Tangan

Pada kasus penggunaan ini, proses verifikasi kehadiran tanda tangan dilakukan *preprocessing* terlebih dahulu. Kemudian didapatkan fitur-fitur melalui ekstraksi fitur. Terdapat 4 fitur pada masing-masing gambar tanda tangan. kecocokan dataset dengan tanda tangan dihitung jaraknya (*euclidean distance*) berdasarkan fitur yang diperoleh. Proses *preprocessing* yaitu *cropping* dan konversi RGB ke *grayscale*. *Cropping* dilakukan untuk mendapatkan gambar dengan ukuran tanda tangan. Kode sumber dan penjelasan dari *cropping* dapat dilihat pada Kode Sumber 4.6 dan Tabel 4.7.

**Tabel 4.7 Penjelasan Kode Sumber 4-6**

No. Baris	Kegunaan
1-10	Mencari titik tinggi maksimal dari tanda tangan
11-20	Mencari titik lebar maksimal dari tanda tangan

21-31	Mencari titik lebar minimal dari tanda tangan
32-42	Mencari tinggi minimal dari tanda tangan
44	Menghitung lebar tanda tangan yang akan <i>dicrop</i>
45	Menghitung tinggi tanda tangan yang akan <i>dicrop</i>
46	Membuat bitmap baru untuk menyimpan gambar binary hasil <i>crop</i>
47-50	Melakukan <i>crop</i> gambar berdasarkan tinggi dan lebar gambar baru ke dalam variabel bitmap baru

```

1. //cari tinggi maksimal
2. for(ya=0; ya<tinggi_img1; ya++) {
3.   for (xa = 0; xa < lebar_img1; xa++) {
4.     int piksel = bitmap3.getPiksel(xa, ya);
5.     R = Color.red(piksel);
6.     if (R > 128) R = 255;
7.     else R = 0;
8.     gray_1.setPiksel(xa, ya, Color.rgb(R, R, R));
9.     if (R == 0) { x1 = ya; break;} }
10.  if (R == 0) break; }
11. //cari lebar maksimum
12. for( xb=0; xb<lebar_img1; xb++){
13.   for(yb=0; yb<tinggi_img1; yb++) {
14.     int piksel = bitmap3.getPiksel(xb, yb);
15.     R = Color.red(piksel);
16.     if (R > 128) R = 255;
17.     else R = 0;
18.     gray_1.setPiksel(xb, yb, Color.rgb(R, R, R));
19.     if (R == 0) { x2 = xb; break; } }
20.   if (R == 0) break;}
21. //cari lebar minimal
22. for(xc=lebar_img1-1; xc >=0; xc--){
23.   for(yc=0; yc < tinggi_img1; yc++){
24.     int piksel = bitmap3.getPiksel(xc, yc);
25.     R = Color.red(piksel);
26.     if (R > 128) R = 255;
27.     else R = 0;
28.     gray_1.setPiksel(xc, yc, Color.rgb(R, R, R));
29.     if (R == 0) {
30.       x3 = xc; break;} }
31.   if (R == 0){break;}}
32. //cari tinggi minimal

```

```

33. for(yd=tinggi_img1-1; yd>=0; yd--){
34.   for(xd=0; xd<lebar_img1; xd++){
35.     int piksel = bitmap3.getPiksel(xd, yd);
36.     R = Color.red(piksel);
37.     if (R > 128) R = 255;
38.     else R = 0;
39.     gray_1.setPiksel(xd, yd, Color.rgb(R, R, R));
40.     if (R == 0) {
41.       x4 = yd; break;} }
42.   if (R == 0)break;}
43. //cropping
44. lebar_new = x3-x2;
45. tinggi_new = x4-x1;
46. new_img_crop = Bitmap.createBitmap(lebar_new,tinggi_new,Bit
map.Config.RGB_565);
47. for(i=0; i<lebar_new; i++){
48.   for(j=0; j<tinggi_new; j++){
49.     int piksel = bitmap3.getPiksel(x2+i, x1+j);
50.     new_img_crop.setPiksel(i,j,piksel); }}

```

#### Kode Sumber 4.6 Melakukan *Cropping* Gambar Tanda Tangan

Kode Sumber 4.7 berfungsi untuk konversi RGB ke *grayscale* terhadap gambar yang telah *dicrop*. Kode sumber dan penjelasan dari *cropping* dapat dilihat pada Tabel 4.8.

**Tabel 4.8 Penjelasan Kode Sumber 4-7**

No. Baris	Kegunaan
2	Mendapat tinggi dari gambar hasil <i>crop</i>
3	Mendapat lebar dari gambar hasil <i>crop</i>
4	Membuat bitmap baru
9	Mengecek jika RGB dari gambar lebih dari 128 maka akan diubah menjadi 255 (warna putih)
10	Mengecek jika RGB dari gambar 0 – 128 maka akan diubah menjadi 0 (warna hitam)
11	Melakukan set piksel hasil pengecekan ke dalam bitmap baru

```

1. //grayscale
2. tinggi_img_baru = new_img_crop.getHeight();
3. lebar_img_baru =new_img_crop.getWidth();
4. Bitmap gray_img_crop = Bitmap.createBitmap( bitmap.g
   etWidth(), bitmap.getHeight(), bitmap.getConfig());
5. for(ya=0; ya<tinggi_img_baru; ya++) {
6.   for (xa = 0; xa < lebar_img_baru; xa++) {
7.     int pikselnya = new_img_crop.getPiksel(xa, ya);
8.     R = Color.red(pikselnya);
9.     if (R > 128) R = 255;
10.    else R = 0;
11.    gray_img_crop.setPiksel(xa, ya, Color.rgb(R, R, R)
   ); } }

```

#### Kode Sumber 4.7 Melakukan *Grayscale* Gambar Tanda Tangan

Kode Sumber 4.8 berfungsi untuk mendapatkan fitur *histogram* horisontal maksimal. Penjelasan kode program dapat dilihat pada Tabel 4.9.

```

1. //ekstraksi fitur //max horisontal
2. for(baris=0; baris<tinggi_img_baru; baris++){
3.   for(kolom=0; kolom<lebar_img_baru; kolom++){
4.     int piksel = gray_img_crop.getPiksel(kolom, baris);
5.     if(piksel == Color.BLACK){
6.       kolom_temp=kolom_temp+1;}}
7.   if(kolom_temp > kolom_max) {
8.     kolom_max = kolom_temp;
9.     horisontal_max = baris; }}

```

#### Kode Sumber 4.8 Mencari *Histogram* Harisontal Maksimum

Tabel 4.9 Penjelasan Kode Sumber 4-8

No. Baris	Kegunaan
2-4	Melakukan pengecekan terhadap piksel warna hitam pada setiap kolom dan baris
5	Menyimpan ke dalam variabel jika piksel yang ditemukan berwarna hitam

No. Baris	Kegunaan
6-8	Melakukan pengecekan untuk memilih baris yang memiliki paling banyak pikses berwarna hitam

Kode Sumber 4.9 berfungsi untuk mendapatkan fitur aspek rasio dari gambar. Penjelasan kode program dapat dilihat pada Tabel 4.10.

```

1. //aspect ratio
2. lebar_new2 =lebar_new;
3. tinggi_new2 = tinggi_new;
4. Ratio = lebar_new2/tinggi_new2;

```

#### Kode Sumber 4.9 Mencari Aspek Rasio

Tabel 4.10 Penjelasan Kode Sumber 4-9

No. Baris	Kegunaan
2	Mendapatkan lebar dari gambar tanda tangan
3	Mendapatkan tinggi dari gambar tanda tangan
4	Membagi lebar gambar dengan tinggi gambar untuk mendapat aspek rasio

Kode Sumber 4.10 berfungsi untuk mendapatkan fitur jumlah *piksel* warna hitam dari gambar. Penjelasan kode program dapat dilihat pada Tabel 4.11.

```

1. //total number black piksel area
2. for(baris=0; baris<tinggi_img_baru; baris++){
3.   for(kolom=0; kolom<lebar_img_baru; kolom++){
4.     int piksel = gray_img_crop.getPiksel(kolom, baris);
5.     if(piksel == Color.BLACK){
6.       piksel_area=piksel_area+1;}}}

```

#### Kode Sumber 4.10 Mencari Jumlah Piksel

**Tabel 4.11 Penjelasan Kode Sumber 4-10**

No. Baris	Kegunaan
4	Mengecek apakah piksel pada koordinat x,y berwarna hitam
5	Jika piksel warna hitam maka disimpan ke dalam variabel

Kode Sumber 4.11 berfungsi untuk mendapatkan fitur kemiringan dari gambar. Penjelasan kode program dapat dilihat pada dan Tabel 4.12.

**Tabel 4.12 Penjelasan Kode Sumber 4-11**

No. Baris	Kegunaan
2	Melakukan grayscale terhadap gambar
3	Memberikan threshold
6	Melakukan inersi warna menjadi warna gambar mnejadi putih sedangkan warna background menjadi warna hitam
11-12	Melakukan rotasi pada gambar
16	Mengirimkan variabel kemiringan gambar

```

1. public double calculate_skew(String image){
2. Mat img = Imgcodecs.imread( image, Imgcodecs.IMREAD
   _GRAYSCALE );
3. Imgproc.threshold( img, img, 200, 255, THRESH_BINAR
   Y );
4. Core.bitwise_not( img, img );
5. Mat element = Imgproc.getStructuringElement(Imgproc
   .MORPH_RECT, new Size(3, 3));
6. Imgproc.erode(img, img, element);
7. //Find all white piksels
8. Mat wLocMat = Mat.zeros(img.size(),img.type());
9. Core.findNonZero(img, wLocMat);
10. matOfPoint.convertTo(mat2f, CvType.CV_32FC2);

```

```

11. RotatedRect rotatedRect = Imgproc.minAreaRect( mat2
    f );
12. Point[] vertices = new Point[4];
13. rotatedRect.points(vertices);
14. Imgproc.drawContours( img, boxContours, 0, new Scal
    ar(128, 128, 128), -1);
15. double resultAngle = rotatedRect.angle;
16. return resultAngle; }

```

#### Kode Sumber 4.11 Skewness

Kode Sumber 4.12 berfungsi untuk melakukan normalisasi terhadap fitur tanda tangan dan dataset tanda tangan. Penjelasan kode program dapat dilihat pada Tabel 4.13.

```

1. for(int index=1; index<=5; index++){
2. for(int i=0; i<7; i++){
3. norm_atas= fitur_ttd[index][i]-min.get(i);
4. norm_bawah= max.get(i)-min.get(i);
5. fitur_ttd_normalisasi[index][i]=norm_atas/norm_bawah
    if(index==1){
6. norm_atas= fitur_ttd_mhs.get(i)-min.get(i);
7. norm_bawah= max.get(i)-min.get(i);
8. fitur_ttd_mhs_normalisasi.add(norm_atas/norm_bawah);
    }}}

```

#### Kode Sumber 4.12 Melakukan Normalisasi Fitur

Tabel 4.13 Penjelasan Kode Sumber 4-12

No. Baris	Kegunaan
2	Melakukan perulangan (normalisasi) sebanyak dataset tanda tangan yang tersimpan di dalam <i>internal storage smartphone</i> (5 data tanda tangan)
3	Menghitung nilai normalisasi atas yaitu dengan cara nilai fitur dibagi dengan nilai maksimal dari masing-masing fitur tanda tangan sampel dan tanda tangan saat melakukan verifikasi kehadiran

No. Baris	Kegunaan
4	Menghitung nilai normalisasi atas yaitu dengan cara nilai maksimal dari masing-masing fitur dikurangi dengan nilai minimal dari masing-masing fitur tanda tangan sampel dan tanda tangan pada saat verifikasi kehadiran
5	Menghitung pembagian hasil dari perhitungan dari kode program nomor 3 dibagi dengan kode program nomor 4
6-8	Menghitung normalisasi dari gambar tanda tangan saat verifikasi

Kode Sumber 4.13 berfungsi untuk menghitung *euclidean distance* antara *dataset* dengan tanda tangan saat verifikasi. Penjelasan kode program dapat dilihat pada Tabel 4.14.

```

1. for(int i=1; i<=5; i++){
2. fitur_a.add(fitur_ttd_normalisasi[i][0]-
   fitur_ttd_mhs_normalisasi.get(0));
3. fitur_b.add(fitur_ttd_normalisasi[i][1]-
   fitur_ttd_mhs_normalisasi.get(1));
4. fitur_c.add(fitur_ttd_normalisasi[i][2]-
   fitur_ttd_mhs_normalisasi.get(2));
5. fitur_d.add(fitur_ttd_normalisasi[i][3]-
   fitur_ttd_mhs_normalisasi.get(3)); }
6. for(int i=0; i<5;i++){
7. f1= Math.pow((fitur_a.get(i)),2.0);
8. f2= Math.pow((fitur_b.get(i)),2.0);
9. f3= Math.pow((fitur_c.get(i)),2.0);
10. f4= Math.pow((fitur_d.get(i)),2.0);
11. euclidean.add(Math.sqrt(f1+f2+f3+f4));}
12. if(euclidean.get(0)<1 || euclidean.get(1)<1 || euclidean.get(2)<1 || euclidean.get(3)<1 || euclidean.get(4)<1)
13. flagStatusPencocokan = 1;
14. else flagStatusPencocokan = 0; }

```

**Kode Sumber 4.13 Mencari *Euclidean Distance***

**Tabel 4.14 Penjelasan Kode Sumber 4-13**

<b>No. Baris</b>	<b>Kegunaan</b>
1-5	Melakukan perulangan untuk menyimpan hasil pengurangan dari masing-masing fitur antara dataset dengan gambar tanda tangan saat verifikasi
7-10	Melakukan perhitungan perpangkatan dari masing-masing fitur hasil pengurangan dataset dengan gambar tanda tangan
11	Melakukan perhitungan akar dari jumlah masing-masing perpangkatan dari fitur sehingga mendapatkan euclidean distance
12-13	Melakukan pengecekan terhadap hasil Euclidean distance, jika kurang dari threshold yang ditentukan maka verifikasi berhasil, jika lebih maka verifikasi gagal

Kode Sumber 4.14 berfungsi untuk mengirimkan status hasil verifikasi ke *server* jika verifikasi kehadiran berhasil dilakukan. Penjelasan kode program dapat dilihat pada Tabel 4.15

**Tabel 4.15 Penjelasan Kode Sumber 4-14**

<b>No. Baris</b>	<b>Kegunaan</b>
2	Mengirimkan string request ke server
3	Mendapatkan respon dari server
6	Menampilkan informasi ketika berhasil melakukan verifikasi beserta kirim status ke server
10	Mengirimkan id perkuliahan
11	Mengirimkan NRP dari mahasiswa
12	Mengirimkan status jika telah berhasil verifikasi kehadiran yaitu M

```
1. private void sendStatus() {
2.   stringRequest = new StringRequest(Request.Method.POST, NetworkUtils.KIRIM_STATUS_KEHADIRAN_MAHASISWA,
3.   new Response.Listener<String>() {
4.     @Override
5.     public void onResponse(String response) {
6.       Toast.makeText(PencocokanTandaTangan.this, "Verifikasi Kehadiran Berhasil", Toast.LENGTH_SHORT).show();
7.     };
8.   }, ... ){
9.     @Override
10.    protected Map<String, String> getParams() throws AuthFailureError {
11.      Map<String, String> params = new HashMap<>();
12.      params.put("id_perkuliahan", idPerkuliahan);
13.      params.put("nrp", idUserTerlogin);
14.      params.put("status", "M");
15.      return params;
16.    };
17.  }
18. }
```

**Kode Sumber 4.14 Mengirimkan Status Verifikasi ke Web Service**

*[Halaman ini sengaja dikosongkan]*

## **BAB V**

### **PENGUJIAN DAN EVALUASI**

Bab ini membahas tentang pengujian dan evaluasi pada perangkat lunak yang dibangun untuk tugas akhir ini. Pengujian dilakukan adalah pengujian terhadap kebutuhan fungsional program serta pengujian kebergunaan program. Pengujian fungsionalitas akan mengacu pada kasus penggunaan program yang sudah dibahas pada bab 3, pengujian kebergunaan dilakukan untuk mengetahui tanggapan pengguna terhadap program yang dikembangkan serta pengujian waktu eksekusi dilakukan untuk mengetahui kecepatan (rata-rata waktu yang dibutuhkan) oleh aplikasi untuk melakukan pencocokan tanda tangan. Hasil dari ketiga pengujian tersebut akan dijabarkan pada bagian evaluasi di akhir bab ini.

#### **5.1 Lingkungan Pengujian**

Pada proses pengujian fungsionalitas perangkat lunak, dibutuhkan suatu lingkungan pengujian yang sesuai dengan standar kebutuhan. Lingkungan pengujian dalam tugas akhir ini dilakukan pada setiap kasus penggunaan. Spesifikasi masing-masing lingkungan pengujian dijabarkan pada Tabel 5.1

**Tabel 5.1 Lingkungan Pengujian Fungsionalitas Perangkat Lunak**

<b>Spesifikasi</b>	<b>Deskripsi</b>
Jenis Perangkat	<i>Smartphone</i>
Merek Perangkat	Oppo A37f
Sistem Operasi	Android Lollipop v5.1.1
Memori Internal	16 GB

Spesifikasi	Deskripsi
RAM	2 GB

## 5.2 Pengujian Fungsionalitas

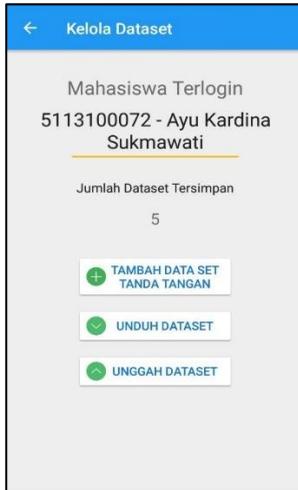
Pengujian fungsionalitas ini adalah pengujian fungsi-fungsi yang berjalan pada aplikasi berdasarkan kasus penggunaan. Pengujian fungsionalitas ini dijelaskan pada Tabel 5.2 dan Tabel 5.3

### 5.2.1 Pengujian Fungsionalitas Menambah Dataset

**Tabel 5.2 Skenario 1 Pengujian Fungsionalitas Menambah Dataset**

<b>No. Pengujian</b>	<b>UJ.UC.01</b>
<b>Referensi Kasus Penggunaan</b>	<b>UC.01</b>
<b>Nama</b>	Pengujian Fungsionalitas Menambah Dataset
<b>Tujuan Pengujian</b>	Mengujian tambah dataset tanda tangan
<b>Skenario Pengujian</b>	<i>Pengguna menambah/ mendaftarkan tanda tangan</i>
<b>Kondisi Awal</b>	Pengguna berada pada halaman Kelola Dataset Tanda Tangan
<b>Data Uji</b>	Nama : Ayu Kardina Sukmawati, NRP : 5113100072, Kelas : Data Mining, Data : 5 gambar tanda tangan
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>4. Pengguna masuk ke halaman Kelola Dataset Tanda Tangan seperti pada Gambar 5.1 Halaman Kelola Dataset Tanda Tangan</li> <li>5. Pengguna menekan tombol yang bertuliskan “Tambah Dataset Tanda Tangan”</li> <li>6. Pengguna melakukan tanda tangan pada kanvas seperti pada Gambar 5.2. Tanda tangan</li> </ol>

	<p>yang telah dilakukan oleh penguji terlihat seperti pada Gambar 5.3</p> <ol style="list-style-type: none"> <li>7. Pengguna menyimpan tanda tangan dengan menekan tombol “Simpan”. Aplikasi menampilkan alert dialog yang menunjukkan dataset tanda tangan berhasil disimpan seperti pada Gambar 5.4</li> <li>8. Pengguna mencoba untuk menghapus tanda tangan yang telah dilakukan pada kanvas dengan menekan tombol ”hapus”. Gambar tanda tangan terlihat seperti pada Gambar 5.5, kemudian pengguna melakukan hapus tanda tangan terlihat hasil seperti pada Gambar 5.6.</li> <li>9. Pengguna melakukan langkah 4-5 sebanyak 5 kali. Kemudian pada simpan dataset tanda tangan kelima, aplikasi akan melakukan unggah secara otomatis ke server. Ketika berhasil aplikasi akan menampilkan informasi dataset tersimpan di internal storage smartphone seperti pada Gambar 5.7</li> </ol>
<b>Hasil yang diharapkan</b>	Program dapat menampilkan: jumlah dataset tersimpan 5
<b>Hasil yang diperoleh</b>	Program dapat menampilkan: jumlah dataset tersimpan 5
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Dataset berhasil disimpan



**Gambar 5.1 Halaman Kelola Dataset Tanda Tangan**



**Gambar 5.2 Halaman Dialog Kanvas**



**Gambar 5.3 Tanda Tangan Penguji**



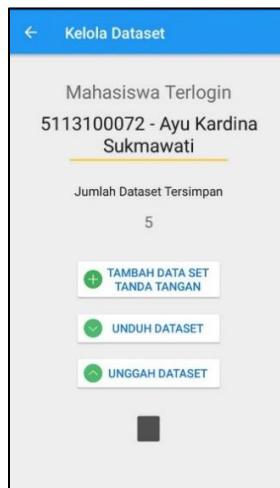
**Gambar 5.4 Alert Dialog Berhasil Simpan Dataset**



**Gambar 5.5 Tanda Tangan pada Kanvas**



**Gambar 5.6 Hapus Tanda Tangan pada Kanvas**



**Gambar 5.7 Menampilkan Informasi Jumlah Dataset**

## 5.2.2 Pengujian Fungsionalitas Melakukan Verifikasi Kehadiran

**Tabel 5.3 Skenario 1 Pengujian Fungsionalitas Melakukan Verifikasi Kehadiran**

<b>No. Pengujian</b>	UJ.UC.02
<b>Referensi Kasus Penggunaan</b>	UC.02
<b>Tujuan Pengujian</b>	Mengujian presensi kehadiran mahasiswa di perkuliahan aktif
<b>Skenario Pengujian</b>	<i>Melakukan verifikasi kehadiran dengan tanda tangan</i>
<b>Kondisi Awal</b>	Mahasiswa belum melakukan verifikasi kehadiran terhadap perkuliahan yang sedang aktif
<b>Data Uji</b>	Nama: Ayu Kardina Sukmawati, NRP : 5113100072, Kelas : Data Mining
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Penguji masuk ke halaman utama Kelola Verifikasi Tanda Tangan</li> <li>2. Penguji memilih pilihan “Verifikasi Kehadiran” kemudian akan muncul kanvas untuk tanda tangan seperti pada Gambar 5.8</li> <li>3. Penguji melakukan tanda tangan seperti pada Gambar 5.9 dan memilih tombol pilihan “simpan” jika selesai melakukan tanda tangan.</li> <li>4. Aplikasi akan menampilkan halaman informasi hasil verifikasi tanda tangan seperti pada Gambar 5.10</li> </ol>
<b>Hasil yang diharapkan</b>	Aplikasi dapat menampilkan informasi hasil verifikasi kehadiran, yaitu Nama: Ayu Kardina Sukmawati, NRP: 5113100072, Status Kehadiran: OK, Keterangan Status Kehadiran: Berhasil Melakukan Presensi
<b>Hasil yang diperoleh</b>	Aplikasi dapat menampilkan informasi hasil verifikasi kehadiran, yaitu Nama: Ayu Kardina Sukmawati, NRP: 5113100072, Status Kehadiran:

	OK, Keterangan Status Kehadiran: Berhasil Melakukan Presensi
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Status presensi kehadiran berhasil disimpan ke server



**Gambar 5.8 Halaman Dialog  
Kanvas Tanda Tangan**



**Gambar 5.9 Tanda Tangan  
Verifikasi Kehadiran**



**Gambar 5.10 Halaman Hasil Verifikasi Tanda Tangan**

### **5.3 Pengujian Kebergunaan**

Pengujian kebergunaan ini dilakukan untuk mengetahui tingkat kebergunaan aplikasi terhadap pengguna. Kebergunaan yang diuji meliputi kemudahan dalam melakukan verifikasi kehadiran dengan menggunakan tanda tangan.

Pengujian dilakukan dengan memberikan kesempatan pada pengguna untuk mencoba sendiri melakukan aplikasi yang telah dikembangkan dengan arahan dari pengembang aplikasi. Pengujian dilakukan oleh 30 pengguna yang mencoba menggunakan aplikasi ini. Pengujian ini dengan menggunakan skenario 5 kelas. Terdapat Uji coba yang dilakukan adalah siswa pada kelas tersebut melakukan verifikasi kehadiran mulai dari mengisi *dataset* hingga melakukan verifikasi. Setelah pengujian selesai pengguna diminta untuk mengisi kuesioner yang diberikan untuk mengetahui tanggapan pengguna. Rincian dari kuesioner yang diberikan dapat dilihat pada Tabel 5.4. Daftar pengguna yang melakukan pengujian dapat dilihat seperti pada Tabel 5.5.

Hasil dari kuesioner yang telah diisi oleh pengguna dapat dilihat pada Tabel 5.11. Hasil kuesioner ini dikelompokkan berdasarkan masing-masing kelas. Penilaian kuesioner ini memiliki skala 1-6 dimana semakin kecil angka skala yang dipilih berarti semakin tidak setuju, sedangkan semakin besar angka skala yang dipilih berarti semakin sangat setuju.

Hasil kuesioner masing-masing kelas dapat dilihat pada Tabel 5.6, Tabel 5.7, Tabel 5.8, Tabel 5.9 dan Tabel 5.10. Rekapitulasi hasil kuesioner kelas A, B, C, D dan E disajikan dalam bentuk prosentase dari masing-masing pertanyaan yang dapat dilihat pada Tabel 5.11.

**Tabel 5.4 Kuesioner Pengujian Kebergunaan**

No	Pertanyaan	Penilaian					
		1	2	3	4	5	6
1	Menurut saya, aplikasi ini mudah untuk digunakan						
2	Menurut saya, tidak sulit memasukkan dataset tanda tangan ke dalam aplikasi						
3	Menurut saya, presensi kehadiran dengan menggunakan aplikasi ini lebih aman dari kecurangan daripada sistem manual (absen di kertas)						
4	Menurut saya, lebih mudah dan efisien melakukan presensi kehadiran dengan menggunakan aplikasi ini daripada sistem manual (absen di kertas)						
5	Menurut saya, tidak sulit menggunakan fitur						

No	Pertanyaan	Penilaian					
		1	2	3	4	5	6
	presensi kehadiran dengan tanda tangan						
6	Menurut saya, hasil presensi kehadiran dengan fitur tanda tangan memiliki hasil yang akurat						
7	Menurut saya, presensi kehadiran dengan menggunakan fitur tanda tangan dalam aplikasi ini lebih aman dari kecurangan daripada sistem manual (absen di kertas)						
8	Menurut saya, lebih mudah dan efisien melakukan presensi kehadiran dengan menggunakan fitur tanda tangan dalam aplikasi ini daripada sistem manual (absen di kertas)						

### Keterangan

- 1 = sangat tidak setuju
- 2 = tidak setuju
- 3 = cukup setuju
- 4 = setuju
- 5 = setuju sekali
- 6 = sangat setuju sekali

**Tabel 5.5 Daftar Penguji Aplikasi**

<b>No</b>	<b>NRP</b>	<b>Nama Peserta</b>	<b>Smartphone dan Sistem Operasi</b>	<b>Kelas Skenario</b>
1	5113100051	Mardiana Sekarsari	SONY C5 + Marshmallow	Kelas A
2	5113100075	Achmad Rizky Haqiqi	SONY C5 + Marshmallow	Kelas A
3	5113100128	Relaci Aprilia Istiqomah	Xperia Z5 + Nougat	Kelas A
4	5113100172	Afif Ishamsyah Hantriono	Samsung A5 + Marshmallow	Kelas A
5	5115100062	Alvin Mudhoffar	Xiaom Redmi Note 2 + Lollipop	Kelas A
6	5113100053	Lophita Y. Napitupulu	Samsung J2 + Lollipop	Kelas B
7	5113100072	Ayu Kardina Sukmawati	Oppo A37 + Lollipop	Kelas B
8	5113100143	Nuning Septiana	Samsung J2 + Lollipop	Kelas B
9	5113100173	Andi Akram Yusuf	Sony Xperia Z1 Compact + Lollipop	Kelas B
10	5113100174	Armirara Refa Aziz	Oppo A37 + Lollipop	Kelas B
11	5113100020	Setiyo A	Samsung J5 2015 + Marshmallow	Kelas C
12	5113100151	Danang Adi N	Samsung S5 + Marshmallow	Kelas C
13	5113100083	Tities N	Lenovo VIBE + Lollipop	Kelas C
14	5113100046	Burhanudin R	Lenovo VIBE + Lollipop	Kelas C
15	5113100152	Achmad Faisal Y	Oppo A37 + Lollipop	Kelas C

No	NRP	Nama Peserta	Smartphone dan Sistem Operasi	Kelas Skenario
16	5113100090	Didit Sepiyanto	Samsung J7 + Marshmallow	Kelas D
17	5113100085	Ni Luh Made Asri Mulyasari	Samsung Galaxy Grand Prime + Lollipop	Kelas D
18	5113100109	Daniel Fablius	Samsung Galaxy A5 + Marshmallow	Kelas D
19	5113100110	I Gusti Ngurah Adi Wicaksana	Samsung Galaxy A5 + Marshmallow	Kelas D
20	5113100022	M. Syaiful Jihad A	Samsung Galaxy A5 + Marshmallow	Kelas D
21	5113100121	Cayza Angka Maulana	Oppo A37 + Lollipop	Kelas D
22	5113100122	Eriko Agnita Masroeri	Oppo A37 + Lollipop	Kelas D
23	5113100176	Abdurrachman Rafif Sayudha	Oppo A37 + Lollipop	Kelas D
24	5113100096	I Dewa Putu Ardi Nusawan	Samsung J2 + Lollipop	Kelas E
25	5113100065	Agha Maulana	Xiomi 4i + Lollipop	Kelas E
26	5113100056	Franky Setiawan Daldiri	Samsung E5 + Lollipop	Kelas E
27	5113100153	Sani Puji Rahayu	Samsung E5 + Lollipop	Kelas E
28	5113100094	Muhammad Ghulam Fajri	Xiomi 4i + Lollipop	Kelas E

No	NRP	Nama Peserta	Smartphone dan Sistem Operasi	Kelas Skenario
29	5113100049	Dhanar Prayoga	Samsung J2 + Lollipop	Kelas E
30	5113100105	Muhammad Fahrul Razi	Samsung J2 + Lollipop	Kelas E

### 5.3.1 Pengujian Kebergunaan Skenario Kelas A

Berikut ini adalah rincian pengujian kebergunaan skenario kelas A :

- Nama Kelas = ERPL (Ekonomi Rekayasa Perangkat Lunak) A
- Jam Mulai - Selesai = 15.00 – 17.30
- Tempat = IF 101

**Tabel 5.6 Tabel Hasil Penilaian Kuesioner Kelas A**

No	NRP	Nomor Pertanyaan Kuesioner							
		1	2	3	4	5	6	7	8
1	5113100172	4	5	4	3	4	4	3	4
2	5113100075	5	6	5	4	6	5	4	4
3	5113100128	5	4	6	5	5	4	3	4
4	5115100062	5	4	6	4	5	4	5	4
5	5113100051	5	6	5	5	5	5	3	4

Tabel 5.6 adalah tabel hasil penilaian kuesioner skenario kelas A dari masing-masing mahasiswa pada setiap nomor pertanyaan kuesioner. Nilai minimal dari masing-masing pertanyaan adalah 1 dan nilai maksimal adalah 6.

### 5.3.2 Pengujian Kebergunaan Skenario Kelas B

Berikut ini adalah rincian pengujian kebergunaan skenario kelas B :

- Nama Kelas = Data Mining A
- Jam Mulai - Selesai = 19.30 – 22.00
- Tempat = IF 103

Tabel 5.7 adalah tabel hasil penilaian kuesioner skenario kelas B dari masing-masing mahasiswa pada setiap nomor pertanyaan kuesioner. Nilai minimal dari masing-masing pertanyaan adalah 1 dan nilai maksimal adalah 6.

**Tabel 5.7 Tabel Hasil Penilaian Kuesioner Kelas B**

No	NRP	Nomor Pertanyaan Kuesioner							
		1	2	3	4	5	6	7	8
1	5113100053	4	4	4	4	4	4	4	4
2	5113100173	5	4	5	3	5	5	5	3
3	5113100072	5	5	6	6	5	4	4	5
4	5113100174	5	2	4	6	2	2	4	3
5	5113100143	4	4	4	4	4	4	4	4

### 5.3.3 Pengujian Kebergunaan Skenario Kelas C

Berikut ini adalah rincian pengujian kebergunaan skenario kelas C :

- Nama Kelas = TK KBJ
- Jam Mulai - Selesai = 16.00 – 18.30
- Tempat = IF 102

Tabel 5.8 adalah tabel hasil penilaian kuesioner skenario kelas C dari masing-masing mahasiswa pada setiap nomor pertanyaan kuesioner. Nilai minimal dari masing-masing pertanyaan adalah 1 dan nilai maksimal adalah 6.

**Tabel 5.8 Tabel Hasil Penilaian Kuesioner Kelas C**

No	NRP	Nomor Pertanyaan Kuesioner							
		1	2	3	4	5	6	7	8
1	5113100020	4	3	3	6	3	3	4	4
2	5113100151	4	4	4	3	4	4	4	4
3	5113100083	4	3	5	5	4	5	5	4
4	5113100046	5	4	6	5	4	4	5	5
5	5113100152	5	6	5	5	3	4	5	5

### 5.3.4 Pengujian Kebergunaan Skenario Kelas D

Berikut ini adalah rincian pengujian kebergunaan skenario kelas D :

- Nama Kelas = Komputasi Pervasif dan Jaringan Komputer
- Jam Mulai - Selesai = 13.00 – 16.00
- Tempat = IF 108

Tabel 5.9 adalah tabel hasil penilaian kuesioner skenario kelas D dari masing-masing mahasiswa pada setiap nomor pertanyaan kuesioner. Nilai minimal dari masing-masing pertanyaan adalah 1 dan nilai maksimal adalah 6.

**Tabel 5.9 Tabel Hasil Penilaian Kuesioner Kelas D**

No	NRP	Nomor Pertanyaan Kuesioner							
		1	2	3	4	5	6	7	8
1	5113100090	4	5	5	5	5	4	6	5
2	5113100085	5	5	5	5	4	4	5	5
3	5113100109	5	4	5	5	4	4	4	3
4	5113100110	4	5	4	4	4	5	4	4
5	5113100022	5	6	4	5	6	5	4	5
6	5113100121	5	5	4	4	5	4	5	5
7	5113100122	6	5	6	6	5	5	6	5

No	NRP	Nomor Pertanyaan Kuesioner							
		1	2	3	4	5	6	7	8
8	5113100176	4	5	5	5	5	4	5	5

### 5.3.5 Pengujian Kebergunaan Skenario Kelas E

Berikut ini adalah rincian pengujian kebergunaan skenario kelas E :

- Nama Kelas = Dasar Pemrograman
- Jam Mulai - Selesai = 11.00 – 13.30
- Tempat = IF 104

Tabel 5.10 adalah tabel hasil penilaian kuesioner skenario kelas E dari masing-masing mahasiswa pada setiap nomor pertanyaan kuesioner. Nilai minimal dari masing-masing pertanyaan adalah 1 dan nilai maksimal adalah 6.

**Tabel 5.10 Tabel Hasil Penilaian Kuesioner Kelas E**

No	NRP	Nomor Pertanyaan Kuesioner							
		1	2	3	4	5	6	7	8
1	5113100096	6	6	6	6	6	6	6	6
2	5113100065	4	5	5	5	4	3	5	4
3	5113100056	5	5	6	5	6	5	5	6
4	5113100153	6	6	6	6	6	6	6	6
5	5113100094	4	6	1	2	5	3	3	3
6	5113100049	4	4	3	2	3	4	2	3
7	5113100105	5	5	4	6	5	5	4	5

Hasil penilaian dari ketiga kelas dilakukan rekapitulasi untuk mendapatkan prosentase dari masing-masing pertanyaan kuesioner yaitu dengan menghitung jumlah nilai masing-masing pertanyaan kuesioner pada skenario kelas A,B,C,D dan E kemudian dibagi jumlah kuesioner yang disebar (30). Hasil

prosentase penilaian masing-masing pertanyaan keusioner dapat dilihat pada Tabel 5.11

**Tabel 5.11 Tabel Rekapitulasi Kuesioner**

No	Pertanyaan	Hasil Penilaian
1	Menurut saya, aplikasi ini mudah untuk digunakan	4.70
2	Menurut saya, tidak sulit memasukkan dataset tanda tangan ke dalam aplikasi	4.70
3	Menurut saya, presensi kehadiran dengan menggunakan aplikasi ini lebih aman dari kecurangan daripada sistem manual (absen di kertas)	4.70
4	Menurut saya, lebih mudah dan efisien melakukan presensi kehadiran dengan menggunakan aplikasi ini daripada sistem manual (absen di kertas)	4.63
5	Menurut saya, tidak sulit menggunakan fitur presensi kehadiran dengan tanda tangan	4.53
6	Menurut saya, hasil presensi kehadiran dengan fitur tanda tangan memiliki hasil yang akurat	4.26
7	Menurut saya, presensi kehadiran dengan menggunakan fitur tanda tangan dalam aplikasi ini lebih aman dari kecurangan daripada	4.40

No	Pertanyaan	Hasil Penilaian
	sistem manual (absen di kertas)	
8	Menurut saya, lebih mudah dan efisien melakukan presensi kehadiran dengan menggunakan fitur tanda tangan dalam aplikasi ini daripada sistem manual (absen di kertas)	4.36

Berdasarkan Tabel 5.11 range nilai 1 sampai 6, 30 responden kelas A,B dan C 4.70 bahwa aplikasi ini mudah untuk digunakan, 4.70 bahwa tidak sulit memasukkan dataset tanda tangan ke dalam aplikasi, 4.70 bahwa presensi kehadiran dengan menggunakan aplikasi ini lebih aman dari kecurangan daripada sistem manual (absen di kertas), 4.63 bahwa lebih mudah dan efisien melakukan presensi kehadiran dengan menggunakan aplikasi ini daripada sistem manual (absen di kertas), 4.53 bahwa tidak sulit menggunakan fitur presensi kehadiran dengan tanda tangan, 4.26 bahwa hasil presensi kehadiran dengan fitur tanda tangan memiliki hasil yang akurat, 4.40 bahwa presensi kehadiran dengan menggunakan fitur tanda tangan dalam aplikasi ini lebih aman dari kecurangan daripada sistem manual (absen di kertas), dan 4.36 bahwa lebih mudah dan efisien melakukan presensi kehadiran dengan menggunakan fitur tanda tangan dalam aplikasi ini daripada sistem manual (absen di kertas).

#### **5.4 Pengujian Waktu Eksekusi**

Pengujian waktu eksekusi ini dilakukan untuk mengetahui tingkat kecepatan aplikasi dalam melakukan proses pencocokan tanda tangan. Pengujian ini dilakukan dengan cara pengguna melakukan presensi kehadiran dengan aplikasi ini. Waktu eksekusi dihitung ketika pengguna memilih tombol

kirim (tombol yang digunakan untuk mengecek validasi tanda tangan) sampai muncul notifikasi hasil verifikasi kehadiran. Pengguna aplikasi pada uji coba ini sama dengan pengguna pada uji coba kebergunaan. Daftar pengguna yang melakukan pengujian dapat dilihat seperti pada Tabel 5.5. Hasil pengujian waktu eksekusi dapat dilihat pada Tabel 5.12

**Tabel 5.12 Hasil Pengujian Waktu Eksekusi**

No	NRP	Smartphone dan Sistem Operasi	Waktu Eksekusi	Kelas
1	5113100051	SONY C5 + Marshmallow	40 detik	Kelas A
2	5113100075	Samsung Galaxy Note 3 + Android Marshmallow	16 detik	Kelas A
3	5113100128	Xperia Z5 + Nougat	43 detik	Kelas A
4	5113100172	Samsung A5 + Marshmallow	49 detik	Kelas A
5	5115100062	Xiaom Redmi Note 2 + Lollipop	15 detik	Kelas A
6	5113100053	Samsung J2 + Lollipop	8 detik	Kelas B
7	5113100072	Oppo A37 + Lollipop	6 detik	Kelas B
8	5113100143	Samsung J2 + Lollipop	8 detik	Kelas B
9	5113100173	Sony Xperia Z1 Compact + Lollipop	6 detik	Kelas B
10	5113100174	Oppo A37 + Lollipop	8 detik	Kelas B
11	5113100020	Samsung J5 2015 + Marshmallow	35 detik	Kelas C
12	5113100151	Samsung S5 + Marshmallow	67 detik	Kelas C

No	NRP	Smartphone dan Sistem Operasi	Waktu Eksekusi	Kelas
13	5113100083	Lenovo VIBE + Lollipop	7 detik	Kelas C
14	5113100046	Lenovo VIBE + Lollipop	8 detik	Kelas C
15	5113100152	Oppo A37 + Lollipop	8 detik	Kelas C
16	5113100090	Samsung J7 + Marshmallow	40 detik	Kelas D
17	5113100085	Samsung Galaxy Grand Prime + Lollipop	8 detik	Kelas D
18	5113100109	Samsung Galaxy A5 + Marshmallow	30 detik	Kelas D
19	5113100110	Samsung Galaxy A5 + Marshmallow	34 detik	Kelas D
20	5113100022	Samsung Galaxy A5 + Marshmallow	35 detik	Kelas D
21	5113100121	Oppo A37 + Lollipop	7 detik	Kelas D
22	5113100122	Oppo A37 + Lollipop	8 detik	Kelas D
23	5113100176	Oppo A37 + Lollipop	8 detik	Kelas D
24	5113100096	Samsung J2 + Lollipop	10 detik	Kelas E
25	5113100065	Xiomi 4i + Lollipop	11 detik	Kelas E
26	5113100056	Samsung E5 + Lollipop	13 detik	Kelas E
27	5113100153	Samsung E5 + Lollipop	10 detik	Kelas E
28	5113100094	Xiomi 4i + Lollipop	12 detik	Kelas E
29	5113100049	Samsung J2 + Lollipop	10 detik	Kelas E

No	NRP	Smartphone dan Sistem Operasi	Waktu Eksekusi	Kelas
30	5113100105	Samsung J2 + Lollipop	9 detik	Kelas E

Berdasarkan data waktu eksekusi pada Tabel 5.12, terdapat perbedaan antara masing-masing kelas dan masing-masing *smartphone*. Untuk kelas A memiliki rata-rata waktu eksekusi yaitu 32.6 detik, kelas B yaitu 7.2 detik, kelas C yaitu 25 detik, kelas D yaitu 21.25 detik, sedangkan kelas E yaitu 10.71 detik. Sehingga rata-rata waktu eksekusi pada pengujian ini adalah 18.96 detik.

## 5.5 Pengujian Keakuratan

Pengujian keakuratan ini dilakukan untuk mengetahui tingkat keakuratan proses pencocokan tanda tangan pada aplikasi. Pengujian ini dilakukan dengan metode *confusion matrix* yaitu melakukan perhitungan akurasi dengan konsep data mining dengan 4 keluaran, yaitu: recall, precision, accuracy dan error rate. Skenario pengujian ini dilakukan dengan cara pengguna melakukan presensi kehadiran dengan aplikasi ini. Pengguna mendaftarkan tanda tangan ke dalam aplikasi, kemudian melakukan tanda tangan yang dianggap benar selama 10 kali kemudian dilihat hasil verifikasi yang diproses oleh aplikasi apakah mengeluarkan hasil benar atau salah. Kemudian menghitungnya. Setelah itu pengguna diminta untuk melakukan tanda tangan yang dianggap salah selama 10 kali dan menghitung hasil verifikasi yang dilakukan oleh aplikasi. Hasil pengujian waktu eksekusi terlihat pada Tabel 5.13.

**Tabel 5.13 Tabel Pengujian Keakuratan**

		Nilai Sebenarnya		
Nilai Prediksi		True	False	
	True		<i>True Positive</i>	<i>False Positive</i>
			8	2
	False		<i>False Negative</i>	<i>True negative</i>
		5	5	

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{True Positif} + \text{True Negatif}}{\text{True Positif} + \text{True Negatif} + \text{False Positif} + \text{False Negatif}} \\
 &= \frac{8+5}{8+5+2+5} \\
 &= 0.65
 \end{aligned}$$

Berdasarkan perhitungan akurasi yang didapat adalah 65 %.

## 5.6 Analisa Threshold untuk Euclidean Distance

Analisa *threshold* ini digunakan untuk menentukan batas toleransi *euclidean distance* antara gambar tanda tangan di dalam *database* dengan tanda tangan mahasiswa saat melakukan presensi. Dalam menentukan *threshold*, digunakan 5 pengguna dengan skenario melakukan tanda tangan secara benar dan melakukan tanda tangan secara sengaja salah. Tanda tangan sengaja salah adalah melakukan tanda tangan yang hampir mirip maupun berbeda dengan tanda tangan asli.

### 5.6.1 Analisa Pengguna 1

Berdasarkan Tabel 5.14 nilai minimal jarak *euclidean* untuk pencocokan gambar tanda tangan benar bernilai benar adalah 0.87, sedangkan nilai minimal jarak *euclidean* untuk tanda tangan salah bernilai salah adalah 1.07

**Tabel 5.14 Hasil Perhitungan Jarak Euclidean Pengguna 1**

No	Data	Jarak <i>Euclidean</i>	
		Tanda Tangan Benar	Tanda Tangan Salah
1	Dataset 1	1.08	1.07
2	Dataset 2	1.08	1.58
3	Dataset 3	1.14	1.33
4	Dataset 4	0.90	1.39
5	Dataset 5	0.87	1.76

### 5.6.2 Analisa Pengguna 2

Berdasarkan Tabel 5.15 nilai minimal jarak *euclidean* untuk pencocokan gambar tanda tangan benar bernilai benar adalah 0.83, sedangkan nilai minimal jarak *euclidean* untuk tanda tangan salah bernilai salah adalah 1.00

**Tabel 5.15 Hasil Perhitungan Jarak Euclidean Pengguna 2**

No	Data	Jarak <i>Euclidean</i>	
		Tanda Tangan Benar	Tanda Tangan Salah
1	Dataset 1	1.61	1.07
2	Dataset 2	1.23	1.15
3	Dataset 3	1.57	1.00
4	Dataset 4	0.83	1.11
5	Dataset 5	0.93	1.09

### 5.6.3 Analisa Pengguna 3

Berdasarkan Tabel 5.16 nilai minimal jarak *euclidean* untuk pencocokan gambar tanda tangan benar bernilai benar adalah 0.31, sedangkan nilai minimal jarak *euclidean* untuk tanda tangan salah bernilai salah adalah 1.15

**Tabel 5.16 Hasil Perhitungan Jarak Euclidean Pengguna 3**

No	Data	Jarak <i>Euclidean</i>	
		Tanda Tangan Benar	Tanda Tangan Salah
1	Dataset 1	1.64	1.43
2	Dataset 2	1.39	1.41
3	Dataset 3	1.14	1.50
4	Dataset 4	0.31	1.15
5	Dataset 5	1.27	1.24

#### 5.6.4 Analisa Pengguna 4

Berdasarkan Tabel 5.17 nilai minimal jarak *euclidean* untuk pencocokan gambar tanda tangan benar bernilai benar adalah 0.82, sedangkan nilai minimal jarak *euclidean* untuk tanda tangan salah bernilai salah adalah 0.89

**Tabel 5.17 Hasil Perhitungan Jarak Euclidean Pengguna 4**

No	Data	Jarak <i>Euclidean</i>	
		Tanda Tangan Benar	Tanda Tangan Salah
1	Dataset 1	1.05	1.32
2	Dataset 2	1.49	1.19
3	Dataset 3	0.82	1.20
4	Dataset 4	1.47	0.89
5	Dataset 5	1.52	0.89

#### 5.6.5 Analisa Pengguna 5

Berdasarkan Tabel 5.18 nilai minimal jarak *euclidean* untuk pencocokan gambar tanda tangan benar bernilai benar adalah 0.86, sedangkan nilai minimal jarak *euclidean* untuk tanda tangan salah bernilai salah adalah 1.27.

**Tabel 5.18 Hasil Perhitungan Jarak Euclidean Pengguna 5**

No	Data	Jarak <i>Euclidean</i>	
		Tanda Tangan Benar	Tanda Tangan Salah
1	Dataset 1	1.36	1.61
2	Dataset 2	0.86	1.27
3	Dataset 3	1.67	1.79
4	Dataset 4	1.52	1.73
5	Dataset 5	1.16	1.48

Berdasarkan perhitungan jarak *euclidean* dari 5 pengguna diperoleh rata-rata bahwa nilai *threshold* maksimal untuk pencocokan tanda tangan antara tanda tangan saat mahasiswa melakukan presensi dengan tanda tangan di database bernilai benar adalah 0.86.

## 5.7 Evaluasi Pengujian Fungsionalitas

Berdasarkan pengujian yang telah dilakukan, terdapat pengujian fungsionalitas memberikan hasil yang sesuai dengan skenario yang direncanakan dan juga memberikan hasil yang belum sesuai dengan skenario yang direncanakan. Evaluasi terhadap pengujian yang telah berhasil dilaksanakan sebagai berikut:

1. Fungsionalitas menambah dataset tanda tangan berjalan sesuai yang diharapkan. Pengujian UJ.UC-01, memperlihatkan bahwa sistem memberikan respon yang sesuai dengan masing-masing skenario pengujian.
2. Fungsionalitas melakukan verifikasi kehadiran dengan tanda tangan berjalan sesuai yang diharapkan. Pengujian UJ.UC-02 memperlihatkan bahwa sistem memberikan respon yang sesuai dengan masing-masing skenario pengujian.

## 5.8 Evaluasi Pengujian Kebergunaan

Berdasarkan hasil kuesioner pada Tabel 5.11 dapat ditarik kesimpulan bahwa 30 responden kelas A,B dan C 4.70 bahwa aplikasi ini mudah untuk digunakan, 4.70 bahwa tidak sulit memasukkan dataset tanda tangan ke dalam aplikasi, 4.70 bahwa presensi kehadiran dengan menggunakan aplikasi ini lebih aman dari kecurangan daripada sistem manual (absen di kertas), 4.63 bahwa lebih mudah dan efisien melakukan presensi kehadiran dengan menggunakan aplikasi ini daripada sistem manual (absen di kertas), 4.53 bahwa tidak sulit menggunakan fitur presensi kehadiran dengan tanda tangan, 4.26 bahwa hasil presensi kehadiran dengan fitur tanda tangan memiliki hasil yang akurat, 4.40 bahwa presensi kehadiran dengan menggunakan fitur tanda tangan dalam aplikasi ini lebih aman dari kecurangan daripada sistem manual (absen di kertas), dan 4.36 bahwa lebih mudah dan efisien melakukan presensi kehadiran dengan menggunakan fitur tanda tangan dalam aplikasi ini daripada sistem manual (absen di kertas).

## 5.9 Evaluasi Pengujian Waktu Eksekusi

Berdasarkan data waktu eksekusi pada Tabel 5.12, terdapat perbedaan antara masing-masing kelas dan masing-masing *smartphone*. Untuk kelas A memiliki rata-rata waktu eksekusi yaitu 32.6 detik, kelas B yaitu 7.2 detik, kelas C yaitu 25 detik, kelas D yaitu 21.25 detik, sedangkan kelas E yaitu 10.71 detik. Sehingga rata-rata waktu eksekusi pada pengujian ini adalah 18.96 detik.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil pengujian yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan perangkat lunak di masa mendatang.

#### **6.1 Kesimpulan**

1. Sistem kehadiran mahasiswa melalui pencocokan tanda tangan dapat diakses melalui perangkat *mobile* berdasarkan hasil uji UJ.UC.01 dan UJ.UC.02.
2. Pencocokan tanda tangan dengan metode normalisasi skala pada fitur gambar tanda tangan dapat digunakan dalam proses validasi kehadiran
3. Penggunaan *euclidean distance* dapat digunakan dalam proses validasi kehadiran.

#### **6.2 Saran**

1. Batas toleransi aplikasi diperbaiki lagi untuk menghindari tanda tangan yang hampir sama namun berbeda terverifikasi oleh aplikasi (akurasi kurang baik).
2. Kecepatan ketika melakukan pengecekan tanda tangan dipercepat.
3. Untuk penerapan aplikasi pada Mahasiswa Baru Teknik Informatika ITS 2017 disarankan penambahan dataset tanda tangan melalui hak akses Tata Usaha sehingga aman dari kecurangan dalam memasukkan tanda tangan.

*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- [1] A. F. Rachman, “Android Kuasai Asia Tenggara, di Indonesia Paling Juara,” *detikinet*. [Daring]. Tersedia pada: <https://inet.detik.com/consumer/d-3054169/android-kuasai-asia-tenggara-di-indonesia-paling-juara>. [Diakses: 20-Jun-2017].
- [2] “Kumpulan Aplikasi Tanda Tangan Digital,” *Tech in Asia Indonesia*, 21-Jan-2016. [Daring]. Tersedia pada: <https://id.techinasia.com/kumpulan-aplikasi-tanda-tangan-digital-gratis>. [Diakses: 12-Jul-2017].
- [3] “Login Aplikasi Menjadi Lebih Mudah dan Aman dengan PrivyID,” *Tech in Asia Indonesia*, 07-Jul-2016. [Daring]. Tersedia pada: <https://id.techinasia.com/login-aplikasi-menjadi-lebih-mudah-dan-aman-dengan-privyid>. [Diakses: 12-Jul-2017].
- [4] “Teknik Informatika ITS.” [Daring]. Tersedia pada: <http://if.its.ac.id/>. [Diakses: 30-Mei-2017].
- [5] “Sistem,” *Wikipedia bahasa Indonesia, ensiklopedia bebas*. 20-Des-2016.
- [6] “Arti kata hadir - Kamus Besar Bahasa Indonesia (KBBI) Online.” [Daring]. Tersedia pada: <http://kbbi.web.id/hadir>. [Diakses: 30-Des-2016].
- [7] “Tanda tangan,” *Wikipedia bahasa Indonesia, ensiklopedia bebas*. 15-Nov-2016.
- [8] “Pengenalan tulisan tangan,” *Wikipedia bahasa Indonesia, ensiklopedia bebas*. 29-Jan-2017.
- [9] “Image processing,” *Wikipedia*. 25-Mei-2017.
- [10] “Grayscale - Wikipedia.” [Daring]. Tersedia pada: [https://en.wikipedia.org/wiki/Grayscale#cite\\_note-1](https://en.wikipedia.org/wiki/Grayscale#cite_note-1). [Diakses: 30-Mei-2017].
- [11] “Cropping (image),” *Wikipedia*. 05-Mei-2017.
- [12] “Skewness,” *Wikipedia*. 25-Mei-2017.

- [13]“Android (operating system) - Wikipedia.” [Daring]. Tersedia pada: [https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)) . [Diakses: 30-Mei-2017].
- [14]N. Purohit, Sapna Purohit, Dan C.S. Satsangi, “Offline Handwritten Signature Verification Using Template Matching And Clustering Technique,” Vol. 2, No. 2, Hal. 4, Nov 2014.

## LAMPIRAN

### SIKEMAS

- Sistem Informasi Kehadiran Mahasiswa -  
Aplikasi dibuat untuk memenuhi Tugas Akhir S1 Teknik Informatika ITS. Aplikasi ini bertujuan untuk memberikan solusi terhadap kecurangan (Titip Absen) dalam melakukan presensi kehadiran mahasiswa di perkuliahan. Ruang lingkup aplikasi ini adalah Teknik Informatika ITS.

**\* Diperlukan**

1. Menurut saya, aplikasi ini mudah untuk digunakan \*

*Tandai hanya satu oval.*

	1	2	3	4	5	6	
Tidak Setuju	<input type="radio"/>	Sangat Setuju					

2. Menurut saya, presensi kehadiran dengan menggunakan aplikasi ini lebih aman dari kecurangan daripada sistem manual (absen di kertas) \*

*Tandai hanya satu oval.*

	1	2	3	4	5	6	
Tidak Setuju	<input type="radio"/>	Sangat Setuju					

3. Menurut saya, lebih mudah dan efisien melakukan presensi kehadiran dengan menggunakan aplikasi ini daripada sistem manual (absen di kertas) \*

*Tandai hanya satu oval.*



8. Menurut saya, presensi kehadiran dengan menggunakan fitur foto wajah dalam aplikasi ini lebih aman dari kecurangan daripada sistem manual (absen di kertas) \*

*Tandai hanya satu oval.*

	1	2	3	4	5	6	
Tidak Setuju	<input type="radio"/>	Sangat Setuju					

9. Kritik dan saran mengenai aplikasi presensi kehadiran ini \*

---

---

---

---

---

10. Kritik dan saran mengenai fitur presensi kehadiran menggunakan fitur tanda tangan\*

---

---

---

---

---

*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS



Novita Retno P L, lahir pada tanggal 3 Nopember 1994 di Sukoharjo, Jawa Tengah. Penulis menempuh pendidikan mulai dari SD Negeri Kenokorejo 1 (2001 – 2007), SMP Negeri 3 Sukoharjo (2007 – 2010) dan SMA Negeri 1 Sukoharjo (2010-2013). Saat ini penulis telah menempuh pendidikan perguruan tinggi di Institut Teknologi Sepuluh Nopember Surabaya di jurusan Teknik Informatika Fakultas Teknologi Informasi angkatan tahun 2013. Semasa kuliah, penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Computer-Informatika (HMTC), diantaranya adalah menjadi staff Departemen Dalam Negeri HMTC ITS 2014-2015, panitia Biro Dana dan Usaha Schematics HMTC ITS 2014, panitia biro Web dan Kesekretariatan Schematics HMTC 2015-2016, dan sekretaris Departemen Dalam Negeri HMTC ITS 2015-2016. Selain itu penulis juga aktif sebagai staff di *Organization Social and Responsibility* Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi (BEM FTIf). Pengalaman yang penulis dapatkan adalah pernah menjadi pemandu Latihan Keterampilan Manajemen Mahasiswa Tingkat Pra-Dasar dan Tingkat Dasar di Fakultas Teknologi Informasi pada tahun 2014- 2016.

Selama kuliah di Teknik Informatika ITS, penulis mengambil bidang minat Rekayasa Perangkat Lunak (RPL) dan menjadi administrator di Laboratorium Rekayasa Perangkat Lunak (RPL). Penulis memiliki ketertarikan pada bidang analisis perancangan sistem, arsitektur perangkat lunak, dan penjaminan mutu perangkat lunak. Penulis dapat dihubungi melalui alamat surel [nobitaretno@gmail.com](mailto:nobitaretno@gmail.com)

*[Halaman ini sengaja dikosongkan]*