

**APLIKASI *FACE RECOGNITION* UNTUK ABSENSI  
MAHASISWA DENGAN MENGGUNAKAN  
METODE *128D EMBEDDING***

**SKRIPSI**

Diajukan sebagai salah satu syarat untuk kelulusan  
Jenjang Strata Satu (S1)  
Pada Program Studi Teknik Informatika

**Disusun Oleh:**

**Anugrah Fahrul Ramadhan**

**361701006**



**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
INDONESIA MANDIRI**

**BANDUNG**

**2022**

**LEMBAR PERSETUJUAN**

**APLIKASI *FACE RECOGNITION* UNTUK ABSENSI  
MAHASISWA DENGAN MENGGUNAKAN  
METODE *128D EMBEDDING***

***FACE RECOGNITION APPLICATION FOR STUDENT  
ATTENDANCE USING 128D  
EMBEDDING METHOD***

Oleh:

**Anugrah Fahrul Ramadhan**  
**361701006**

Skripsi ini telah di terima dan disahkan  
Untuk memenuhi persyaratan mencapai gelar

SARJANA TEKNIK INFORMATIKA

Pada

PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
INDONESIA MANDIRI

Bandung, 20 Januari 2022

Disahkan Oleh

**Ketua Program Studi Teknik  
Informatika,**

**Dosen Pembimbing,**

**Chalifa Chazar, S.T., M.T.**

**NIDN: 0421098704**

**Dr. Chairuddin, Ir., M.M., M. T**

**NIDN: 0426076901**

**LEMBAR PERSETUJUAN REVISI TUGAS AKHIR**

**APLIKASI *FACE RECOGNITION* UNTUK ABSENSI  
MAHASISWA DENGAN MENGGUNAKAN  
METODE *128D EMBEDDING***

***FACE RECOGNITION APPLICATION FOR STUDENT  
ATTENDANCE USING 128D  
EMBEDDING METHOD***

Telah melakukan sidang Skripsi pada hari Rabu, 02 Februari 2022 dan telah melakukan revisi sesuai dengan masukan pada saat sidang Tugas Akhir

Menyetujui,

No	Nama	Keterangan	Tanda Tangan
1	Dr. Chairuddin, Ir., M.M., M. T .,	Pembimbing	
2	Moch. Ali Ramdhani, S.T., M.Kom.	Penguji 1	
3	Chalifa Chazar, S.T., M.T.	Penguji 2	

Bandung, 02 Maret 2022

Mengetahui,

Ketua program studi

**Chalifa Chazar, S.T., M.T.**  
NIDN. 0421098704

## **SURAT PERNYATAAN**

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar Akademik, baik di Sekolah Tinggi Manajemen & Komputer Indonesia Mandiri maupun Perguruan Tinggi lainnya.
2. Skripsi ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
3. Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah diluar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, 02 Maret 2022  
Yang Membuat Pernyataan

**Anugrah Fahrul Ramadhan**

361701006

# **APLIKASI *FACE RECOGNITION* UNTUK ABSENSI MAHASISWA DENGAN MENGGUNAKAN METODE *128D EMBEDDING***

## **ABSTRAK**

Sistem absensi saat ini memiliki beragam bentuk mulai dari tanda tangan secara manual, kartu identitas dan sidik jari. Namun, penggunaan sistem absensi ini tampaknya belum menjamin bahwa mahasiswa benar – benar hadir di kampus. Faktanya, masih ada mahasiswa yang tidak hadir dapat menitipkan absen pada temannya. Berdasarkan hal tersebut, penulis membangun aplikasi program absensi dengan menggunakan identifikasi wajah menggunakan metode *128D Embedding*, sistem kerja aplikasi yaitu dengan melakukan pendeteksian wajah secara *real-time*. Dengan cara tersebut memungkinkan untuk tidak menandatangani absen kehadiran, menunjukkan kartu identitas, dan melakukan pemindaian sidik jari. Hanya dengan menunjukkan wajah terhadap kamera mahasiswa sudah mengisi absen saat perkuliahan akan dimulai. Dengan begitu diharapkan mahasiswa tidak bisa melakukan kecurangan seperti menitipkan absen. Implementasi *Face Recognition* menggunakan metode *128D Embedding* untuk mengenali wajah – wajah yang sudah terdaftar dalam database. Kamera yang dilengkapi dengan teknologi pengenalan wajah akan bisa mengenali wajah seseorang berdasarkan data yang sudah disimpan dalam database. Dengan ditambah nya sensor suhu dapat meminimalisir kecurangan saat proses absensi sedang berlangsung. Mahasiswa yang sudah melakukan absensi kehadiran akan tersimpan dalam *website* laporan absensi.

**Kata kunci:** *Face Recognition*, *128D Embedding*, Absensi, Mahasiswa.

# **FACE RECOGNITION APPLICATION FOR STUDENT ATTENDANCE USING 128D EMBEDDING METHOD**

## **ABSTRACT**

*The current attendance system has various forms ranging from manual signatures, identity cards and fingerprints. However, the use of this attendance system does not seem to guarantee that students are actually present on campus. In fact, in this way there are still students who are not present, such as entrusting their absences to their friends. Based on this, the authors build an attendance program application using face identification using the 128D Embedding method, the application's working system is to perform real-time face detection. In this way it is possible not to sign timesheets, show identity cards, and perform fingerprint scans. Only by showing their faces to the camera, students have already filled in the absences when the lecture is about to start. In this way, it is hoped that students will not be able to commit fraud such as entrusting absenteeism. The implementation of Face Recognition uses the 128D Embedding method to recognize faces that have been registered in the database. Cameras equipped with facial recognition technology will be able to recognize a person's face based on data that has been stored in the database. With the addition of a temperature sensor, it can minimize cheating when the attendance process is in progress. Students who have made attendance attendance will be stored in the attendance report website.*

**Keyword:** *Face Recognition, 128D Embedding, Attendance, Student.*

## UCAPAN TERIMAKASIH

Dengan mengucapkan syukur Alhamdulillah, penelitian ini dapat diselesaikan untuk memenuhi syarat tugas akhir. Tugas akhir Skripsi ini dapat tersusun berkat bantuan, bimbingan dan saran-saran serta masukan dari berbagai pihak sampai selesainya laporan ini. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT dengan segala rahmat dan karunia-Nya yang memberikan kekuatan dan kelancaran bagi peneliti dalam menyelesaikan skripsi ini.
2. Kedua Orang Tua yang sangat saya cintai yang telah memberikan doa, bantuan, dukungan, kasih sayang, pengorbanan dan semangat kepada Penulis dalam menyelesaikan tugas akhir skripsi.
3. Bapak Dr. Chairuddin, Ir., M.M., M. T ., selaku ketua Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri sekaligus juga sebagai Dosen Pembimbing. Terimakasih telah meluangkan waktu, memberikan bimbingan, arahan, dan saran kepada penulis dalam menyelesaikan skripsi ini.
4. Ibu Chalifa Chazar, S,T., M.M., M.T., selaku Ketua Program Jurusan Teknik Informatika.
5. Seluruh dosen dan staf Sekolah Tinggi Indonesia Mandiri Bandung yang telah memberikan ilmu dan fasilitas pada penulis hingga dapat menyelesaikan skripsi ini.

6. Sepupu saya Naufal Dzaki Hafizhan yang telah membantu penulis selama dalam proses pembuatan skripsi.
7. Seluruh teman - teman penulis yang sama - sama berjuang dalam menyusun skripsi, memberikan semangat dan saling membantu satu sama lain.
8. Semua pihak yang tidak dapat disebutkan satu persatu, terima kasih atas segalanya.

Penulis mengucapkan terima kasih kepada pihak-pihak yang telah membantu dalam penyelesaian laporan tugas akhir ini. Semoga laporan tugas akhir ini dapat bermanfaat dan berguna bagi semua pihak.



## KATA PENGANTAR

Bismillahirrahmaanirrahim, Puji dan Syukur penulis panjatkan kepada Allah SWT karena atas taufiq dan hidayah-Nya, penulis dapat menyelesaikan skripsi yang berjudul “**APLIKASI *FACE RECOGNITION* UNTUK ABSENSI MAHASISWA DENGAN MENGGUNAKAN METODE *128D EMBEDDING***”. Skripsi ini disusun sebagai syarat untuk menyelesaikan jenjang pendidikan Strata Satu (S1) di Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri. Penulis menyadari dalam penulisan Laporan Tugas Akhir ini, baik bentuk maupun isinya masih terdapat banyak kekurangan karena keterbatasan pengetahuan dan kemampuan serta pengalaman yang penulis miliki. Untuk itu dengan kerendahan hati penulis mengharapkan kritik dan saran yang membangun untuk kemajuan penulis dikemudian hari. Penulis berharap tugas akhir ini bisa menjadi salah satu sumber yang dapat bermanfaat.

Bandung, 02 Maret 2022  
Yang Membuat Pernyataan

**Anugrah Fahrul Ramadhan**

361701006

## DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PERSETUJUAN REVISI TUGAS AKHIR.....	ii
ABSTRAK .....	iv
ABSTRACT.....	v
UCAPAN TERIMA KASIH.....	vi
KATA PENGANTAR .....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR .....	xv
BAB I .....	1
PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Identifikasi Masalah .....	4
1.3 Tujuan Penelitian.....	4
1.4 Batasan Masalah.....	4
1.5 Metode Penelitian.....	5
1.5.1 Metode Pengumpulan Data.....	5
1.5.2 Metode Pengembangan Sistem.....	6
1.6 Sistematika Penulisan.....	10
BAB II.....	12
LANDASAN TEORI .....	12

2.1 Wajah.....	12
2.2 <i>Face Recognition</i> .....	12
2.3 <i>Machine Learning</i> .....	14
2.4 Klasifikasi <i>Machine Learning</i> .....	18
2.5 Sistem Intejelensi Visual .....	20
2.6 Visi Komputer .....	21
2.7 Pengolahan Citra .....	22
2.8 <i>Object Detection</i> .....	23
2.9 <i>Deep-Learning</i> .....	24
2.10 <i>Python</i> .....	26
2.11 <i>128D Embedding</i> .....	27
2.12 <i>FaceNet</i> .....	28
2.13 <i>Microcontroller</i> .....	31
2.14 <i>Arduino</i> .....	34
2.15 <i>Website</i> .....	34
2.16 Basis Data ( <i>Data Base</i> ) .....	35
2.16.1 <i>MySQL</i> .....	36
2.17 Entity Relationship Diagram (ERD) .....	37
2.18 UML ( <i>Unified Modeling Language</i> ) .....	38
2.19 Metode Waterfall.....	52
2.20 PHP .....	55
2.21 <i>Laravel</i> .....	56
2.22 <i>Blackbox Testing</i> .....	56

BAB III .....	58
PEMBAHASAN .....	58
3.1 <i>Requirement</i> (Analisis Kebutuhan) .....	59
3.1.2 Analisis Permasalahan.....	62
3.1.3 Gambaran Umum Sistem Yang Diusulkan .....	63
3.2 <i>Design System</i> (Desain Sistem).....	80
3.2.1 Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak .....	82
3.3 <i>Coding &amp; Testing</i> (Penulisan Sinkode Program / Implementasi) .....	83
3.3.1 <i>Use Case Diagram</i> .....	84
3.3.2 Activity Diagram .....	99
3.3.3 <i>Sequence Diagram</i> .....	109
3.3.4 Class Diagram .....	118
3.3.5 Deployment Diagram .....	120
3.3.6 <i>Entity Relationship Diagram (ERD)</i> .....	120
3.3.7 <i>Design Interface</i> .....	124
BAB IV .....	129
IMPLEMENTASI DAN UJI COBA .....	129
4.1 <i>Integration &amp; Testing</i> (Penerapan / Pengujian Program) .....	129
4.1.1 Implementasi .....	129
4.1.1.1 Tampilan Aplikasi.....	129
4.1.1.2 Tampilan Website.....	132
4.2 Testing.....	139
BAB V.....	144

PENUTUP.....	144
5.1 Kesimpulan.....	144
5.2 Saran.....	145
DAFTAR PUSTAKA .....	146
LAMPIRAN.....	151

## DAFTAR TABEL

<b>Tabel 2.1</b> Hasil Penelitian tingkat persentase error (Jingtuo Liu, Yafeng Deng, Tao Bai, Zhengping Wei, & Chang Huang, 2015).....	28
<b>Tabel 2.2</b> Kardinalitas pada ERD versi James Martin .....	38
<b>Tabel 2.3</b> Simbol-simbol <i>Use Case Diagram</i> (Rosa dan Shalahuddin, 2016:155)....	39
<b>Tabel 2.4</b> Simbol-simbol <i>Activity Diagram</i> (Rosa dan Shalahuddin, 2016:162).....	43
<b>Tabel 2.5</b> Simbol-simbol <i>Sequence Diagram</i> (Rosa dan Shalahuddin, 2018:165)....	45
<b>Tabel 2.6</b> Simbol-simbol <i>Class Diagram</i> (Rosa dan Shalahuddin, 2016:146) .....	47
<b>Tabel 2.7</b> Simbol-simbol <i>Statechart Diagram</i> (Rosa dan Shalahuddin, 2016:163)...	49
<b>Tabel 2.8</b> Simbol-simbol <i>Deployment Diagram</i> (Rosa dan Shalahuddin, 2016:154) 51	
<b>Tabel 3.1</b> Referensi Penelitian.....	60
<b>Tabel 3.2</b> Penjadwalan Penelitian.....	81
<b>Tabel 3.3</b> Spesifikasi Kebutuhan Perangkat Keras.....	82
<b>Tabel 3.4</b> Spesifikasi Kebutuhan Perangkat Lunak.....	83
<b>Tabel 3.5</b> Deskripsi <i>Actor</i> .....	84
<b>Tabel 3.6</b> Deskripsi <i>Use Case</i> .....	85
<b>Tabel 3.7</b> <i>Flow of Event Scan Object</i> .....	88
<b>Tabel 3.8</b> <i>Flow of Event Face Recognition</i> .....	89
<b>Tabel 3.9</b> <i>Flow of Event</i> Deteksi Suhu .....	90
<b>Tabel 3.10</b> <i>Flow of Event</i> Hasil Absensi Kehadiran .....	91
<b>Tabel 3.11</b> <i>Flow of Event Login</i> .....	92
<b>Tabel 3.12</b> <i>Flow of Event</i> Laporan Absensi <i>User</i> .....	93
<b>Tabel 3.13</b> <i>Flow of Event</i> Laporan Absensi <i>Admin</i> .....	94
<b>Tabel 3.14</b> <i>Flow of Event</i> Data Mahasiswa.....	95
<b>Tabel 3.15</b> <i>Flow of Event Input</i> Absensi .....	96
<b>Tabel 3.16</b> <i>Flow of Event</i> Histori Absensi <i>User</i> .....	97
<b>Tabel 3.17</b> <i>Flow of Event</i> Histori Absensi <i>Admin</i> .....	98
<b>Tabel 3.18</b> Tabel Mahasiswa.....	121

<b>Tabel 3.19</b> Tabel Dosen.....	121
<b>Tabel 3.20</b> Tabel Matakuliah.....	122
<b>Tabel 3.21</b> Tabel Ruangan.....	122
<b>Tabel 3.22</b> Tabel Data Absensi .....	122
<b>Tabel 3.23</b> Tabel Laporan Absensi.....	123
<b>Tabel 3.24</b> Tabel Histori Absensi.....	123
<b>Tabel 4.1</b> Rencana Pengujian .....	140
<b>Tabel 4.2</b> Tabel Hasil Pengujian <i>Blackbox</i> .....	141

## DAFTAR GAMBAR

<b>Gambar 1.1</b> Metode <i>Waterfall</i> (Trisianto, 2018).....	7
<b>Gambar 2.1</b> Manusia vs mesin dalam mengambil keputusan (Ibnu Daqiqil Id, 2021) .....	15
<b>Gambar 2.2</b> Menganalisa Kemiripan wajah dengan <i>128D Embedding</i> .....	30
<b>Gambar 2.3</b> Mengubah dari gambar menjadi 128 angka <i>vector</i> (Arsfutura, 2019)...	31
<b>Gambar 2.4</b> Contoh <i>Use Case Diagram</i> .....	42
<b>Gambar 2.5</b> Contoh <i>Activity Diagram</i> .....	44
<b>Gambar 2.6</b> Contoh <i>Sequence Diagram</i> .....	46
<b>Gambar 2.7</b> Contoh <i>Class Diagram</i> .....	48
<b>Gambar 2.8</b> Contoh <i>Statechart Diagram</i> .....	50
<b>Gambar 2.9</b> Contoh <i>Deployment Diagram</i> .....	52
<b>Gambar 2.10</b> Metode <i>Waterfall</i> (Trisianto, 2018).....	53
<b>Gambar 3.1</b> Arsitektur Teknologi Dari Aplikasi <i>Face Recognition</i> .....	64
<b>Gambar 3.2</b> Proses Pengambilan <i>Image</i> .....	65
<b>Gambar 3.3</b> Membuat Labelling <i>Image</i> .....	66
<b>Gambar 3.4</b> Penginputan NIM dan Nama Mahasiswa.....	67
<b>Gambar 3.5</b> Hasil Penginputan Disimpan Di <i>Database Mysql</i> .....	67
<b>Gambar 3.6</b> Proses Pembuatan Notifikasi Suara.....	68
<b>Gambar 3.7</b> <i>Hardware</i> Arduino Uno R3.....	69
<b>Gambar 3.8</b> Kabel <i>Jumper Male to Female</i> .....	70
<b>Gambar 3.9</b> Sensor Suhu MLX90614.....	71
<b>Gambar 3.10</b> Arduino Uno Sebelum Pemasangan Sensor Suhu.....	72
<b>Gambar 3.11</b> Arduino Uno Sesudah Pemasangan Sensor Suhu.....	73
<b>Gambar 3.12</b> <i>Code</i> Program Suhu.....	74
<b>Gambar 3.13</b> Hasil Uji Coba Sensor Suhu.....	75
<b>Gambar 3.14</b> <i>Code</i> Program <i>128D Embedding</i> .....	76
<b>Gambar 3.15</b> Sistem Mempelajari Sample.....	77



<b>Gambar 3.16</b>	Hasil <i>Learning Sample</i> Oleh Sistem.....	78
<b>Gambar 3.17</b>	Arsitektur <i>128D Embedding</i> (Arsfutura, 2019).....	79
<b>Gambar 3.18</b>	<i>Use Case Diagram</i> Aplikasi S.M.A.R.T Sense.....	86
<b>Gambar 3.19</b>	<i>Use Case Diagram Website</i> S.M.A.R.T Sense.....	87
<b>Gambar 3.20</b>	<i>Activity Diagram Scan Object</i> .....	99
<b>Gambar 3.21</b>	<i>Activity Diagram Face Recognition</i> .....	100
<b>Gambar 3.22</b>	<i>Activity Diagram Deteksi Suhu</i> .....	101
<b>Gambar 3.23</b>	<i>Activity Diagram Hasil Absensi Kehadiran</i> .....	102
<b>Gambar 3.24</b>	<i>Activity Diagram Login</i> .....	103
<b>Gambar 3.25</b>	<i>Activity Diagram Laporan Absensi User</i> .....	104
<b>Gambar 3.26</b>	<i>Activity Diagram Laporan Absensi Admin</i> .....	105
<b>Gambar 3.27</b>	<i>Activity Diagram Data Mahasiswa</i> .....	106
<b>Gambar 3.28</b>	<i>Activity Diagram Input Absensi</i> .....	107
<b>Gambar 3.29</b>	<i>Activity Diagram Histori Absensi User</i> .....	108
<b>Gambar 3.30</b>	<i>Activity Diagram Histori Absensi Admin</i> .....	109
<b>Gambar 3.31</b>	<i>Sequence Diagram Scan Object</i> .....	110
<b>Gambar 3.32</b>	<i>Sequence Diagram Face Recognition</i> .....	111
<b>Gambar 3.33</b>	<i>Sequence Diagram Deteksi Suhu</i> .....	112
<b>Gambar 3.34</b>	<i>Sequence Diagram Hasil Face Recognition</i> .....	113
<b>Gambar 3.35</b>	<i>Sequence Diagram Hasil Deteksi Suhu</i> .....	113
<b>Gambar 3.36</b>	<i>Sequence Diagram Login</i> .....	114
<b>Gambar 3.37</b>	<i>Sequence Diagram Laporan Absensi User</i> .....	114
<b>Gambar 3.38</b>	<i>Sequence Diagram Laporan Absensi Admin</i> .....	115
<b>Gambar 3.39</b>	<i>Sequence Diagram Data Mahasiswa</i> .....	116
<b>Gambar 3.40</b>	<i>Sequence Diagram Input Absensi</i> .....	117
<b>Gambar 3.41</b>	<i>Sequence Diagram Histori Absensi User</i> .....	117
<b>Gambar 3.42</b>	<i>Sequence Diagram Histori Absensi Admin</i> .....	118
<b>Gambar 3.43</b>	<i>Class Diagram S.M.A.R.T Sense</i> .....	119
<b>Gambar 3.44</b>	<i>Deployment Diagram Aplikasi S.M.A.R.T Sense</i> .....	120

<b>Gambar 3.45</b> <i>Entity Relationship Diagram</i> Aplikasi S.M.A.R.T Sense .....	121
<b>Gambar 3.46</b> Perancangan <i>Interface</i> Aplikasi S.M.A.R.T Sense.....	124
<b>Gambar 3.47</b> Perancangan <i>Home</i> .....	125
<b>Gambar 3.48</b> Perancangan <i>Login</i> .....	125
<b>Gambar 3.49</b> Perancangan <i>Dashboard</i> .....	126
<b>Gambar 3.50</b> Perancangan Menu Laporan Absensi .....	126
<b>Gambar 3.51</b> Perancangan Menu Data Mahasiswa .....	127
<b>Gambar 3.52</b> Perancangan Menu <i>Input</i> Absensi .....	127
<b>Gambar 3.53</b> Perancangan Menu Histori Absensi .....	128
<b>Gambar 4.1</b> Tampilan Wajah Dikenal.....	130
<b>Gambar 4.2</b> Tampilan Wajah Dikenal.....	130
<b>Gambar 4.3</b> Tampilan Wajah Dikenal dan Sensor Suhu.....	131
<b>Gambar 4.4</b> Tampilan Wajah Dikenal Saat Memakai Masker.....	131
<b>Gambar 4.5</b> Tampilan Wajah Tidak Dikenal.....	132
<b>Gambar 4.6</b> Tampilan <i>Home</i> .....	133
<b>Gambar 4.7</b> Tampilan <i>Login</i> .....	133
<b>Gambar 4.8</b> Tampilan <i>Dashboard</i> Laporan Absensi.....	134
<b>Gambar 4.9</b> Tampilan <i>Dashboard</i> Data Mahasiswa .....	135
<b>Gambar 4.10</b> Tampilan <i>Dashboard Input</i> Absensi.....	135
<b>Gambar 4.11</b> Tampilan <i>Dashboard</i> Histori Absensi .....	136
<b>Gambar 4. 12</b> Tampilan Menu Laporan Absensi .....	137
<b>Gambar 4.13</b> Tampilan Menu Data Mahasiswa .....	138
<b>Gambar 4.14</b> Tampilan Menu <i>Input</i> Absensi .....	138
<b>Gambar 4.15</b> Tampilan Menu Histori.....	139

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan ilmu dan teknologi yang ada pada saat ini, banyak karakteristik data biologis manusia yang digunakan dalam berbagai macam keperluan. Hal ini dikarenakan ciri biologis setiap manusia berbeda yang dapat memberikan informasi berkaitan dengan identifikasi masing - masing individu, seperti yang ada pada tubuh manusia berupa sidik jari, retina, pola suara dan pola wajah (*face recognition*). *Face recognition* merupakan salah satu teknik pengenalan wajah yang sama seperti sidik jari dan retina mata, dimana hasil tangkapan kamera akan dicocokkan dengan foto dan lekuk wajah yang sudah ada di dalam database. *Face recognition* juga termasuk salah satu teknologi biometrik yang telah dipelajari dan dikembangkan oleh para ahli, karena menggunakan algoritma pengenalan wajah untuk membedakan individu yang satu dengan lainnya berdasarkan data yang sudah ada didalam database wajah. Teknologi biometrik merupakan salah satu ciri khas yang dapat digunakan pada suatu sistem keamanan, yakni pengenalan wajah sebagai identitas data. Wajah manusia mempunyai banyak informasi dan mempunyai karakteristik paling khas serta banyak digunakan untuk identitas seseorang (Munawir, Liza Fitria, & Muhammad Hermansyah, 2020).

Teknologi biometrik menawarkan autentikasi secara biologis yang memungkinkan sistem dapat mengenali penggunanya lebih tepat. Sistem pengenalan biometrika (*biometrics recognition sistem*) merupakan sistem otentikasi (*authentication sistem*) dengan menggunakan biometrika. Sistem biometrika akan melakukan pengenalan secara otomatis atas identitas seseorang berdasarkan suatu ciri biometrika dengan mencocokkan ciri tersebut dengan ciri biometrika yang telah disimpan pada basis data. Sebagai suatu sistem otentikasi, sistem biometrika mampu memutuskan apakah hasil pengenalan itu sah atau tidak sah, diterima atau ditolak, dikenali atau tidak dikenali (Ni Wayan Marti & Kadek Yota Ernanda Aryanto, 2016).

Pada dasarnya, sistem pengenalan wajah bekerja dengan membandingkan citra masukan (*input*) dengan citra yang telah tersimpan dalam sebuah database dan menemukan kecocokan wajah yang paling sesuai dengan data masukan yang ada sebelumnya (Muhammad Athoillah, 2017).

Sistem absensi adalah hal yang penting dalam sebuah perkuliahan sebagai pencatat kehadiran mahasiswa. Sistem absensi dapat diartikan sebagai catatan kehadiran atau keikutsertaan mahasiswa dalam mengikuti aktivitas di perkuliahan. Pencatatan kehadiran dalam dunia pendidikan sangat penting dilakukan untuk mengetahui dan mengontrol kehadiran para mahasiswa dalam proses belajar mengajar.

Proses absensi mahasiswa di STMIK - IM masih menggunakan sistem absensi manual yakni menggunakan kertas sebagai catatan kehadiran mahasiswa, metode absensi tersebut masih belum efektif karena menimbulkan kecurangan saat melakukan proses absensi seperti menitipkan absen pada temannya dan juga material kertas yang mudah rusak. Apabila sistem absensi menggunakan *fingerprint* akan muncul beberapa permasalahan yang biasanya disebabkan oleh *human error*, misalnya ada kotoran pada jari, luka pada jari, jari dalam kondisi basah, dan lain – lain. Hal tersebut menyebabkan tidak dapat dikenalnya sidik jari dengan baik. Dan juga di mesin pemindaian sidik jari akan ada bakteri dan kuman hasil dari pemakaian beberapa pengguna. Dalam kondisi pandemi Covid-19 kebersihan harus dijaga dengan ketat khususnya pada tangan agar terhindar dari berbagai kuman dan bakteri. Maka dari itu, penulis memilih *face recognition* karena tidak perlu berhubungan langsung dengan perangkat keras, hanya dengan menunjukkan wajah pada kamera maka pengguna langsung terdaftar pada laporan catatan kehadiran mahasiswa.

Dengan menggunakan metode *128D Embedding* diharapkan sistem dapat mengenali wajah mahasiswa secara *real-time* dan juga penulis menambahkan sensor suhu dengan menggunakan mikrokontroler Arduino Uno untuk meminimalisir kecurangan apabila ada yang menggunakan foto wajah. Serta dengan adanya sistem tersebut dapat mengefisienkan waktu. Maka judul penelitian yang diambil adalah “***APLIKASI FACE RECOGNITION UNTUK***

## ***ABSENSI MAHASISWA DENGAN MENGGUNAKAN METODE 128D EMBEDDING”.***

### **1.2 Identifikasi Masalah**

Berdasarkan latar belakang di atas, maka identifikasi masalah dari penelitian ini adalah:

1. Bagaimana cara merancang sistem absensi dengan pengenalan wajah dan suhu objek menggunakan metode *128D Embedding* dan Arduino Uno?
2. Bagaimana cara mengimplementasikan metode pengenalan wajah *128D Embedding* kedalam sistem absensi *face recognition*?

### **1.3 Tujuan Penelitian**

Berdasarkan identifikasi masalah di atas, maka tujuan dari penelitian ini adalah:

1. Mengetahui cara pembuatan rancangan *face recognition* dengan menggunakan metode *128D Embedding* serta alat – alat apa saja yang dibutuhkan untuk membuat absensi berbasis *face recognition* untuk meminimalisir kecurangan dengan menggunakan foto dari *handphone*.
2. Dapat melakukan absen dengan menekan tombol absen secara *real-time* serta mengintegrasikan sistem absensi dengan database.

### **1.4 Batasan Masalah**

Dalam melakukan suatu penelitian di perlukan adanya batasan suatu masalah agar penelitian tersebut lebih terarah dan memudahkan dalam pembahasan

sehingga tujuan penelitian akan tercapai. Beberapa batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Aplikasi yang dibangun menggunakan metode *128D Embedding*, berbasis desktop, dan menggunakan bahasa pemrograman *Python*.
2. Sistem yang dibangun berupa prototipe dan tidak terintegrasi dengan sistem informasi yang ada di STMIK – IM.
3. Sistem yang dibangun tidak terikat dengan dosen, mata kuliah dan ruangan. Data – data tersebut hanya *diinput* secara manual.
4. Wajah yang akan dideteksi tidak terhalangi objek apapun seperti kacamata, rambut, masker, dan lain – lain.
5. Pencahayaan ruangan tidak boleh terlalu terang maupun gelap.
6. Penelitian dilakukan hanya sampai tahap pengujian.
7. Data – data yang digunakan penelitian ini adalah data *dummy*.

## **1.5 Metode Penelitian**

### **1.5.1 Metode Pengumpulan Data**

Metode pengumpulan data yang dilakukan untuk mendapatkan data data dan referensi yang dibutuhkan dalam menyusun penelitian ini, yakni:

1. Studi Kepustakaan

Dilakukan dengan cara membaca dan mempelajari buku-buku, referensi yang mendukung dengan topik dengan menghimpun informasi yang relevan pada topik atau masalah yang menjadi obyek penelitian. Informasi tersebut dapat diperoleh dari buku-buku, karya ilmiah, tesis, disertasi, ensiklopedia, internet, dan sumber-sumber lain yang akan dibahas dalam penyusunan skripsi ini.

## 2. Studi Lapangan

### a. Observasi

Melakukan observasi ke kampus STMIK – IM untuk mendapatkan data – data yang diperlukan.

### b. Wawancara

Melakukan wawancara kepada pihak yang bersangkutan untuk mendapatkan data yang diperlukan dalam penelitian.

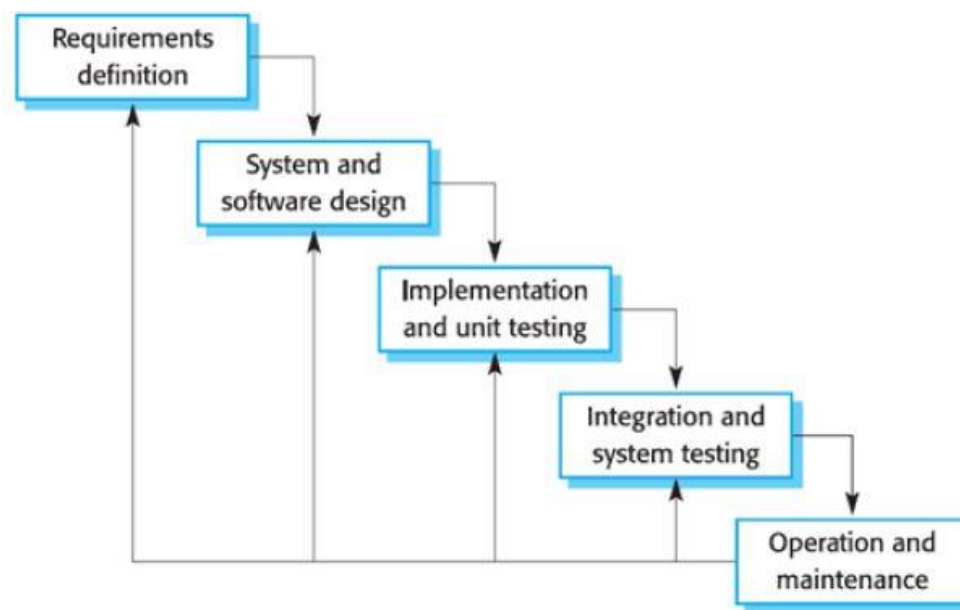
### **1.5.2 Metode Pengembangan Sistem**

Dalam penelitian ini peneliti menggunakan metode penelitian kuantitatif. Prosedur ilmiah terstandar berarti itu adalah satu seri tahap-tahap yang dirancang dan diikuti, dengan tujuan menemukan jawaban untuk isu atau masalah yang menjadi perhatian peneliti. Tahap-tahap diikuti secara logis dalam aluran. Tahap-tahap tersebut adalah identifikasi masalah yang dipelajari, melakukan telaah literatur untuk memahami bagaimana hal itu dipahami dan



didekati, mengumpulkan data, menganalisis data dan membuat kesimpulan dan generalisasi (Ulber Silalahi, 2015).

Kerangka berpikir dalam penelitian ini menggunakan metode pengembangan SDLC (*Systems Development Life Cycle*) dengan Model *Waterfall*.



**Gambar 1.1** Metode Waterfall (Trisianto, 2018)

Berikut ini adalah penjelasan dari tahapan-tahapan yang dilakukan dalam Model *Waterfall*:

1. *Requirement definition* (Analisis Kebutuhan)

Dalam langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah

penelitian, wawancara atau *study literatur*. Seseorang *system analisis* akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas – tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan sistem analisis untuk menterjemahkan kedalam bahasa pemrograman (Trisianto, 2018).

## 2. *System and software design* (Desain Sistem)

Proses *design* akan menterjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya (Trisianto, 2018).

## 3. *Implementation and unit testing* (Penulisan Sinkode Program / Implementation)

*Coding* merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menterjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem.

Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan - kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki (Trisianto, 2018).

4. *Intergration and system testing* (Penerapan / Pengujian Program)

Tahapan ini bisa dikatakan *final* dalam pembuatan sebuah sistem. Setelah melakukan analisa, *design* dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user* (Trisianto, 2018).

5. *Operation and Maintenance* (Operasi dan Pemeliharaan)

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional (Trisianto, 2018).

Penulis tidak akan melakukan penelitian pada tahap *Operation and Maintenance*, penulis hanya melakukan hingga tahap empat yaitu Penerapan / Pengujian Program.

## 1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan skripsi ini adalah sebagai berikut:

### **BAB I PENDAHULUAN**

Bab ini berisi latar belakang, identifikasi masalah, tujuan, batasan masalah, metode penelitian dan sistematika penulisan mengenai Aplikasi *Face Recognition* dengan menggunakan metode *128d Embedding*.

### **BAB II LANDASAN TEORI**

Bab ini menjelaskan landasan teori dan kumpulan studi pustaka yang relevan dengan topik penelitian. Terdapat landasan teori yang mendasari pembangunan aplikasi seperti pengertian mengenai *face recognition*, *machine learning*, absensi, analisis sistem, pengolahan data, dan komponen – komponen yang digunakan untuk perancangan dan pembangunan aplikasi absensi *Face Recognition*.

### **BAB III ANALISIS DAN PERANCANGAN SISTEM**

Bab ini menjelaskan proses awal pembangunan perangkat lunak yang terdiri dari analisis kebutuhan dan perancangan. Penulis memaparkan perancangan dari sistem yang akan dibuat dalam bentuk diagram UML, analisis kebutuhan, alur sistem yang sedang berjalan dan yang diusulkan, perancangan basis data serta perancangan antarmuka.

#### **BAB IV IMPLEMENTASI DAN UJI COBA**

Bab ini membahas tentang tahapan akhir dari pembangunan perangkat lunak, yaitu pengujian perangkat lunak berdasarkan analisis dan perancangan yang telah dibuat. Pengujian dan implementasi menggunakan *Blackbox Testing*.

#### **BAB V KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dari hasil penelitian yang dilakukan dan membahas saran – saran untuk pengembangan perangkat lunak kedepannya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Wajah**

Wajah merupakan salah satu ciri khas identik yang dimiliki setiap manusia agar dapat dikenali orang lain. Wajah terutama digunakan untuk menentukan ekspresi, penampilan, serta identitas dari seseorang. Tidak ada satu wajah pun yang serupa mutlak, bahkan pada manusia kembar identik sekalipun. (Fajri, Rizky, T. Rahmad Effendi, & Nurul Fadillah, 2020).

#### **2.2 Face Recognition**

Pendeteksian wajah merupakan proses segmentasi area wajah dengan latar belakang dari suatu citra masukan. Proses ini bekerja dengan cara memeriksa citra yang dimasukkan parameternya apakah memiliki ciri wajah atau tidak. Jika memiliki ciri wajah, maka akan dilakukan proses pemisahan citra wajah dengan latar belakang citra yang dimasukkan (Muhammad Karunia Rahmadhika & Ahmad Muhammad Thantawi. 2021).

Pengenalan Wajah adalah teknologi biometrik yang memungkinkan untuk memverifikasi wajah seseorang melalui gambar digital dengan cara mencocokkan tekstur lengkung wajah dengan data wajah yang disimpan di database (Mutaqqin. 2020).

Berbeda dengan teknologi biometrik lainnya, pengenalan wajah dapat digunakan dalam kasus-kasus yang melibatkan banyak orang sekaligus, misalnya dalam pencarian orang hilang atau dalam kasus DPO (*People Search List*). Ada 3 tahapan dalam melakukan Pengenalan Wajah menurut Komal Vora. (2015:1) yaitu:

1. *Face detection*: pada tahap ini mendeteksi ada tidaknya wajah pada gambar atau video yang dimasukkan.
2. *Feature Extraction*: setelah wajah terdeteksi, ekstraksi fitur dilakukan untuk mendapatkan fitur wajah.
3. *Face Recognition*: tahap terakhir adalah pengenalan wajah dengan membandingkan wajah yang memiliki karakteristik yang diketahui dengan wajah yang ada di database.

Skenario pengenalan wajah dapat diklasifikasikan menjadi 2 jenis, yaitu:

1. *Face Verification* ("Am I what I say I am?") Adalah pencocokan satu-ke-satu yang membandingkan gambar *Query Face* dengan gambar wajah kerangka yang diklaim benar. Untuk mengevaluasi tingkat verifikasi kemampuan verifikasi (parameter yang diberikan akses kepada pengguna yang valid) dibandingkan dengan tingkat penerimaan palsu (parameter yang diberikan akses kepada pengguna palsu) kemudian diplot ke dalam kurva *ROC*. Sistem verifikasi yang baik adalah sistem yang memiliki perbandingan yang seimbang antara dua parameter dalam pengoperasiannya.

2. Identifikasi wajah ("*Who am I?*") Adalah pencocokan satu-ke-banyak yang membandingkan gambar wajah kueri dengan semua templat yang ada dalam database untuk menentukan identitas gambar kueri. Identifikasi dilakukan dengan mencari wajah di database yang memiliki tingkat kemiripan paling tinggi dengan citra *query*. Fitur yang dinormalisasi dari subjek uji dibandingkan dengan fitur lain yang ada di database dan akan diperoleh angka kecocokan untuk setiap perbandingan. Angka, citra yang memiliki tingkat kemiripan tertinggi akan diperoleh, yang merupakan identitas yang dicari.

### **2.3 Machine Learning**

Menurut Ibnu Daqiqil Id. (2021:7) *Machine Learning* (ML) merupakan bidang studi yang fokus kepada desain dan analisis algoritma sehingga memungkinkan komputer untuk dapat belajar. Menurut Samuel, ML berisi sebuah algoritma yang bersifat *generic* (umum) dimana algoritma tersebut dapat menghasilkan sesuatu yang menarik atau bermanfaat dari sejumlah data tanpa harus menulis kode yang spesifik. Pada intinya, algoritma yang generik tersebut ketika diberikan sejumlah data maka ia dapat membangun sebuah aturan atau model atau inferensi dari data tersebut. Sebagai contoh sebuah Algoritma untuk mengenali tulisan tangan dapat digunakan untuk mendeteksi email yang berisi spam dan bukan spam tanpa mengganti kode. Algoritma yang sama ketika diberikan data pelatihan yang berbeda menghasilkan logika klasifikasi yang berbeda.



Machine learning juga dapat diartikan sebuah komputer yang memiliki kemampuan belajar tanpa diprogram secara eksplisit. Program tersebut memanfaatkan data untuk membangun model dan mengambil keputusan berdasarkan model yang telah dibangun.



**Gambar 2.1** Manusia vs mesin dalam mengambil keputusan (Ibnu Daqiqil Id, 2021)

Selain Samuel, Mitchel(1997) juga memberikan sebuah definisi ringkas dan jelas mengenai ML dimana *Machine Learning* adalah “Satu program komputer yang dikatakan telah melakukan pembelajaran dari pengalaman **E** (*Experience*) terhadap tugas **T** (*Task*) dan mengukur peningkatan kinerja **P** (*Performance Measure*), jika kinerja Tugas T diukur oleh kinerja P, maka meningkatkan pengalaman E”. Dari definisi ini Mitchel dapat dikatakan sebuah aplikasi Machine Learning memiliki 3 komponen yaitu *Task T*, *Performance Measure P*, dan *Experience E*. Oleh karena itu, untuk membangun sebuah aplikasi ML maka komponen T, P dan E harus dapat diidentifikasi.

*Task T* merupakan objektif dari program ML yang kita buat. Program ML membantu kita untuk mengatasi masalah yang sulit untuk diselesaikan oleh program

tradisional yang bersifat statis. Berikut ini beberapa contoh *Task* yang ada program ML:

1. **Klasifikasi** adalah suatu pengelompokan data di mana data yang digunakan tersebut mempunyai kelas label atau target. Pada jenis task ini, program komputer diminta untuk “menebak” atau memilih kelompok/kategori/kelas dari data. Pada task ini kelas telah ditetapkan sebelumnya. Contohnya adalah klasifikasi emosi pada foto manusia dengan tiga kelas yaitu *Happy*, *Angry* dan *Neutral*. Aplikasi tersebut harus memilih salah satu (dalam kasus tertentu bisa lebih dari satu) dari tiga kelas tersebut berdasarkan foto yang diberikan.
2. **Regresi** sederhananya adalah sebuah tugas di mana aplikasi ML diminta untuk menebak angka (bilangan continue) berdasarkan sejumlah data. Sebagai contoh kita ingin melakukan prediksi harga saham, maka luaran dari prediksi tersebut adalah sebuah nilai harga. Jadi salah satu perbedaan antara regresi dan klasifikasi adalah objek yang diprediksi, regresi memprediksi angka sedangkan klasifikasi memprediksi kelas.
3. **Transkripsi** adalah sebuah tipe *Task* dimana aplikasi ML mencoba “memahami” sebuah data yang tidak terstruktur yang merepresentasikan data lain. Sebagai contoh pengenalan tulisan pada gambar dimana gambar berisi data pixel yang tidak terstruktur, tetapi data tersebut merepresentasikan sebuah tulisan atau huruf-huruf. Contoh lainnya adalah aplikasi text to speech dimana

data berupa gelombang berisi teks. Proses mengubah gelombang suara menjadi teks adalah transkripsi.

4. ***Machine Translation***. *Task* ini merupakan proses mengubah sekumpulan input terurut yang berisi simbol-simbol tertentu menjadi simbol tertentu di bahasa lainnya. Contoh aplikasinya penerjemah bahasa Indonesia dan bahasa Inggris.
5. ***Anomaly Detection***. *Task* ini bertujuan untuk menggunakan aplikasi ML untuk memeriksa sejumlah data atau event lalu menandainya sebagai sesuatu yang tidak biasa.
6. ***Syntesis dan Sampling***. *Task* ini bertujuan untuk membuat atau generate sesuatu berdasarkan contoh. Misalnya aplikasi ML yang dapat menghasilkan sebuah gambar pemandangan berdasarkan sketsa tertentu.

Untuk mengevaluasi kinerja aplikasi ML maka diperlukan sebuah metode pengukuran kinerja yang kualitatif. Setiap *task* memiliki metode pengukuran yang berbeda sebagai contoh antara klasifikasi dan regresi cara mengukur performanya berbeda. Klasifikasi mengukur kinerja berdasarkan perbandingan jumlah tebakan benar dan salah sedangkan regresi dinilai berdasarkan kedekatannya dengan nilai asli dari suatu tebakan.

Aplikasi atau model ML mendapatkan *experience* berdasarkan dataset yang disediakan pada proses pelatihan. Sebuah dataset adalah kumpulan contoh-contoh yang harus dipelajari oleh komputer agar dapat menyelesaikan *Task*-nya.

## 2.4 Klasifikasi *Machine Learning*

*Machine learning* dapat dikelompokkan berdasarkan bagaimana cara belajar sehingga dapat melakukan tugasnya. Pembagian *Machine learning* berdasarkan cara belajarnya dibagi menjadi tiga kelompok yaitu :

### 1. *Supervised Learning*

Secara bahasa, supervised learning adalah pembelajaran terarah/terawasi. Jika kita analogikan pada proses pembelajaran, komputer atau mesin akan mempelajari data training yang berisi label. Jika dianalogikan ke siswa dan guru, dimana komputer sebagai siswa yang belajar maka guru akan meminta siswa untuk belajar dari soal yang sudah memiliki solusi dan kunci jawaban.

Selain klasifikasi, regresi juga dapat dimasukkan sebagai pembelajaran terarah. Adapun contoh-contoh algoritma yang termasuk ke dalam kategori supervised adalah *Linear Regression* (Regresi Linear), *Logistic Regression* (Regresi Logistik), *Linear Discriminant Analysis*, *k-Nearest Neighbors*, *Support Vector Machines* (SVMs), *Random Forests*, dan *Neural networks*.

### 2. *Unsupervised Learning*

Berbeda dengan *supervised learning*, *unsupervised learning* adalah kebalikannya dimana proses pembelajaran dilakukan tanpa petunjuk. Algoritma dalam komputerlah yang bekerja untuk menemukan pola pola di dalam data. Secara matematis, *unsupervised learning* terjadi ketika kita

memiliki sejumlah data masukan (X) dan tanpa variabel output yang berhubungan. Jika kita menggunakan analogi yang sama dengan siswa tadi maka, siswa belajar tanpa ada solusi dan jawaban yang benar, siswa harus menemukan sendiri jawabannya. Masalah *Unsupervised learning* dapat dibagi menjadi dua jenis yaitu *asosiasi* dan *clustering*.

### 3. *Reinforcement Learning*

Sebuah komputer akan berinteraksi dengan sebuah lingkungan yang sangat dinamis dimana komputer tersebut harus melakukan sebuah tugas tertentu contohnya pada permainan catur atau *self-driving car*. Melalui sebuah algoritma, mesin akan mempelajari bagaimana membuat keputusan yang spesifik berdasarkan lingkungan yang berubah-ubah.

Selain berdasarkan cara belajar, machine learning dapat dikategorikan berdasarkan proses pembelajaran dilakukan secara bertahap (*Batch Learning*) atau secara langsung (*on the fly - Online Learning*). Prinsip dasar dari *online learning* adalah menghasilkan sebuah model yang melakukan proses pembelajaran data baru secara realtime atau mendekati realtime. Sedangkan *Batch Learning*, data training akan dipecah-pecah menjadi beberapa bagian lalu setiap bagiannya akan dipelajari secara terpisah pada waktu yang berbeda.

*Machine learning* berdasarkan bagaimana cara kerjanya dapat dikolompokkan menjadi dua yaitu

- *Instance-based learning* (atau sering disebut *memory-based learning*) adalah sebuah kelompok algoritma ML yang bekerja dengan membandingkan data testing dengan data yang telah dipelajari pada proses training. Algoritma kelompok ini tidak membuat sebuah generalisasi tetapi lebih ke arah perbandingan dengan data yang disimpan di memori. Contoh algoritma yang masuk ke kelompok ini adalah *k-nearest neighbors*, *kernel machines*, dan *RBF network*.
- *Model based learning* adalah kebalikan dari instance-based dimana menggunakan memori untuk melakukan pemecahan masalah, algoritma ini membuat sebuah model yang bersifat generik.

## 2.5 Sistem Intejelensi Visual

Menurut Bloei Zhou (2018:24) Sistem intelijensi visual merupakan model kecerdasan buatan yang berusaha menyamai tingkat pengenalan visual manusia, bisa mempelajari, mengidentifikasi, dan menguraikan struktur yang mendasari faktor penjelasan pada *input* data yang diamati.

Memahami dunia visual dalam satu tatapan adalah salah satu prestasi kecerdasan manusia yang paling menakjubkan. Dengan mengambil sampel adegan puluhan kali per detik, kami terpapar jutaan gambar alami dalam setahun. Pengalaman visual yang kaya dan beragam yang terkandung dalam konteks adegan ini memandu penafsiran kita tentang dunia dan membentuk pengetahuan kita tentang realitas. Terinspirasi oleh ini, model pembelajaran mendalam baru-baru ini membangun

representasi internal mereka dalam pendekatan pembelajaran berbasis data dari jutaan sampel pelatihan dari dataset adegan skala besar. Digeneralisasi dari jutaan sampel pelatihan, banyak model pengenalan visual yang didukung oleh jaringan saraf yang dalam telah mencapai kinerja tingkat manusia.

Model AI mencapai pengakuan visual tingkat manusia, sehingga terdapat kemungkinan besar sistem dapat belajar mengidentifikasi dan menguraikan faktor-faktor penjelas yang mendasari data input yang diamati. Pertanyaan utama tentang pembelajaran representasi yang dapat ditafsirkan adalah apa saja faktor penjelas yang mendasari ketika sistem dilatih untuk mengenali objek atau adegan, dan lebih jauh lagi, apakah faktor-faktor ini bermakna dan dapat ditafsirkan bagi manusia atau tidak.

Dalam pelaksanaannya, sistem intelijensi visual sangatlah erat dengan visi komputer.

## **2.6 Visi Komputer**

Visi Komputer merupakan sub disiplin ilmu dari kecerdasan buatan yang mempelajari bagaimana mesin dapat mengenali objek yang diamati atau diobservas. Dapat juga disebut sebagai machine vision, ilmu ini mengembangkan teori-teori dan algoritma dimana informasi yang berguna diekstraksi dan dianalisis dari sebuah citra penelitian, sekumpulan citra, atau citra yang berurutan dari sebuah komputasi yang dibuat oleh komputer.

Visi komputer dapat juga diartikan sebagai berikut :

1. Otomatis dan integrasi sebuah range yang luas yang terdiri dari proses-proses dan representasi-representasi terhadap persepsi visual.
2. Visi komputer berhubungan dengan perolehan gambar, pemrosesan, klasifikasi, pengenalan, dan menjadi penggabungan, pengurutan pembuatan keputusan menuju pengenalan.
3. Visi komputer sesuai dengan sifatnya, merupakan suatu subyek yang merangkul berbagai disiplin tradisional secara luas guna mendasari prinsip-prinsip formalnya, dan dalam mengembangkan suatu metodologi yang berlainan dari apa yang dimilikinya, pertama-tama harus mengembangkan dan secara berurutan membangun materi yang mendasari ini.
4. Visi komputer adalah suatu bidang yang bertujuan untuk membuat suatu keputusan yang berguna mengenai objek fisik nyata dan keadaan berdasarkan atas sebuah citra. Visi komputer merupakan kombinasi antara pengolahan citra dan pengenalan pola. Hasil keluaran dari proses visi komputer adalah pengertian tentang citra. Boyle dan Thomas (C & Boyle, 1988), mengatakan bahwa computer vision lebih daripada pengenalan, computer vision melakukan operasi “low level processing” sebagai algoritma image processing yang murni. Mereka juga yang menggolongkan image processing ke dalam computer vision.

## **2.7 Pengolahan Citra**

Menurut Muwardi & Fadlil (2018:125) Pengolahan citra merupakan bidang yang bersifat multi disiplin, yang terdiri dari banyak aspek, antara lain: fisika (optik,



nuklir, gelombang, dll), elektronika, matematika, seni, fotografi, dan teknologi komputer.

Pengolahan citra (*image processing*) mempunyai keterikatan yang erat dengan disiplin ilmu yang jika sebuah disiplin ilmu dinyatakan dalam bentuk proses suatu *input* menjadikan *output*, maka pengolahan citra memiliki input berupa citra serta *output* berupa citra.

Pengolahan citra digital atau yang lebih dikenal dengan pengolahan citra merupakan proses memodifikasi citra digital. Pengolahan citra bertujuan memperbaiki kualitas citra digital sehingga dapat digunakan pada proses selanjutnya seperti pengenalan pola, computer vision atau hanya sekedar memperbaiki kualitasnya saja (R. C. Gonzalez dan R. E. Woods, 2018). Pada pengolahan citra terdapat beberapa prinsip dasar untuk mengolah sebuah citra digital yakni, peningkatan citra, restorasi citra, pemrosesan citra berwarna, penghilangan derau, segmentasi, representasi dan deskripsi serta pencarian bentuk atau pengenalan objek (R. M. Thanki dan A. M. Kothari, 2018).

## **2.8 Object Detection**

Menurut Schneiderman, Henry dan Kanade (2000:746) *Object Detection* adalah bagian besar dari kehidupan orang-orang. Kita, sebagai manusia, terus-menerus "mendeteksi" berbagai benda seperti orang, bangunan, dan mobil. Namun itu tetap menjadi misteri bagaimana kita mendeteksi objek secara akurat dan dengan sedikit usaha yang jelas. Penjelasan komperhensif telah menentang psikolog dan fisiolog selama lebih dari satu abad.

*Object Detection* memiliki banyak kegunaan potensial termasuk pengambilan gambar. Koleksi gambar digital telah tumbuh secara dramatis ditahun terakhir. Corbis memperkirakan memiliki lebih dari 67 juta gambar dalam koleksinya. The Associated Press mengumpulkan dan mengarsipkan sekitar 1.000 foto sehari. Saat ini, kegunaan dari koleksi ini dibatasi oleh kurangnya metode retrieval yang efektif. Untuk menemukan gambar tertentu dalam koleksi semacam itu, orang harus mencari menggunakan teks berbasis keterangan dan fitur gambar primitif seperti warna dan tekstur.

Object Detection dapat digunakan untuk mengekstraksi lebih banyak informasi dari gambar-gambar dan membantu memberi label dan mengkategorikannya. Metode pencarian yang ditingkatkan akan membuat database dapat diakses oleh kelompok pengguna yang lebih luas, seperti lembaga penegak hukum, praktisi medis, desainer grafis dan multimedia, dan seniman.

Object Detection juga bisa berguna dalam fotografi. Ketika teknologi kamera berubah dari film ke capture digital, kamera akan menjadi bagian optik dan bagian computer.

## **2.9 Deep-Learning**

Menurut Danukusumo (2017:2) *Deep Learning* merupakan salah satu bidang dari *Machine Learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik *Deep Learning* memberikan arsitektur yang sangat kuat untuk *Supervised Learning*. Dengan menambahkan lebih

banyak lapisan maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik. Pada *Machine Learning* terdapat teknik untuk menggunakan ekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasi citra maupun untuk mengenali suara. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi.

Aplikasi konsep jaringan syaraf tiruan yang dalam (banyak lapisan) dapat ditanggihkan pada algoritma *Machine Learning* yang sudah ada sehingga komputer sekarang bisa belajar dengan kecepatan, akurasi, dan skala yang besar. Prinsip ini terus berkembang hingga *Deep Learning* semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak masalah data besar seperti *Computer vision*, *Speech recognition*, dan *Natural Language Processing*. *Feature Engineering* adalah salah satu fitur utama dari Deep Learning untuk mengekstrak pola yang berguna dari data yang akan memudahkan model untuk membedakan kelas. *Feature Engineering* juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi. Namun, sulit untuk dipelajari dan dikuasai karena kumpulan data dan jenis data yang berbeda memerlukan pendekatan teknik yang berbeda juga.

Algoritma yang digunakan pada *Feature Engineering* dapat menemukan pola umum yang penting untuk membedakan antara kelas. Dalam *Deep Learning*, metode CNN atau *Convolutional Neural Network* sangatlah bagus dalam menemukan fitur yang baik pada citra ke lapisan berikutnya untuk membentuk hipotesis nonlinier yang

dapat meningkatkan kekompleksitasan sebuah model. Model yang kompleks tentunya akan membutuhkan waktu pelatihan yang lama sehingga di dunia *Deep Learning* penggunaan GPU sudah sangatlah umum.

### **2.10 Python**

*Python* menurut Akbar Nur Syahrudin dan Tedi Kurniawan (2018:2) adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. *Python* juga didukung oleh komunitas yang besar.

*Python* mendukung multi paradigma pemrograman, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada *python* adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, *python* umumnya digunakan sebagai bahasa *script* meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa *script*. *Python* dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi.

Saat ini kode *python* dapat dijalankan di berbagai platform sistem operasi, beberapa diantaranya adalah:

- Linux/Unix
- Windows
- Mac OS X
- Java Virtual Machine
- Amiga
- Palm
- Symbian (untuk produk-produk Nokia)

*Python* didistribusikan dengan beberapa lisensi yang berbeda dari beberapa versi. Namun pada prinsipnya Python dapat diperoleh dan dipergunakan secara bebas, bahkan untuk kepentingan komersial. Lisensi *Python* tidak bertentangan baik menurut definisi Open Source maupun *General Public License* (GPL).

### **2.11 128D Embedding**

Menurut Semiha Makinist , Betül Ay, dan Galip Aydın (2020:523) *Embedding* adalah representasi dari dokumen, kata atau gambar dalam ruang 2D atau 3D. Dengan kata lain, dokumen, kata-kata atau gambar (benda, manusia, wajah, dll) direpresentasikan secara vektor dalam ruang dua dimensi. Representasi wajah sebagai vektor numerik ini disebut “*Face Embeddings*”. Beberapa metode untuk membuat Face Embedding, salah satunya yaitu *Deep Neural Networks*.

Dalam penelitian yang berjudul *Targeting Ultimate Accuracy: Face Recognition via Deep Embedding*, Jingtuo Liu, Yafeng Deng, Tao Bai, Zhengping Wei,

dan Chang Huang. (2015) mereka melakukan penelitian *Face Recognition* melalui *Deep Embedding*. Mereka mengumpulkan sampel sebanyak 1,2 Juta foto wajah, berikut table hasil penelitian:

<i>Identities</i>	<i>Faces</i>	<i>Error rate</i>
1.5K	150K	3.1%
9K	450K	1.35%
18K	1.2M	0.87%

**Tabel 2.1** Hasil Penelitian tingkat persentase error (Jingtuo Liu, Yafeng Deng, Tao Bai, Zhengping Wei, & Chang Huang, 2015)

Dari sini dapat disimpulkan bahwa membuat *Face Recognition* dengan menggunakan metode *128D Embedding* semakin banyak sampel foto wajah ditambahkan maka semakin kecil tingkat *error* untuk mengenali wajah.

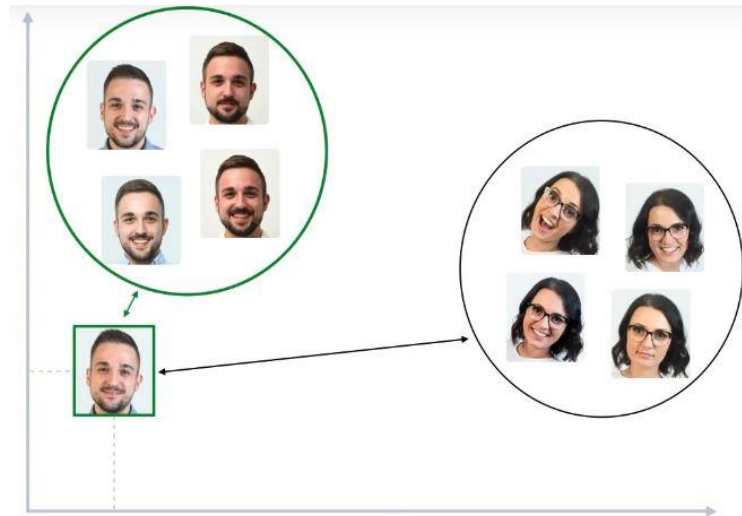
### 2.12 FaceNet

Menurut Preetha S. dan Sheela S. V. (2021:5) Teknologi *FaceNet* menggunakan gambar untuk mengubah wajah menjadi *128D Euclidian* atau bisa juga disebut *Embedding*. Model *FaceNet* adalah model yang dilatih untuk membedakan perbedaan wajah apakah wajah tersebut mirip dengan sampel atau berbeda. Gambar wajah dapat dikelompokkan secara efektif dengan memanfaatkan penyematan *128D Embedding*. Dengan vektor ruang yang dibuat terdiri dari penyematan di tempat pengelompokan, verifikasi, dan pengenalan wajah, maka untuk implementasi akan lebih mudah dengan menggunakan model tersebut.

*Embedding* dapat dibuat saat model dilatih untuk mengenal wajah dengan cara menggunakan foto wajah sebagai sampel. Untuk menyamakan dua gambar, *embedding* untuk kedua foto wajah dibuat terpisah setelah itu rumus di atas untuk mencari jarak nilai terkecil untuk wajah serupa dan nilai tertinggi untuk wajah berbeda.

Menurut Arsfutura (2019) *FaceNet* adalah jaringan saraf dalam yang digunakan untuk mengekstrak fitur dari gambar wajah seseorang. *FaceNet* mengambil gambar wajah orang tersebut sebagai *input* dan *output* vektor 128 angka yang mewakili fitur paling penting dari wajah. Dalam *Machine Learning*, vektor ini disebut *embedding*. Semua informasi penting dari sebuah gambar disematkan ke dalam vektor ini. Pada dasarnya, *FaceNet* mengambil wajah seseorang dan mengkompresnya menjadi vektor 128 angka. *Embeddings* adalah vektor dan kita dapat menafsirkan vektor sebagai titik dalam sistem koordinat *Cartesian*.

Salah satu cara yang mungkin untuk mengenali seseorang pada gambar yang tidak terlihat adalah dengan menghitung penyisipannya, memperhitungkan jarak kemiripan pada gambar orang yang dikenal dan jika penyematan wajah cukup mirip dengan penyematan orang A, kita katakan bahwa gambar ini berisi wajah orang A.



**Gambar 2.2** Menganalisa Kemiripan wajah dengan *128D Embedding*

(Arsfutura, 2019)

Untuk melatih *FaceNet* kita membutuhkan banyak gambar wajah. Untuk lebih sederhana nya, kita memiliki dua buah gambar wajah orang. Logika yang sama dapat diterapkan jika kita memiliki ribuan gambar orang yang berbeda. Pada awal pelatihan, *FaceNet* menghasilkan vektor acak untuk setiap gambar yang berarti gambar tersebar secara acak saat diplot.

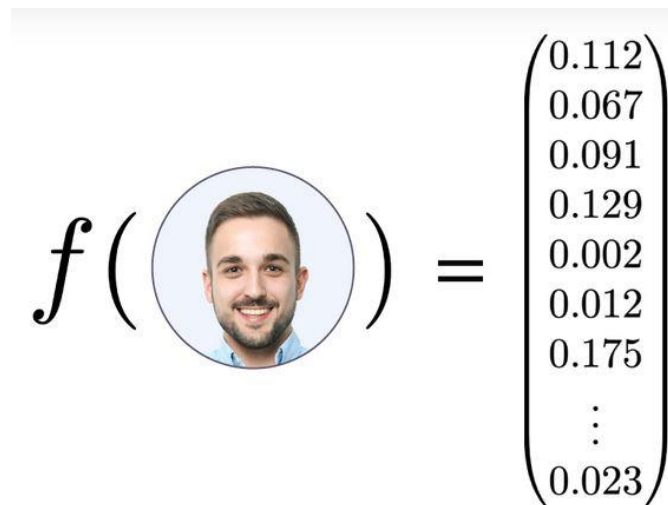
*FaceNet* belajar dengan cara berikut:

1. Memilih gambar sesuai dengan sampel.
2. Secara acak memilih gambar orang yang mirip dengan sampel gambar. (Contoh Positif).
3. Secara acak memilih gambar seseorang yang berbeda dari sampel gambar. (Contoh Negatif).



4. Menyesuaikan parameter jaringan *FaceNet* sehingga contoh positif lebih dekat ke sampel gambar dibandingkan dengan contoh negatif.

*FaceNet* perlu mengidentifikasi fitur utama wajah seseorang yang membedakannya dari wajah yang berbeda. *FaceNet* mencoba banyak kombinasi berbeda selama *training* hingga menemukan yang paling cocok dengan sampel gambar.



$$f\left(\text{Image of a man's face}\right) = \begin{pmatrix} 0.112 \\ 0.067 \\ 0.091 \\ 0.129 \\ 0.002 \\ 0.012 \\ 0.175 \\ \vdots \\ 0.023 \end{pmatrix}$$

**Gambar 2.3** Mengubah dari gambar menjadi 128 angka vector (Arsfutura, 2019)

Dari gambar diatas yakni sebuah fungsi yang mengambil gambar sebagai *input* dan *output* berupa angka vektor dari *face embedding*. *FaceNet* akan membandingkan wajah yang tertera pada kamera dengan gambar sampel yang paling mirip.

### 2.13 Microcontroller

Menurut Angki Nopebriansyah Pratama (2015:18) *Microcontroller* merupakan komputer di dalam chip yang digunakan untuk mengontrol peralatan elektronik, yang

menekankan efisiensi dan efektifitas biaya. Secara teknis *microcontroller* terbagi 2 jenis yaitu RISC dan CISC yang masing – masing mempunyai keluarga. RISC (*Reduced Instruction Set Computer*) yaitu instruksi terbatas namun dengan fasilitas yang lebih banyak. CISC kepanjangan dari (*Complex Instruction Set Computer*) yaitu instruksi lebih lengkap namun dengan fasilitas terbatas.

Jadi, *microcontroller* adalah sebuah alat yang mengerjakan instruksi yang dibuat oleh programmer. Program menginstruksikan *microcontroller* untuk melakukan jalinan yang panjang dari aksi – aksi sederhana untuk melakukan tugas yang lebih kompleks sesuai keinginan programmer. Beberapa fitur yang umumnya ada didalam *microcontroller*, yaitu:

1. RAM

RAM merupakan singkatan dari *Random Acces Memory*. RAM digunakan sebagai tempat penyimpanan variabel. *Memory* ini bersifat *volatile* yang berarti akan kehilangan semua datanya jika tidak mendapat catu daya.

2. ROM

ROM merupakan singkatan dari *Read Only Memory*. ROM juga sering disebut sebagai *code memory* karena berfungsi sebagai tempat penyimpanan program yang diberikan *programmer*.

3. Register

Register adalah tempat penyimpanan nilai – nilai yang akan digunakan dalam proses. Data - data yang disimpan dalam register bersifat sementara.

#### 4. SFR

SFR merupakan singkatan dari *Special Function Register*. SFR adalah register khusus yang berfungsi mengatur jalannya *microcontroller*. SFR ini terletak pada RAM.

#### 5. *Input dan Output Pin*

Pin input berfungsi sebagai penerima signal dari luar, pin ini dapat dihubungkan dengan media *input* seperti sensor. Pin *output* adalah bagian yang berfungsi untuk mengeluarkan signal dari hasil proses algoritma *microcontroller*.

#### 6. *Interrupt*

*Interrupt* berfungsi sebagai bagian yang dapat melakukan intrupsi. Ketika program utama sedang berjalan, program utama tersebut dapat diinterupsi secara internal.

#### 7. *External Interrupt*

*External Interrupt* adalah interupsi yang berasal dari luar sistem komputer. Interupsi akan terjadi bila ada input dari pin *interrupt*.

#### 8. *Interrupt Timer*

Intrupsi akan terjadi pada saat tertentu sesuai waktu yang ditentukan. Misalnya digunakan untuk penundaan selama satu detik yang dalam Bahasa pemrograman dituliskan dengan kata “*delay*” dengan satuan *milisecond*.

#### 9. *Interrupt serial*

*Intrupsi serial* terjadi ketika menerima data pada saat komunikasi serial atau ketika register dalam keadaan penuh pada proses *receive*. Proses *receive* yaitu dimana processor menerima data *serial* dari luar.

#### **2.14 Arduino**

*Arduino* adalah *kit elektronik* atau papan rangkaian elektronik *open source* yang didalamnya terdapat komponen utama, yaitu sebuah *chip microcontroller* dengan jenis AVR dari perusahaan Atmel. *Microcontroller* itu sendiri adalah *chip* atau IC (*integrated circuit*) yang biasa di program dengan komputer tujuan menanamkan program pada *microcontroller* adalah agar rangkaian elektronik dapat membaca *input*, memproses *input* tersebut dan kemudian menghasilkan *output* sesuai yang diinginkan. (Rosmanila, Radillah, & Sofiyan, 2018).

#### **2.15 Website**

Menurut Arief (2011:8) Pengertian *Website* adalah kumpulan dari halaman web yang sudah dipublikasikan di jaringan internet dan memiliki domain/URL (*Uniform Resource Locator*) yang dapat diakses semua pengguna internet dengan cara mengetikan alamatnya. Hal ini dimungkinkan dengan adanya teknologi *World Wide Web* (WWW).

Halaman *website* biasanya berupa dokumen yang ditulis dalam *format Hyper Text Markup Language* (HTML), yang bisa diakses melalui HTTP, HTTPS adalah

suatu protokol yang menyampaikan berbagai informasi dari *server website* untuk ditampilkan kepada para *user* atau pemakai melalui *web browser*.

Menurut Taufik (2017:2) menyatakan bahwa website atau situs merupakan tempat penyimpanan data dan informasi dengan menggunakan topik tertentu. Diumpamakan situs web ini adalah sebuah buku yang berisikan sebuah topik tertentu, website atau situs web juga merupakan kumpulan dari halaman-halaman web yang saling berkaitan didalam web tersebut.

Secara umum ada beberapa bahasa pemrograman yang digunakan untuk membuat aplikasi website. Adapun bahasa program yang di pakai sebagai berikut: HTML (*Hyper Text Markup Language*), PHP (*PHP Hypertext Preprocessor*), CSS (*Cascading Style Sheet*), *Javascript*, *Mysql*, *Jquery*.

### **2.16 Basis Data (*Data Base*)**

Menurut Fathansyah (2012:2) Basis data terdiri dari 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah represtasi fakta nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Sebagai satu kesatuan istilah, Basis Data (*Database*) sendiri dapat didefinisikan, himpunan kelompok data (arsip) yang saling berhubungan yang

diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

Berikut merupakan nama beberapa aplikasi database yang dikategorikan menurut komersi dan nonkomersil:

1. Aplikasi database komersil seperti: *DB2, Microsoft SQL Server, Oracle, Sybase* dan *Teradata*.
2. Aplikasi database nonkomersi seperti: *MySQL, Firebird, PostgreSQL*.

### **2.16.1 MySQL**




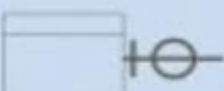
*MySQL* adalah sebuah *server database open source* yang terkenal yang digunakan berbagai aplikasi terutama untuk server atau membuat *webservice*. *MySQL* berfungsi sebagai *Structured Query Language (SQL)* yang dimiliki sendiri dan sudah diperluas oleh *MySQL* yang umumnya digunakan bersamaan dengan *PHP* untuk membuat aplikasi *server* yang dinamis dan *powerfull*. Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja dalam melakukan proses perintah-perintah *SQL* yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai pengoperasi basisdata, *MySQL* mendukung operasi basisdata transaksional maupun operasi basisdata nontransaksional. Pada modus operasi nontransaksional, *MySQL* dapat dikatakan unggul dalam hal untuk kerja di bandingkan perangkat lunak pengoperasi basisdata kompetitor lainnya. Namun demikian pada modus nontransaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan,

karenanya modus nontransaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi *blogging* berbasis *web* (*wordpress*), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja *MySQL* pada modus transaksional tida secepat unjuk kerja pada modus non-transaksional. (Muhammad Irsan, 2016:2)

### **2.17 Entity Relationship Diagram (ERD)**

Menurut Aprilia Cerry Natalin Rovita (2017:3) ERD adalah diagram yang menggambarkan keterhubungan antar data secara konseptual. Penggambaran keterhubungan antar data ini didasarkan pada anggapan bahwa dunia nyata dari kumpulan objek yang disebut entitas (*entity*), dan hubungan yang terjadi diantaranya yang disebut relasi (*relationship*).

**Tabel 2.2** Kardinalitas pada ERD versi James Martin

Simbol	Arti
	Satu dan hanya satu (One and only one)
	Satu atau lebih (One or more)
	Tidak ada atau lebih (Zero or more)
	Tidak ada atau satu (Zero or one)

### 2.18 UML (*Unified Modeling Language*)

Rosa dan Shalahudin (2016:133), menjelaskan bahwa UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan *design*, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori *object oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar team programmer maupun dengan pengguna.



UML mempunyai sejumlah elemen grafis yang bias dikombinasikan menjadi diagram. UML memiliki sejumlah aturan untuk menggabungkan / mengkombinasikan elemen-elemen tersebut.

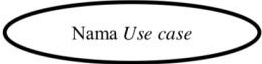
Berikut adalah diagram yang terdapat di dalam UML:




### 1. *Use Case Diagram*



*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

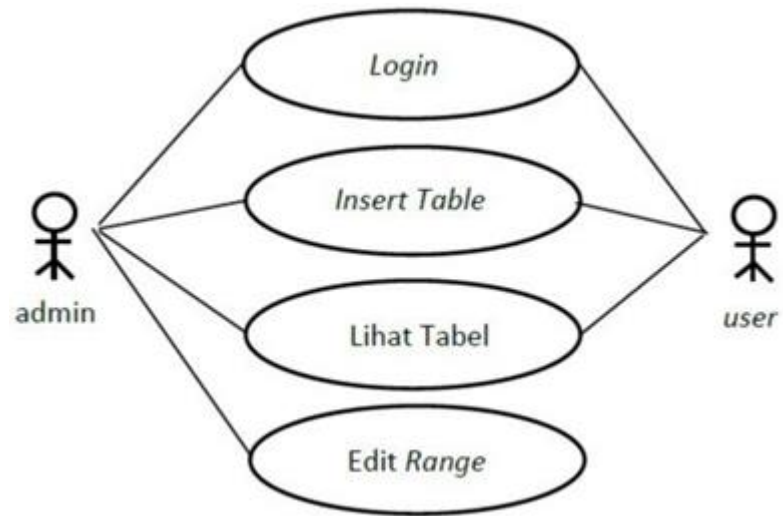
**Tabel 2.3** Simbol-simbol *Use Case Diagram* (Rosa dan Shalahuddin, 2016:155)

<p>Use Case</p> 	<p>Fungsionalitas yang disediakan <i>system</i> sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i>.</p>
---	---

<p>Aktor / Actor</p> 	<p>Orang, proses, atau <i>system</i> lain yang berinteraksi dengan system informasi yang akan dibuat di luar <i>system</i> informasi yang akan dibuat itu sendiri jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Asosiasi / Association</p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p>Ekstensi / Extend</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri</p>

	sendiri walau tanpa <i>use case</i> tambahan itu. Mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.
<p>Generalisasi / Generalization</p> 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<p>Menggunakan / include</p> 	Relasi use case tambahan ke sebuah use case di mana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.

Contoh *Use Case* diagram sederhana:




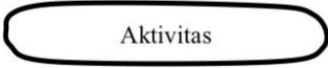
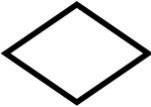


**Gambar 2.4** Contoh *Use Case Diagram*

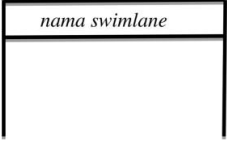
## 2. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor.

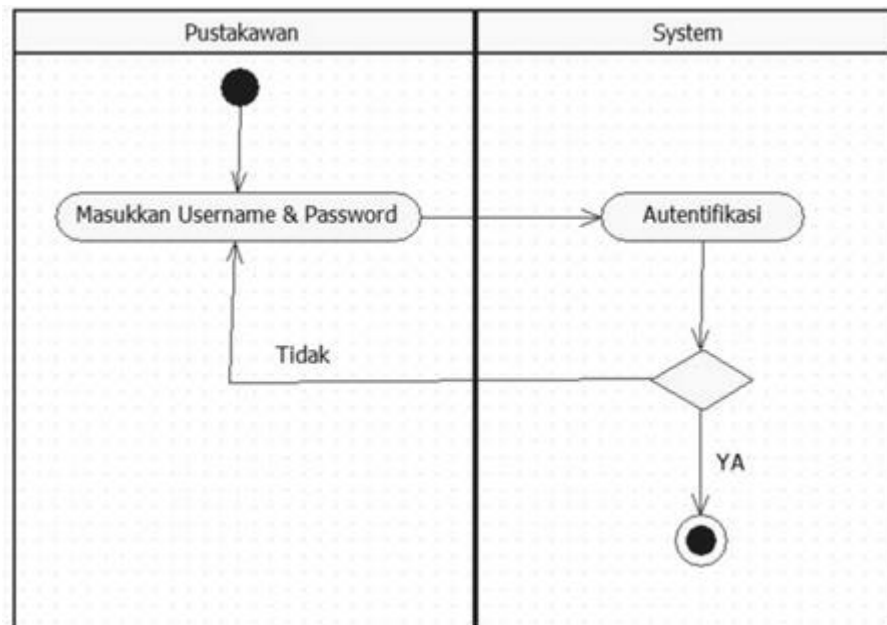
Simbol-simbol yang digunakan dalam *Activity Diagram* yaitu:

**Tabel 2.4** Simbol-simbol Activity Diagram (Rosa dan Shalahuddin, 2016:162)

<p>Status awal</p> 	<p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.</p>
<p>Aktivitas</p> 	<p>Aktivitas yang dilakukan <i>system</i>, aktivitas biasanya diawali dengan kata kerja.</p>
<p>Percabangan / decision</p> 	<p>Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.</p>
<p>Penggabungan / join</p> 	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.</p>
<p>Status akhir</p> 	<p>Status akhir yang dilakukan <i>system</i>, sebuah diagram aktivitas memiliki sebuah status akhir.</p>

<p>Swimlane</p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>
---	---

Contoh *Activity Diagram* sederhana:






**Gambar 2.5** Contoh *Activity Diagram*

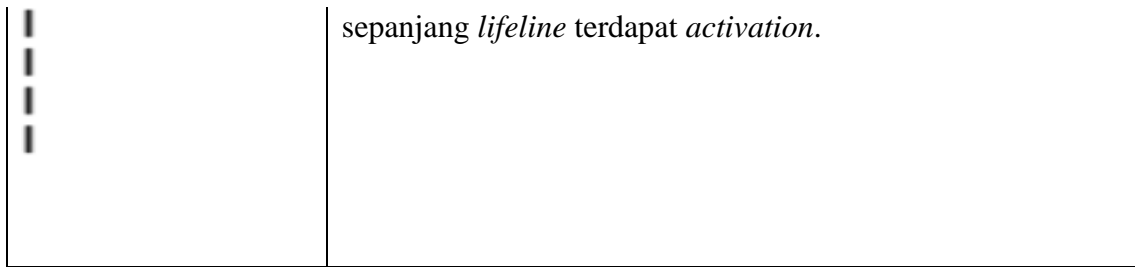
### 3. *Sequence Diagram*

Diagram *Sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

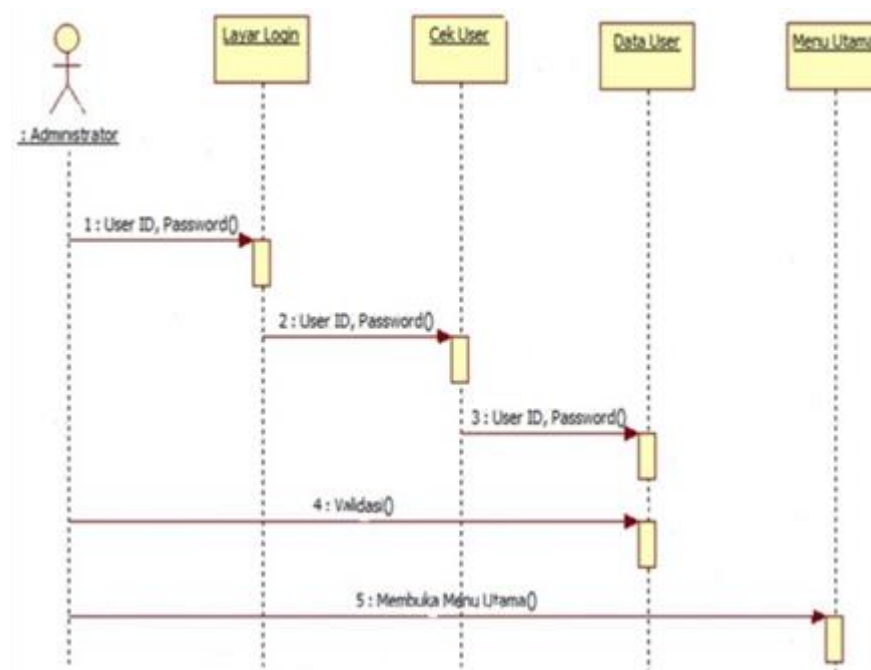
Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

**Tabel 2.5** Simbol-simbol *Sequence Diagram* (Rosa dan Shalahuddin, 2018:165)

<< <i>Entity Class</i> >>	<i>Entity class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
<< <i>Boundary Class</i> >>	<i>Boundary class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak.
<< <i>Control Class</i> >>	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
<i>Message</i> 	<i>Message</i> , symbol mengirim pesan antar <i>class</i> .
<i>Recursive</i> 	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
<i>Activation</i> 	<i>Activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
<i>Lifeline</i>	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek,



Contoh *sequence diagram* sederhana:



**Gambar 2.6** Contoh *Sequence Diagram*

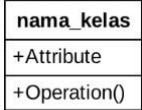




#### 4. *Class Diagram*



Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.



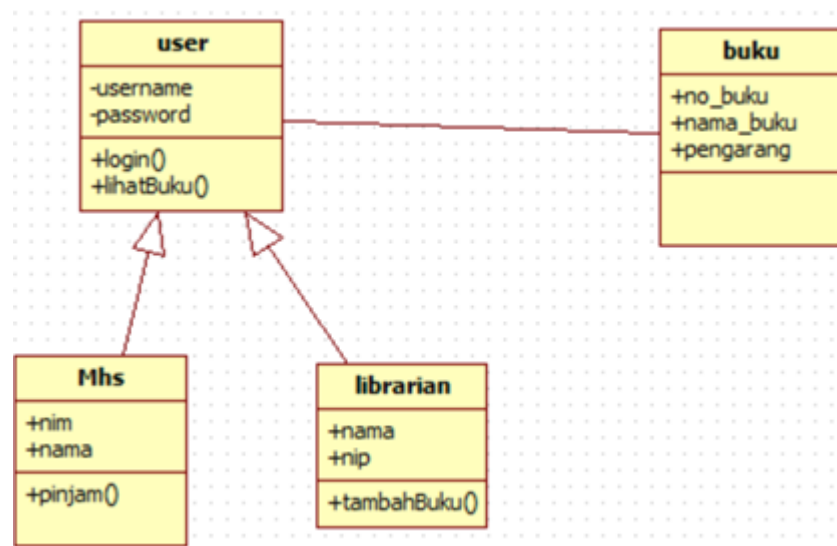
Simbol-simbol yang digunakan dalam *Class Diagram* yaitu:

**Tabel 2.6** Simbol-simbol *Class Diagram* (Rosa dan Shalahuddin, 2016:146)

<p>Kelas</p> 	<p>Kelas pada struktur sistem</p>
<p>Antarmuka / <i>interface</i></p> 	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek</p>
<p>Asosiasi / <i>association</i></p> 	<p>Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Asosiasi berarah / <i>directed association</i></p> 	<p>Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Generalisasi</p> 	<p>Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)</p>
<p>Kebergantungan / <i>dependency</i></p>	<p>Relasi antarkelas dengan makna kebergantungan antarkelas</p>

	
	
Agregasi / <i>aggregation</i>	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )

Contoh *class diagram* sederhana:






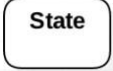
**Gambar 2.7** Contoh *Class Diagram*

## 5. *Statechart Diagram*

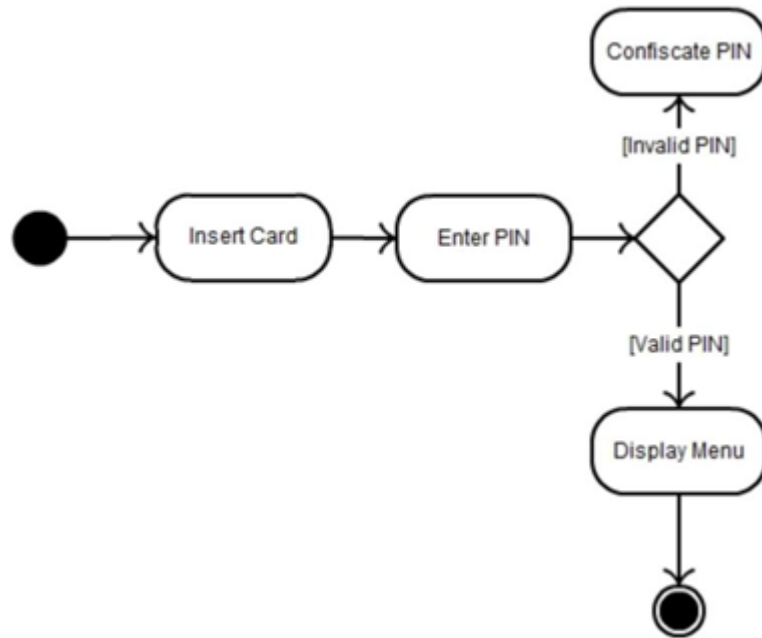
*Statechart diagram* digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek. Jika *diagram* sekuen digunakan untuk interaksi antar objek. Perubahan tersebut digambarkan dalam suatu graf berarah.

Simbol-simbol yang digunakan dalam *Statechart Diagram* yaitu:

**Tabel 2.7** Simbol-simbol *Statechart Diagram* (Rosa dan Shalahuddin, 2016:163)

<p><i>Start / Status Awal (Initial State)</i></p> 	<p><i>Start</i> atau <i>initial state</i> adalah <i>state</i> atau keadaan awal pada saat sistem mulai hidup.</p>
<p><i>End / Status Akhir (Final State)</i></p> 	<p><i>End</i> atau <i>final state</i> adalah <i>state</i> keadaan akhir dari daur hidup suatu sistem.</p>
<p><i>Event</i></p> 	<p><i>Event</i> adalah kegiatan yang menyebabkan berubahnya status mesin.</p>
<p><i>State</i></p> 	<p>Sistem pada waktu tertentu. <i>State</i> dapat berubah jika ada event tertentu yang memicu perubahan tersebut.</p>

Contoh *statechart diagram* sederhana:



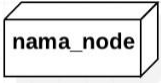


**Gambar 2.8** Contoh *Statechart Diagram*

## 6. *Statechart Diagram*

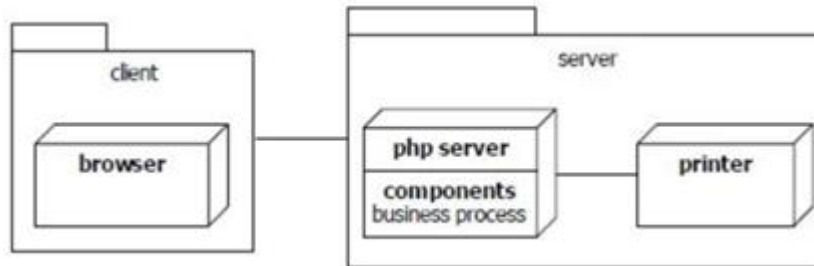
Diagram deployment atau deployment diagram menunjukkan konfigurasi komponen dalam prose eksekusi aplikasi.

Simbol-simbol yang digunakan dalam Deployment Diagram yaitu:

**Tabel 2.8** Simbol-simbol *Deployment Diagram* (Rosa dan Shalahuddin, 2016:154)

<p><i>Package</i></p> 	<p><i>Package</i> merupakan sebuah bungkus dari satu atau lebih node.</p>
<p><i>Node</i></p> 	<p>Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak tidak dibuat sendiri (<i>software</i>).</p>
<p>Kebergantungan / <i>dependency</i></p> 	<p>Kebergantungan antar <i>node</i>, arah panah mengarah pada <i>node</i> yang dipakai.</p>
<p><i>Link</i></p> 	<p>Relasi antar <i>node</i></p>

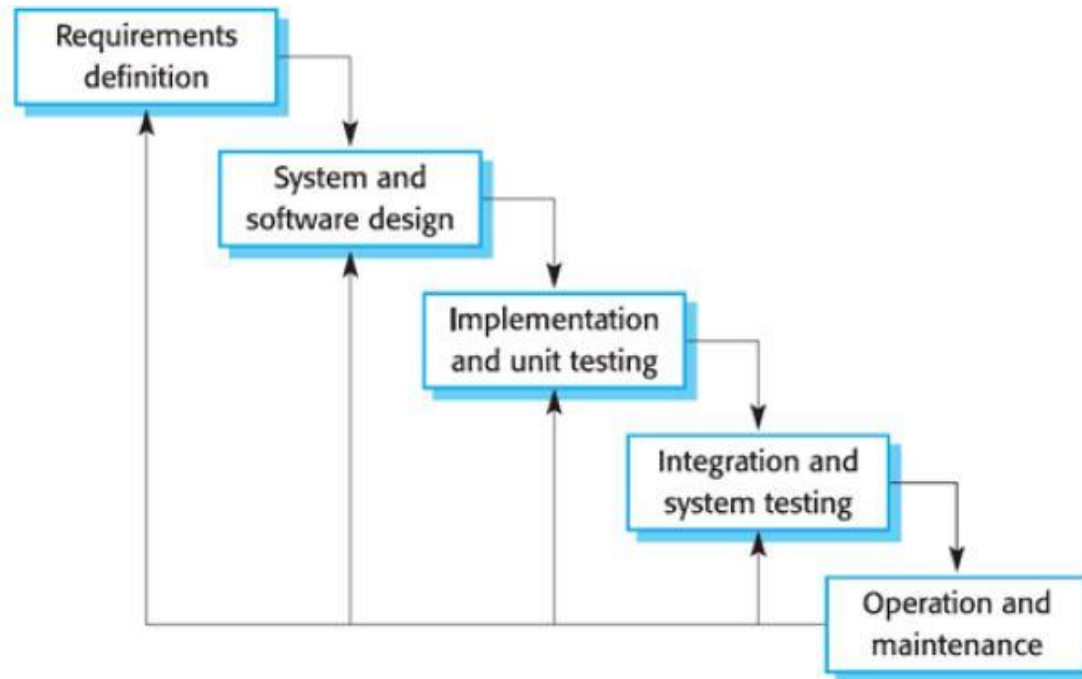
Contoh *Deployment Diagram* sederhana:



**Gambar 2.9** Contoh *Deployment Diagram*

## 2.19 Metode Waterfall

Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian. Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang runtut: *requirement* (analisis kebutuhan), desain sistem (*system design*), *Coding & Testing*, Penerapan Program, pemeliharaan (Trisianto, 2018).



**Gambar 2.10** Metode *Waterfall* (Trisianto, 2018)

Chrisantus Trisianto menyatakan bahwa model *waterfall* memiliki tahapan - tahapan sebagai berikut:

1. *Requirement definition* (Analisis Kebutuhan)

Dalam langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau *study literatur*. Seseorang *system* analisis akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas – tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang

berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan sistem analisis untuk menterjemahkan kedalam bahasa pemrograman (Trisianto, 2018).

## 2. *System and software design* (Desain Sistem)

Proses *design* akan menterjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya (Trisianto, 2018).

## 3. *Implementation and unit testing* (Penulisan Sinkode Program / Implementation)

*Coding* merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan meterjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan - kesalahan terhadap system tersebut dan kemudian bisa diperbaiki (Trisianto, 2018).

## 4. *Intergration and system testing* (Penerapan / Pengujian Program)



Tahapan ini bisa dikatakan *final* dalam pembuatan sebuah sistem. Setelah melakukan analisa, *design* dan pengkodean maka sistem yang sudah jadi digunakan oleh *user* (Trisianto, 2018).

#### 5. *Operation and Maintenance* (Operasi dan Pemeliharaan)

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau system operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional (Trisianto, 2018).

### **2.20 PHP**

Menurut Yudho Yudhanto dan Helmi Adi Prasetyo (2018:7) PHP atau *Hypertext Preprocessor* adalah bahasa pemrograman *script server side* yang sengaja dirancang lebih cenderung untuk membuat dan mengembangkan web. Bahasa pemrograman ini memang dirancang untuk para pengembang web agar dapat menciptakan suatu halaman web yang bersifat dinamis.

Sekilas tentang PHP, PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995 dan terus dikembangkan hingga saat ini. Ada banyak sekali web termasuk CMS yang dibuat menggunakan bahasa PHP, seperti WordPress dan lain-lain.

### **2.21 *Laravel***

*Laravel* adalah sebuah MVC web development framework yang didesain untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan serta meningkatkan produktifitas pekerjaan dengan sintak yang bersih dan fungsional yang dapat mengurangi banyak waktu untuk implementasi (Farizan Luthfi, 2017:1)

### **2.22 *Blackbox Testing***

*Black box* atau uji fungsional, kondisi uji dikembangkan atas program atau fungsi sistem yaitu tester membutuhkan informasi tentang data *input* dan *output* yang diamati, tetapi tidak tahu bagaimana program atau sistem bekerja. Tester berfokus pada pengujian fungsi program terhadap spesifikasi tersebut. Dengan pengujian *black-box*, tester memandang program sebagai kotak hitam dan benarbenar tidak peduli dengan struktur internal dari program atau sistem. (Farizan Luthfi, 2017:3).

Proses rekayasa perangkat lunak engineering harus dilakukan pada saat pengujian perangkat lunak. Referensi yang diusulkan dalam beberapa tahapan pengujian software dengan menyiapkan template untuk pengujian bagi pembuat software.

Tahapan dalam pengujian software antara lain (Sutanto, Utomo, & Perbawa, 2018):

1. *Unit testing*, merupakan suatu aktifitas mencoba alur proses aplikasi untuk memastikan fungsionalitas secara lengkap dan memaksimalkan pencarian kesalahan.
2. *Integration testing*, merupakan sekumpulan dari permasalahan yang ditemukan pada proses verifikasi.
3. *High-order test*, aplikasi yang sudah selesai akan dirancang menjadi bagian yang lengkap dan menjadi satu kesatuan.
4. *Validation test*, memberikan jaminan akhir bahwa aplikasi yang dirancang telah memenuhi semua kriteria yang diperlukan.

## **BAB III**

### **PEMBAHASAN**

Dalam penelitian ini digunakan metode pengembangan SDLC (*System Development Life Cycle*) dengan model *Waterfall*. Ada lima tahapan untuk pengembangan sistem ini, namun untuk penelitian ini dibatasi sampai dengan tahap keempat (Trisianto, 2018).

Berikut adalah tahapan metode *Waterfall* yang digunakan untuk penelitian ini:

1. *Requirement* (Analisis Kebutuhan)
2. *Design System* (Desain Sistem)
3. *Coding & Testing* (Penulisan Sinkode Program / Implementasi)
4. *Integration & Testing* (Penerapan / Pengujian Program)
5. *Operation & Maintenance* (Operasi dan Pemeliharaan)

Penulis tidak akan melakukan penelitian pada tahap *Operation and Maintenance*, penulis hanya melakukan hingga tahap empat yaitu Penerapan / Pengujian Program.

### **3.1 Requirement (Analisis Kebutuhan)**

Untuk tahapan ini merupakan analisa terhadap kebutuhan sistem. proses pengumpulan informasi berupa data-data yang berkaitan dengan penelitian.

#### **1. Metode Pengumpulan Data**

Metode pengumpulan data untuk penelitian ini penulis mengambil foto wajah orang sebagai sampel untuk penelitian.

#### **2. Metode Wawancara**

Metode wawancara yang dilakukan untuk penelitian ini adalah dengan mewawancarai beberapa dosen dan mahasiswa di lingkungan kampus. Dalam wawancara tersebut berbicara mengenai bagaimana sistem bekerja, fitur – fitur yang diperlukan untuk membuat sistem, serta solusi agar sistem tidak dicurangi oleh pengguna saat sistem sudah jadi / siap dipakai.

#### **3. Metode Studi Literatur**

Studi literatur merupakan teknik yang digunakan untuk memperoleh informasi atau data dengan cara mempelajari buku-buku, jurnal, dan juga internet. Dalam skripsi ini sumber informasi juga didapat dari buku-buku serta jurnal yang berkaitan dengan karakteristik wajah, *object detection*, *face recognition*, metode *128D Embedding*, python dan informasi lainnya yang berkaitan dengan skripsi, berikut adalah tabel literatur yang menjadi referensi:

**Tabel 3.1** Referensi Penelitian

No	Literature	Pembahasan
1	E.Aishwarya, K.Kumaravel, R.MohamedSuthesh, S. Poornima & R.Poonguzhali. “ <i>Smart Attendance System Using Deep Learning Technique</i> ”.	Penelitian ini membahas tentang sistem absensi berbasis <i>face recognition</i> dengan menggunakan metode <i>Haar-like</i> dan <i>Deep Learning</i> .
2	Jonathan S. Talahua, Jorge Buele, P. Calvopiña, & José Varela-Aldás. “ <i>Facial Recognition System for People with and without Face Mask in Times of the COVID-19 Pandemic</i> ”.	Penelitian ini membahas tentang cara sistem mengenali wajah baik memakai masker maupun tidak. Dalam penelitian ini menggunakan metode <i>MobileNetV2</i> untuk meneliti 13,359 foto wajah dengan tingkat akurasi memakai masker sebesar 99.52%, sedangkan tingkat akurasi tidak memakai masker sebesar 99.96%.
3	Jingtuo Liu, Yafeng Deng, Tao Bai, Zhengping Wei, & Chang Huang. “ <i>Targeting Ultimate Accuracy: Face Recognition via Deep Embedding</i> ”.	Penelitian ini berisi tentang pengenalan wajah menggunakan metode Deep CNN ( <i>Convolutional Neural Network</i> ) sampel foto wajah sebanyak 1,2 juta

		dengan tingkat akurasi mengenal wajah 99.77%.
4	Musfirah Putri Lukman, Junaedy, & Yosua Friendly Yorendy Rieuwpassa. “ <i>Sistem Lampu Otomatis Dengan Sensor Gerak, Sensor Suhu Dan Sensor Suara Berbasis Mikrokontroler</i> ”.	Penelitian ini melakukan riset mengenai sistem lampu otomatis. Perangkat keras yang digunakan untuk penelitian yaitu mikrokontroler Arduino Uno, sensor gerak PIR, sensor suhu MLX90614, sensor suara KY - 038, relay, dan lampu LED 3W. <i>Software</i> untuk membuat program yaitu Arduino IDE bahasa pemrograman yang digunakan yaitu bahasa pemrograman C. Hasil pengujian menunjukkan bahwa sensor PIR dapat mendeteksi gerakan orang yang memasuki atau meninggalkan ruangan sedangkan sensor KY - 038 dan sensor MLX90614 masih memiliki kelemahan dalam pendeteksian suara dan suhu.
5	Semiha Makinist, Betül Ay, & Galip Aydın. “ <i>Average Neural Face</i>	Penelitian ini berisi tentang cara sistem mengetahui <i>gender</i> /jenis kelamin

	<i>Embeddings for Gender Recognition</i> ".	berdasarkan pengenalan wajah dengan menggunakan metode ANFE ( <i>Average Neural Face Embeddings</i> ).
--	---	--

### 3.1.2 Analisis Permasalahan

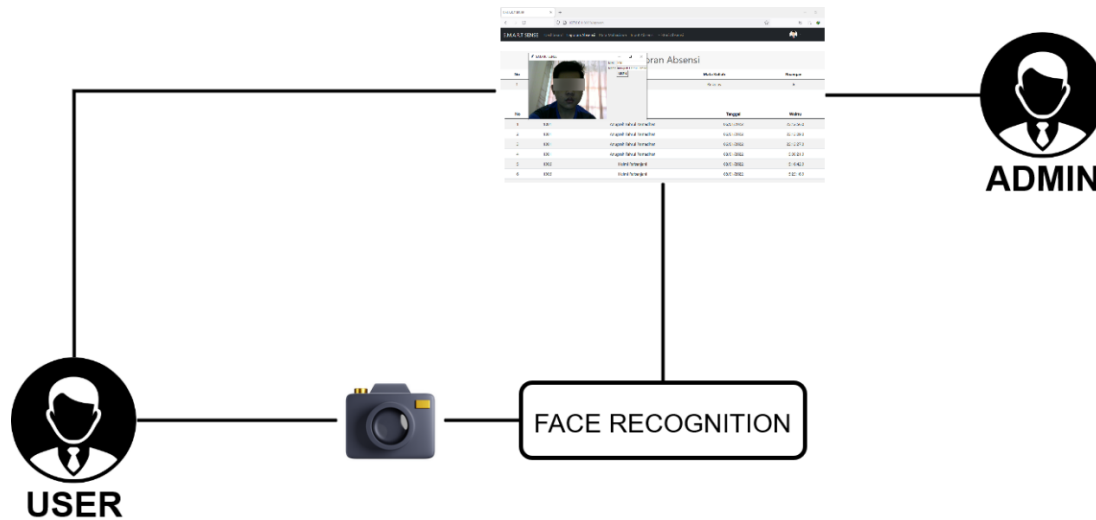
Teknologi *Face Recognition* merupakan salah satu sistem identifikasi biometrik yang sangat efektif dalam menghasilkan informasi. Sistem *Face Recognition* mengidentifikasi seseorang dengan fitur-fitur khusus pada wajah yang membedakan satu orang dengan orang lainnya. Dengan ditambahkan sensor suhu maka akan mengurangi kecurangan saat mengisi kehadiran. Sistem akan mendeteksi suhu, apakah objek memiliki suhu atau tidak, apabila objek memiliki suhu maka absen kehadiran akan terisi oleh sistem, sebaliknya apabila objek tidak memiliki suhu atau suhu sama dengan suhu lingkungan maka sistem tidak akan mengisi absensi kehadiran. *Hardware* yang dibutuhkan untuk menjalankan sistem ini yaitu komputer, kamera, dan Arduino uno untuk sensor suhu. Masalah yang sering dihadapi pada *face recognition* adalah sulitnya membedakan wajah yang mirip/kembar, apabila ada seseorang memiliki wajah yang mirip/kembar dengan yang sudah terdaftar tetapi orang tersebut belum terdaftar maka orang tersebut masih tetap bisa terdeteksi dan bisa mengisi kehadiran oleh sistem. Dan juga *face recognition* memerlukan banyak sample untuk mengurangi tingkat error pendeteksian wajah.



### 3.1.3 Gambaran Umum Sistem Yang Diusulkan

Dalam skripsi ini aplikasi yang diusulkan merupakan sebuah aplikasi yang akan mengisi absensi kehadiran dengan menggunakan *face recognition*. Sistem *face recognition* akan mengenali wajah saat wajah berada dihadapan *webcam* saat sistem sudah mengenali wajah tersebut maka selanjutnya sistem akan memeriksa suhu objek, apabila suhu objek tersebut lebih dari suhu lingkungan sekitar maka sistem akan mengenali objek tersebut adalah manusia. Lalu sistem akan mengeluarkan notifikasi suara untuk memberitahu bahwa *user* telah dikenali oleh sistem. Setelah wajah tersebut sudah dikenali oleh sistem maka secara otomatis sistem akan mengisi kehadiran di *website* laporan kehadiran.

Aplikasi ini akan dibangun dengan menggunakan *python* dimana *user* harus menjalankannya melalui konsol CMD (*Command Prompt*), setelah itu ketika aplikasi memunculkan layar kamera *user* harus melakukan scan wajah agar kamera dapat menangkap wajah yang sedang di *scan* dan menghasilkan *output* nim dan nama pada aplikasi tersebut. Gambar 3.1 menjelaskan gambaran arsitektur teknologi dari aplikasi *face recognition* yang diusulkan.



**Gambar 3.1** Arsitektur Teknologi Dari Aplikasi *Face Recognition*

Dalam proses *face recognition*, sebelumnya aplikasi *machine learning* ini membutuhkan *dataset* foto wajah orang untuk dijadikan data *sample*, sedangkan untuk mendapatkan hasil *face recognition* maka *user* harus melakukan *scan object* dengan mengarahkan wajah ke layar kamera. Berikut adalah langkah-langkah *face recognition*:

1. Mengumpulkan *Dataset*

Pada penelitian ini proses absensi kehadiran berbasis *face recognition* memerlukan *dataset* berupa struktur foto wajah. Proses dari mengumpulkan *dataset* yaitu sebagai berikut:

a. *Data Image*

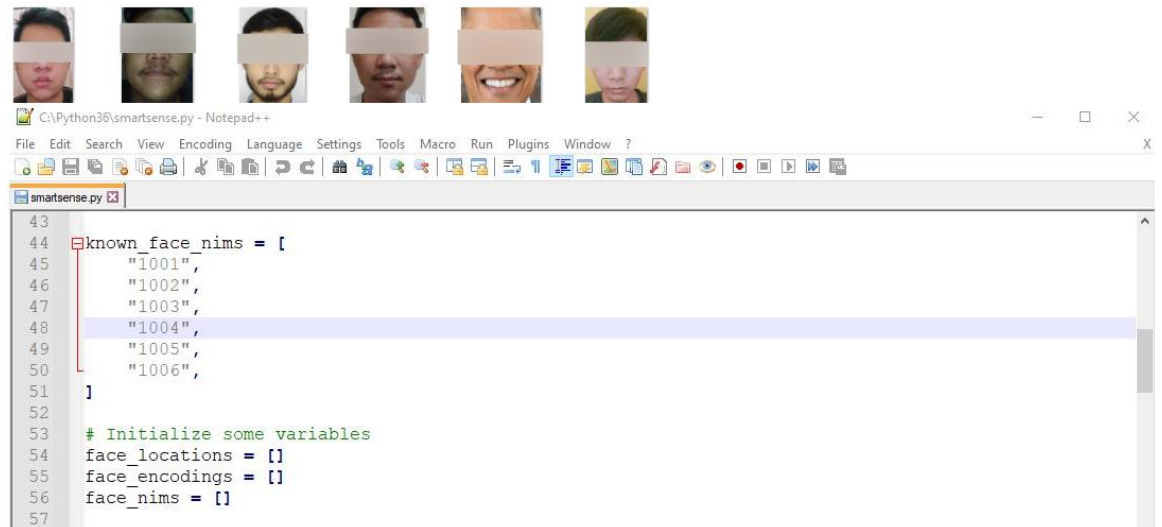
Tahapan pertama yaitu dengan mengumpulkan *dataset image* yang di perlukan dalam penelitian, pengambilan *data image* dengan memfotokan wajah orang dan mengambil satu foto wajah dari internet. Gambar 3.2 menjelaskan proses pengambilan image.



**Gambar 3.2** Proses Pengambilan *Image*

b. *Labelling Image*

Dalam tahap ini penulis mengambil foto dengan wajah yang berbeda - beda, terdapat beberapa jumlah sample foto wajah. Foto tersebut akan diberikan label dengan cara memberikan NIM pada setiap foto tersebut, dikarenakan apabila label diberikan nama dikhawatirkan akan ada nama yang sama tetapi memiliki wajah yang berbeda apabila dengan NIM sudah dipastikan tidak akan ada NIM yang sama. Dan hasil *labelling* foto ini akan disimpan dalam pemrograman *python* Gambar 3.3 menjelaskan gambaran *labelling* image.



**Gambar 3.3** Membuat *Labelling Image*

c. Membuat Nama

Dalam tahap ini semua sample foto wajah yang telah diberi label NIM akan diberi nama untuk output pada aplikasi dan notifikasi suara apabila mahasiswa tersebut telah melakukan absensi kehadiran. Tidak perlu khawatir apabila ada nama yang sama, karena nama – nama tersebut mengikuti/berkaitan dengan NIM, meskipun terdapat nama yang sama tetapi NIM tetap berbeda. NIM dan nama foto tersebut akan disimpan dalam *database mysql*. Penginputan NIM dan nama dilakukan dengan program *python* dan dijalankan oleh CMD (*Command Prompt*). Gambar 3.4 menjelaskan tentang pembuatan nama pada foto yang telah diberi label.

Gambar 3.5 menjelaskan hasil pemberian NIM dan nama lalu disimpan dalam *database mysql*.

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Python36>python inputdatamahasiswa.py
NIM: 1001
Nama: Anugrah Fahrul Ramadhan

C:\Python36>

```

**Gambar 3.4** Penginputan NIM dan Nama Mahasiswa

Server: 127.0.0.1 » Database: dbdemo » Tabel: mahasiswa

Menampilkan baris 0 - 6 (total 7, Pencarian dilakukan dalam 0,0021 detik.)

SELECT \* FROM `mahasiswa`

Profil [ Edit dikotak ] [ Ubah ] [ Jelaskan SQL ] [ Buat kode PHP ] [ Segarkan ]

Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini | Sort by key: Tidak ada

	NIM	Nama	Id
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1001	Anugrah Fahrul Ramadhan	1
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1002	Obama	2
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1003	Naufal Dzaki	3
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1004	Yoga Pramana	4
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1005	Moch. Dimmas Saputra	5
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1006	Helmi Rafsanjani	6

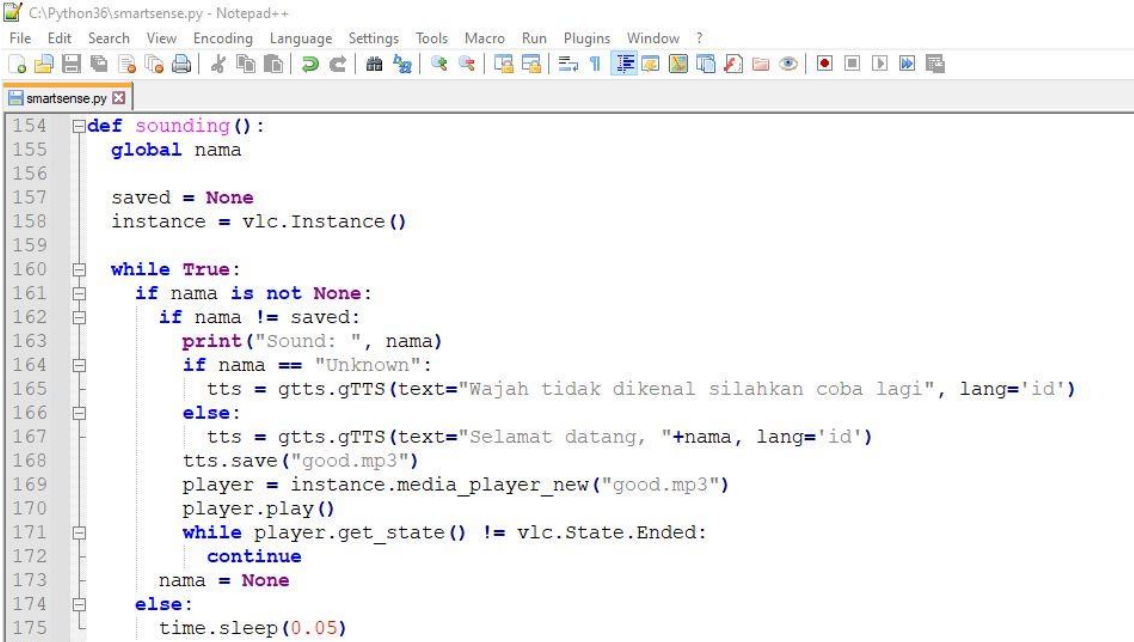
Pilih Semua | Dengan pilihan:  Ubah  Salin  Hapus  Ekspor

Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini | Sort by key: Tidak ada

**Gambar 3.5** Hasil Penginputan Disimpan Di *Database Mysql*

## 2. Membuat Notifikasi Suara

Dalam tahap ini penulis membuat notifikasi suara apabila sudah melakukan absensi kehadiran maka sistem akan mengeluarkan suara “Selamat datang (nama mahasiswa)”, apabila wajah tidak terdaftar dalam database atau tidak ada wajah didepan kamera maka sistem akan mengeluarkan suara “Wajah tidak dikenal silahkan coba lagi”. Gambar 3.6 menjelaskan proses pembuatan suara.



```

C:\Python36\smartsense.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
smartsense.py
154 def sounding():
155     global nama
156
157     saved = None
158     instance = vlc.Instance()
159
160     while True:
161         if nama is not None:
162             if nama != saved:
163                 print("Sound: ", nama)
164                 if nama == "Unknown":
165                     tts = gtts.gTTS(text="Wajah tidak dikenal silahkan coba lagi", lang='id')
166                 else:
167                     tts = gtts.gTTS(text="Selamat datang, "+nama, lang='id')
168                 tts.save("good.mp3")
169                 player = instance.media_player_new("good.mp3")
170                 player.play()
171                 while player.get_state() != vlc.State.Ended:
172                     continue
173                 nama = None
174             else:
175                 time.sleep(0.05)

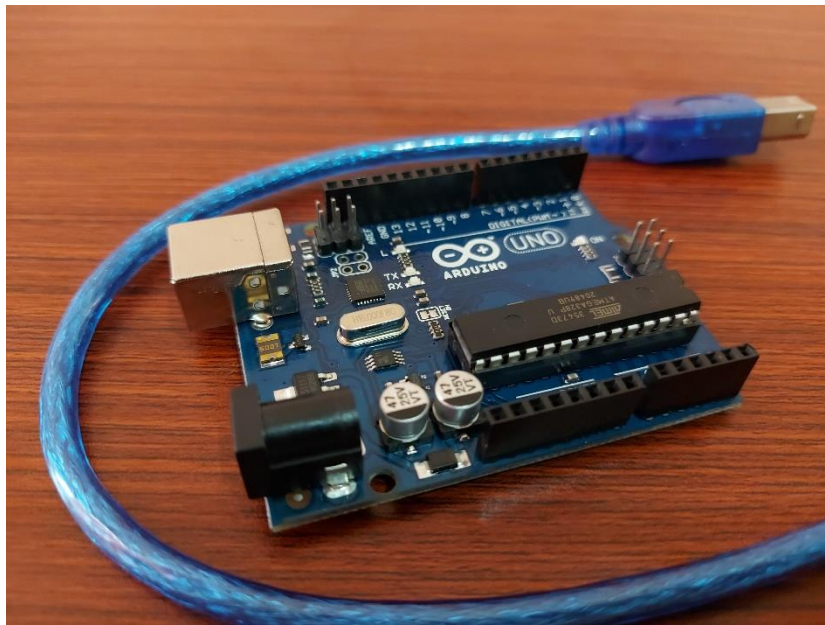
```

**Gambar 3.6** Proses Pembuatan Notifikasi Suara.

### 3. Membuat Sensor Suhu

Selanjutnya dalam tahap ini penulis membuat sensor suhu, sensor ini digunakan untuk meminimalisir kecurangan absensi kehadiran, jadi pada saat melakukan absensi kehadiran sistem memeriksa terlebih dahulu apakah objek tersebut memiliki suhu atau tidak. Untuk membuat sensor suhu diperlukan beberapa hardware, yaitu : Arduino Uno R3 ATMEGA328P ATMEGA16U2, Kabel *Jumper Male To Female*, dan Sensor Suhu MLX90614. Berikut penjelasannya :

#### a. Arduino Uno R3 ATMEGA328P ATMEGA16U2

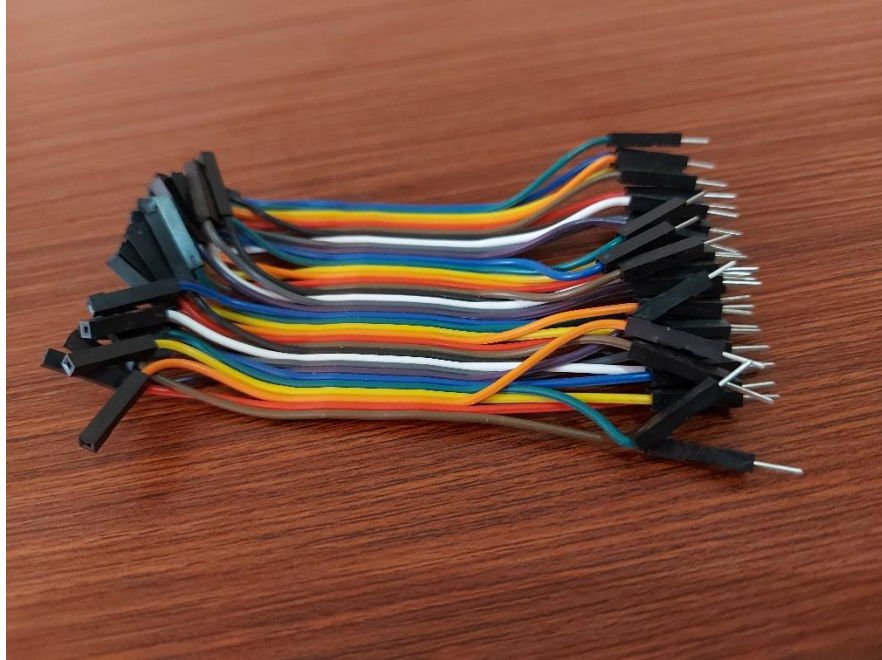


**Gambar 3.7** Hardware Arduino Uno R3

Pada Gambar 3.7 terdapat Arduino Uno dan kabel USB sebagai penghubung antara Arduino Uno dan komputer. Untuk membuat sensor suhu

diperlukan perangkat Arduino Uno sebagai saluran komunikasi melalui USB dan sebagai perangkat lunak pada computer.

b. Kabel *Jumper Male to Female*

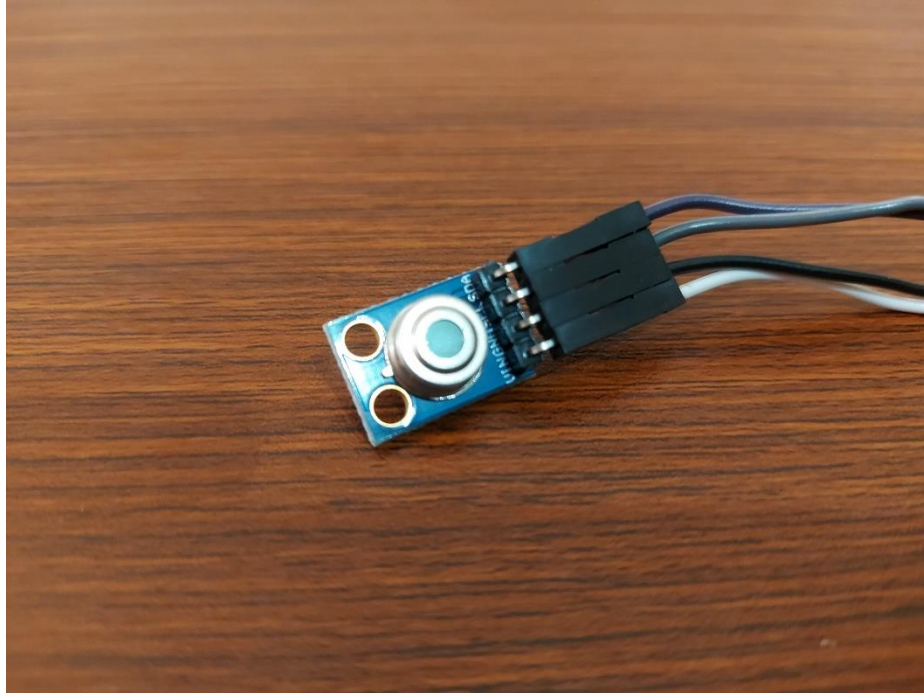


**Gambar 3.8** Kabel *Jumper Male to Female*

Pada Gambar 3.8 terdapat kabel *jumper male to female*, kabel ini diperlukan sebagai konduktor aliran listrik dan juga bisa sebagai penghubung antara alat *prototype* dengan Arduino Uno.



c. Sensor Suhu MLX90614



**Gambar 3.9** Sensor Suhu MLX90614

Pada Gambar 3.9 terdapat sensor suhu dengan jenis MLX90614, sensor suhu dengan jenis adalah sensor suhu non-kontak, sensor suhu non-kontak sangat menguntungkan karena dapat mengukur suhu tinggi tanpa merusak sistem. Sensor ini dapat merasakan suhu suatu benda tanpa menyentuh benda tersebut. Adapun *pin out* dari *board* GY-906 adalah sebagai berikut :

- *Vin* : Tegangan *supply* dari modul
- *GND* : Sinyal *Ground*
- *SCL* : *Serial Clock*
- *SDA* : *Serial Data*

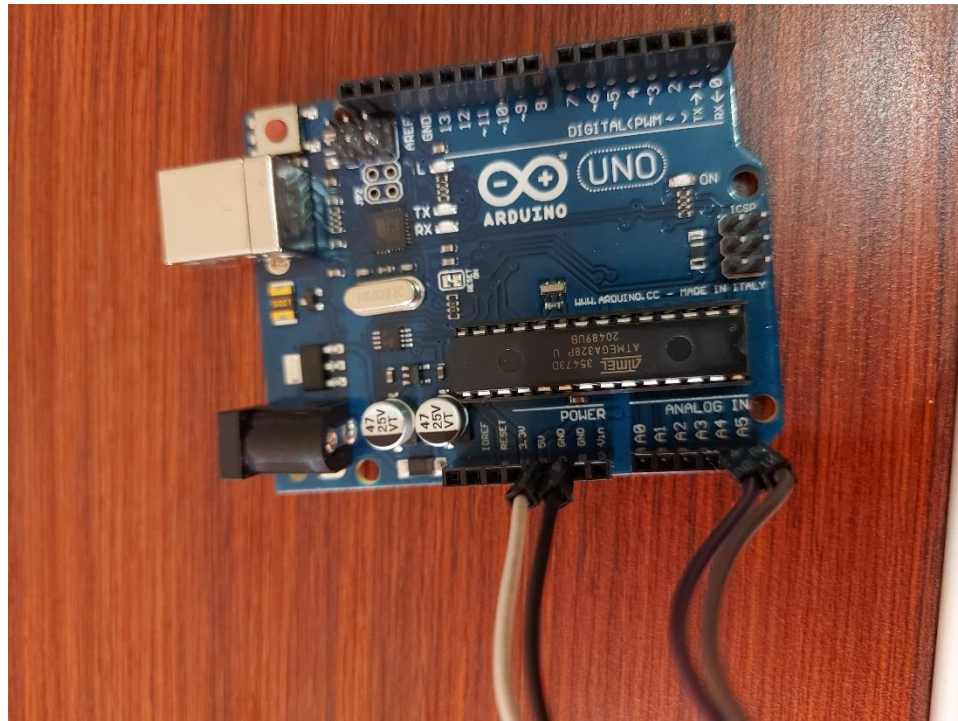
Setelah semua perangkat tersedia maka langkah selanjutnya adalah merangkai alat – alat tersebut. Pertama – tama lihat terlebih dahulu pada sensor suhu dengan jenis MLX90614 terdapat *pin out* yaitu Vin, GND, SCL, dan SDA. Dari keempat *pin out* dibagi 2 kelompok, kelompok pertama yaitu Vin dan GND sebagai *power supply* tegangan listrik dan kelompok kedua yaitu SCL dan SDA sebagai penyalur data dari Arduino Uno menuju ke sensor suhu. Berikut penjelasan pada Gambar 3.10.



**Gambar 3.10** Arduino Uno Sebelum Pemasangan Sensor Suhu

Saat pemasangan sesuaikan dengan kelompok yang sudah dibagi, untuk menyambungkan perangkat antara sensor suhu dan Arduino Uno menggunakan kabel *jumper male to female*. Untuk kelompok *power supply* yaitu Vin

disambungkan ke 5 volt dan untuk GND disambungkan ke GND. Kelompok penyalur data yaitu SCL disambungkan ke A5 dan SDA disambungkan ke A4. Berikut Gambar 3.11 hasil pemasangan kedua perangkat Arduino Uno dan sensor suhu



**Gambar 3.11** Arduino Uno Sesudah Pemasangan Sensor Suhu

Setelah kedua perangkat disambungkan maka selanjutnya adalah menambahkan *code* program agar perangkat dapat berfungsi. Berikut pada Gambar 3.12 penjelasan *code* program.

The image shows a screenshot of a code editor with two windows. The left window, titled 'smartsense.py', contains Python code for a GUI application. The right window, titled 'mkxtest | Arduino 1.8.18', contains C++ code for an Arduino sketch.

```

91 self.window.after(self.interval, self.
92
93 def clickMe():
94 global captured, conn, cursor, nama, por
95
96 port.write(b'a')
97 sensor = port.readline().decode()
98 sensor = sensor.strip().split()
99 ambient = float(sensor[0])
100 objek = float(sensor[1])
101 print(ambient, objek)
102 if objek < ambient + 5.0: #logika false
103     print("Unknown")
104     return
105
106 small_frame = cv2.resize(captured, (0, 0
107 rgb_small_frame = small_frame[:, :, :-1
108
109 # Find all the faces and face encodings
110 face_locations = face_recognition.face_l
111 face_encodings = face_recognition.face_e
112

```

```

mkxtest
.. .. .. .. ..
while (1):
}
}

void loop() {
if (Serial.available() > 0) {
char m = Serial.read();
if (m == 'a') {
float ambient = mlx.readAmbientTempC();
float temp = mlx.readObjectTempC();
Serial.print(ambient);
Serial.print(" ");
Serial.println(temp);
digitalWrite(13,HIGH);
delay(500);
digitalWrite(13,LOW);
}
}
}

```

**Gambar 3.12** Code Program Suhu

Penjelasan *code* program pada Gambar 3.12 yaitu terdapat dua variabel yaitu objek dan *ambient*. Objek adalah suhu objek yang terdeteksi sedangkan *ambient* adalah suhu lingkungan. Apabila suhu objek lebih panas dari suhu lingkungan maka sistem akan mendeteksi bahwa objek yang di depan sensor suhu adalah manusia. Untuk lebih jelasnya penulis akan menguji sensor suhu tersebut. Berikut Gambar 3.13 hasil uji coba sensor suhu.

```

C:\Windows\System32\cmd.exe - python smartnose.py
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Python36>python smartnose.py
Load apps and learning ...
30.41 33.55
Suhu: Anurrah Fahrul Ramadhan
  
```

**Gambar 3.13** Hasil Uji Coba Sensor Suhu

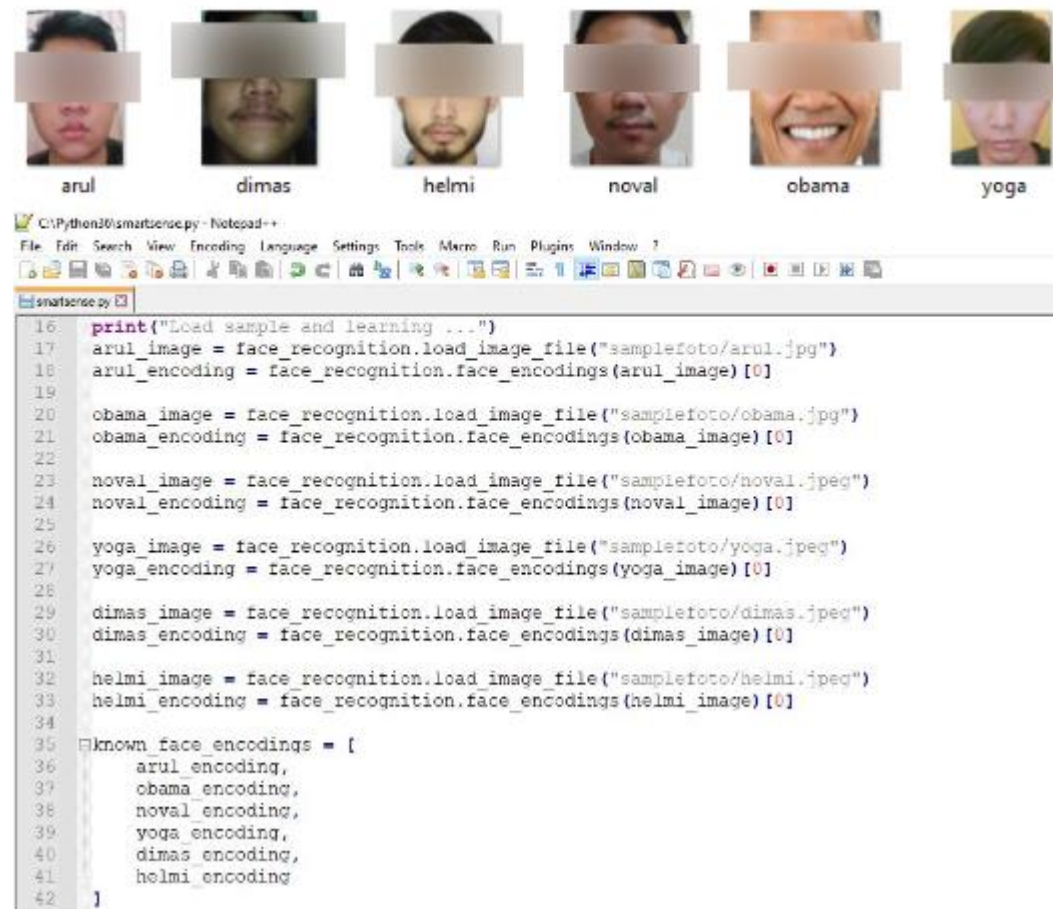
Dapat terlihat bahwa sensor suhu berfungsi dengan baik, untuk menjalankan sensor suhu harus melalui CMD (*Command Prompt*). Hasil pengukuran suhu dengan menggunakan sensor suhu MLX90614 bahwa suhu lingkungan terdeteksi 30.41 derajat *celcius* dan suhu objek terdeteksi 33.55 derajat *celcius*.

#### 4. *Learning Sample*

Dalam tahap ini seluruh sample foto wajah yang akan dijadikan absensi kehadiran akan dipelajari (*learning*) oleh sistem. Tidak perlu meng*training* sample foto wajah tersebut dikarenakan dengan metode *128D Embedding* didalam library *Face Net* sudah meng*training* 3 juta foto wajah manusia yang dilakukan oleh Davis E. King. Seluruh sample foto wajah akan dipelajari oleh sistem setelah itu aplikasi dapat mendeteksi wajah manusia dengan akurat dan

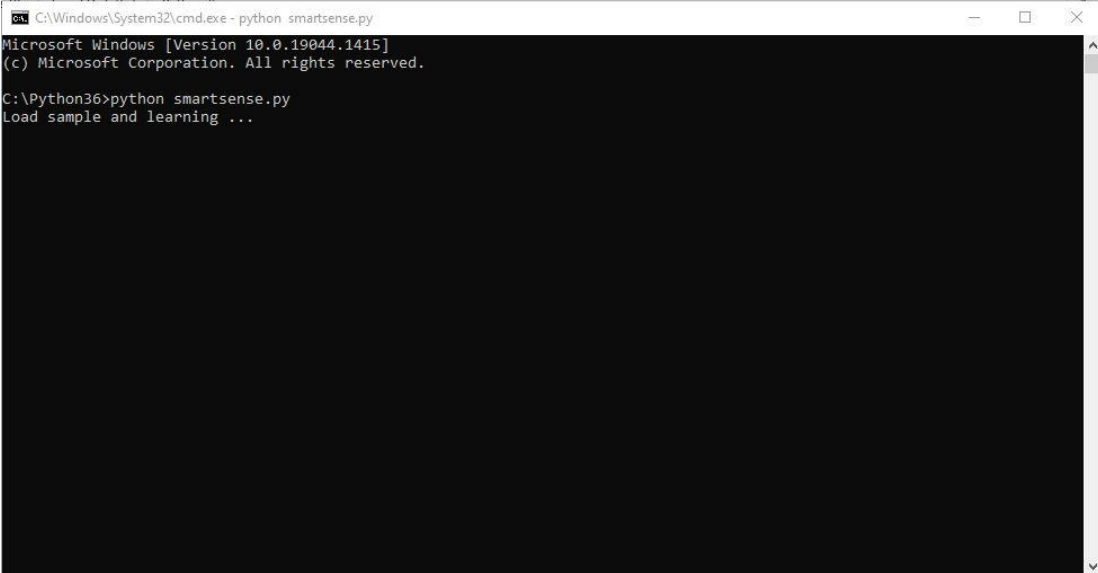


sesuai dengan bentuk struktur wajah. Gambar 3.14 menjelaskan *code* program *128D Embedding*.



**Gambar 3.14** Code Program *128D Embedding*

Dalam codingan Gambar 3.14 sistem akan mempelajari struktur wajah dari sample foto dengan *library FaceNet* sistem akan mempelajari sample foto tanpa harus meng*training*. Berikut Gambar 3.15 sistem akan dijalankan oleh CMD (*Command Prompt*).

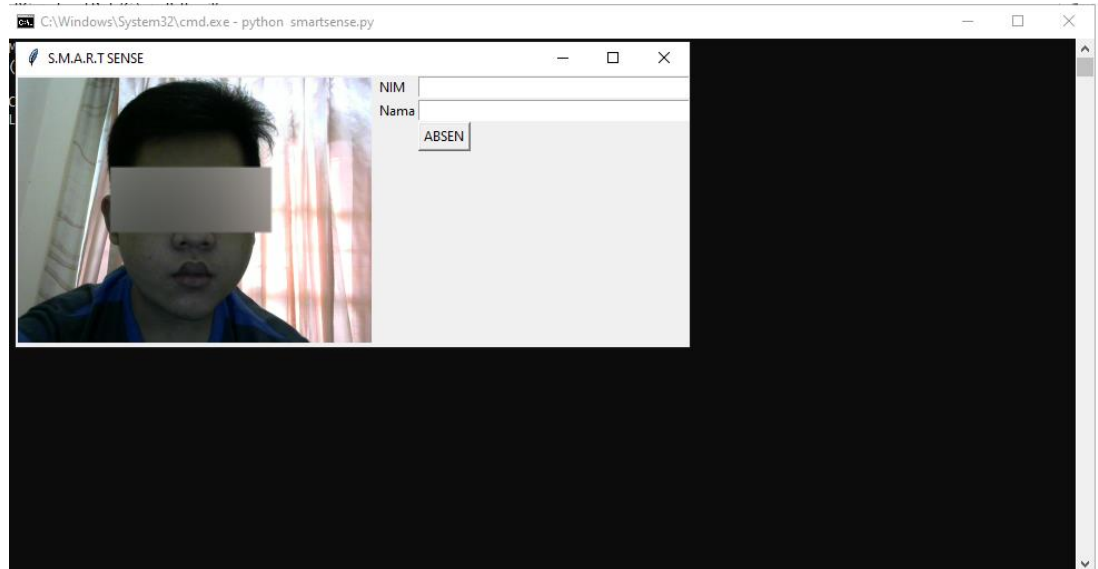


```
C:\Windows\System32\cmd.exe - python smartsense.py
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Python36>python smartsense.py
Load sample and learning ...
```

**Gambar 3.15** Sistem Mempelajari Sample

Pada Gambar 3.15 sistem sedang mempelajari dari sample foto wajah, apabila sistem selesai mempelajari struktur sample foto maka sistem akan memunculkan aplikasi S.M.A.R.T Sense dan siap digunakan untuk absensi kehadiran. Berikut Gambar 3.16 hasil *learning* sample oleh sistem.

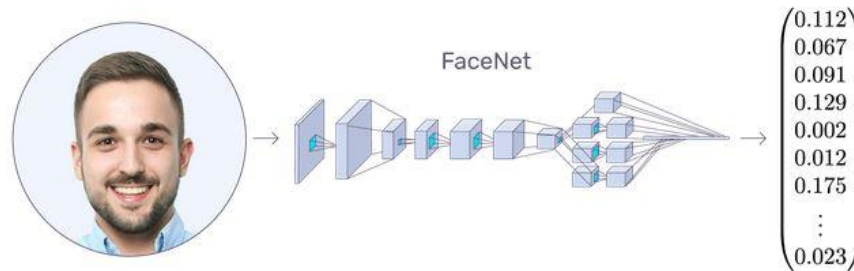


**Gambar 3.16** Hasil *Learning* Sample Oleh Sistem.

#### 5. *Face Recognition* Menggunakan Metode *128D Embedding*

Dalam melakukan *face recognition* diperlukan sebuah metode, yaitu metode *128D Embedding* untuk mempelajari struktur wajah dari foto sample wajah manusia. Sistem akan mempelajari wajah dari sample foto wajah lalu mengubah nya menjadi angka *vector* sebanyak 128 angka. Gambar 3.17 menjelaskan mengenai gambaran arsitektur jaringan *128D Embedding*.





*FaceNet takes an image of a face as input and outputs the embedding vector.*

**Gambar 3.17** Arsitektur *128D Embedding* (Arsfutura, 2019).

Berikut ini cara kerja *128D Embedding*:

1. Sistem akan mengambil sample foto wajah secara *random* lalu mengekstrak atau merubah nya ke dalam bentuk angka vektor sebanyak 128 angka.
2. *FaceNet* menghasilkan vektor acak untuk setiap gambar yang berarti gambar tersebar secara acak saat diplot
3. Sistem ini membagi 3 kategori, yaitu *anchor*, *picture positif*, dan *picture negative*. Berikut penjelasannya:
  - *Anchor* adalah wajah orang yang ingin diidentifikasi oleh sistem.
  - *Picture positif* adalah kategori gambar yang memiliki angka vektor mendekati *anchor*.
  - *Picture negative* adalah kategori gambar yang memiliki angka vector jauh atau berbeda dari *anchor*.

4. Saat wajah orang menghadapi didepan kamera sistem akan mencari sample foto wajah yang memiliki *output* vektor mendekati *anchor* secara acak.
5. Setelah sistem menemukan sample foto yang mendekati vektor *anchor* maka sistem akan melakukan pemisahan. Apabila sample foto memiliki *output* vektor mendekati atau sama dengan *anchor* maka sistem akan memasukan sample foto tersebut ke kategori *picture positive* dan apabila foto sample tersebut memiliki *output* vektor yang berbeda jauh dengan *anchor* maka sistem akan memasukan sample foto tersebut ke kategori *picture negative*. Teknik proses tersebut bisa juga disebut *Triplet Loss*.

### **3.2 Design System (Desain Sistem)**

Untuk tahapan ini penelitian akan memfokuskan pada penjadwalan pengerjaan penelitian. Pada Penelitian ini terdapat beberapa proses yang harus dilakukan dari tahap *requirement* hingga *integration & testing*, maka dari itu diperlukan penjadwalan yang tepat agar penelitian ini dapat selesai tepat pada waktunya, berikut penjadwalan penelitian berdasarkan aktifitas yang dilakukan dengan skala waktu per minggu. Tabel 3.2 Akan melihat jadwal pengerjaan dalam penelitian ini.

Tabel 3.2 Penjadwalan Penelitian

Tahapan	No	Aktifitas	Oktober				November				Desember				Januari	
			1	2	3	4	1	2	3	4	1	2	3	4	1	2
<i>Requirement</i>	1	Pengumpulan Data														
	2	Analisis														
	3	Pendefinisian Fungsi														
<i>Design System</i>	1	Penjadwalan														
	2	Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak														
<i>Coding &amp; Testing</i>	1	Perancangan Sistem menggunakan (UML)														
	2	User Interface Webcam														
	3	Pengkodean														
<i>Integration &amp; Testing</i>	2	Testing (Blackbox)														

### 3.2.1 Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak

Untuk membangun dan menjalankan aplikasi ini dibutuhkan persyaratan minimal perangkat keras agar aplikasi dapat berjalan dengan semestinya dan tanpa kendala, Tabel 3.3 menjelaskan persyaratan minimal perangkat keras yang digunakan untuk membangun dan menjalankan aplikasi.

**Tabel 3.3** Spesifikasi Kebutuhan Perangkat Keras

	<i>Developer</i>	<i>User</i>
Perangkat Keras	<i>Processor Intel Core i5 Gen 7 2.50GHz</i>	<i>Processor Intel Core i3 Gen 7 1.80GHz</i>
	<i>RAM 8 Gb</i>	<i>RAM 4 Gb</i>
	<i>Free Storage Harddisk 5 Gb</i>	<i>Free Storage Harddisk 2 Gb</i>
	<i>Grafis NVIDIA Geforce 940MX 2 Gb</i>	<i>Grafis NVIDIA GeForce GT 940MX 2 Gb</i>
	<i>GPU HD Graphics 620</i>	<i>GPU HD Graphics 505</i>
	<i>Monitor dengan resolusi layar 1280x768</i>	<i>Monitor dengan resolusi layar 1280x768</i>
	<i>Keyboard</i>	<i>Keyboard</i>
	<i>Mouse</i>	<i>Mouse</i>

Selain perangkat keras untuk membangun sebuah aplikasi dan menjalankan aplikasi, membutuhkan spesifikasi perangkat lunak agar aplikasi dapat berjalan dengan lancar dan tanpa kendala, Tabel 3.4 menjelaskan spesifikasi minimal perangkat lunak yang digunakan untuk membangun dan menjalankan aplikasi.

**Tabel 3.4** Spesifikasi Kebutuhan Perangkat Lunak

	<i>Developer</i>	<i>User</i>
Perangkat Lunak	<i>Sistem Operasi Windows 10</i>	<i>Sistem Operasi Windows 10</i>
	<i>Python 3.6.8 versi Windows Operating System</i>	<i>Browser (Mozilla Firefox / Google Chrome / Opera / internet Explorer)</i>
	<i>Notepad++</i>	<i>CMD (Command Prompt)</i>
	<i>Visual Studio Code</i>	
	<i>XAMPP (Apache, MySQL, PHP)</i>	
	<i>Browser (Mozilla Firefox / Google Chrome / Opera / internet Explorer)</i>	
	<i>CMD (Command Prompt)</i>	

### 3.3 Coding & Testing (Penulisan Sinkode Program / Implementasi)

Dalam tahapan ini akan dilakukan perancangan dengan menggunakan Diagram *Unified Modeling Language (UML)* diantaranya *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, *deployment diagram* dan *Entity Relationship Diagram (ERD)*.

### 3.3.1 Use Case Diagram

Use case diagram menggambarkan proses dari sebuah *system*, hubungan antara *use case* dan *actor* berdasarkan kebutuhan *system* dan menggambarkan fungsionalitas yang diharapkan dari sebuah *system*.

#### 1. Deskripsi Actor

Tabel 3.5 menjelaskan deskripsi aktor yang terlibat dalam aplikasi ini.

**Tabel 3.5** Deskripsi Actor

No	Aktor	Keterangan
1	<i>User</i>	<i>User</i> adalah orang yang melakukan absen kehadiran berdasarkan hasil <i>face recognition</i> . <i>User</i> dapat menjalankan aplikasi <i>face recognition</i> dengan menggunakan konsol CMD ( <i>Command Prompt</i> ), lalu hasil dari <i>face recognition</i> tersebut akan otomatis tersimpan di <i>website</i> laporan absensi.
2	<i>Admin</i>	<i>Admin</i> merupakan orang yang mengelola informasi absensi kehadiran di <i>website</i> S.M.A.R.T Sense.

#### 2. Deskripsi Use Case

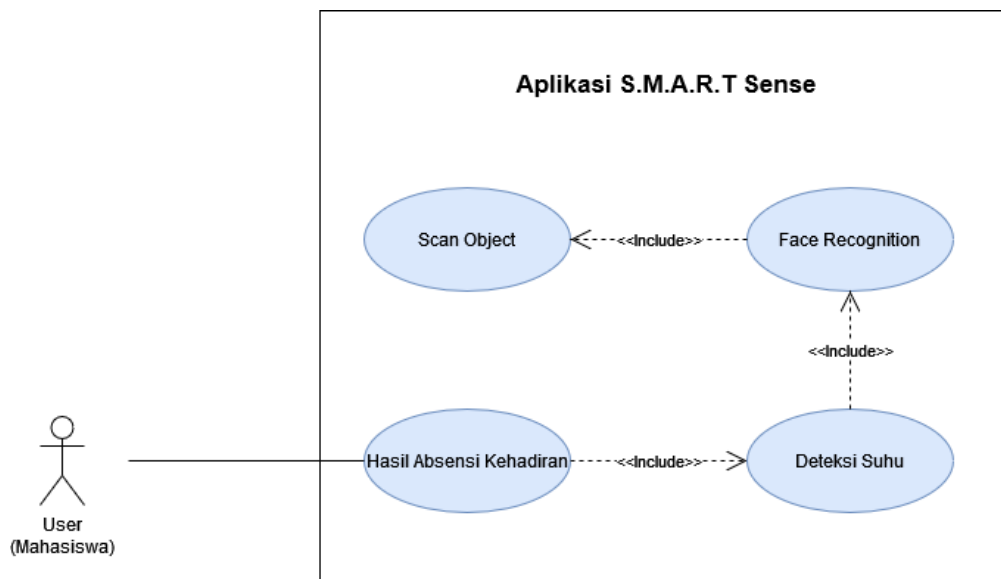
Tabel 3.6 menjelaskan lebih lanjut deskripsi *use case* pada aplikasi ini.

Tabel 3.6 Deskripsi *Use Case*

No	<i>Use Case</i>	Deskripsi
1	<i>Scan Object</i>	Merupakan fitur yang dapat dilakukan oleh <i>user</i> sebelum mengisi absensi kehadiran, <i>user</i> harus mengarahkan wajah tepat didepan kamera yang selanjutnya akan diidentifikasi oleh sistem. Fitur ini dapat dijalankan dengan menggunakan konsol CMD ( <i>Command Prompt</i> ).
2	<i>Face Recognition</i>	Merupakan proses dimana wajah yang ditangkap oleh kamera akan dikenali, diidentifikasi dan diproses oleh <i>machine learning</i> untuk dapat menghasilkan NIM, nama, waktu, dan tanggal saat melakukan absensi kehadiran. Fitur ini dapat dijalankan dengan menggunakan aplikasi S.M.A.R.T Sense yang dijalankan melalui konsol CMD ( <i>Command Prompt</i> ).
3	Deteksi Suhu	Merupakan fitur dimana sistem akan memulai pengukuran terhadap jumlah energi panas yang dihasilkan dari suhu lingkungan dan suhu objek. Sistem ini akan mendeteksi suhu <i>user</i> dengan mengarahkan tangan atau dahi pada sensor suhu. Sistem akan memasukan <i>user</i> ke dalam data absensi kehadiran apabila suhu <i>user</i> melebihi dari suhu lingkungan. Fitur ini dapat dijalankan dengan menggunakan aplikasi S.M.A.R.T Sense yang dijalankan melalui konsol CMD ( <i>Command Prompt</i> ).
4	Hasil Absensi Kehadiran	Merupakan fitur dimana <i>user</i> dapat melihat langsung hasil absensi kehadiran, lengkap dengan NIM, nama, perbandingan suhu lingkungan dan <i>user</i> serta notifikasi pemberitahuan suara di aplikasi S.M.A.R.T Sense. Fitur ini dapat dijalankan dengan menggunakan konsol CMD ( <i>Command Prompt</i> ).
5	Login	Merupakan fitur <i>login</i> dimana <i>admin</i> harus mengisi <i>username</i> dan <i>password</i> untuk <i>login</i> ke <i>website</i> S.M.A.R.T Sense agar dapat mengelola informasi laporan absensi kehadiran.

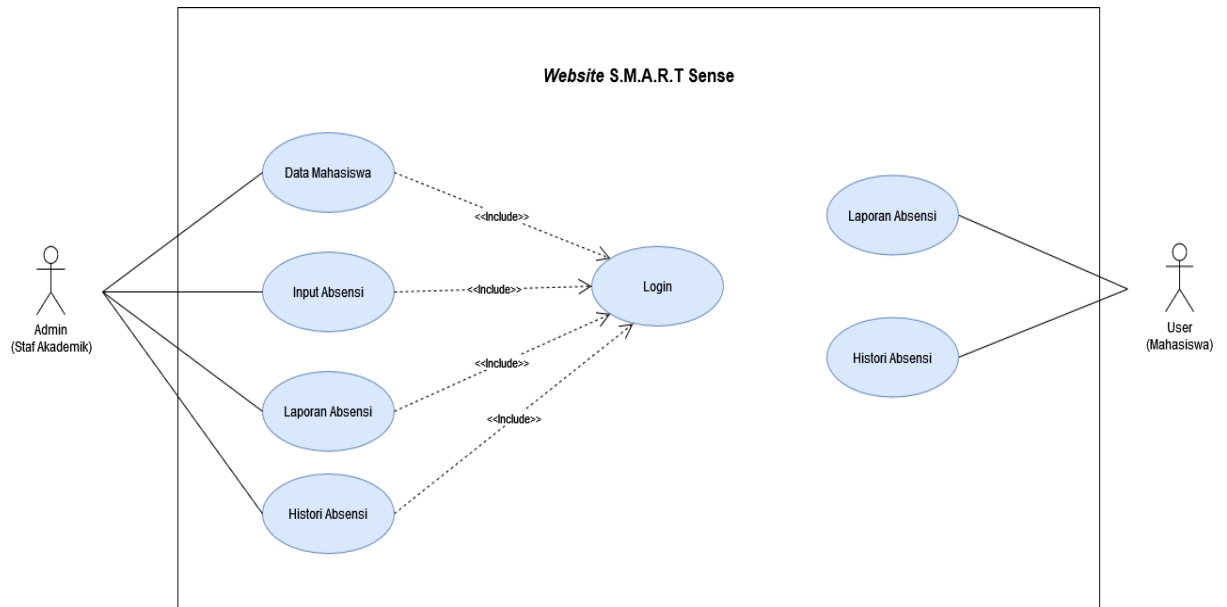
6	Laporan Absensi	Merupakan fitur dimana <i>user</i> bisa melihat laporan absensi kehadiran dengan menggunakan wajah lengkap dengan NIM, nama, waktu, tanggal, NP dosen, nama dosen, mata kuliah, jam keberapa dan ruangan. Sedangkan <i>admin</i> untuk mengelola absen kehadiran seperti menyimpan ke histori dan mereset untuk laporan absensi yang baru.
7	Data Mahasiswa	Merupakan fitur dimana <i>admin</i> dapat melihat daftar mahasiswa yang terdaftar di aplikasi S.M.A.R.T Sense.
8	Input Absensi	Merupakan fitur dimana <i>admin</i> mempersiapkan absensi kehadiran dengan mengisi form seperti nama dosen, mata kuliah, jam keberapa dan ruangan.
9	Histori Absensi	Merupakan fitur dimana <i>user</i> dapat melihat semua absensi baik itu di waktu sebelumnya maupun waktu sekarang. Sedangkan untuk <i>admin</i> dapat menghapus histori absensi kehadiran apabila sudah tidak diperlukan.

Gambar 3.18 dan Gambar 3.19 menjelaskan *use case diagram* aplikasi.



**Gambar 3.18** Use Case Diagram Aplikasi S.M.A.R.T Sense





**Gambar 3.19** Use Case Diagram Website S.M.A.R.T Sense

Tabel 3.7 menggambarkan *flow of event* (skenario) dari tiap *use case diagram* untuk menjelaskan proses apa saja yang terjadi dan bagaimana respon yang dikeluarkan oleh sistem.

1. *Scan Object***Tabel 3.7** *Flow of Event Scan Object*

<b>Identifikasi</b>			
<b>Nomor Use Case</b>	1		
<b>Nama Use Case</b>	<i>Scan Object</i>		
<b>Deskripsi</b>	Merupakan fitur yang dapat dilakukan oleh <i>user</i> sebelum mengisi absensi kehadiran, <i>user</i> harus mengarahkan wajah tepat didepan kamera yang selanjutnya akan diidentifikasi oleh sistem. Fitur ini dapat dijalankan dengan menggunakan konsol CMD ( <i>Command Prompt</i> ).		
<b>Aktor</b>	<i>User</i>		
<b>Skenario</b>			
<b>Kondisi Awal</b>	Konsol CMD ( <i>Command Prompt</i> )		
No	Aksi Aktor	No	Respon Sistem
1	Jalankan aplikasi <i>S.M.A.R.T SENSE</i> melalui CMD	2	Menampilkan <i>S.M.A.R.T Sense</i> dan membuka layar kamera
3	Mengarahkan wajah tepat didepan kamera	4	<i>Scan object</i>
<b>Kondisi Akhir</b>	Wajah di <i>scan</i> oleh kamera		

2. *Face Recognition***Tabel 3.8** *Flow of Event Face Recognition*

<b>Identifikasi</b>			
<b>Nomor Use Case</b>		2	
<b>Nama Use Case</b>		<i>Face Recognition</i>	
<b>Deskripsi</b>		Merupakan proses dimana wajah yang ditangkap oleh kamera akan dikenali, diidentifikasi dan diproses oleh <i>machine learning</i> untuk dapat menghasilkan NIM, nama, waktu, dan tanggal saat melakukan absensi kehadiran. Fitur ini dapat dijalankan dengan menggunakan aplikasi <i>S.M.A.R.T Sense</i> yang dijalankan melalui konsol CMD ( <i>Command Prompt</i> ).	
<b>Aktor</b>		<i>User</i>	
<b>Skenario</b>			
<b>Kondisi Awal</b>		Menampilkan Aplikasi <i>Face Recognition</i>	
No	Aksi Aktor	No	Respon Sistem
1	Mengarahkan wajah didepan kamera	2	Mendeteksi apakah ada wajah yang tertangkap kamera dan menampilkan wajah yang terdeteksi kamera
3	Menunggu hasil deteksi	4	Memproses hasil
<b>Kondisi Akhir</b>		Hasil deteksi wajah diidentifikasi sistem	

## 3. Deteksi Suhu

**Tabel 3.9** *Flow of Event* Deteksi Suhu

<b>Identifikasi</b>			
<b>Nomor Use Case</b>		3	
<b>Nama Use Case</b>		Deteksi Suhu	
<b>Deskripsi</b>		Merupakan fitur dimana sistem akan memulai pengukuran terhadap jumlah energi panas yang dihasilkan dari suhu lingkungan dan suhu objek. Sistem ini akan mendeteksi suhu <i>user</i> dengan mengarahkan tangan atau dahi pada sensor suhu. Sistem akan memasukan <i>user</i> ke dalam data absensi kehadiran apabila suhu <i>user</i> lebih dari suhu lingkungan. Fitur ini dapat dijalankan dengan menggunakan aplikasi S.M.A.R.T Sense yang dijalankan melalui konsol CMD ( <i>Command Prompt</i> ).	
<b>Aktor</b>		<i>User</i>	
<b>Skenario</b>			
<b>Kondisi Awal</b>		Menampilkan Aplikasi <i>Face Recognition</i>	
No	Aksi Aktor	No	Respon Sistem
1	Mengarahkan tangan ke sensor suhu	2	Menampilkan hasil deteksi perbandingan suhu lingkungan dan suhu <i>user</i>
3	Melihat hasil hasil deteksi perbandingan suhu lingkungan dan suhu user	4	Memproses hasil
<b>Kondisi Akhir</b>		Tampilan informasi perbandingan antara suhu lingkungan dan suhu <i>user</i>	

## 4. Hasil Absensi Kehadiran

Tabel 3.10 *Flow of Event* Hasil Absensi Kehadiran

<b>Identifikasi</b>			
<b>Nomor Use Case</b>	4		
<b>Nama Use Case</b>	Hasil Absensi Kehadiran		
<b>Deskripsi</b>	Merupakan fitur dimana <i>user</i> dapat melihat langsung hasil absensi kehadiran, lengkap dengan NIM, nama, perbandingan suhu lingkungan dan <i>user</i> serta notifikasi pemberitahuan suara di aplikasi S.M.A.R.T Sense. Fitur ini dapat dijalankan dengan menggunakan konsol CMD ( <i>Command Prompt</i> ).		
<b>Aktor</b>	<i>User</i>		
<b>Skenario</b>			
<b>Kondisi Awal</b>	Menampilkan Aplikasi <i>Face Recognition</i>		
No	Aksi Aktor	No	Respon Sistem
1	Menunggu hasil deteksi	2	Menampilkan hasil <i>Face Recognition</i> yaitu NIM, nama, perbandingan suhu lingkungan dan <i>user</i> serta notifikasi pemberitahuan suara.
3	Melihat hasil Absensi Kehadiran	4	Hasil <i>Face Recognition</i> disimpan
<b>Kondisi Akhir</b>	Hasil deteksi wajah dilihat oleh <i>user</i> dan otomatis tersimpan di <i>website</i> laporan absensi kehadiran.		

## 5. Login

Tabel 3.11 Flow of Event Login

Identifikasi			
<b>Nomor Use Case</b>		5	
<b>Nama Use Case</b>		<i>Login</i>	
<b>Deskripsi</b>		Merupakan fitur <i>login</i> dimana <i>admin</i> harus mengisi <i>username</i> dan <i>password</i> untuk <i>login</i> ke <i>website S.M.A.R.T Sense</i> agar dapat mengelola informasi laporan absensi kehadiran.	
<b>Aktor</b>		<i>Admin</i>	
Skenario			
<b>Kondisi Awal</b>		Menu <i>Login</i>	
No	Aksi Aktor	No	Respon Sistem
1	Melihat menu <i>Login</i> .	2	Menampilkan form <i>Login</i> .
3	Memasukkan <i>username</i> dan <i>password</i> . Lalu menekan tombol masuk.	4	Memvalidasi data ( <i>username</i> & <i>password</i> ) <i>user</i> ke dalam <i>database</i> . Jika data <i>user valid</i> maka sistem akan menampilkan menu utama ( <i>dashboard</i> ) sesuai hak akses <i>user</i> , jika tidak <i>valid</i> maka sistem akan menampilkan peringatan kesalahan dan <i>user</i> harus menginputkan data <i>Login</i> kembali.
<b>Kondisi Akhir</b>		<i>Admin</i> masuk ke dalam menu utama ( <i>Dashboard</i> ).	

## 6. Laporan Absensi

**Tabel 3.12** *Flow of Event* Laporan Absensi *User*

<b>Identifikasi</b>			
<b>Nomor Use Case</b>		6	
<b>Nama Use Case</b>		Laporan Absensi	
<b>Deskripsi</b>		Merupakan fitur dimana <i>user</i> bisa melihat laporan absensi kehadiran dengan menggunakan wajah lengkap dengan NIM, nama, waktu, tanggal, NP dosen, nama dosen, mata kuliah, jam keberapa dan ruangan.	
<b>Aktor</b>		<i>User</i>	
<b>Skenario</b>			
<b>Kondisi Awal</b>		<i>Website</i> Menu Laporan Absensi	
No	Aksi Aktor	No	Respon Sistem
1	Membuka <i>website</i> laporan absensi <i>S.M.A.R.T Sense</i>	2	Menampilkan hasil absensi kehadiran yang disimpan dalam <i>website</i> laporan absensi dan menampilkan NIM, nama, waktu, tanggal, NP, nama dosen, mata kuliah, jam keberapa, dan ruangan
3	Melihat hasil absensi kehadiran.		
<b>Kondisi Akhir</b>		Tampilan informasi mengenai laporan absensi kehadiran.	

**Tabel 3.13** *Flow of Event* Laporan Absensi Admin

<b>Identifikasi</b>			
<b>Nomor Use Case</b>		7	
<b>Nama Use Case</b>		Laporan Absensi	
<b>Deskripsi</b>		Merupakan fitur dimana <i>admin</i> mengelola absen kehadiran seperti menyimpan ke histori dan mereset untuk laporan absensi yang baru.	
<b>Aktor</b>		<i>Admin</i>	
<b>Skenario</b>			
<b>Kondisi Awal</b>		<i>Website</i> Menu Laporan Absensi	
No	Aksi Aktor	No	Respon Sistem
1	Membuka website laporan absensi <i>S.M.A.R.T Sense</i>	2	Menampilkan hasil absensi kehadiran yang disimpan dalam website laporan absensi dan menampilkan NIM, nama, waktu, tanggal, NP, nama dosen, mata kuliah, jam keberapa, ruangan, tombol <i>reset</i> , dan tombol simpan
3	Klik simpan	4	Data laporan absensi pindah ke menu histori
5	Klik <i>reset</i>	6	Data laporan absensi dihapus
<b>Kondisi Akhir</b>		Data dikelola oleh <i>admin</i> apabila mengklik simpan data absensi kehadiran pindah ke menu histori, apabila <i>admin</i> mengklik <i>reset</i> data absensi kehadiran dihapus.	



## 7. Data Mahasiswa

**Tabel 3.14** Flow of *Event* Data Mahasiswa

<b>Identifikasi</b>			
<b>Nomor Use Case</b>		8	
<b>Nama Use Case</b>		Data Mahasiswa	
<b>Deskripsi</b>		Merupakan fitur dimana <i>admin</i> dapat melihat daftar mahasiswa yang terdaftar di aplikasi <i>S.M.A.R.T Sense</i> .	
<b>Aktor</b>		<i>Admin</i>	
<b>Skenario</b>			
<b>Kondisi Awal</b>		<i>Website</i> Menu Data Mahasiswa	
No	Aksi Aktor	No	Respon Sistem
1	Membuka <i>website</i> menu data mahasiswa <i>S.M.A.R.T Sense</i>	2	Menampilkan daftar mahasiswa yang terdaftar di aplikasi <i>S.M.A.R.T Sense</i> dalam menu tersebut tertera NIM dan nama mahasiswa.
3	Melihat daftar mahasiswa yang terdaftar di aplikasi <i>S.M.A.R.T Sense</i>		
<b>Kondisi Akhir</b>		Tampilan informasi mengenai data mahasiswa yang terdaftar di aplikasi <i>S.M.A.R.T Sense</i> .	

## 8. Input Absensi

Tabel 3.15 *Flow of Event Input Absensi*

<b>Identifikasi</b>			
<b>Nomor Use Case</b>		9	
<b>Nama Use Case</b>		Input Absensi	
<b>Deskripsi</b>		Merupakan fitur dimana <i>admin</i> mempersiapkan absensi kehadiran dengan mengisi form seperti nama dosen, mata kuliah, jam keberapa dan ruangan.	
<b>Aktor</b>		<i>Admin</i>	
<b>Skenario</b>			
<b>Kondisi Awal</b>		<i>Website</i> Menu Input Absensi	
No	Aksi Aktor	No	Respon Sistem
1	Membuka <i>website</i> menu input absen <i>S.M.A.R.T Sense</i>	2	Menampilkan form yang harus diisi dengan cara mengklik tombol dropdown. Tertera nama dosen, matakuliah, jam keberapa, ruangan dan tombol <i>submit</i> .
3	Klik <i>dropdown</i> dosen	4	Menampilkan daftar nama – nama dosen
5	Klik <i>dropdown</i> Jam ke	6	Menampilkan daftar jam keberapa saja
7	Klik <i>dropdown</i> matakuliah	8	Menampilkan daftar nama – nama matakuliah
9	Klik dropdown ruangan	10	Menampilkan daftar nama – nama ruangan
11	Klik tombol <i>submit</i>	12	Data disimpan ke menu laporan absensi.
<b>Kondisi Akhir</b>		Data yang sudah terisi akan disimpan ke menu laporan absensi.	

## 9. Histori Absensi

**Tabel 3.16** *Flow of Event* Histori Absensi *User*

<b>Identifikasi</b>			
<b>Nomor Use Case</b>		10	
<b>Nama Use Case</b>		Histori Absensi	
<b>Deskripsi</b>		Merupakan fitur dimana <i>user</i> dapat melihat semua absensi baik itu di waktu sebelumnya maupun waktu sekarang.	
<b>Aktor</b>		<i>User</i>	
<b>Skenario</b>			
<b>Kondisi Awal</b>		<i>Website</i> Menu Histori Absensi	
<b>No</b>	<b>Aksi Aktor</b>	<b>No</b>	<b>Respon Sistem</b>
1	Membuka <i>website</i> menu histori absensi <i>S.M.A.R.T Sense</i>	2	Menampilkan histori absensi kehadiran diwaktu sebelumnya dan waktu sekarang.
3	Melihat histori absensi		
<b>Kondisi Akhir</b>		Tampilan informasi mengenai histori absensi kehadiran.	

Tabel 3.17 *Flow of Event* Histori Absensi Admin

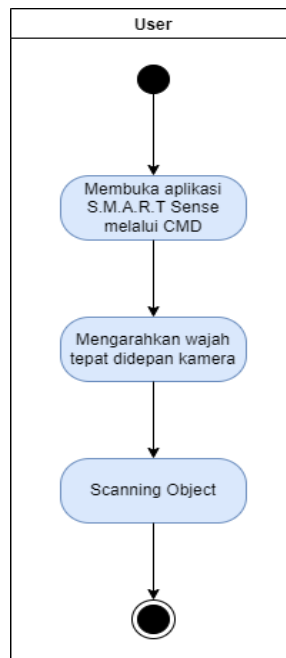
<b>Identifikasi</b>			
<b>Nomor Use Case</b>		11	
<b>Nama Use Case</b>		Histori Absensi	
<b>Deskripsi</b>		Merupakan fitur dimana <i>admin</i> dapat menghapus histori absensi kehadiran apabila sudah tidak diperlukan.	
<b>Aktor</b>		<i>Admin</i>	
<b>Skenario</b>			
<b>Kondisi Awal</b>		<i>Website</i> Menu Histori Absensi	
No	Aksi Aktor	No	Respon Sistem
1	Membuka <i>website</i> menu histori absensi <i>S.M.A.R.T Sense</i>	2	Menampilkan histori absensi kehadiran diwaktu sebelumnya, waktu sekarang dan tombol reset.
3	Klik tombol reset	4	Seluruh data histori absensi dihapus
<b>Kondisi Akhir</b>		Semua data histori absensi terhapus lalu di <i>website</i> menampilkan notifikasi “Tidak ada data”	

### 3.3.2 Activity Diagram

*Activity diagram* digunakan untuk menggambarkan *event – event* yang terjadi dalam *use case diagram*. Berdasarkan *use case diagram* aplikasi S.M.A.R.T Sense pada Gambar 3.18 dan Gambar 3.19, maka dibuat *activity diagram* dibawah ini.

#### 1. *Scan Object*

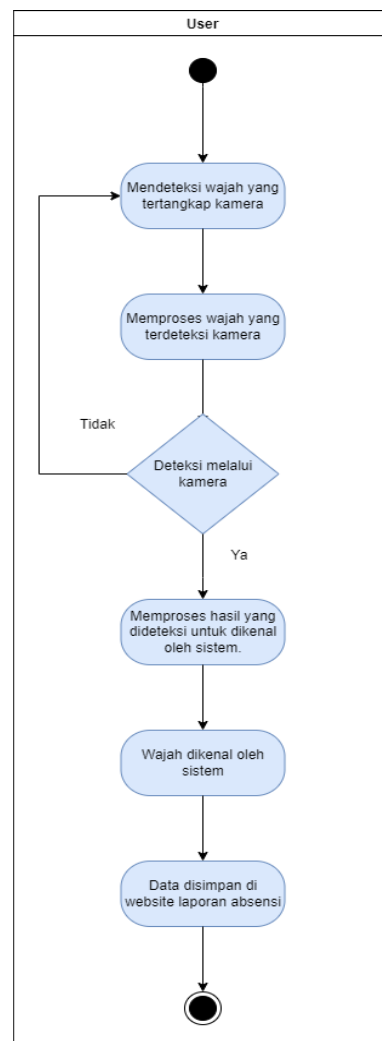
Pada Gambar 3.20 menunjukkan aktifitas pada saat *user* menjalankan aplikasi S.M.A.R.T Sense. Ketika *user* ingin menjalankan aplikasi S.M.A.R.T Sense maka *user* harus menjalankan perintah dengan CMD untuk membuka aplikasi S.M.A.R.T Sense tersebut, setelah itu maka sistem akan merespon dengan membuka aplikasi S.M.A.R.T Sense dan menampilkan layar kamera untuk dapat melakukan *scan object* (wajah).



**Gambar 3.20** *Activity Diagram Scan Object*

## 2. Face Recognition

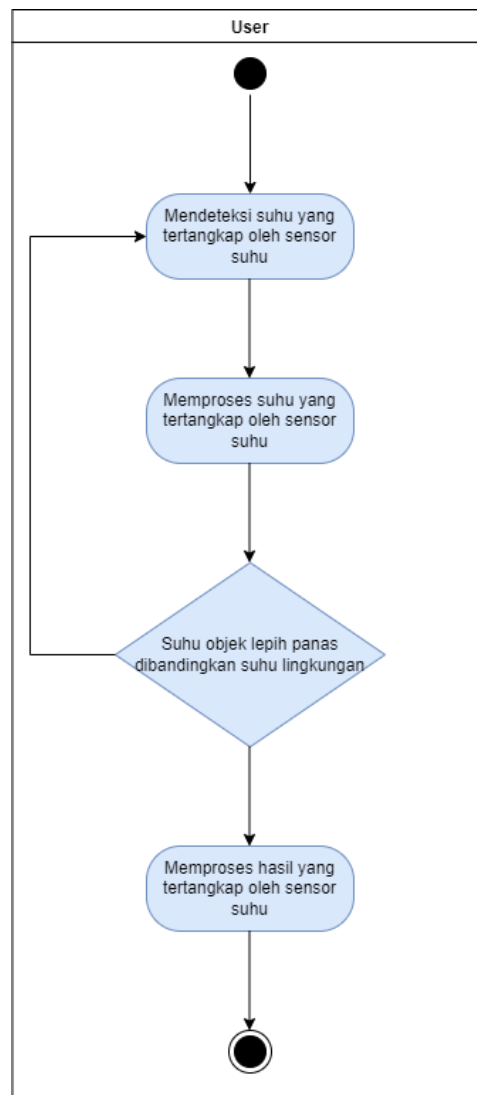
Pada Gambar 3.21 menunjukkan aktifitas ketika proses *scan object* sudah dilakukan dan selanjutnya wajah akan di deteksi oleh sistem melalui kamera, jika ada wajah yang terdeteksi maka wajah tersebut akan diproses untuk dikenali dan diproses hasilnya. Hasil absensi kehadiran tersebut sudah otomatis akan tersimpan di *website* laporan absensi.



**Gambar 3.21** Activity Diagram Face Recognition

### 3. Deteksi Suhu

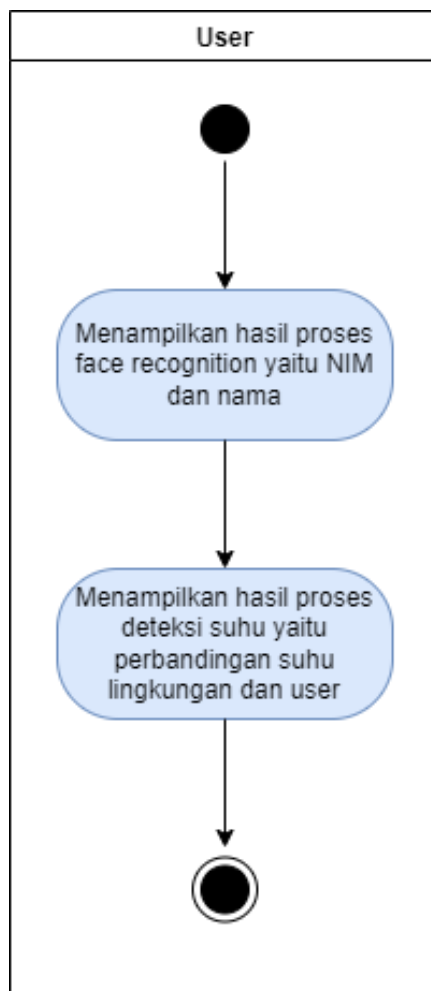
Pada Gambar 3.22 Menunjukkan aktifitas disaat proses *face recognition* sudah dilakukan dan selanjutnya melakukan pendeteksian suhu oleh sensor suhu, sensor suhu akan mendeteksi suhu lingkungan dan suhu objek dan akan diproses hasilnya.



**Gambar 3.22** Activity Diagram Deteksi Suhu

#### 4. Hasil Absensi Kehadiran

Pada Gambar 3.23 Menunjukkan aktifitas disaat proses *face recognition* dan deteksi suhu sudah dilakukan dan hasilnya sudah diproses, maka hasil *face recognition* berupa NIM dan nama akan ditampilkan oleh sistem melalui aplikasi S.M.A.R.T Sense sedangkan untuk hasil suhu yaitu perbandingan suhu lingkungan sekitar dan *user* akan ditampilkan oleh sistem melalui CMD (Command Prompt).

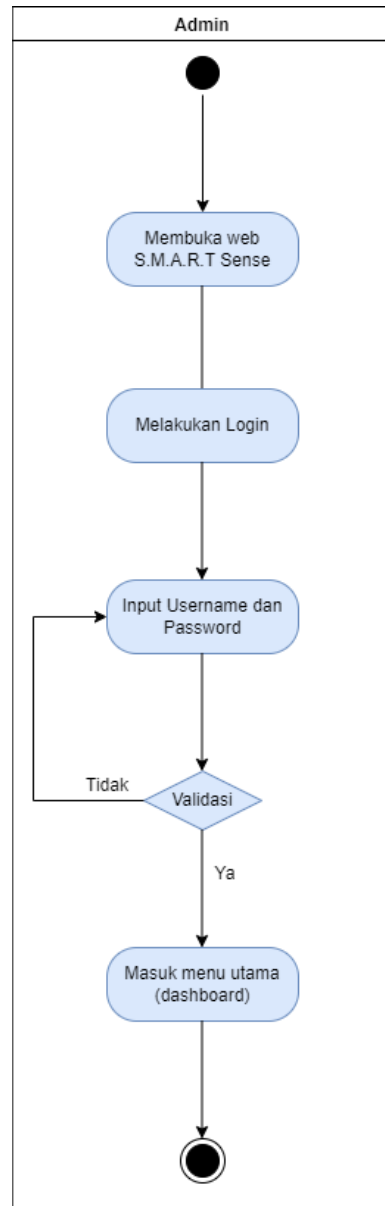


**Gambar 3.23** Activity Diagram Hasil Absensi Kehadiran



## 5. Login

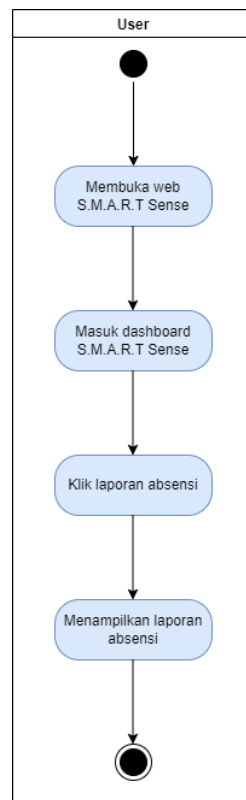
Pada Gambar 3.24 Menunjukkan aktifitas ketika *admin login* untuk masuk ke dalam menu utama (*dashboard*).



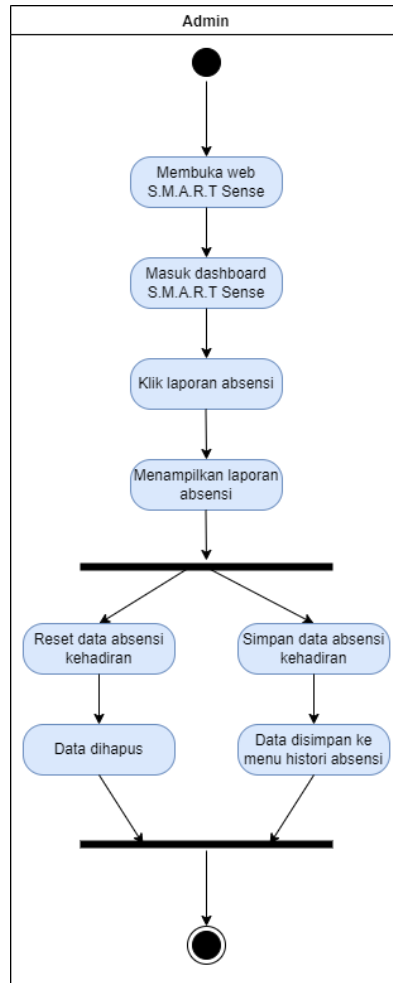
**Gambar 3.24** Activity Diagram Login

## 6. Laporan Absensi

Pada Gambar 3.25 Menunjukkan aktifitas ketika *user* ingin melihat laporan absensi dari hasil absensi kehadiran *face recognition*. *User* harus membuka *website* laporan absensi untuk dapat melihat laporan absensi tersebut. Sedangkan pada Gambar 3.26 Menunjukkan aktifitas ketika *admin* ingin mengelola laporan absensi. *Admin* bisa menyimpan laporan absensi dengan mengklik tombol simpan maka data absensi kehadiran akan tersimpan di menu histori absensi, apabila ingin memulai laporan absensi baru dengan mengklik tombol reset maka data absensi kehadiran akan terhapus.



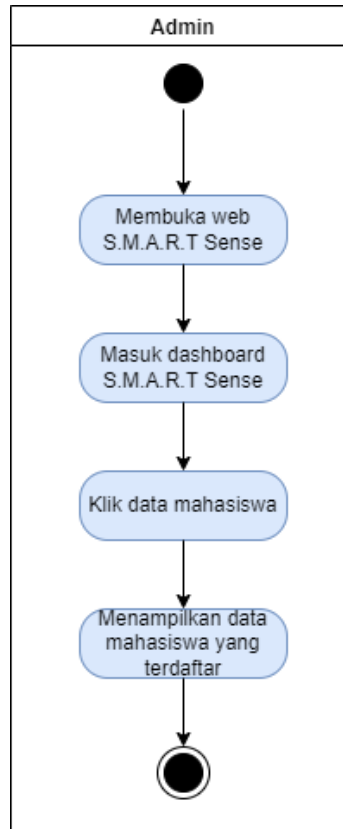
**Gambar 3.25** Activity Diagram Laporan Absensi User



**Gambar 3.26** Activity Diagram Laporan Absensi Admin

## 7. Data Mahasiswa

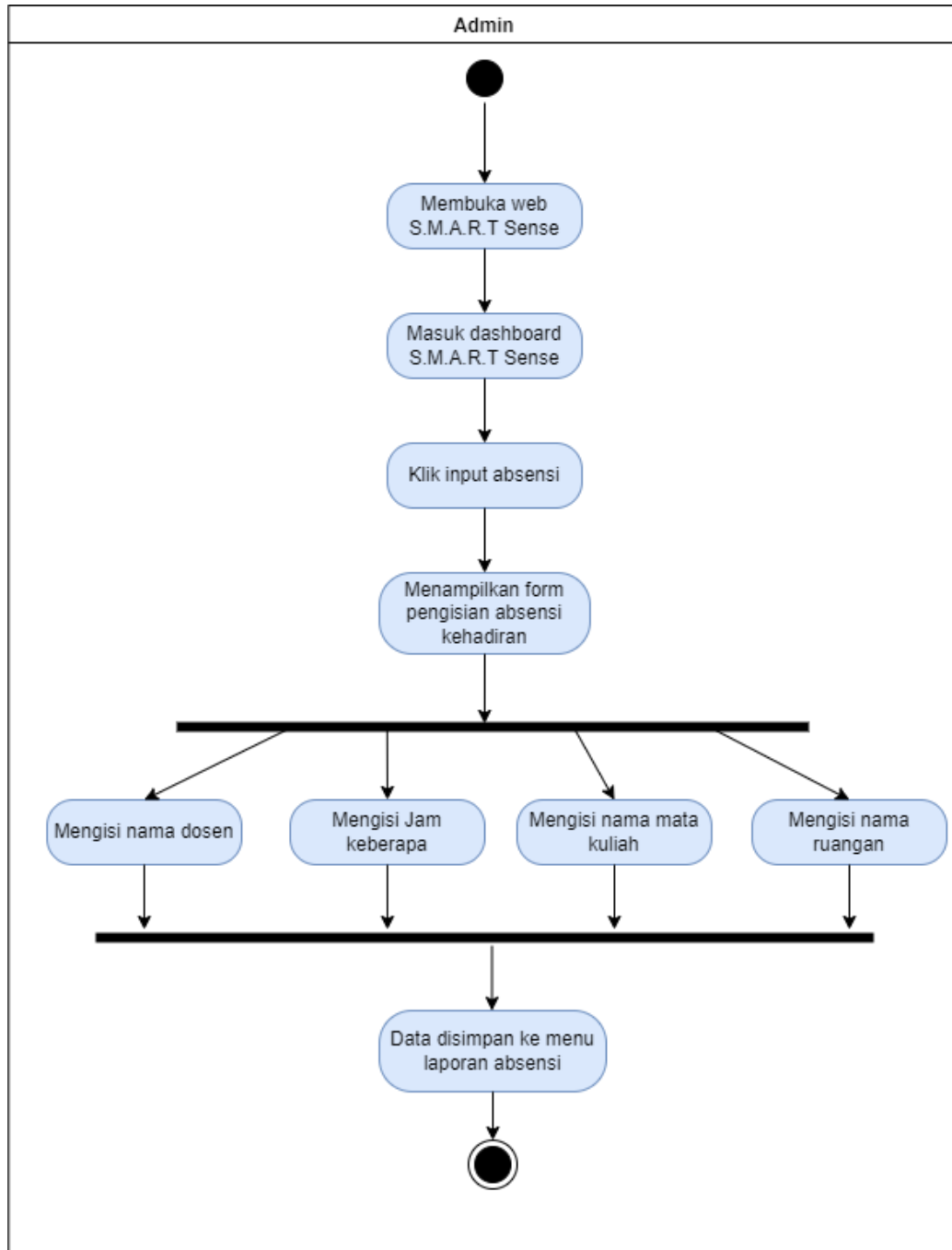
Pada Gambar 3.27 Menunjukkan aktifitas ketika *admin* ingin melihat data mahasiswa yang terdaftar di aplikasi S.M.A.R.T Sense. *Admin* harus membuka *website* daftar mahasiswa untuk dapat melihat siapa saja yang terdaftar di aplikasi tersebut.



**Gambar 3.27** Activity Diagram Data Mahasiswa

## 8. Input Absensi

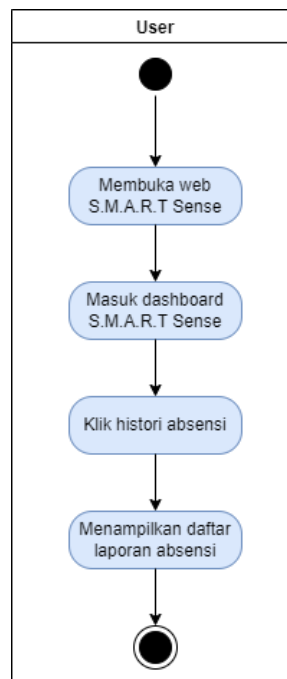
Pada Gambar 3.28 Menunjukkan aktifitas ketika *admin* ingin mempersiapkan absensi kehadiran. *Admin* harus mengisi form seperti nama dosen, mata kuliah, jam keberapa, dan ruangan.



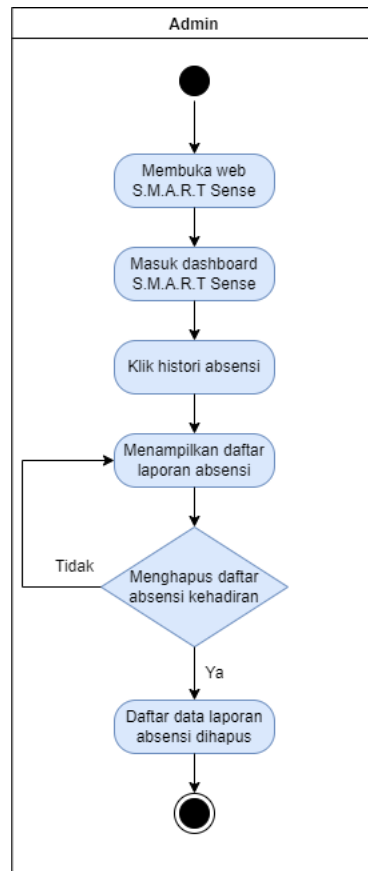
**Gambar 3.28** Activity Diagram Input Absensi

## 9. Histori Absensi

Pada Gambar 3.29 Menunjukkan aktifitas ketika *user* ingin melihat kumpulan laporan absensi baik itu di waktu sebelumnya maupun waktu sekarang. *User* harus membuka *website* histori absensi untuk dapat melihat kumpulan absensi tersebut. Sedangkan pada Gambar 3.30 Menunjukkan aktifitas ketika *admin* ingin melihat kumpulan laporan absensi baik itu di waktu sebelumnya maupun waktu sekarang serta ingin menghapus data tersebut apabila sudah tidak diperlukan. *Admin* harus membuka *website* histori absensi untuk dapat melihat dan menghapus kumpulan data laporan absensi tersebut.



**Gambar 3.29** Activity Diagram Histori Absensi User



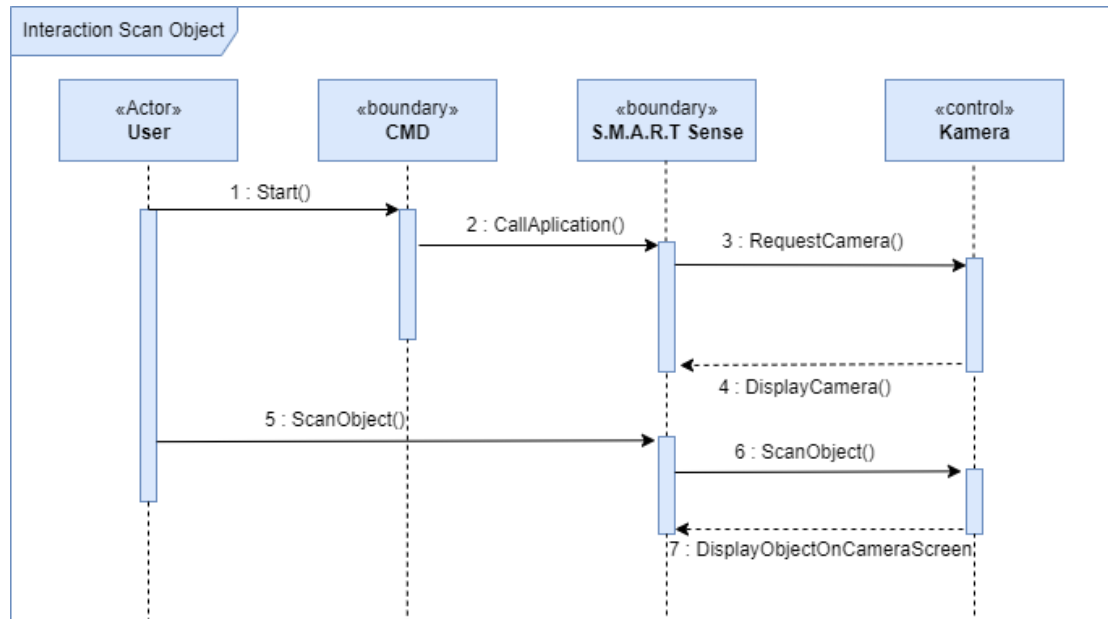
**Gambar 3.30** *Activity Diagram Histori Absensi Admin*

### 3.3.3 *Sequence Diagram*

*Sequence diagram* atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Objek-objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan.

### 1. Sequence Diagram Scan Object

Pada diagram *Scan Object* menjelaskan alur interaksi saat proses melakukan *scan wajah* pada wajah *user*.

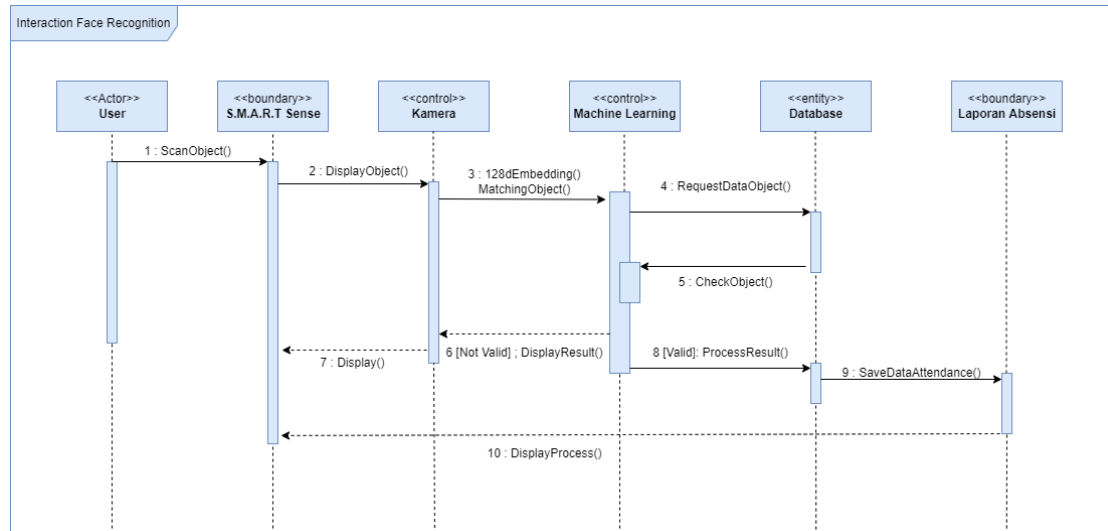


**Gambar 3.31** Sequence Diagram Scan Object

### 2. Sequence Diagram Face Recognition

Pada diagram *face recognition* menjelaskan alur interaksi saat proses melakukan pengenalan oleh sistem dengan menggunakan metode *128D Embedding* pada *machine learning* pada data yang terkait untuk menampilkan informasi.

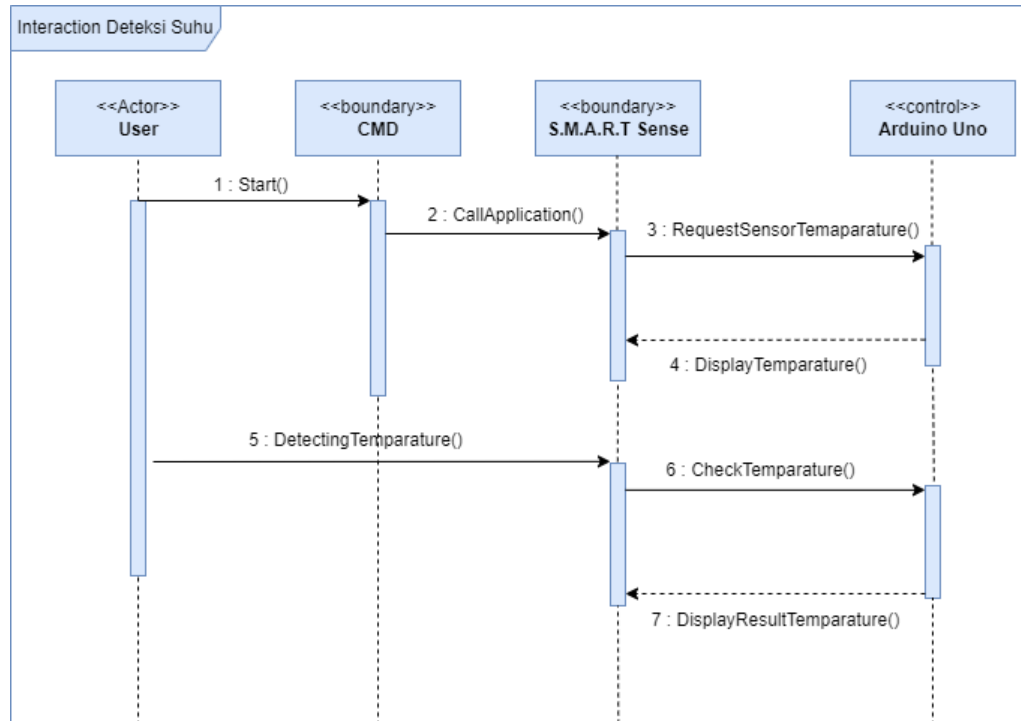




**Gambar 3.32** *Sequence Diagram Face Recognition*

### 3. *Sequence Diagram* Deteksi Suhu

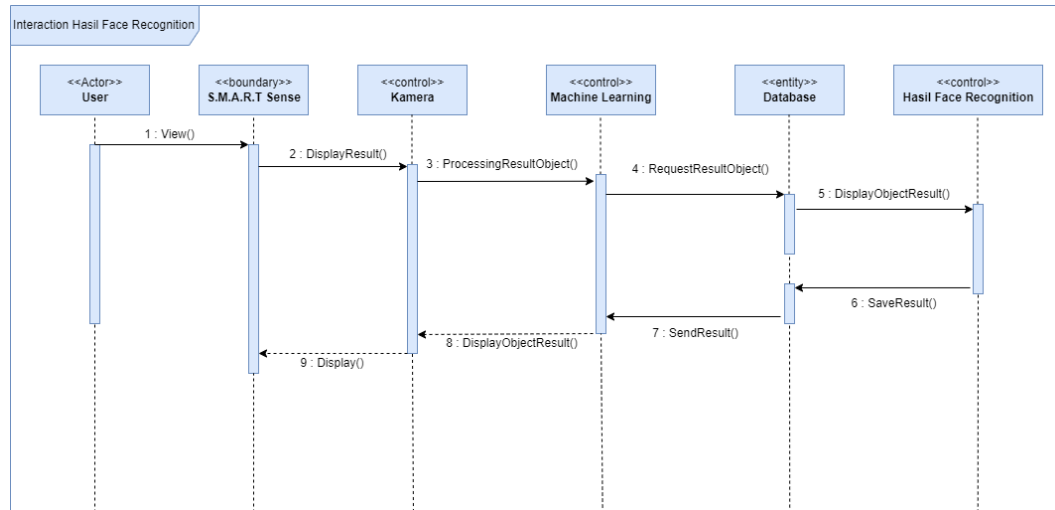
Pada diagram deteksi suhu menjelaskan alur interaksi saat proses mendeteksi suhu dengan menggunakan Arduino Uno



**Gambar 3.33** *Sequence Diagram* Deteksi Suhu

#### 4. *Sequence Diagram* Hasil *Face Recognition*

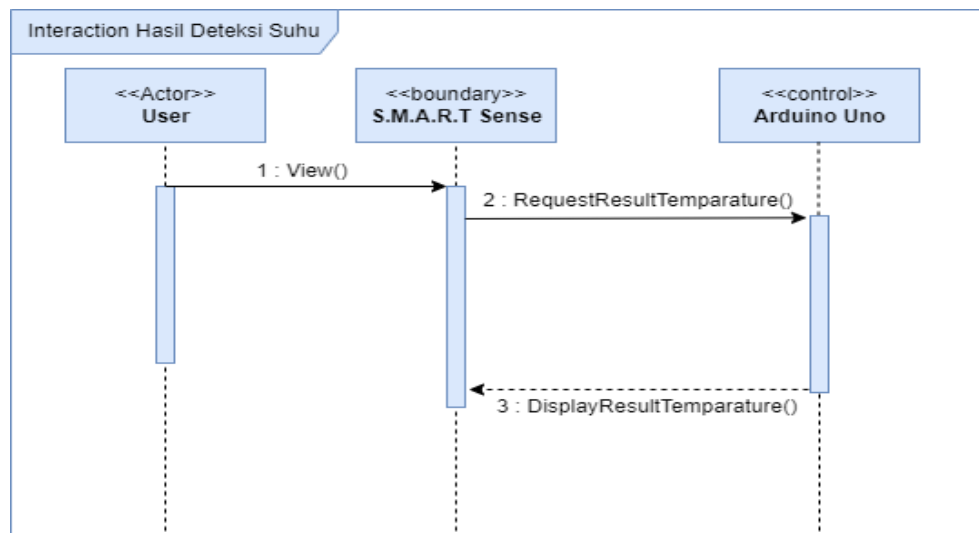
Pada diagram hasil *face recognition* menjelaskan alur interaksi saat proses program menampilkan informasi data *user*.



**Gambar 3.34** Sequence Diagram Hasil Face Recognition

#### 5. Sequence Diagram Hasil Deteksi Suhu

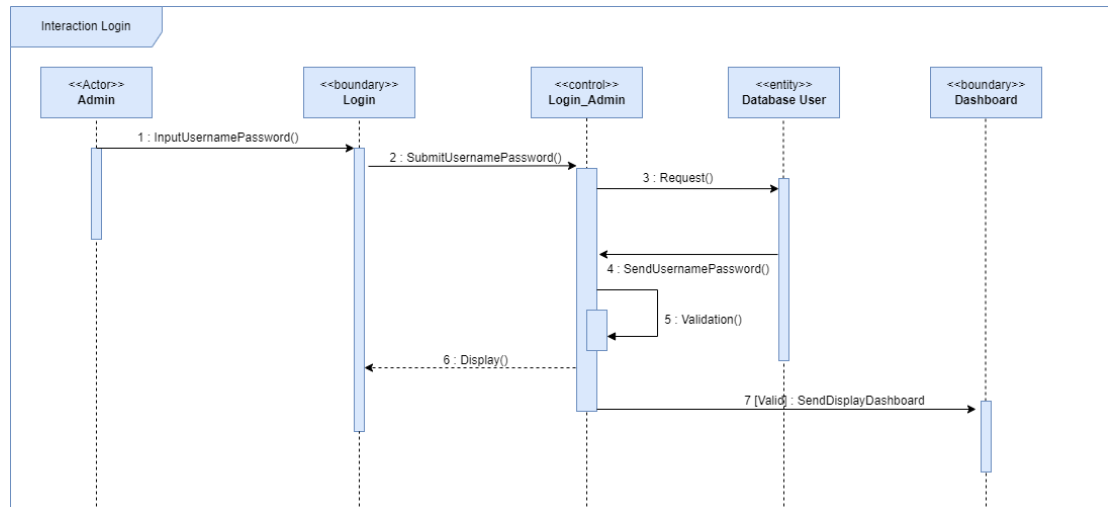
Pada diagram hasil deteksi suhu menjelaskan alur interaksi saat proses program menampilkan informasi suhu.



**Gambar 3.35** Sequence Diagram Hasil Deteksi Suhu

## 6. Sequence Diagram Login

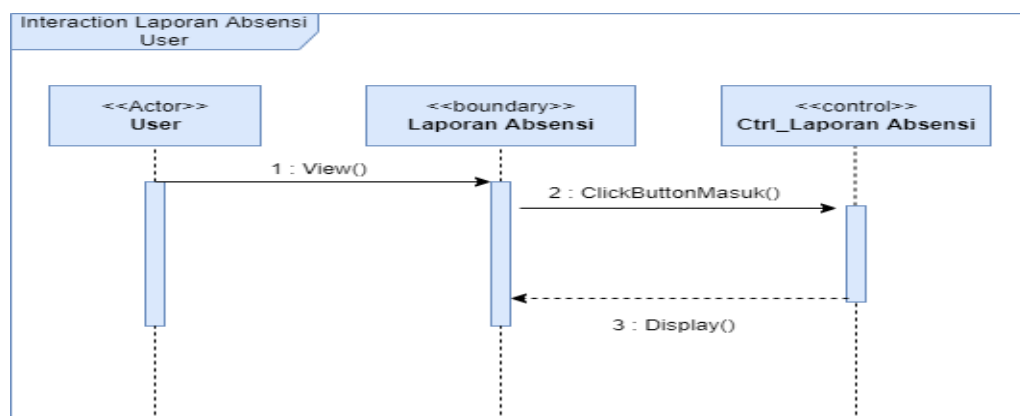
Pada diagram *login* menjelaskan alur interaksi saat proses *login* yang dilakukan oleh *admin*.



**Gambar 3.36** Sequence Diagram Login

## 7. Sequence Diagram Laporan Absensi User

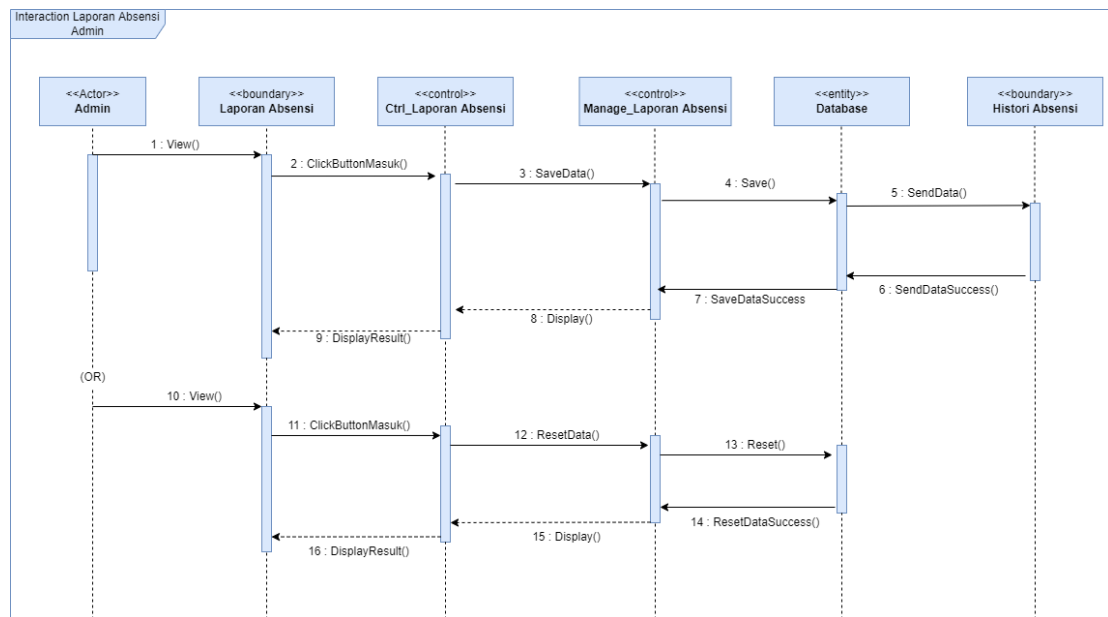
Pada diagram laporan absensi menjelaskan alur interaksi saat *user* ingin melihat laporan absensi di *website* S.M.A.R.T Sense.



**Gambar 3.37** Sequence Diagram Laporan Absensi User

## 8. Sequence Diagram Laporan Absensi Admin

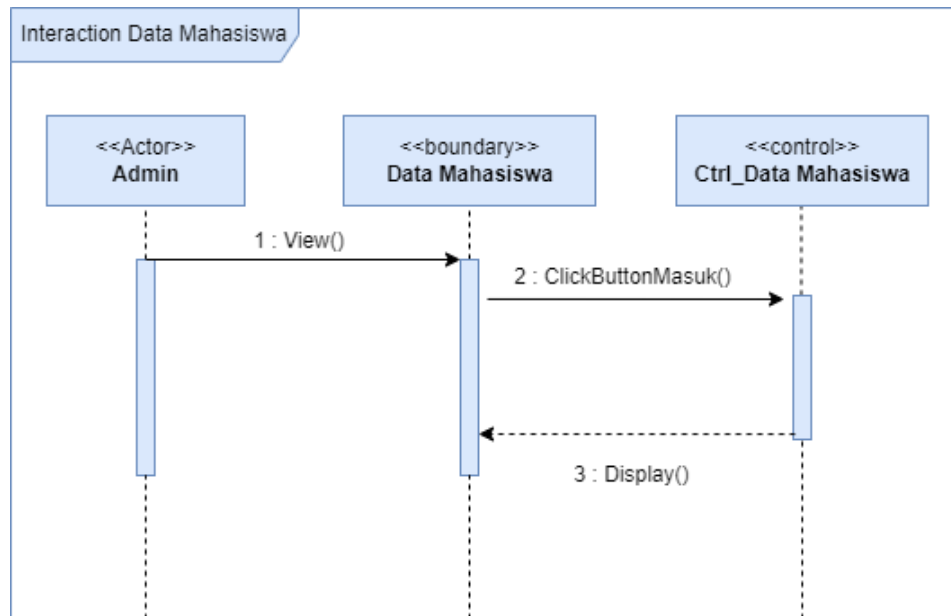
Pada diagram hasil laporan absensi *admin* menjelaskan alur interaksi saat *admin* mengelola data laporan absensi yaitu mereset dan menyimpan data absensi kehadiran.



**Gambar 3.38** Sequence Diagram Laporan Absensi Admin

## 9. Sequence Diagram Data Mahasiswa

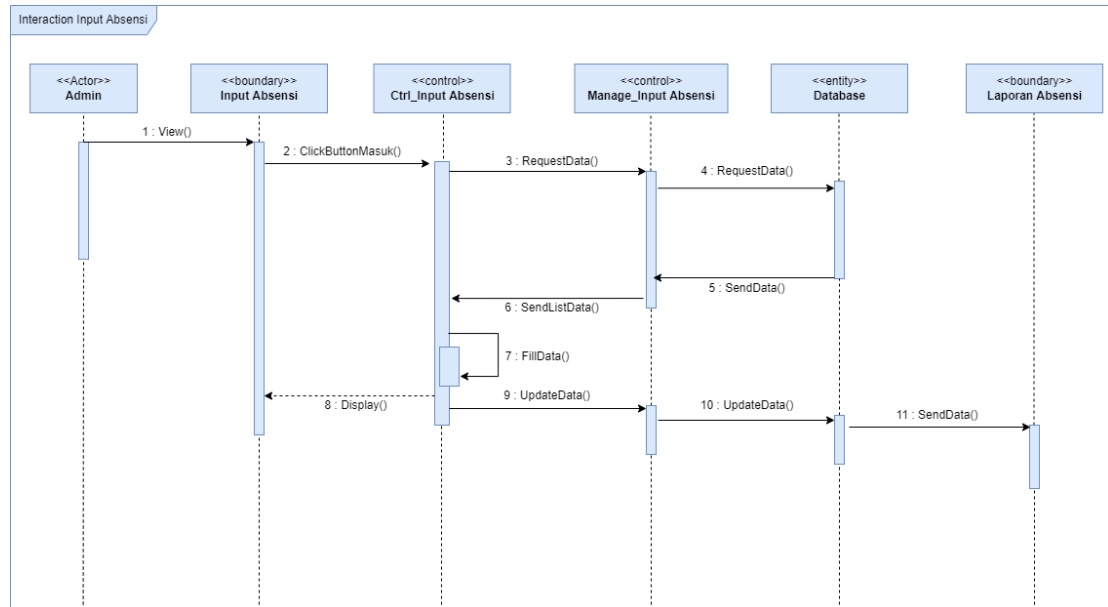
Pada diagram hasil data mahasiswa *admin* menjelaskan alur interaksi saat *admin* melihat mahasiswa yang terdaftar di aplikasi S.M.A.R.T Sense.



**Gambar 3.39** *Sequence Diagram* Data Mahasiswa

#### 10. *Sequence Diagram Input Absensi*

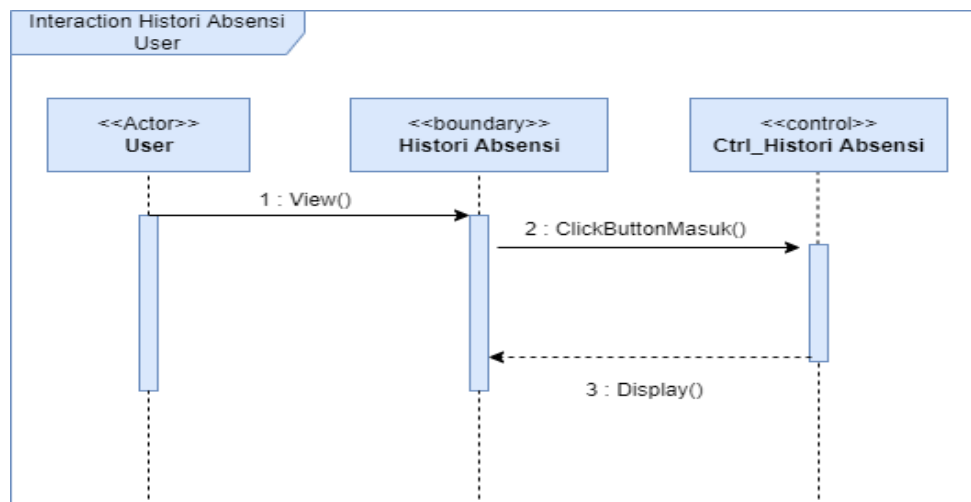
Pada diagram hasil *input* absensi *admin* menjelaskan alur interaksi saat *admin* mengelola data absensi yaitu mengisi form absensi kehadiran, apabila form sudah terisi data tersebut akan tersimpan di menu laporan absensi.



**Gambar 3.40** *Sequence Diagram Input Absensi*

### 11. *Sequence Diagram Histori Absensi User*

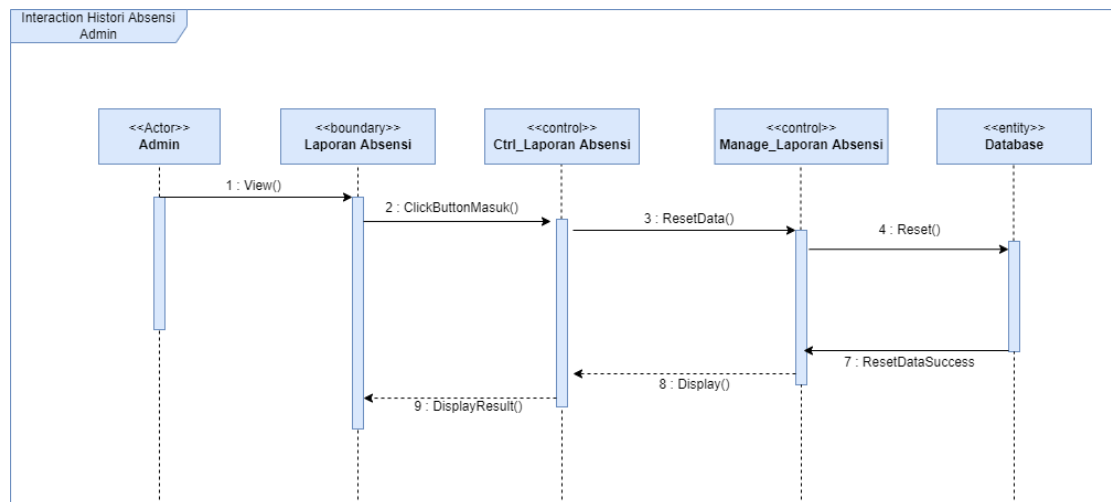
Pada diagram laporan absensi menjelaskan alur interaksi saat *user* ingin melihat histori absensi di *website* S.M.A.R.T Sense.



**Gambar 3.41** *Sequence Diagram Histori Absensi User*

## 12. Sequence Diagram Histori Absensi Admin

Pada diagram hasil laporan absensi *admin* menjelaskan alur interaksi saat *admin* mengelola data histori absensi yaitu mereset data absensi kehadiran apabila sudah tidak diperlukan.

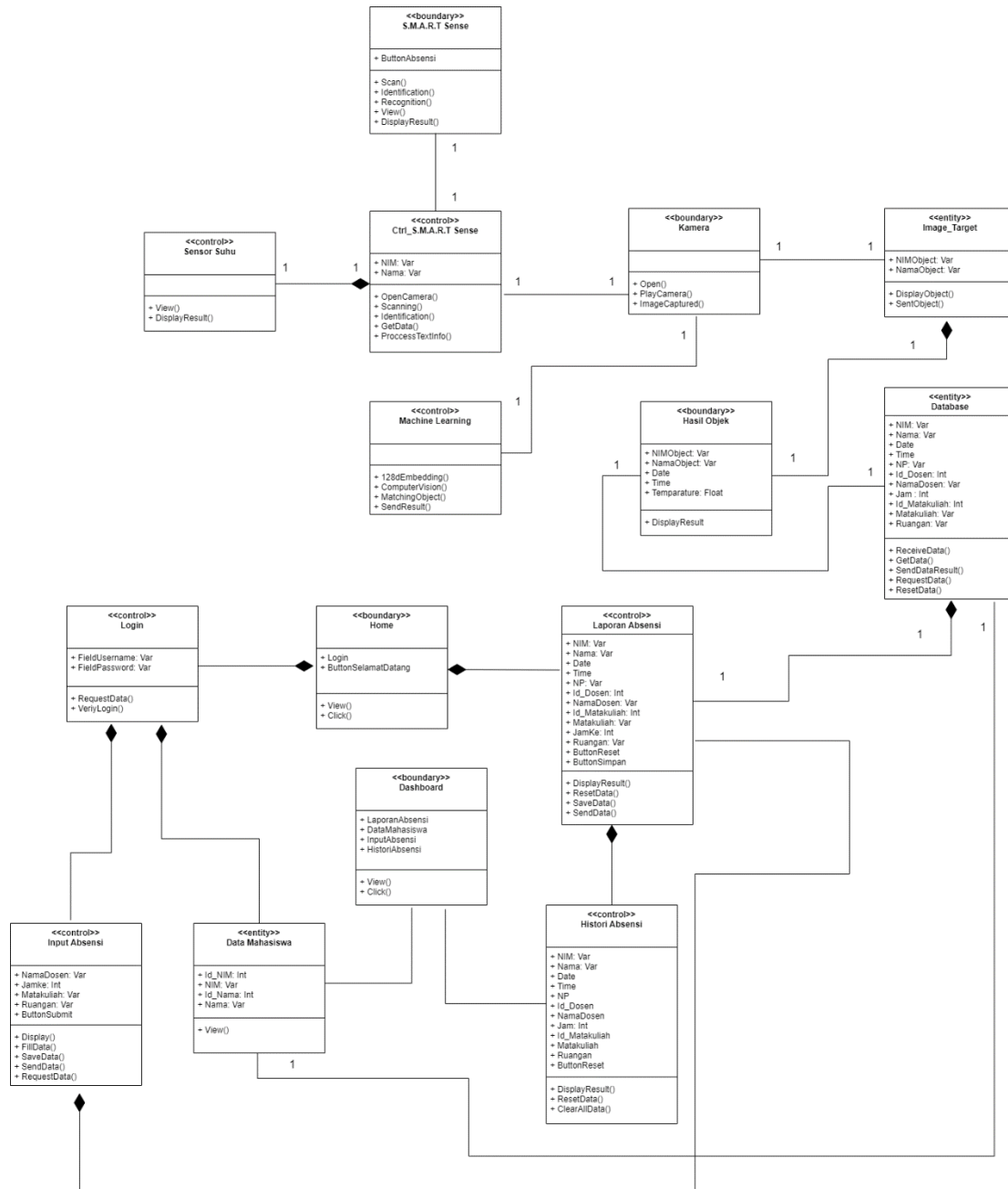


**Gambar 3.42** Sequence Diagram Histori Absensi Admin

### 3.3.4 Class Diagram

*Class diagram* menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode, dan hubungan dari setiap objek. Berikut ini gambar 3.43 yang merupakan class diagram dari aplikasi S.M.A.R.T Sense.

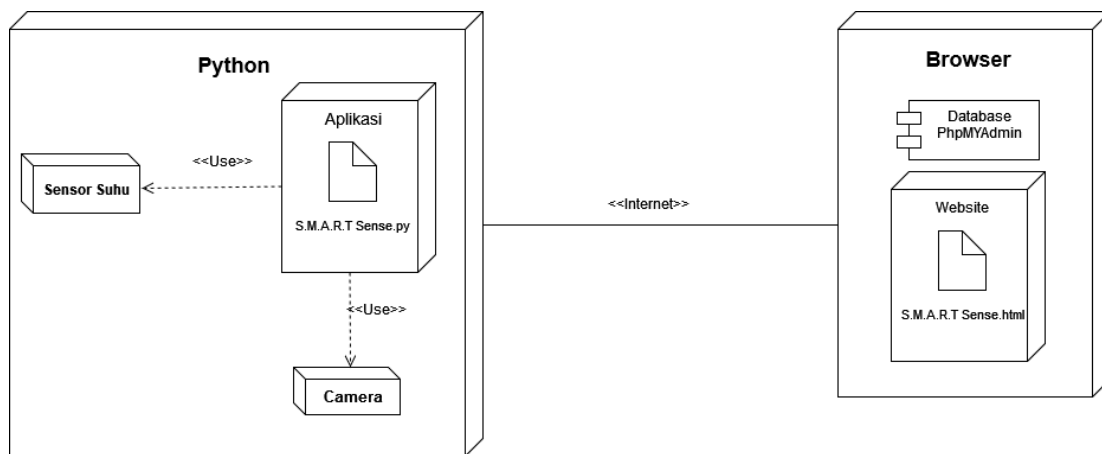




Gambar 3.43 Class Diagram S.M.A.R.T Sense

### 3.3.5 Deployment Diagram

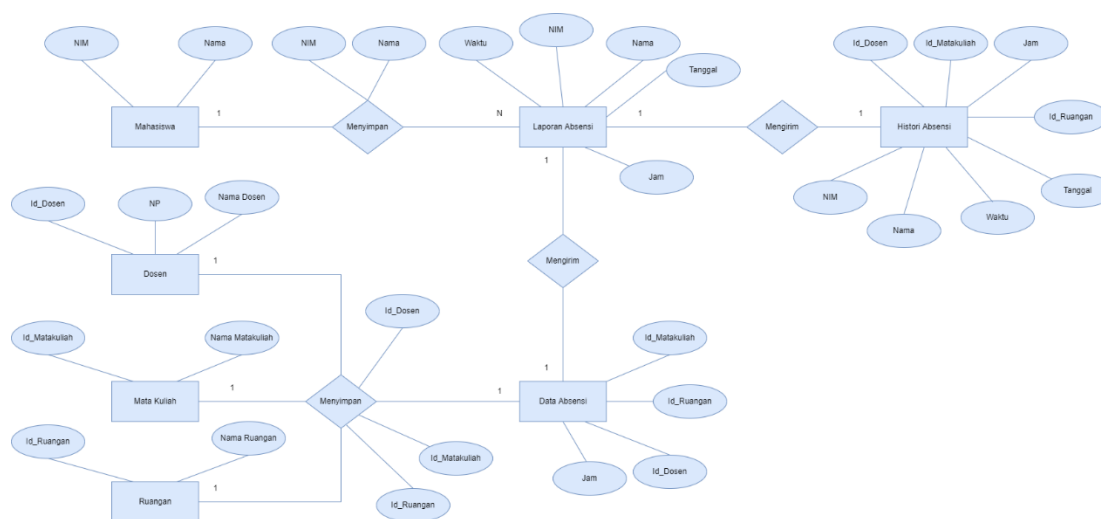
*Deployment diagram* S.M.A.R.T Sense menjelaskan hubungan antar user, aplikasi dan website. Model *deployment diagram* aplikasi S.M.A.R.T Sense dapat dilihat pada Gambar 3.44.



**Gambar 3.44** *Deployment Diagram* Aplikasi S.M.A.R.T Sense

### 3.3.6 Entity Relationship Diagram (ERD)

*Entity Relationship Diagram (ERD)* adalah diagram yang digunakan untuk perancangan suatu database dan menunjukkan relasi antar objek atau entitas beserta atribut-atributnya secara detail. Berikut gambar 3.45 adalah rancangan diagram ERD pada aplikasi S.M.A.R.T Sense.



**Gambar 3.45** Entity Relationship Diagram Aplikasi S.M.A.R.T Sense

Terdapat beberapa tabel yang ada pada aplikasi S.M.A.R.T Sense diantaranya tabel Mahasiswa, Dosen, Matakuliah, Ruangan, Data absensi, Laporan absensi dan Histori absensi. Pada Tabel 3.18 menunjukkan penjelasan dari isi tabel Mahasiswa.

**Tabel 3.18** Tabel Mahasiswa

Nama	Tipe Data	Ukuran	Key
NIM	Var	20	<i>Primary-Key</i>
Nama	Var	30	

Tabel Dosen berisikan data dan informasi mengenai data dosen yang akan ditampilkan.

**Tabel 3.19** Tabel Dosen

Nama	Tipe Data	Ukuran	Key
Id_Dosen	Int	20	<i>Primary-Key</i>

Nama Dosen	Var	30	
NP	Var	20	

Tabel Matakuliah berisikan data dan informasi mengenai data matakuliah yang akan ditampilkan.

**Tabel 3.20** Tabel Matakuliah

<b>Nama</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Key</b>
Id_Matakuliah	Int	20	<i>Primary-Key</i>
Nama Matakuliah	Var	30	

Tabel Ruangan berisikan data dan informasi mengenai data ruangan yang akan ditampilkan.

**Tabel 3.21** Tabel Ruangan

<b>Nama</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Key</b>
Id_Ruangan	Int	20	<i>Primary-Key</i>
Nama Ruangan	Var	2	

Tabel Data absensi berisikan data dan informasi mengenai data dosen, matakuliah, jam keberapa dan ruangan yang akan ditampilkan.

**Tabel 3.22** Tabel Data Absensi

<b>Nama</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Key</b>
Id_Dosen	Int	20	<i>Primary-Key</i>
Id_Matakuliah	Int	20	

Id_Ruangan	Int	20	
Jam	Int	2	

Tabel Laporan Absensi berisikan data dan informasi mengenai data mahasiswa, waktu, dan tanggal yang akan ditampilkan.

**Tabel 3.23** Tabel Laporan Absensi

<b>Nama</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Key</b>
NIM	Var	20	
Nama	Var	30	
Tanggal	Date	-	
Waktu	Time	-	

Tabel Histori Absensi berisikan data dan informasi mengenai data laporan absensi yang akan ditampilkan.

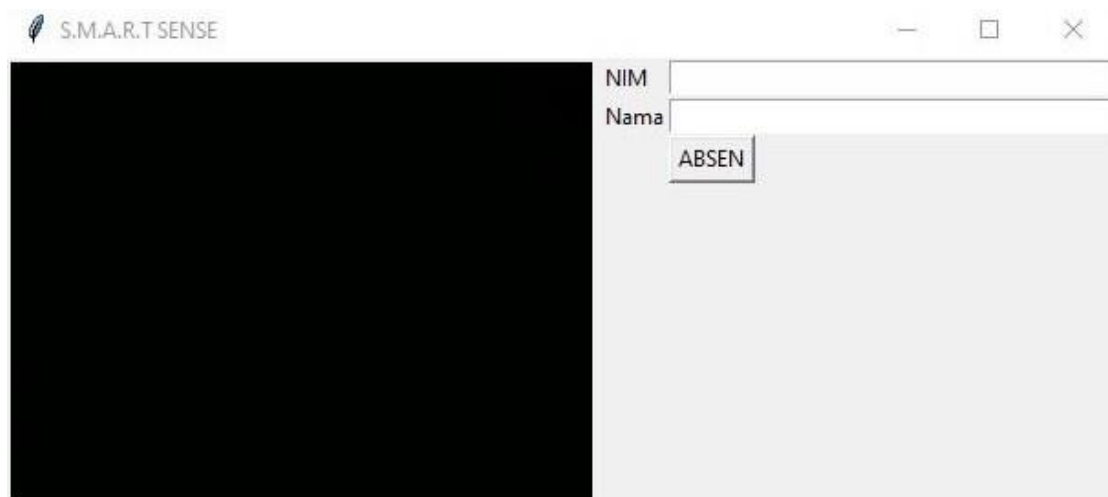
**Tabel 3.24** Tabel Histori Absensi

<b>Nama</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Key</b>
NIM	Var	20	
Nama	Var	30	
Tanggal	Date	-	
Waktu	Time	-	
Id_Dosen	Int	20	
Jam	Int	2	
Id_Matakuliah	Int	20	
Id_Ruangan	Int	20	

### 3.3.7 Design Interface

Setelah melakukan analisis diagram pada maka pada tahap selanjutnya adalah desain *interface* yang akan dibangun, *interface* yang akan dibangun terdiri dari aplikasi S.M.A.R.T Sense dan *website* Laporan Absensi S.M.A.R.T Sense, untuk *website* Laporan Absensi S.M.A.R.T Sense terdiri dari: Home, Menu Login, Dashboard, Menu Laporan Absensi, Menu Input Absensi dan Menu Histori Absensi.

#### 1. Perancangan Aplikasi S.M.A.R.T Sense



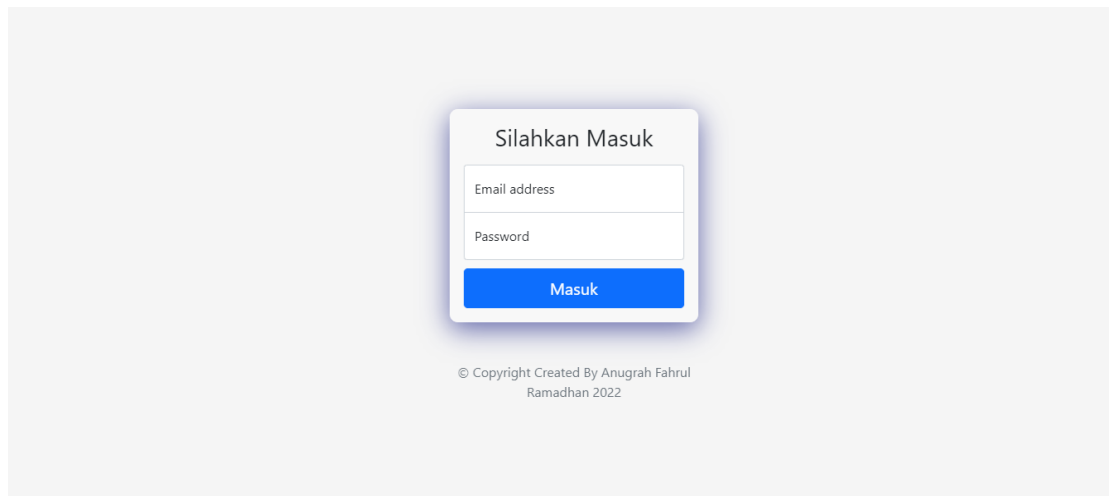
**Gambar 3.46** Perancangan *Interface* Aplikasi S.M.A.R.T Sense

## 2. Perancangan website S.M.A.R.T Sense (Home)



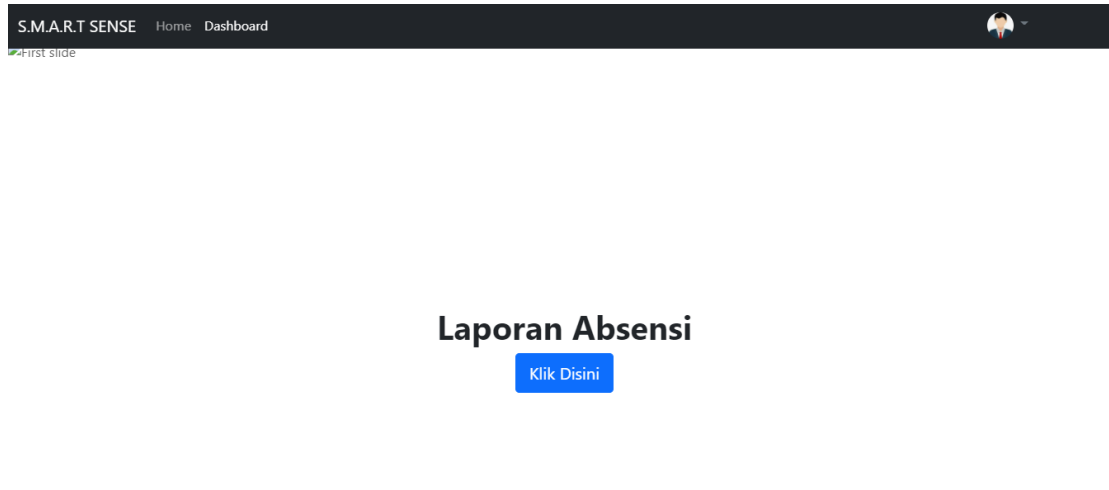
**Gambar 3.47** Perancangan *Home*

## 3. Perancangan website S.M.A.R.T Sense (*Login*)



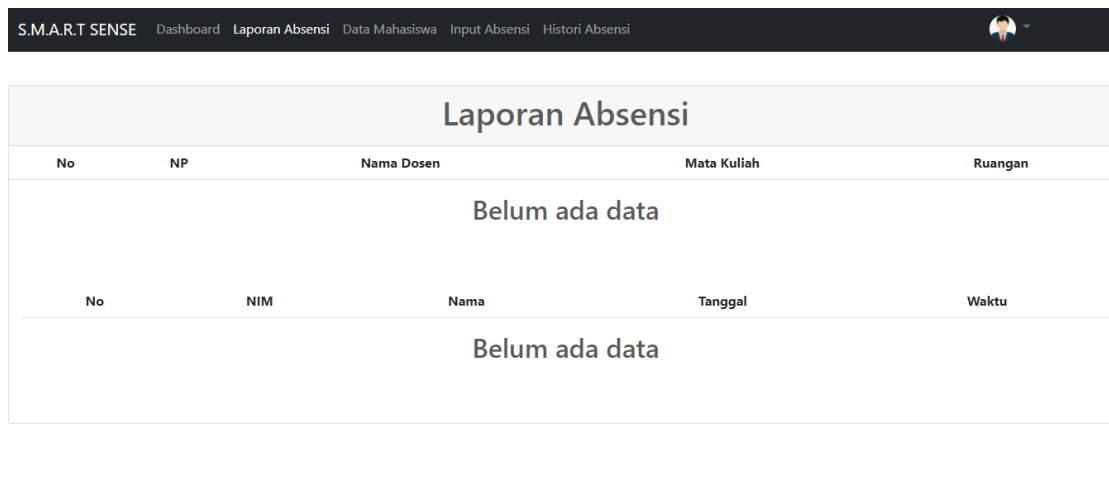
**Gambar 3.48** Perancangan *Login*

#### 4. Perancangan website S.M.A.R.T Sense (*Dashboard*)



**Gambar 3.49** Perancangan *Dashboard*

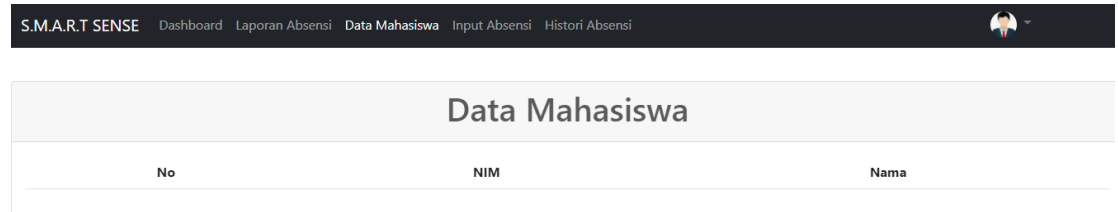
#### 5. Perancangan website S.M.A.R.T Sense (Menu Laporan Absensi)



**Gambar 3.50** Perancangan Menu Laporan Absensi



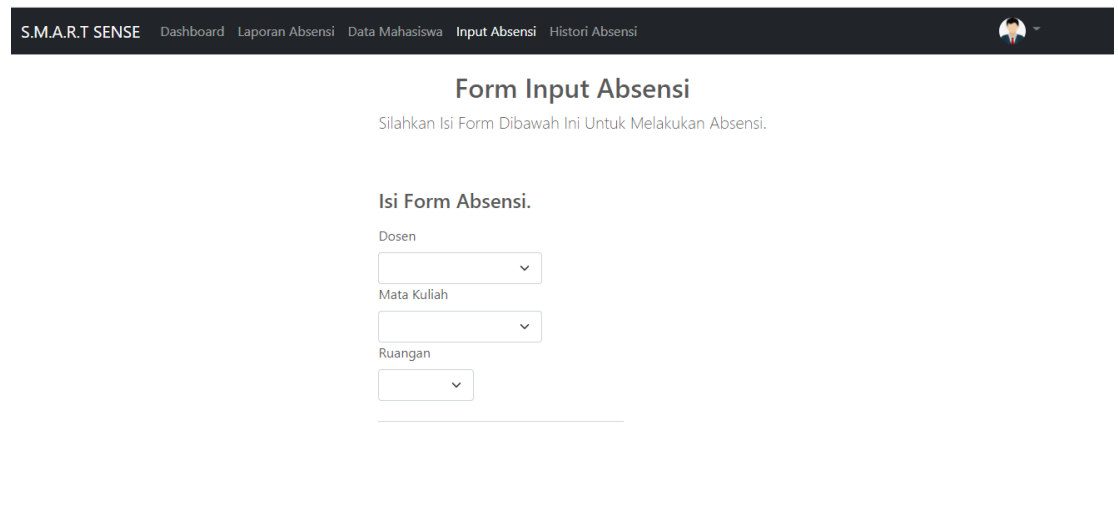
## 6. Perancangan website S.M.A.R.T Sense (Menu Data Mahasiswa)



Data Mahasiswa		
No	NIM	Nama

**Gambar 3.51** Perancangan Menu Data Mahasiswa

## 7. Perancangan website S.M.A.R.T Sense (Menu Input Absensi)



S.M.A.R.T SENSE Dashboard Laporan Absensi Data Mahasiswa **Input Absensi** Histori Absensi

### Form Input Absensi

Silahkan Isi Form Dibawah Ini Untuk Melakukan Absensi.

**Isi Form Absensi.**

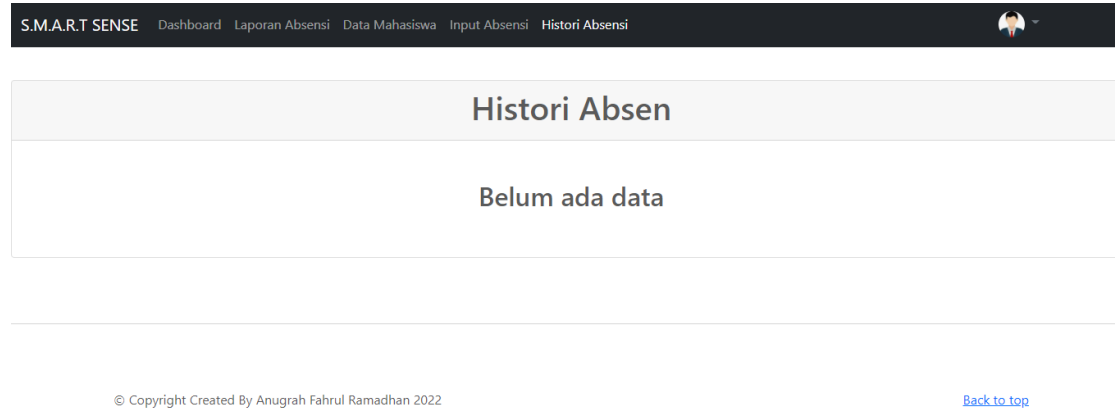
Dosen

Mata Kuliah

Ruangan

**Gambar 3.52** Perancangan Menu Input Absensi

## 8. Perancangan website S.M.A.R.T Sense (Menu Histori Absensi)



**Gambar 3.53** Perancangan Menu Histori Absensi

## **BAB IV**

### **IMPLEMENTASI DAN UJI COBA**

#### **4.1 *Integration & Testing* (Penerapan / Pengujian Program)**

Dalam tahapan ini penelitian berfokus pada penerapan, pengujian program menggunakan bahasa *Python* dan *Laravel*. Setelah itu dilakukan pengujian sampel wajah yang sudah diterapkan dalam kode pemrograman sebelumnya untuk *recognition*/pengenalan terhadap objek yaitu wajah manusia dan pengujian fungsi menggunakan metode *blackbox testing*. *Blackbox testing* ini untuk menguji spesifikasi suatu fungsi atau modul, apakah berjalan sesuai yang diharapkan atau tidak.

##### **4.1.1 Implementasi**

Dalam tahapan ini metode *128D Embedding* akan diimplementasikan ke dalam aplikasi menggunakan bahasa pemrograman *Python* dan untuk hasil dari *face recognition* serta laporan absensi diimplementasikan ke *website* menggunakan bahasa pemrograman *Laravel* sesuai dengan perancangan yang dilakukan. Berikut tampilan sistem aplikasi S.M.A.R.T Sense:

##### **4.1.1.1 Tampilan Aplikasi**

###### **1. Wajah Dikenal**

Pada gambar 4.1 dan gambar 4.2 merupakan tampilan ketika sistem mengenali objek (wajah).



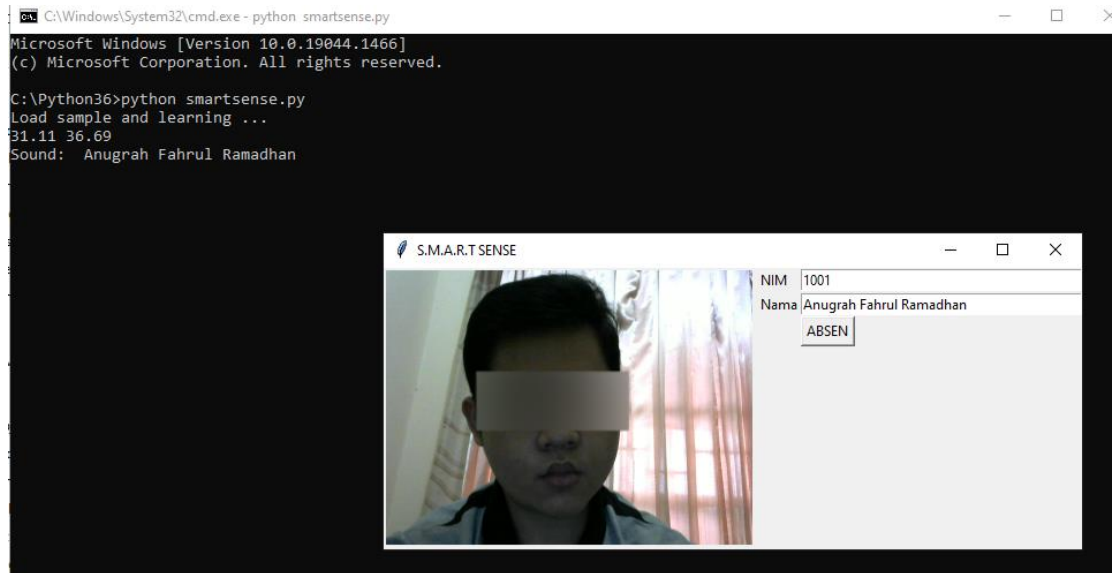
**Gambar 4.1** Tampilan Wajah Dikenal



**Gambar 4.2** Tampilan Wajah Dikenal

## 2. Tampilan Wajah Dikenal dan Sensor Suhu

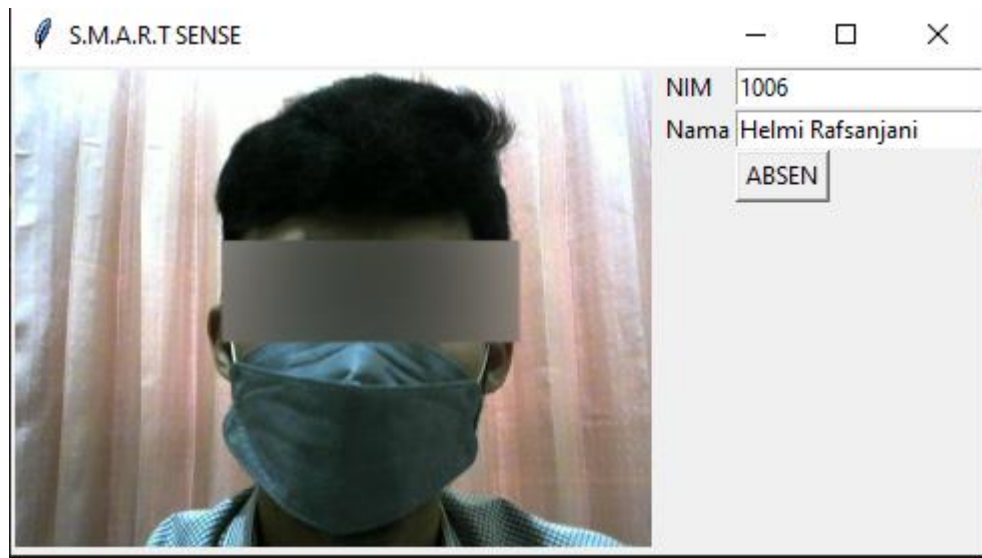
Pada gambar 4.3 merupakan tampilan saat sistem mengenali wajah dan menampilkan suhu lingkungan sekitar dan suhu objek.



**Gambar 4.3** Tampilan Wajah Dikenal dan Sensor Suhu

### 3. Tampilan Wajah Memakai Masker

Pada gambar 4.4 merupakan tampilan saat sistem mengenali objek (wajah) ketika *user* sedang memakai masker.



**Gambar 4.4** Tampilan Wajah Dikenal Saat Memakai Masker

#### 4. Tampilan Wajah Tidak Dikenal

Pada gambar 4.5 merupakan tampilan saat sistem tidak mengenali objek (wajah).

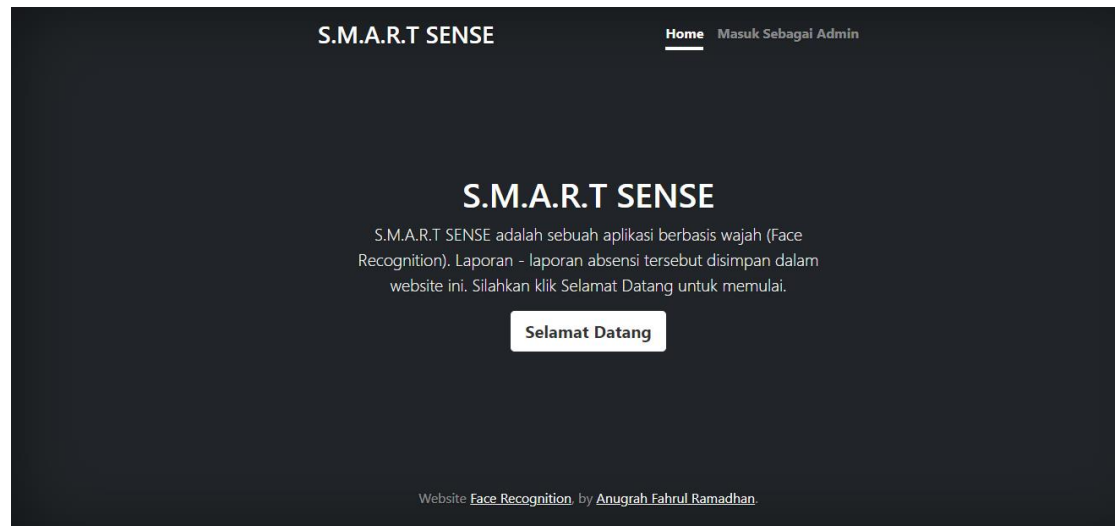


**Gambar 4.5** Tampilan Wajah Tidak Dikenal

##### 4.1.1.2 Tampilan Website

###### 1. Tampilan *Home*

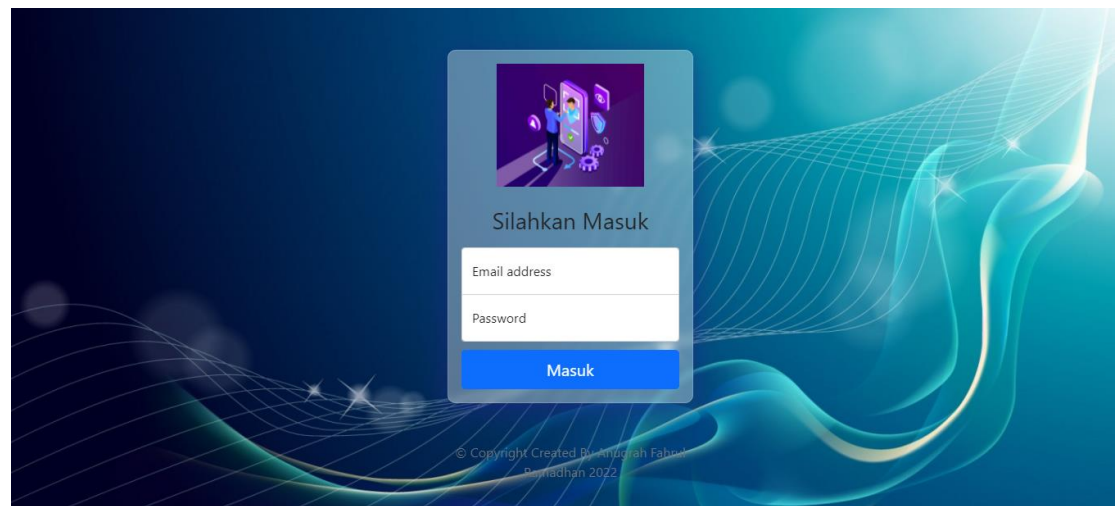
Pada gambar 4.6 merupakan tampilan awal saat *user* memasuki *website* S.M.A.R.T Sense. Dalam tampilan tersebut terdapat pengertian tentang aplikasi S.M.A.R.T Sense, terdapat tombol untuk memasuki *dashboard* dan *login* sebagai *admin*.



**Gambar 4.6** Tampilan *Home*

## 2. Tampilan *Login*

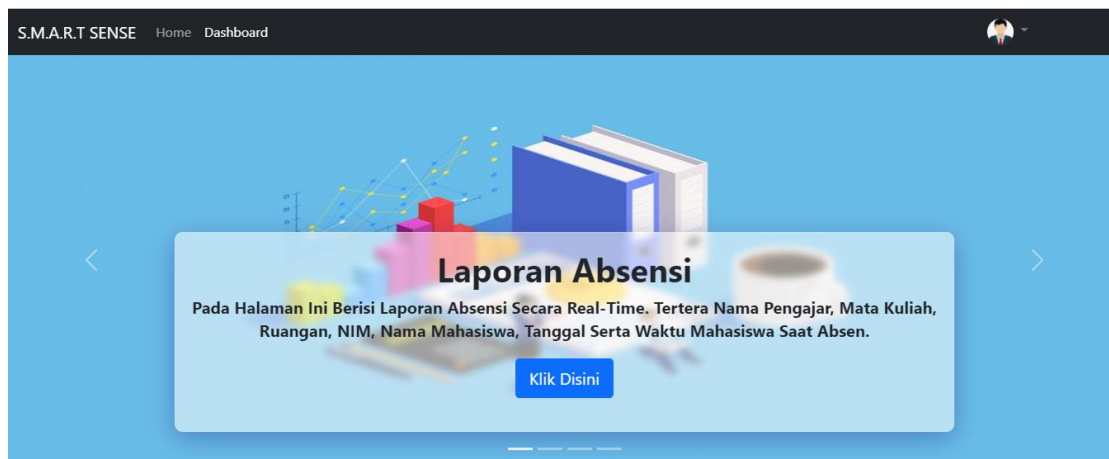
Pada gambar 4.7 merupakan halaman untuk masuk sebagai *admin*.



**Gambar 4.7** Tampilan *Login*

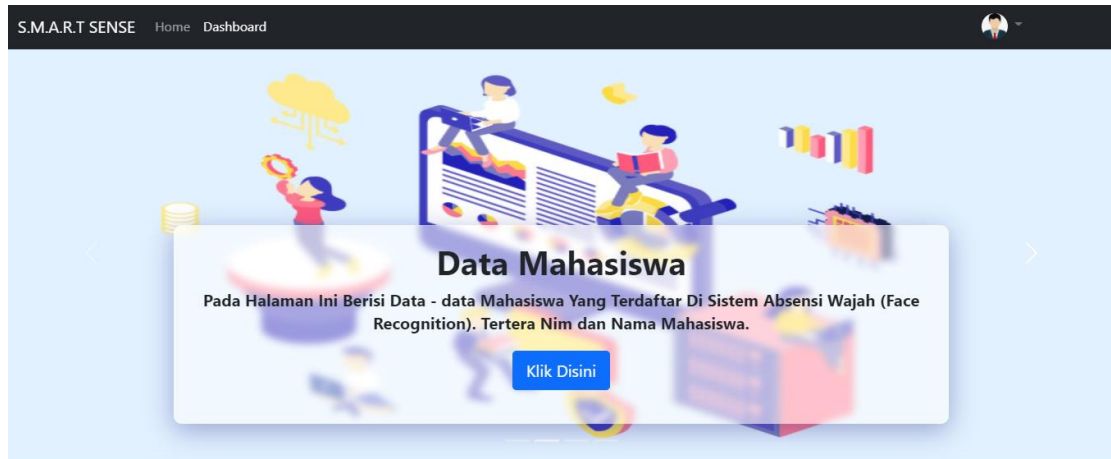
### 3. Tampilan Dashboard

Pada gambar 4.8, gambar 4.9, gambar, gambar 4.10 dan gambar 4.11 merupakan menu – menu terdapat di *dashboard* yang bisa diakses serta terdapat penjelasan pada menu – menu tersebut seperti: Laporan Absensi, Data Mahasiswa, Input Data, dan Histori Absensi.

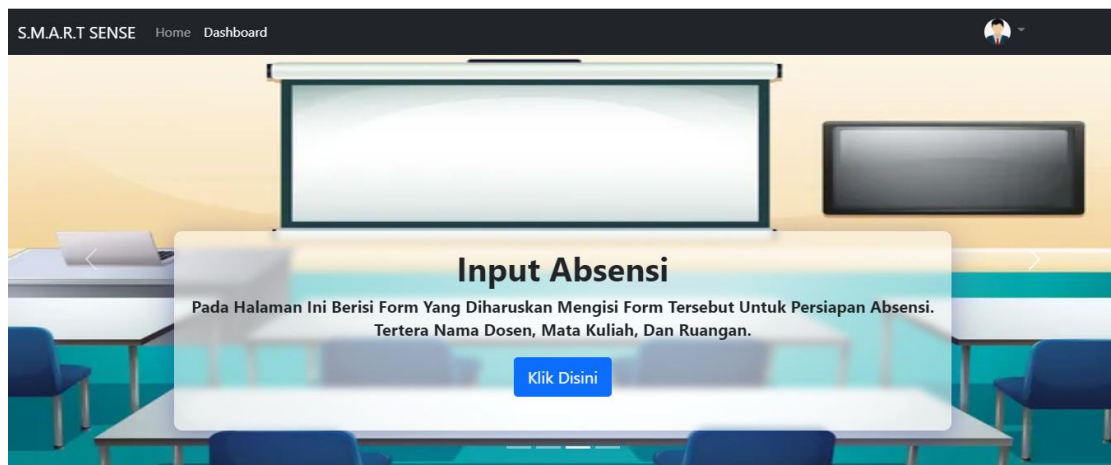


**Gambar 4.8** Tampilan *Dashboard* Laporan Absensi

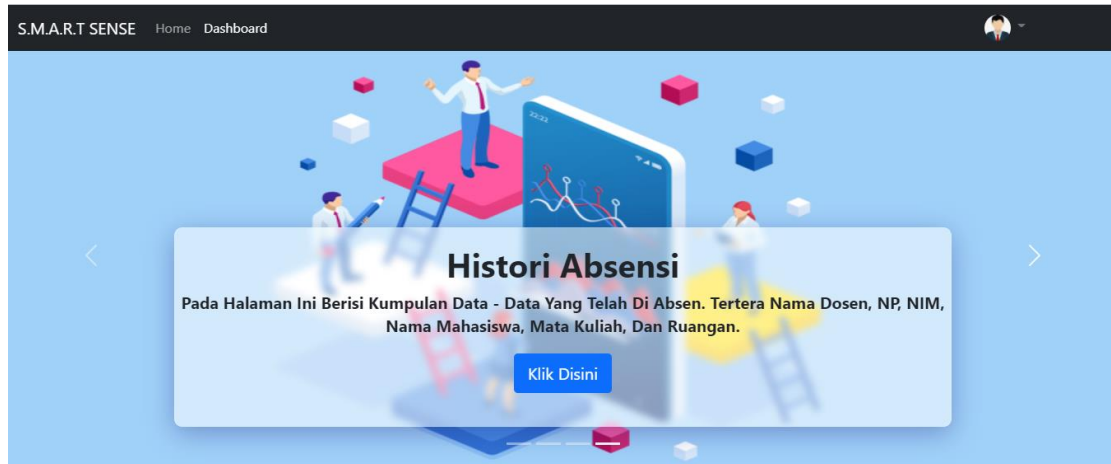




**Gambar 4.9** Tampilan *Dashboard* Data Mahasiswa



**Gambar 4.10** Tampilan *Dashboard* Input Absensi



**Gambar 4.11** Tampilan *Dashboard* Histori Absensi

#### **4. Tampilan Menu Laporan Absensi**

Pada gambar 4.12 merupakan halaman yang menampilkan informasi mahasiswa yang telah melakukan absensi kehadiran melalui aplikasi S.M.A.R.T Sense dan hasil pengisian form dari menu Input Absensi. Data – data tersebut secara otomatis langsung tersimpan di halaman ini.

S.M.A.R.T SENSE Dashboard Laporan Absensi Data Mahasiswa Input Absensi Histori Absensi

### Laporan Absensi

No	NP	Nama Dosen	Mata Kuliah	Jam Ke	Ruangan
1	2222	Agus	Fisika	3	C

No	NIM	Nama	Tanggal	Waktu
1	1001	Anugrah Fahrul Ramadhan	23/01/2022	10:25:46.0
2	1003	Naufal Dzaki	23/01/2022	10:26:14.0

Reset
Simpan

**Gambar 4. 12** Tampilan Menu Laporan Absensi

## 5. Tampilan Menu Data Mahasiswa

Pada gambar 4.13 merupakan tampilan informasi mengenai daftar mahasiswa yang sudah terdapat di aplikasi S.M.A.R.T Sense, seperti NIM dan nama mahasiswa.

No	NIM	Nama
1	1001	Anugrah Fahrul Ramadhan
2	1002	Obama
3	1003	Naufal Dzaki
4	1004	Yoga Pramana
5	1005	Moch. Dimmas Saputra
6	1006	Helmi Rafsanjani

**Gambar 4.13** Tampilan Menu Data Mahasiswa

## 6. Tampilan Menu Input Absensi

Pada gambar 4.14 merupakan tampilan dimana *admin* mengisi form untuk memulai absensi kehadiran, seperti nama dosen, matakuliah dan ruangan. Data tersebut akan tersimpan di menu Laporan Absensi.

**Form Input Absensi**  
Silahkan Isi Form Dibawah Ini Untuk Melakukan Absensi.

Isi Form Absensi.

Dosen

Jam Ke -

Mata Kuliah

Ruangan

**Gambar 4.14** Tampilan Menu *Input Absensi*

## 7. Tampilan Menu Histori

Pada gambar 4.15 merupakan tampilan histori absensi kehadiran. Data dari menu Laporan Absensi akan tersimpan ke halaman ini apabila kegiatan mengabsen telah selesai. Pada halaman ini terdapat informasi mengenai data – data absensi baik itu di waktu sebelumnya maupun waktu sekarang.

NIM	Nama	Tanggal	Waktu
1001	Anugrah Fahrul Ramadhan	2022-01-16	20:20:17
1005	Moch. Dimmas Saputra	2022-01-16	20:20:36

**Gambar 4.15** Tampilan Menu Histori

## 4.2 Testing

Untuk melakukan pemeriksaan terhadap sistem yang sudah dibangun, maka dilakukan pengujian sistem menggunakan metode *black box*. Tabel 4.1 menunjukkan rencana pengujian *black box*.

**Tabel 4.1** Rencana Pengujian

No	Kelas Uji	Butir Uji	Jenis Pengujian
1	Pengujian <i>scan object</i>	Validasi Layar Kamera dan <i>Object</i>	<i>Blackbox</i>
2	Pengujian <i>face recognition</i>	Validasi Data <i>Object</i>	<i>Blackbox</i>
3	Pengujian suhu	Validasi <i>Object</i>	<i>Blackbox</i>
4	Pengujian tombol dan <i>link</i> yang terdapat di menu <i>home</i>	Validasi Tombol & Data <i>Object</i>	<i>Blackbox</i>
5	Pengujian <i>form login</i>	Validasi Data <i>Object</i>	<i>Blackbox</i>
6	Pengujian tombol menu – menu yang terdapat di <i>dashboard</i>	Validasi Tombol & Data <i>Object</i>	<i>Blackbox</i>
7	Pengujian menu laporan absensi	Validasi Tombol & Data <i>Object</i>	<i>Blackbox</i>
8	Pengujian menu <i>input</i> absensi	Validasi Tombol & Data <i>Object</i>	<i>Blackbox</i>
9	Pengujian menu histori absensi	Validasi Tombol & Data <i>Object</i>	<i>Blackbox</i>

Setelah dilakukan pengujian dengan menggunakan metode *black box* maka didapatkan hasil pengujian pada Tabel 4.2.

**Tabel 4.2** Tabel Hasil Pengujian *Blackbox*

No	Kelas Uji	Data Masukan	Hasil Yang Diharapkan	Respon sistem	Kesimpulan
1	Pengujian <i>scan object</i>	Objek wajah menghadap kamera	Objek wajah muncul di layar kamera depan	Aplikasi menampilkan objek wajah di layar kamera	Valid
2	Pengujian <i>face recognition</i>	Objek wajah menghadap kamera	Saat mengklik tombol maka sistem mengenali objek wajah dan menghasilkan <i>output</i> NIM dan nama	Aplikasi menampilkan NIM dan nama objek serta memberikan notifikasi suara dari hasil <i>face recognition</i>	Valid
3		Objek wajah memakai masker saat menghadap kamera	Saat mengklik tombol maka sistem mengenali objek wajah dan menghasilkan <i>output</i> NIM dan nama	Aplikasi menampilkan NIM dan nama objek serta memberikan notifikasi suara dari hasil <i>face recognition</i>	Valid
4		Objek wajah tidak terdaftar di aplikasi S.M.A.R.T Sense menghadap kamera	Saat mengklik tombol maka sistem tidak mengenali objek wajah	Sistem aplikasi tidak mengenali objek wajah dihadapan kamera dan memberikan notifikasi “Wajah tidak dikenal atau tidak terdeteksi” serta memberikan notifikasi suara bahwa sistem tidak mengenali objek tersebut	Valid
5		Tidak ada objek wajah dihadapan kamera	Saat mengklik tombol maka sistem tidak bisa melakukan absensi kehadiran	Sistem aplikasi tidak bisa melakukan absensi kehadiran dan memberikan notifikasi suara	Valid

6	Penguujian suhu	Suhu objek didekatkan ke sensor suhu	Saat mengklik tombol maka sistem akan mendeteksi suhu objek setelah itu sistem menyimpan data <i>user</i> ke Laporan Absensi	Sistem aplikasi mendeteksi suhu objek dan suhu lingkungan sekitar lalu sistem menyimpan data <i>user</i> ke Laporan Absensi	Valid
7		Objek tidak mendekati sensor suhu	Data <i>User</i> tidak akan masuk ke Laporan Absensi saat melakukan absensi kehadiran	Sistem aplikasi mendeteksi suhu objek sama dengan suhu lingkungan dan sistem tidak menyimpan data <i>user</i> ke Laporan Absensi	Valid
8	Penguujian tombol dan <i>link</i> yang terdapat di menu <i>home</i>	Mengklik tombol “Selamat Datang”	Muncul tampilan <i>dashboard</i> setelah mengklik tombol “Selamat Datang”	Muncul tampilan <i>dashboard</i> setelah tombol di klik	Valid
9		Mengklik <i>Link</i> “Masuk Sebagai Admin”	Muncul tampilan <i>Login</i> untuk masuk sebagai <i>admin</i> setelah mengklik <i>Link</i> “Masuk Sebagai Admin”	Muncul tampilan <i>Login</i> setelah tombol di klik	Valid
10	Penguujian <i>form login</i>	Memasukan <i>username</i> dan <i>password</i> dengan benar	Data <i>email</i> dan <i>password</i> benar dan akan masuk ke halaman <i>dashboard</i>	Data <i>email</i> dan <i>password</i> benar sistem menampilkan halaman <i>dashboard</i>	Valid
11		Memasukan <i>username</i> dan <i>password</i> dengan salah	Tidak bisa masuk ke <i>dashboard</i> dan akan menampilkan notif “ <i>Email</i> atau <i>Password</i> Salah! Silahkan Coba Lagi”	Tidak bisa masuk ke <i>dashboard</i> dan akan menampilkan notif “ <i>Email</i> atau <i>Password</i> Salah! Silahkan Coba Lagi”	Valid
12	Penguujian tombol menu – menu yang terdapat di <i>dashboard</i>	Mengklik tombol “Klik Disini” pada tampilan Laporan Absensi di menu <i>dashboard</i>	Masuk ke menu Laporan Absensi setelah mengklik tombol “Klik Disini”	Masuk ke menu Laporan Absensi setelah tombol di klik	Valid
13		Mengklik tombol “Klik Disini” pada tampilan Data Mahasiswa di menu <i>dashboard</i>	Masuk ke menu Data Mahasiswa setelah mengklik tombol “Klik Disini”	Masuk ke menu Data Mahasiswa setelah tombol di klik	Valid



14		Mengklik tombol “Klik Disini” pada tampilan <i>Input</i> Absensi di menu <i>dashboard</i>	Masuk ke menu <i>Input</i> Absensi setelah mengklik tombol “Klik Disini”	Masuk ke menu <i>Input</i> Absensi setelah tombol di klik	Valid
15		Mengklik tombol “Klik Disini” pada tampilan Histori Absensi di menu <i>dashboard</i>	Masuk ke menu Histori Absensi setelah mengklik tombol “Klik Disini”	Masuk ke menu Histori Absensi setelah tombol di klik	Valid
16	Pengujian menu laporan absensi	Mengklik tombol “Reset”	Menghapus data absensi	Sistem menghapus data absensi	Valid
17		Mengklik tombol “Simpan”	Data akan disimpan ke menu Histori Absensi	Sistem menyimpan data absensi ke menu Histori Absensi	Valid
18	Pengujian menu <i>input</i> absensi	Mengklik tombol dropdown dosen	Muncul daftar nama dosen	Sistem memunculkan daftar nama – nama dosen yang tersimpan dalam <i>database</i>	Valid
19		Mengklik tombol dropdown Jam Ke	Muncul daftar Jam beberapa saja untuk urutan dimulainya kuliah	Sistem memunculkan daftar angka untuk urutan dimulainya kuliah	Valid
20		Mengklik tombol dropdown matakuliah	Muncul daftar nama matakuliah	Sistem memunculkan daftar matakuliah yang tersimpan dalam <i>database</i>	Valid
21		Mengklik tombol dropdown ruangan	Muncul daftar nama ruangan	Sistem memunculkan daftar ruangan yang tersimpan dalam <i>database</i>	Valid
22	Pengujian menu histori absensi	Mengklik tombol “Reset	Semua data yang tersimpan di menu Histori Absensi akan terhapus	Sistem akan menghapus semua data absensi yang tersimpan dalam menu Histori Absensi	Valid

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan hasil pembahasan beserta penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan, diantaranya:

1. Absensi kehadiran berbasis *face recognition*/berbasis wajah dapat melakukan proses absensi tanpa diperlukan tanda tangan di kertas dan *fingerprint*. Dengan menghadapkan wajah didepan kamera maka absensi kehadiran secara otomatis akan terisi. Dengan metode *128D Embedding* membuat proses pengenalan wajah lebih praktis. Sistem ini dapat mengenali wajah dengan mempelajari struktur dari sebuah sampel foto. Sensor suhu yang ditambahkan dapat mengetahui suhu dari objek ketika proses absensi sedang berlangsung.
2. Metode *128D Embedding* mempelajari sampel foto wajah dengan cara Sistem akan mempelajari wajah dari sample foto wajah lalu mengubah nya menjadi angka *vector* sebanyak 128 angka, kemudian sistem akan membandingkan wajah yang ada didepan kamera dengan sampel yang memiliki nilai yang mendekati atau sama dengan wajah objek.

## 5.2 Saran

Dengan adanya kesimpulan diatas, ada beberapa saran yang dapat dikemukakan sebagai bahan pertimbangan lebih lanjut guna meningkatkan produktifitas kerja dari aplikasi S.M.A.R.T Sense ini.

1. Agar sistem bekerja dengan baik dan akurasi untuk mengenali wajah menjadi lebih akurat maka sistem perlu diberikan sampel data foto yang lebih banyak. Semakin banyak sampel data foto maka tingkat akurasi dalam mengidentifikasi wajah semakin meningkat.
2. Aplikasi ini bersifat *prototype*, sehingga diharapkan untuk kedepannya bisa dikembangkan lebih jauh lagi agar aplikasi *face recognition* bisa diterapkan ke dalam sistem absensi sehingga saat melakukan absensi kehadiran tidak lagi secara manual tetapi melalui teknologi *face recognition*.

## DAFTAR PUSTAKA

- A. Shah, K Vora, & J Mehta. (2015). *A review paper on currency recognition system*.  
Int. J. Comput. Appl.
- A.S, Rosa & Shalahuddin, M. *Rekayasa Perangkat Lunak Terstruktur dan Beroientasi Objek*. Bandung: Informatika, 2016.
- Akbar Nur Syahrudin & Tedi Kurniawan. (2018). *Input Dan Output Pada Bahasa Pemrograman Python*. Program Studi Teknik Informatika STMIK Sumedang, Jl. Angkrek Situ No.19, Sumedang.
- Arief M Rudianto. (2011). *Pemrograman Web Dinamis menggunakan PHP dan MySQL*. C.V ANDI OFFSET. Yogyakarta.
- Arsfutura, 2019. *Face Recognition with FaceNet and MTCNN*. [online] available at: <https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/>  
[diakses 23 oktober 2021].
- Bolei Zhou. (2018). *Interpreting Deep Visual Representations via Network Dissection*. Massachusetts Institute Of Technology. 1.
- Danukusumo, Kevin Pudi. (2017). *Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU*. Tugas Akhir. Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Atma Jaya Yogyakarta.

- E.Aishwarya, K.Kumaravel, R.MohamedSuthesh, S. Poornima & R.Poonguzhali. (2021). Smart Attendance System Using Deep Learning Technique. *Turkish Journal of Computer and Mathematics Education Vol.12 No.10*, 1367 – 1373.
- Fajri, Rizky, T. Rahmad Effendi, & Nurul Fadillah. (2020). *Sistem Absensi Berbasis Pengenalan Wajah secara Real Time Menggunakan Metode Fisherface*. InfoTekJar Vol. 4, No. 2.
- Fathansyah. (2012). *Basis Data*. Bandung: Informatika
- Ibnu Daqiqil Id. (2021). *MACHINE LEARNING: Teori, Studi Kasus dan Implementasi Menggunakan Python*. Universitas Riau UR PRESS Jl. Pattimura No. 9, Gobah Pekanbaru 28132, Riau, Indonesia.
- Iswandy, Eka. (2015). *Sistem Penunjang Keputusan Untuk Menentukan Penerimaan Dana Santunan Sosial Anak Nagari dan Penyalurannya Bagi Mahasiswa dan Pelajar Kurang Mampu di Kenagarian*. Barung-Barung Balantai Timur. Jurnal Teknoif. Vol. 3, No. 2.
- Jingtuo Liu, Yafeng Deng, Tao Bai, Zhengping Wei, & Chang Huang. (2015). *Targeting Ultimate Accuracy: Face Recognition via Deep Embedding*. Baidu Research – Institute of Deep Learning.
- Jonathan S. Talahua, Jorge Buele, P. Calvopiña, & José Varela-Aldás. (2021). Facial recognition system for people with and without face mask in times of the covid-19 pandemic. *Sustainability*, 13(12), 6900.

- Lutfi, Farizan. (2017). Penggunaan Framework Laravel Dalam Rancangan Bangun Model Back-end Artikel Website Bisnisbinis.ID. *Jurnal Teknik Informatika dan Komputer*. Vol. 2, No. 1.
- Makinist, S., Betül, A. Y., & Aydin, G. (2020). Average Neural Face Embeddings for Gender Recognition. *Avrupa Bilim ve Teknoloji Dergisi*, 522-527.
- Muhammad Athoillah. (2017). *Pengenalan Wajah Menggunakan SVM Multi Kernel dengan Pembelajaran yang Bertambah*. Universitas PGRI Adi Buana.
- Munawir, Liza Fitria, & Muhammad Hermansyah. (2020). *Implementasi Face Recognition pada Absensi Kehadiran Mahasiswa Menggunakan Metode Haar Cascade Classifier*. Fakultas Teknik Informatika, Universitas Samudra, Jln. Meurandeh Kota Langsa.
- Musfirah Putri Lukman, Junaedy, & Yosua Friendly Yorendy Rieuwpassa. (2018). *Sistem Lampu Otomatis Dengan Sensor Gerak, Sensor Suhu Dan Sensor Suara Berbasis Mikrokontroler*. STMIK KHARISMA Makassar.
- Muttaqin et al. (2020). *Biometrika: Teknologi Identifikasi*, I. Yayasan Kita Menulis.
- Muwardi, F., & Fadlil, A. (2018). *Sistem Pengenalan Bunga Berbasis Pengolahan Citra dan Pengklasifikasi Jarak*. *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, 3(2), 124.

- Ni Wayan Marti & Kadek Yota Ernanda Aryanto. (2016). *Prototipe Sistem Absensi Berbasis Face Recognition dengan Metode Eigenface*. FTK UNDIKSHA.
- Pratama. (2015). *Pengembangan Midi Controller Berbasis Microcontroller dengan Mekanisme sentuh*. Yogyakarta. Universitas Negeri Yogyakarta.
- R. C. Gonzalez & R. E. Woods. (2018). *Digital Image Processing*, 4 ed. New York: Pearson Education, Inc.
- R. M. Thanki & A. M. Kothari. (2018). *Digital image processing using SCILAB*. Cham: Springer.
- Rahmadhika, M. K., & Thantawi, A. M. (2021). *Rancang Bangun Aplikasi Face Recognition Pada Pendekatan CRM Menggunakan Opencv Dan Algoritma Haarcascade*. IKRA-ITH INFORMATIKA: Jurnal Komputer dan Informatika, 5(1), 109-118.
- Rosmanila, Radillah, T., & Sofiyani, A. (2018). *Informatika : Jurnal Informatika, Manajemen Dan Komputer*, 10(1), 32–38.
- Rovita, C.N Aprilia. (2017). *Sistem Manajemen Kepegawaian*. Politeknik Telkom
- Schneiderman, Henry, Kanade, T. (2000) „A statistical method for 3D object detection applied to faces and cars“, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (1), pp. 746–751.

- Sutanto, Y., Utomo, A., & Perbawa, D. S. (2018). Pengujian Aplikasi Website Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : Aplikasi website praktekdokter). *Jurnal SAINSTECH Politeknik Indonusa Surakarta ISSN : 2355 5009 Volume 5 Nomer 2, 1-6.*
- Trisianto, C. (2018). *Penggunaan Metode Waterfall Untuk Pengembangan Sistem Monitoring Dan Evaluasi Pembangunan Pedesaan*, Jurnal Teknologi informasi, 12(1).
- Ulber Silalahi. (2015). *Metode Penelitian Sosial Kuantitatif*. Universitas Parahyangan Bandung.
- Yudho Yudhanto & Helmi Adi Prasetyo. (2018). *Panduan Mudah Belajar Framework Laravel*. PT Elex Media Komputindo, Jakarta.



## LAMPIRAN

### *Lampiran Source Code*

#### *1. Face Recognition (Python)*

```
import tkinter as tk
from PIL import Image, ImageTk
import cv2
import face_recognition
import numpy as np
import pymysql
import gtts
import vlc
import time
import threading
import pymysql
import serial

port = serial.Serial("COM3", 9600, timeout=1)

print("Load sample and learning ...")
arul_image = face_recognition.load_image_file("samplefoto/arul.jpg")
arul_encoding = face_recognition.face_encodings(arul_image)[0]

obama_image = face_recognition.load_image_file("samplefoto/obama.jpg")
obama_encoding = face_recognition.face_encodings(obama_image)[0]
```

```
noval_image = face_recognition.load_image_file("samplefoto/noval.jpeg")
noval_encoding = face_recognition.face_encodings(noval_image)[0]

yoga_image = face_recognition.load_image_file("samplefoto/yoga.jpeg")
yoga_encoding = face_recognition.face_encodings(yoga_image)[0]

dimas_image = face_recognition.load_image_file("samplefoto/dimas.jpeg")
dimas_encoding = face_recognition.face_encodings(dimas_image)[0]

helmi_image = face_recognition.load_image_file("samplefoto/helmi.jpeg")
helmi_encoding = face_recognition.face_encodings(helmi_image)[0]

known_face_encodings = [
    arul_encoding,
    obama_encoding,
    noval_encoding,
    yoga_encoding,
    dimas_encoding,
    helmi_encoding
]

known_face_nims = [
    "1001",
    "1002",
    "1003",
```

```
"1004",  
"1005",  
"1006",  
]
```

```
face_locations = []  
face_encodings = []  
face_nims = []
```

```
conn = pymysql.connect(host="localhost", user="root", passwd="", db="dbdemo")  
cursor = conn.cursor()
```

```
captured = None  
nama = None
```

```
class MainWindow():  
    def __init__(self, window, cap):  
        self.window = window  
        self.cap = cap  
        self.interval = 20  
        self.canvas01 = tk.Canvas(self.window, width=320, height=240)  
        self.canvas01.grid(row=0, column=0)  
        self.update_image()  
  
    def update_image(self):
```

```

global captured
self.frame = self.cap.read()[1]
captured = self.frame.copy()
self.image = cv2.cvtColor(self.frame, cv2.COLOR_BGR2RGB)
self.scale_percent = 50
self.width = int(self.image.shape[1] * self.scale_percent / 100)
self.height = int(self.image.shape[0] * self.scale_percent / 100)
self.dim = (self.width, self.height)
self.image = cv2.resize(self.image, self.dim, interpolation = cv2.INTER_AREA)
self.image = Image.fromarray(self.image)
self.image = ImageTk.PhotoImage(self.image)
self.canvas01.create_image(0, 0, anchor=tk.NW, image=self.image)
self.window.after(self.interval, self.update_image)

def clickMe():
    global captured, conn, cursor, nama, port

    port.write(b'a')
    sensor = port.readline().decode()
    sensor = sensor.strip().split()
    ambient = float(sensor[0])
    objek = float(sensor[1])
    print(ambient, objek)
    if objek < ambient + 2.0:
        print("Unknown")
    return

```

```
small_frame = cv2.resize(captured, (0, 0), fx=0.25, fy=0.25)
rgb_small_frame = small_frame[:, :, ::-1]

face_locations = face_recognition.face_locations(rgb_small_frame)
face_encodings = face_recognition.face_encodings(rgb_small_frame,
face_locations)

id = "Unknown"

for face_encoding in face_encodings:
    matches = face_recognition.compare_faces(known_face_encodings,
face_encoding)
    nim = "Unknown"
    face_distances = face_recognition.face_distance(known_face_encodings,
face_encoding)
    best_match_index = np.argmin(face_distances)

    if matches[best_match_index]:
        nim = known_face_nims[best_match_index]
        id = nim
        break

if id != "Unknown":
    mnim.set(id)
    nama = None
    sql = "SELECT nama FROM mahasiswa WHERE nim = '%s'" % ( nim )
```

```
cursor.execute(sql)
rows = cursor.fetchone()
if len(rows) > 0:
    nama = rows[0]
    ruang = 101
    mnama.set(nama)

    sql = "INSERT INTO absensi ( nim, tanggal, waktu, ruang ) VALUES ( '%s',
CURRENT_DATE(), CURRENT_TIME(), %s )" % ( nim, ruang )
    cursor.execute(sql)
    conn.commit()
else:
    mnama.set("Unknown")
else:
    mnim.set("Wajah Tidak Dikenal atau Tidak Terdeteksi")
    mnama.set("Wajah Tidak Dikenal atau Tidak Terdeteksi")
    nama = "Unknown"

def sounding():
    global nama

    saved = None
    instance = vlc.Instance()

    while True:
        if nama is not None:
            if nama != saved:
```

```
print("Sound: ", nama)
if nama == "Unknown":
    tts = gtts.gTTS(text="Wajah tidak dikenal silahkan coba lagi", lang='id')
else:
    tts = gtts.gTTS(text="Selamat datang, "+nama, lang='id')
tts.save("good.mp3")
player = instance.media_player_new("good.mp3")
player.play()
while player.get_state() != vlc.State.Ended:
    continue
nama = None
else:
    time.sleep(0.05)

th01 = threading.Thread(target=sounding)
th01.daemon = True
th01.start()

if __name__ == "__main__":
    window = tk.Tk()
    window.title("S.M.A.R.T SENSE")
    MainWindow(window, cv2.VideoCapture(0))

mnama = tk.StringVar()
mnim = tk.StringVar()
frame = tk.Frame(window)
```

```

frame.grid(row=0, column=1, sticky="n")
label01 = tk.Label(frame, text="NIM")
label01.grid(row=0,column=1, sticky="w")
entry01 = tk.Entry(frame, textvariable=mnim, width=40)
entry01.grid(row=0,column= 2,sticky = "w")
label02 = tk.Label(frame, text="Nama")
label02.grid(row=1,column=1, sticky="w")
entry02 = tk.Entry(frame, textvariable=nama, width=40)
entry02.grid(row=1,column= 2,sticky="w")

button01 = tk.Button(frame, text="ABSEN", command=clickMe)
button01.grid(row=2,column=2, sticky="w")

window.mainloop()

```

## 2. Laporan Absensi Kehadiran (*Laravel*)

```
<!doctype html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
  <meta name="description" content="">
```

```
  <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">
```



```
<meta name="generator" content="Hugo 0.88.1">
<title>S.M.A.R.T SENSE</title>

<link rel="canonical"
href="https://getbootstrap.com/docs/5.1/examples/carousel/">

<!-- Bootstrap core CSS -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW
3" crossorigin="anonymous">

<style>
  .bd-placeholder-img {
    font-size: 1.125rem;
    text-align: middle;
    -webkit-user-select: none;
    -moz-user-select: none;
    user-select: none;
  }
  @media (min-width: 768px) {
    .bd-placeholder-img-lg {
      font-size: 3.5rem;
    }
  }
</style>
```

```

<!-- Custom styles for this template -->
<link href="/css/carousel.css" rel="stylesheet">
</head>

<body>
  <header>
    <nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">
      <div class="container-fluid">
        <a class="navbar-brand" href="/">S.M.A.R.T SENSE</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarCollapse"
          aria-controls="navbarCollapse" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarCollapse">
          <ul class="navbar-nav me-auto mb-2 mb-md-0">
            <li class="nav-item">
              <a class="nav-link" href="/welcome">Dashboard</a>
            </li>
            <li class="nav-item">
              <a class="nav-link active" aria-current="page"
href="/laporan">Laporan Absensi</a>
            </li>
            <a href="#">@auth

```

```

    <li class="nav-item">
      <a class="nav-link" href="/mahasiswa">Data Mahasiswa</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="/inputabsen">Input Absensi</a>
    </li>
  @endauth

  <li class="nav-item">
    <a class="nav-link" href="/histori">Histori Absensi</a>
  </li>
</ul>
</div>

@auth
<div class="dropdown text-end" style="margin-right: 100px">
  <a href="#" class="d-block link-secondary text-decoration-none
dropdown-toggle" id="dropdownUser1"
  data-bs-toggle="dropdown" aria-expanded="false">
    
  </a>
  <ul class="dropdown-menu text-small" aria-
labelledby="dropdownUser1">

    <li>
      <form action="/logout" method="post">
        @csrf

```

```

        <button type="submit" class="dropdown-item">Keluar</button>
    </form>
</li>
</ul>
</div>
@else
    <div class="text-end">
        <a href="/login" class="nav-link btn btn-primary text-light">Masuk</a>
    </div>
@endauth
</div>
</nav>
</header>

<main>
    <div class="card text-center mt-5">
        <div class="card-header">
            <h1>Laporan Absensi</h1>
        </div>
        <table class="table table-striped">
            <thead>
                <tr>
                    <th class="text-center" scope="col">No</th>
                    <th class="text-center" scope="col">NP</th>
                    <th class="text-center" scope="col">Nama Dosen</th>
                    <th class="text-center" scope="col">Mata Kuliah</th>
                </tr>
            </thead>
        </table>
    </div>

```

```

        <th class="text-center" scope="col">Jam Ke</th>
        <th class="text-center" scope="col">Ruangan</th>
    </tr>
</thead>

<tbody>

    @foreach ($data_absen as $dt_absen)
        <tr>
            <td class="text-center">{{ $loop->iteration }}</td>
            <td class="text-center">{{ $dt_absen->dosen->np }}</td>
            <td class="text-center">{{ $dt_absen->dosen->dosen }}</td>
            <td class="text-center">{{ $dt_absen->matkul->nama_matkul
}}</td>

            <td class="text-center">{{ $dt_absen->jam }}</td>
            <td class="text-center">{{ $dt_absen->ruangan->ruangan }}</td>
        </tr>
    @endforeach

</tbody>
</table>

@if ($data_absen->count() === 0)
    <h2>Belum ada data</h2>
@endif

<div class="card-body mt-5 mb-5">
    <table class="table table-striped">

```

```

<thead>
  <tr>
    <th class="text-center" scope="col">No</th>
    <th class="text-center" scope="col">NIM</th>
    <th class="text-center" scope="col">Nama</th>
    <th class="text-center" scope="col">Tanggal</th>
    <th class="text-center" scope="col">Waktu</th>
  </tr>
</thead>
<tbody>
  @foreach ($data as $item)
    <tr>
      <td class="text-center">{{ $loop->iteration }}</td>
      <td class="text-center">{{ $item->nim }}</td>
      <td class="text-center">{{ $item->nama }}</td>
      <td class="text-center">{{ $item->tanggal->format('d/m/Y')
}}</td>
      <td class="text-center">{{ $item->>waktu }}</td>
    </tr>
  @endforeach
</tbody>
</table>
@if ($data->count() === 0)
  <h2>Belum ada data</h2>
@endif
</div>

```

```

</div>
@auth
  <div class="text-center mt-4 d-flex justify-content-center">
    <form action="/laporan" method="post">
      @csrf
      <button type="button" class="btn btn-danger" data-bs-toggle="modal"
data-bs-target="#exampleModal">
        Reset
      </button>
    </form>
    <a class="btn btn-success mx-5" data-bs-toggle="modal" data-bs-
target="#exampleModal1">Simpan</a>
    {{-- <form action="{{ route('shiftdata') }}" method="get">
      @csrf
      <button type="submit" class="btn btn-danger">Save</button>
    </form> --}}
  </div>
@endauth
<div class="modal fade" id="exampleModal" tabindex="-1" aria-
labelledby="exampleModalLabel"
aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Reset Laporan
Absensi</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal"

```

```

        aria-label="Close"></button>
    </div>
    <div class="modal-body">
        Tindakan ini akan menghapus laporan absensi kehadiran. Apakah ingin
        menghapusnya?
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-
        dismiss="modal">Batalkan</button>
        <form action="/laporan" method="post">
            @csrf
            <button class="btn btn-danger" type="submit">Reset</button>
        </form>
    </div>
</div>
</div>
</div>

<div class="modal fade" id="exampleModal1" tabindex="-1" aria-
labelledby="exampleModalLabel1"
    aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel1">Menyimpan Data
                Laporan Absensi</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal"

```



```

        aria-label="Close"></button>
    </div>
    <div class="modal-body">
        Tindakan ini akan menyimpan data laporan absensi ke histori absensi.
        Apakah ingin menyimpannya?
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Batalkan</button>
        <form action="/laporan" method="post">
            @csrf
            <a class="btn btn-success mx-8" href="{{ route('shiftdata') }}"
>Simpan</a>
        </form>
    </div>
</div>
</div>
<hr class="featurette-divider">

<!-- FOOTER -->
<footer class="container">
    <p class="float-end"><a href="#">Back to top</a></p>
    <p>&copy; Copyright Created By Anugrah Fahrul Ramadhan 2022</p>
</footer>
</main>

```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IIRH9sENBO0LRn5q+8nbTov4+
1p" crossorigin="anonymous">
</script>
</body>
</html>
```