

**RANCANG BANGUN SISTEM MANAJEMEN PENGUNJUNG
LABORATORIUM PROGRAM TEKNOLOGI INFORMASI DAN ILMU
KOMPUTER (PTIIK)
BERBASIS PENGENALAN WAJAH**

SKRIPSI

Konsentrasi Rekayasa Perangkat Lunak

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer



Disusun oleh :

ARI AGUSTINA

NIM. 0910680069

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
MALANG**

2013

LEMBAR PERSETUJUAN

RANCANG BANGUN SISTEM MANAJEMEN PENGUNJUNG LABORATORIUM PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER (PTIIK) BERBASIS PENGENALAN WAJAH

SKRIPSI

Konsentrasi Rekayasa Perangkat Lunak

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

Ari Agustina

NIM. 0910680069

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Ismiarta Aknuranda, ST., M.Sc., Ph.D

NIK. 740719 06 1 1 0079

Ahmad Afif Supianto, S.Si., M.Kom

NIK. 820623 16 1 1 0425

LEMBAR PENGESAHAN

**RANCANG BANGUN SISTEM MANAJEMEN PENGUNJUNG
LABORATORIUM PROGRAM TEKNOLOGI INFORMASI DAN
ILMU KOMPUTER (PTIIK)
BERBASIS PENGENALAN WAJAH**

SKRIPSI

Konsentrasi Rekayasa Perangkat Lunak

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh :

Ari Agustina

NIM. 0910680069

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 18 Juli 2013

Penguji I

Penguji II

Dr. Eng Herman Tolle, ST., MT.
NIP. 197408232000121001

Fajar Pradana, S.ST, M.Eng
NIK. 85090516110371

Penguji III

Aswin Suharsono, ST., MT.
NIK. 840919 06 1 1 0251

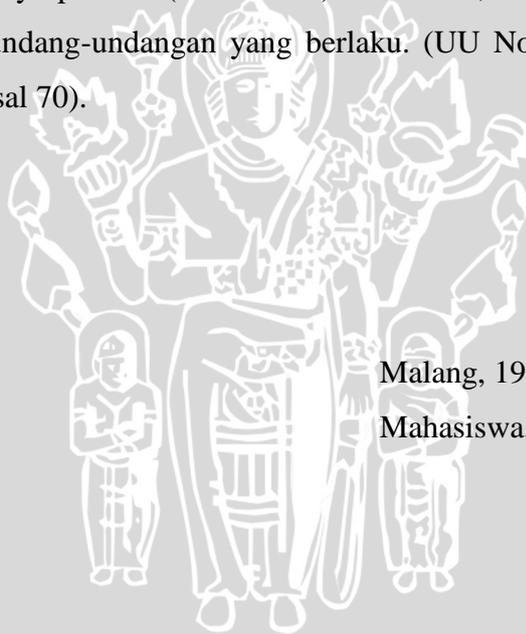
Mengetahui
Ketua Program Studi Teknik Informatika

Drs. Marji, MT.
NIP. 19670801 199203 1 001

**PERNYATAAN
ORISINALITAS SKRIPSI**

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

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



Malang, 19 Juli 2013

Mahasiswa,

Ari Agustina

NIM 0910680069

KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih dan Penyayang. Segala puji bagi Allah SWT karena atas rahmat dan hidayahNya-lah penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Sistem Manajemen Pengunjung Laboratorium Program Teknologi Informasi dan Ilmu Komputer (PTIIK) Berbasis Pengenalan Wajah”. Shalawat serta salam atas junjungan besar kita Nabi Muhammad S.A.W. beserta keluarga dan para sahabat sekalian. Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Studi Teknik Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.

Dalam menyelesaikan skripsi ini, banyak bantuan, bimbingan, dorongan dan do'a yang telah diberikan kepada penulis. Oleh karena itu, dalam kesempatan ini penulis ingin menyampaikan rasa terima kasih penulis kepada :

1. Bapak Subroto atas segala nasehat, perhatian, dukungan dan kesabaran dalam mendidik penulis, serta do'a yang selalu diberikan kepada penulis dalam menyelesaikan skripsi ini.
2. Mas Yanto, Mas Yoyok, Mbak Tutik, Mas Yudi, Mbak Ari dan seluruh keluarga besar yang selalu memberikan semangat dan motivasi dalam menyelesaikan skripsi ini.
3. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Bapak Drs. Marji, M.T dan Bapak Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
5. Bapak Ismiarta Aknuranda, ST., M.Sc.,Ph.D dan Bapak Ahmad Afif Supianto, S.Si.,M.Kom selaku dosen pembimbing yang telah dengan sabar

- membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
6. Bapak Suprpto, S.T, M.T selaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
 7. Mas Soni (laboran Lab. Komputasi dan Multimedia dan Lab. Komputer Dasar), Mas Adit (laboran Lab. Sistem Informasi), Mas Broto (laboran Lab. Rekayasa Perangkat Lunak dan Lab. Jaringan), dan Mas Didit (laboran Lab. Sistem Komputer dan Robotika) yang telah membantu penulis dalam pengujian *usability*.
 8. Seluruh Dosen Teknik Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
 9. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
 10. Pertamina Foundation yang sudah memberikan bantuan beasiswa biaya hidup dan SPP selama tiga tahun.
 11. Yang tercinta Kukuh Heru Irawan yang sudah membantu dengan sabar dan memberikan semangat selama penyusunan skripsi ini.
 12. Seluruh keluarga besar Ibu Sulasmi yang selalu membantu dan memberikan semangat kepada penulis selama mengerjakan skripsi.
 13. Himyatul Milah yang telah memberikan semangat dan tumpangan tempat tinggal selama proses mengerjakan skripsi.
 14. Sahabat-sahabat TIF'09 yang tercinta, Arianty Anggraeni, Fauziah Mayasari, Winda Ayu, Novelia Kharisma, Dian Arisandy, Milani Winangga, Adestia Rahmawati, Hanifa Vidya dan Ervin Yohanes yang selalu memberikan bantuan, semangat dan motivasi untuk menyelesaikan skripsi ini.
 15. Seluruh teman-teman TIF'09 yang tidak bisa penulis sebutkan satu persatu, teman-teman yang sudah memberikan data latihan untuk pengujian, terima kasih atas segala bantuan yang kalian berikan, baik selama penulis

menempuh studi di Teknik Informatika ini maupun pada saat penulis menyelesaikan skripsi.

16. Mas Adam, Mas Aldi, Mas Wisnu, dan Mbak Puput terima kasih atas segala bantuan, semangat dan do'a yang telah diberikan kepada penulis dalam menyelesaikan skripsi ini.

17. Alifia, Mas Deta, Mas Kukuh, Mas Aldi, Mas Wisnu, Ririn, Ervin, Nina, Hendro, Nisa, Meitika, Prima, Yugo, Astri, Mayang, Mas Hoyi, Heni, Margareth, dan Cahya yang sudah membantu penulis dalam melakukan pengujian skripsi ini.

18. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun yang tidak langsung demi terselesaikannya skripsi ini.

Hanya doa yang bisa penulis berikan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga skripsi ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 19 Juni 2013

Penulis

ABSTRAK

Ari Agustina. 2013: Rancang Bangun Sistem Manajemen Pengunjung Laboratorium Program Teknologi Informasi dan Ilmu Komputer (PTIHK) Berbasis Pengenalan Wajah. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Dosen Pembimbing: Ismiarta Aknuranda, ST., M.Sc., Ph.D dan Ahmad Afif Supianto, S.Si., M.Kom.

Kamera tidak dapat terlepas dari sistem keamanan dimanapun. Pada sistem keamanan konvensional, citra dari kamera disimpan dalam bentuk rekaman. Melalui rekaman tersebut pemantau bekerja keras untuk menganalisis dan mengidentifikasi setiap pengunjung. Berdasarkan permasalahan tersebut, diperlukan sistem yang membantu mengidentifikasi pengunjung melalui karakteristik citra yang ditangkap oleh kamera. Laboratorium PTIHK merupakan tempat yang membutuhkan sistem keamanan karena terdapat alat-alat praktik yang harus dijaga. Sistem Manajemen Pengunjung Laboratorium PTIHK Berbasis Pengenalan Wajah dibuat untuk membantu pemantau dalam mengidentifikasi pengunjung yang masuk dalam pengawasan kamera. Sistem dirancang menggunakan OOAD (*Object Oriented Analysis and Design*) yang dimodelkan dengan UML (*Unified Modelling Language*) dan diimplementasikan dengan bahasa pemrograman C#. Metode pengembangan sistem yang digunakan adalah *Reuse Oriented Model*. Pada saat pengujian unit dan integrasi, disimpulkan bahwa unit modul dan integrasi dari beberapa unit modul dari sistem sudah memenuhi kebutuhan fungsional yang telah dirancang pada tahap perancangan. Pada saat pengujian validasi, disimpulkan bahwa implementasi dan fungsionalitas sistem telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan. Pada pengujian performa diperoleh nilai akurasi modul pengenalan wajah sebesar 80,59% dan *error rate* sebesar 19,40%. Pada pengujian *usability*, diperoleh nilai *usability* sistem sebesar 75%.

Kata Kunci : Sistem Manajemen Pengunjung, Pengenalan Wajah, *Reuse Oriented*

ABSTRACT

Ari Agustina. 2013. : *Developing PTIIK Laboratory Visitor Management System Based On Face Recognition.*

Advisor : Ismiarta Aknuranda, ST., M.Sc., Ph.D and Ahmad Afif Supianto, S.Si., M.Kom

Camera can not be separated from security system everywhere. In the conventional security systems, the images obtained from the camera are stored in the form of video recording. By observing the recording, the observer working hard to analyze and identify each visitor. Based on these problems, we need a system that can help identify visitors through the characteristics of the image captured by the camera. PTIIK laboratory is a place that requires a security system because there are practice tools that should be kept. PTIIK Laboratory Visitor Management System Based On Face Recognition made to assist observers in identifying lab visitors who enter the surveillance camera that has been determined. The system is designed using OOAD (Object Oriented Analysis and Design), modeled with UML (Unified Modeling Language) and implemented with the C# programming language. Systems development method used is Reuse Oriented Model. At unit testing and integration testing, it was concluded that the unit module and the integration of some unit modules of the system meets the functional requirements that have been designed at the design stage. At validation testing, it was concluded that the implementation and functionality of the system meets the requirements that have been outlined in the requirements analysis phase. In the performance testing, the value of the face recognition module's accuracy is 80.59% and the error rate is 19.40%. In the usability testing, the value of the system's usability obtained is 75%.

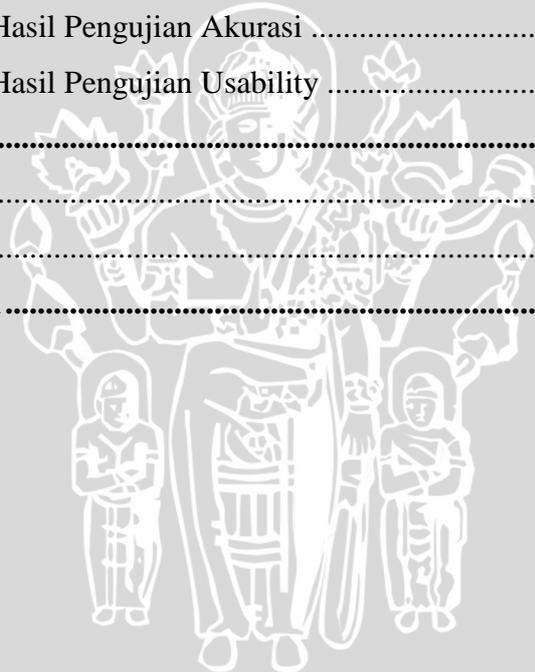
Keywords : Visitor Management System, Face Recognition, Reuse Oriented

DAFTAR ISI

KATA PENGANTAR	i
ABSTRAK	iv
ABSTRACT	v
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Batasan Masalah.....	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	4
BAB II KAJIAN PUSTAKA DAN DASAR TEORI	5
2.1 Pengenalan Wajah.....	5
2.2 Emgu CV.....	6
2.3 OpenCV	6
2.3.1 Deteksi Wajah pada OpenCV	8
2.3.2 Pengenalan Wajah Pada OpenCV	10
2.4 Reuse-Oriented Model	10
2.5 Pengujian.....	12
2.5.1 Pengujian White-Box	13
2.5.2 Pengujian Black-Box	14
2.5.3 Pengujian Akurasi	15
2.5.4 Pengujian Usability	16
BAB III METODE PENELITIAN DAN PERANCANGAN	17
3.1 Metode Penelitian.....	17
3.1.1 Studi Literatur	18

3.1.2	Analisis Kebutuhan	18
3.1.3	Analisis Komponen	18
3.1.4	Modifikasi Kebutuhan	18
3.1.5	Perancangan Model dengan Komponen	19
3.1.6	Implementasi	19
3.1.7	Pengujian dan Analisis	20
3.2	Analisis Kebutuhan dan Perancangan Perangkat Lunak	22
3.2.1	Analisis Kebutuhan	22
3.2.2	Perancangan Perangkat Lunak	39
BAB IV	IMPLEMENTASI	59
4.1	Spesifikasi Sistem	59
4.1.1	Spesifikasi Perangkat Keras	59
4.1.2	Spesifikasi Perangkat Lunak	59
4.2	Batasan-batasan Implementasi	60
4.3	Implementasi Basis Data	60
4.4	Implementasi <i>Class</i> dan Antarmuka pada <i>File</i> Program	61
4.5	Implementasi Method	61
4.5.1	Implementasi Method <i>getLabel</i>	62
4.5.2	Implementasi Method <i>Recognize</i>	62
4.5.3	Implementasi Method <i>BukaData</i>	63
4.5.4	Implementasi Method <i>simpanWajah</i>	64
4.5.5	Implementasi Method <i>ProsesCapture</i>	65
4.6	Implementasi Antarmuka	66
4.6.1	Halaman Login	66
4.6.2	Halaman Utama	67
4.6.3	Halaman Data Pengunjung	68
4.6.4	Halaman Latih Wajah	68
4.6.5	Halaman Latih Wajah Unknown	69
4.6.6	Halaman Daftar Pencarian Pengunjung	69
4.6.7	Halaman Daftar Pengguna	70
4.6.8	Halaman Setting Motion	71

BAB V PENGUJIAN DAN ANALISIS.....	72
5.1 Pengujian.....	72
5.1.1 Pengujian Unit.....	72
5.1.2 Pengujian Integrasi.....	79
5.1.3 Pengujian Validasi	81
5.1.4 Pengujian Akurasi	86
5.1.5 Pengujian Usability	96
5.2 Analisis.....	98
5.2.1 Analisis Hasil Pengujian Unit.....	99
5.2.2 Analisis Hasil Pengujian Integrasi	99
5.2.3 Analisis Hasil Pengujian Validasi.....	99
5.2.4 Analisis Hasil Pengujian Akurasi	100
5.2.5 Analisis Hasil Pengujian Usability	101
BAB VI PENUTUP	103
6.1 Kesimpulan	103
6.2 Saran.....	104
DAFTAR PUSTAKA	105



DAFTAR GAMBAR

Gambar 2.1 Sistem Manajemen Pengunjung	5
Gambar 2.2 Tinjauan arsitektur Emgu CV	7
Gambar 2.3 Struktur dasar OpenCV	8
Gambar 2.4 <i>Reuse-Oriented Model</i>	11
Gambar 2.5 <i>Flow Graph</i>	14
Gambar 3.1 Diagram alir runtutan pengerjaan skripsi secara umum (adaptasi dari [SOM-11:35])	17
Gambar 3.2 Contoh Pose Wajah Data Latih	21
Gambar 3.3 Cara Kerja Sistem Secara Umum	25
Gambar 3.4 Proses Pelatihan	26
Gambar 3.5 Proses Pengujian	27
Gambar 3.6 Diagram <i>Use Case</i> Sistem Manajemen Pengunjung	30
Gambar 3.7 Fungsi .NET Framework 4	38
Gambar 3.8 Fungsi EmguCV	38
Gambar 3.9 Fungsi Aforge .NET	38
Gambar 3.10 Fungsi DirectShowLib .NET	39
Gambar 3.11 Diagram Arsitektur Sistem Manajemen Pengunjung	40
Gambar 3.12 Diagram <i>Entity Relationship</i> Sistem Manajemen Pengunjung	40
Gambar 3.13 <i>Class Diagram</i> Sistem Manajemen Pengunjung	43
Gambar 3.14 Diagram Sekuensial Login	45
Gambar 3.15 Diagram Sekuensial Menambah Data Pengunjung	46
Gambar 3.16 Diagram Sekuensial <i>Edit</i> Data Pengunjung	46
Gambar 3.17 Diagram Sekuensial Menghapus Data Pengunjung	46
Gambar 3.18 Diagram Sekuensial Menambah Data Wajah	47
Gambar 3.19 Diagram Sekuensial Menghapus Data Wajah	47
Gambar 3.20 Diagram Sekuensial Menambah Data Pencarian Pengunjung	48
Gambar 3.21 Diagram Sekuensial Menghapus Data Pencarian Pengunjung	48
Gambar 3.22 Diagram Sekuensial Menambah Data Pengguna	49
Gambar 3.23 Diagram Sekuensial <i>Edit</i> Data Pengguna	49

Gambar 3.24 Diagram Sekuensial Menghapus Data Pengguna.....	49
Gambar 3.25 Struktur Hirarki Menu untuk <i>Administrator</i>	50
Gambar 3.26 Struktur Hirarki Menu untuk Pemantau	51
Gambar 3.27 Tampilan Antarmuka Halaman Login.....	51
Gambar 3.28 Tampilan Antarmuka Halaman Utama.....	52
Gambar 3.29 Tampilan Antarmuka Halaman Data Pengunjung.....	53
Gambar 3.30 Tampilan Antarmuka Halaman Latih Wajah Data Pengunjung	54
Gambar 3.31 Tampilan Antarmuka Halaman Latih Wajah <i>Unknown</i>	55
Gambar 3.32 Tampilan Antarmuka Daftar Pencarian Pengunjung	55
Gambar 3.33 Tampilan Antarmuka Halaman Daftar Pengguna.....	56
Gambar 3.34 Tampilan Antarmuka Halaman <i>Setting Motion</i>	57
Gambar 4.1 Diagram ER konseptual dari Sistem Manajemen Pengunjung	61
Gambar 4.2 Implementasi Antarmuka Halaman Login	66
Gambar 4.3 Implementasi Antarmuka Halaman Utama <i>Administrator</i>	67
Gambar 4.4 Implementasi Antarmuka Halaman Utama Pemantau.....	67
Gambar 4.5 Implementasi Antarmuka Halaman Data Pengunjung	68
Gambar 4.6 Implementasi Antarmuka Halaman Latih Wajah	68
Gambar 4.7 Implementasi Antarmuka Halaman Latih Wajah <i>Unknown</i>	69
Gambar 4.8 Implementasi Antarmuka Halaman Daftar Pencarian Pengunjung .	69
Gambar 4.9 Implementasi Antarmuka Tampilan Pesan Pemberitahuan	70
Gambar 4.10 Implementasi Antarmuka Halaman Daftar Pengguna	70
Gambar 4.11 Implementasi Antarmuka Halaman <i>Setting Motion</i>	71
Gambar 5.1 <i>Flow Graph</i> Method <i>getLabel</i>	73
Gambar 5.2 <i>Flow Graph</i> Method <i>Recognize</i>	74
Gambar 5.3 <i>Flow Graph</i> Method <i>BukaData</i>	76
Gambar 5.4 <i>Flow Graph</i> Method <i>simpanWajah</i>	78
Gambar 5.5 <i>Flow Graph</i> Method <i>ProsesCapture</i>	80
Gambar 5.6 Citra yang salah prediksi	101

DAFTAR TABEL

Tabel 2.1 <i>Confusion matrix</i> untuk permasalahan 2 class	15
Tabel 3.1 Contoh hasil prediksi	21
Tabel 3.2 Identifikasi Aktor	28
Tabel 3.3 Spesifikasi Kebutuhan Pemantau dan <i>Administrator</i>	29
Tabel 3.4 Spesifikasi Kebutuhan <i>Administrator</i>	29
Tabel 3.5 Spesifikasi Kebutuhan Pengunjung	29
Tabel 3.6 Spesifikasi Kebutuhan Non-Fungsional	30
Tabel 3.7 <i>Use Case Login</i>	30
Tabel 3.8 <i>Use Case</i> Mengelola Data Pengunjung	31
Tabel 3.9 <i>Use Case</i> Mengelola Data Wajah	33
Tabel 3.10 <i>Use Case</i> Mengelola Data Pencarian Pengunjung	34
Tabel 3.11 <i>Use Case</i> Mengelola Data Pengguna	35
Tabel 3.12 <i>Use Case</i> Mengarahkan Wajah Kekamera	36
Tabel 3.13 Daftar Komponen	37
Tabel 3.14 Tabel Pengguna	41
Tabel 3.15 Tabel Pengunjung	41
Tabel 3.16 Tabel Wajah	41
Tabel 3.17 Tabel Log	42
Tabel 3.18 Tabel Pencarian	42
Tabel 3.19 Deskripsi Class	43
Tabel 4.1 Spesifikasi Perangkat Keras Komputer	59
Tabel 4.2 Spesifikasi Perangkat Lunak Komputer	60
Tabel 4.3 Implementasi <i>class</i> pada kode program	61
Tabel 4.4 Implementasi <i>method</i> getLabel	62
Tabel 4.5 Implementasi <i>method</i> Recognize	62
Tabel 4.6 Implementasi <i>method</i> BukaData	63
Tabel 4.7 Implementasi <i>method</i> simpanWajah	64
Tabel 4.8 Implementasi <i>method</i> ProsesCapture	65

Tabel 5.1 Pengujian Unit untuk Method <code>getLabel</code>	72
Tabel 5.2 <i>Test Case</i> untuk Pengujian Unit Method <code>getLabel</code>	74
Tabel 5.3 Pengujian Unit untuk Method <code>Recognize</code>	74
Tabel 5.4 <i>Test Case</i> untuk Pengujian Unit Method <code>Recognize</code>	75
Tabel 5.5 Pengujian Unit untuk Method <code>BukaData</code>	75
Tabel 5.6 <i>Test Case</i> Pengujian Unit untuk Method <code>BukaData</code>	77
Tabel 5.7 Pengujian Unit untuk Method <code>simpanWajah</code>	77
Tabel 5.8 <i>Test Case</i> untuk Pengujian Unit Method <code>Simpan Wajah</code>	79
Tabel 5.9 Pengujian Integrasi untuk Method <code>ProsesCapture</code>	79
Tabel 5.10 <i>Test Case</i> untuk Pengujian Integrasi Method <code>ProsesCapture</code>	81
Tabel 5.11 Kasus Uji Login	82
Tabel 5.12 Kasus Uji Mengelola Data Pengunjung	82
Tabel 5.13 Kasus Uji Mengelola Data Wajah.....	83
Tabel 5.14 Kasus Uji Mengelola Data Pencarian Pengunjung	84
Tabel 5.15 Kasus Uji Mengelola Data Pengguna.....	84
Tabel 5.16 Kasus Uji Menghadap Kamera	85
Tabel 5.17 Hasil Pengujian Validasi	85
Tabel 5.18 Data Latih Pengunjung.....	87
Tabel 5.19 Hasil Pengujian Akurasi.....	88
Tabel 5.20 <i>Confusion Matrix</i> untuk hasil pengujian akurasi.....	95
Tabel 5.21 Pernyataan pengujian <i>usability</i>	96
Tabel 5.22 Pernyataan pengujian <i>usability</i>	97
Tabel 5.23 Komponen Pernyataan	98
Tabel 5.24 Citra yang salah prediksi	100

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kamera tidak dapat terlepas dari sistem keamanan dimana pun. Pengunjung perlu dipantau untuk mencegah hal-hal yang tidak diinginkan. Pada sistem konvensional, citra dari kamera disimpan dalam bentuk rekaman. Melalui rekaman tersebut seorang pemantau dapat menganalisis dan mengidentifikasi setiap pengunjung. Dari hasil analisis dan identifikasi tersebut, pemantau dapat melakukan tindakan-tindakan yang diperlukan dalam menjaga keamanan. Pemantau harus bekerja keras memeriksa seluruh hasil rekaman kamera pada saat yang dibutuhkan. Oleh karena itu, selain adanya pemantau, juga diperlukan adanya sistem yang membantu mengidentifikasi pengunjung melalui karakteristik citra yang ditangkap oleh kamera. Salah satu bagian tubuh manusia yang dapat dianalisis dan dikenali melalui pemrosesan citra adalah wajah.

Wajah merupakan salah satu pengenal bagi setiap manusia. Hal ini menyebabkan pengenalan wajah (*face recognition*) terus dikembangkan dan menjadi alternatif dalam berbagai sistem atau aplikasi yang membutuhkan identifikasi seseorang, karena wajah merupakan bagian langsung dari tubuh manusia yang tidak mudah dicuri atau diduplikasi. Menurut Patil A.M., Kolhe S.R., dan Patil P.M., pengenalan wajah merupakan salah satu biometrik paling penting yang tampaknya menjadi kesatuan yang baik antara aktualitas dan penerimaan sosial serta keseimbangan antara keamanan dan privasi [PAT-10:1].

Laboratorium PTIHK merupakan tempat praktik untuk mahasiswa. Di dalam laboratorium terdapat alat-alat praktik yang sangat penting untuk proses praktikum dan pembelajaran. Alat-alat praktik tersebut merupakan aset laboratorium yang harus dijaga keberadaannya. Setiap pengunjung laboratorium harus mengisi *form* masuk laboratorium yang berisi nim, nama, alasan, waktu masuk, dan waktu keluar. *Form* tersebut digunakan sebagai arsip laboratorium dan dapat dijadikan bahan pemeriksaan pada saat terjadi hal-hal yang tidak diinginkan seperti pencurian.

Laboratorium selalu dijaga oleh laboran. Namun, laboran tidak selalu dapat mengawasi secara terus menerus. Oleh karena itu, selain dijaga oleh laboran, laboratorium juga harus didukung oleh sistem keamanan yang mampu merekam keadaan laboratorium secara otomatis. Sistem keamanan yang dimaksud adalah sistem keamanan berbasis kamera yang dapat memantau keadaan yang ada di dalam laboratorium PTIIK dan juga dapat mengidentifikasi pengunjung yang masuk ke dalam laboratorium PTIIK melalui citra wajah pengunjung. Hasil identifikasi tersebut akan dicatat oleh sistem ke dalam *database* dan akan ditampilkan pada aplikasi sebagai *log* yang mencatat pemberitahuan tempat dan waktu pengunjung.

Jika prototipe sistem keamanan yang mampu mengidentifikasi pengunjung diimplementasikan secara nyata, maka dapat berpotensi mempermudah pekerjaan pemantau atau laboran dalam hal identifikasi pengunjung. Selain itu juga dapat diterapkan pencarian orang menggunakan data yang telah tersimpan di *database* sistem. Pencarian orang dibutuhkan karena mahasiswa PTIIK telah mencapai 1000 orang sehingga aplikasi pencarian orang perlu ditanamkan dalam sistem keamanan. *Log* yang dihasilkan oleh sistem keamanan juga dapat melengkapi data yang telah ditulis dalam *form* masuk laboratorium.

Berdasarkan latar belakang tersebut, penulis mengambil judul “**Rancang Bangun Sistem Manajemen Pengunjung Laboratorium Program Teknologi Informasi dan Ilmu Komputer (PTIIK) Berbasis Pengenalan Wajah**” dengan harapan aplikasi ini dapat membantu meningkatkan fungsi sistem pemantauan keamanan berbasis kamera yang diterapkan dalam laboratorium PTIIK.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan permasalahannya, yaitu:

1. Merancang Sistem Manajemen Pengunjung
2. Mengimplementasikan Sistem Manajemen Pengunjung dengan fitur pengenalan wajah
3. Menguji Sistem Manajemen Pengunjung terhadap kebutuhan fungsional dan nonfungsional

1.3 Tujuan

Tujuan penyusunan skripsi ini adalah merancang, mengimplementasi, dan menguji prototipe Sistem Manajemen Pengunjung yang diharapkan dapat membantu meningkatkan fungsi sistem pemantauan keamanan berbasis kamera yang diterapkan dalam laboratorium PTIIK.

1.4 Batasan Masalah

Mengacu pada permasalahan yang telah dirumuskan, maka aspek yang dibahas berkaitan dengan skripsi ini antara lain:

1. Aplikasi yang dibuat menerima masukan berupa citra wajah yang di-*capture* menggunakan *webcam* pada saat pengunjung melewati kamera *webcam* tersebut.
2. Pengujian pengenalan wajah dilakukan kepada 20 orang yang berbeda dengan 10 data wajah yang sudah disimpan dalam *database* dan 10 data wajah yang belum tersimpan dalam *database* atau “unknown”.
3. Pembahasan ditekankan pada bagaimana menggunakan fungsi pengenalan wajah yang telah ada untuk membangun aplikasi.

1.5 Manfaat

- a. Bagi penulis:
 1. Menerapkan ilmu yang telah diperoleh dari Program Studi Teknik Informatika Universitas Brawijaya.
 2. Mendapatkan pemahaman tentang deteksi dan pengenalan wajah sehingga dapat mengimplementasikannya sesuai dengan kebutuhan.
- b. Bagi pengguna (pemantau) :
 1. Meningkatkan fungsi sistem pemantauan keamanan dengan kamera.
 2. Mempermudah pengolahan data hasil tangkapan kamera sesuai dengan kebutuhan.

1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

BAB I Pendahuluan

Menguraikan tentang latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat tujuan, dan sistematika penulisan.

BAB II Kajian Pustaka dan Dasar Teori

Menguraikan tentang dasar teori dan referensi yang mendasari pembangunan aplikasi Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah.

BAB III Metode Penelitian dan Perancangan

Menguraikan tentang metode dan langkah kerja dalam merancang aplikasi Sistem Manajemen Pengunjung. Langkah-langkah yang dijelaskan dalam bab ini meliputi studi literatur, analisis kebutuhan, analisis komponen, modifikasi kebutuhan, perancangan perangkat lunak, implementasi perangkat lunak, dan pengujian perangkat lunak.

BAB IV Implementasi

Membahas implementasi dari aplikasi Sistem Manajemen Pengunjung sesuai dengan perancangan perangkat lunak yang telah dibuat.

BAB V Pengujian dan Analisis

Memuat hasil pengujian dan analisis terhadap aplikasi yang telah direalisasikan.

BAB VI Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran – saran untuk pengembangan lebih lanjut.

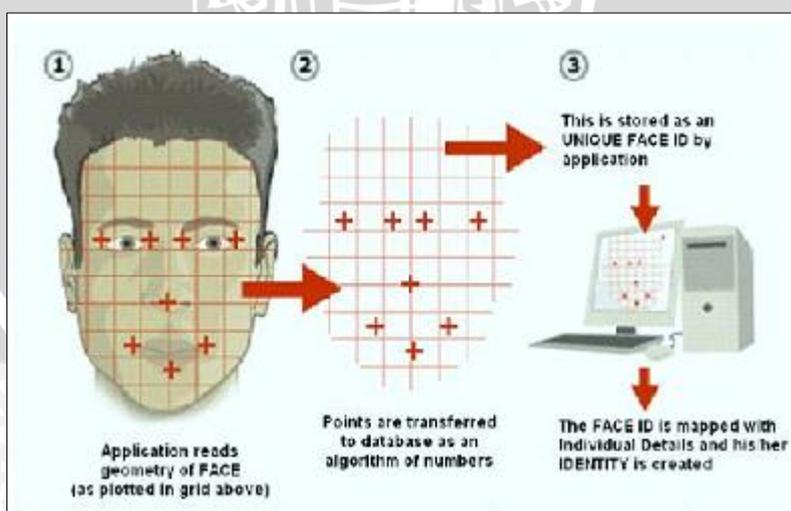
BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

2.1 Pengenalan Wajah

Pengenalan wajah adalah pengenalan seseorang dengan menggunakan gambar wajah secara digital [JAI-12:1]. Teknologi ini dapat digunakan untuk mempercepat proses pencarian informasi mengenai seseorang berdasarkan foto atau gambar wajah seseorang, dengan syarat data yang dicari harus sudah dimasukkan terlebih dahulu ke dalam sistem [INT-09:3]. Kemudian, gambar yang tertangkap oleh kamera akan dicocokkan oleh sistem sesuai data yang ada di *database*.

Pengenalan wajah menganalisis karakteristik citra wajah seseorang melalui kamera. Berdasarkan citra wajah tersebut, sistem menilai keseluruhan struktur wajah, termasuk jarak antara mata, hidung, mulut, dan dagu. Karakteristik unik ini yang digunakan untuk melakukan pengenalan wajah. Autentikasi berbasis *password* atau tanda pengenal sangat mudah untuk diretas. Biometrik merepresentasikan alternatif yang valid tetapi juga memiliki kesulitan. Pengenalan wajah memberikan sebuah perspektif baru yang bernama “Sistem Manajemen Pengunjung” seperti ilustrasi yang ditunjukkan pada Gambar 2.1 [BAN-12:1].



Gambar 2.1 Sistem Manajemen Pengunjung

Sumber : [BAN-12:2]

Untuk beberapa organisasi, manajemen pengunjung yang efektif membantu untuk menghindari pembobolan. Cara konvensional yang menggunakan kertas dan pena tidak dapat memberikan solusi keamanan yang komplit. Manajemen pengunjung berbasis pengenalan wajah menyediakan solusi keamanan dengan mengidentifikasi setiap wajah pengunjung. Jadi, setiap wajah baru diidentifikasi oleh sistem dan ID pengunjung yang unik di-*generate* dan sistem ini juga memberi fasilitas untuk mencetak kartu identitas pengunjung [BAN-12:1]. Dalam skripsi ini, tidak dibahas bagaimana proses pengenalan wajah dilakukan secara spesifik berdasarkan algoritma tertentu, namun lebih ditekankan bagaimana menggunakan fungsi pengenalan wajah yang telah ada untuk membangun aplikasi.

2.2 Emgu CV

Emgu CV adalah pembungkus (*wrapper*) *library* pemrosesan citra Open CV dalam .NET. Emgu CV ditulis dalam bahasa C#. Akan tetapi Emgu CV juga dapat digunakan dari beberapa bahasa, seperti VB.NET, C++ dan IronPython [EMG-12:1].

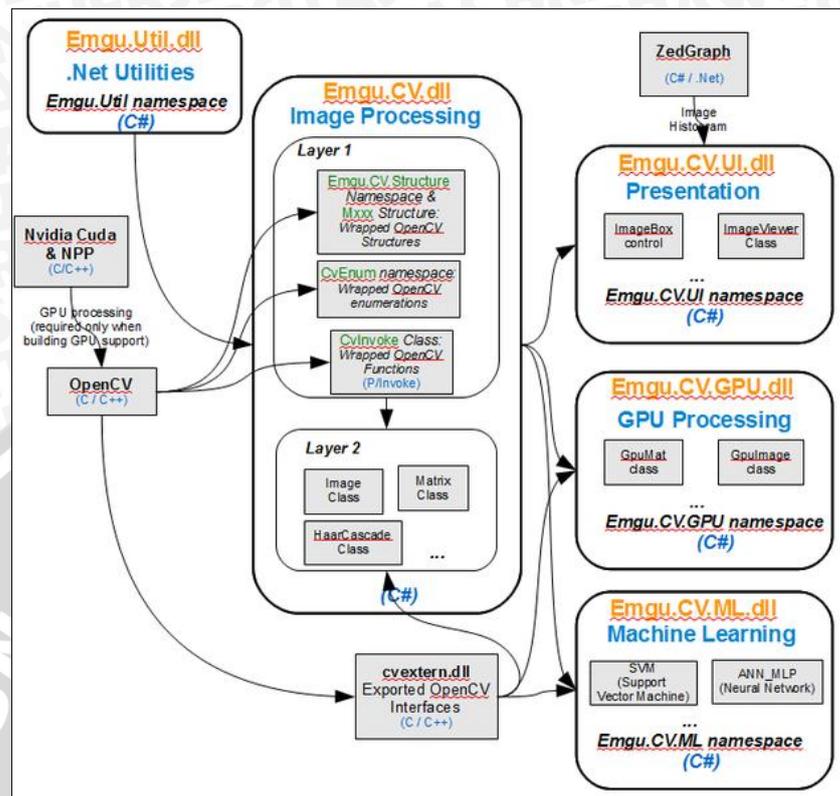
Emgu CV mempunyai dua lapisan pembungkus seperti yang ditunjukkan pada Gambar 2.2, yaitu :

- Lapisan dasar berisi pemetaan fungsi, struktur dan enumerasi yang secara langsung merefleksikan OpenCV.
- Lapisan kedua berisi kelas-kelas yang menggabungkan kelebihan dari .NET.

Sebagai pembungkus *library* OpenCV, Emgu CV menggunakan *library* OpenCV yang kemudian meng-*compile*-nya sehingga menjadi *library* yang dapat digunakan dalam .NET.

2.3 OpenCV

OpenCV adalah *library open source* untuk *computer vision*. *Library* tersebut ditulis dalam bahasa C dan C++ dan berjalan pada Linux, Windows dan MacOS X [BRA-08:1].

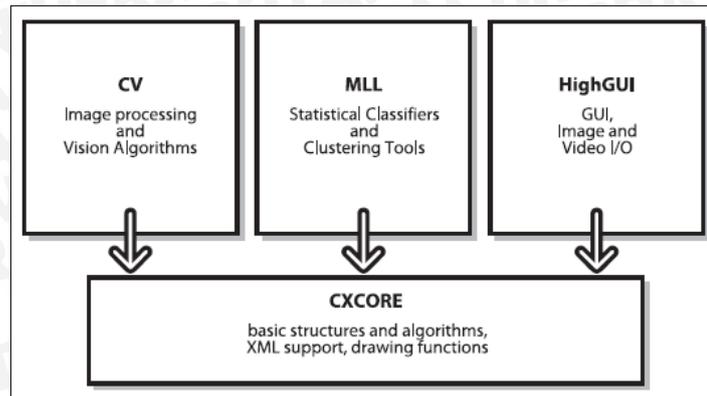


Gambar 2.2 Tinjauan arsitektur Emgu CV

Sumber : [EMG-12:1]

Sebagian besar programmer mengetahui beberapa peran *computer vision*, tapi hanya sedikit dari mereka yang mengetahui bagaimana cara menggunakannya. Contohnya, beberapa orang tahu jika *computer vision* digunakan untuk pengembangan penggunaan citra dan video dalam Web. Dan sedikit yang tahu menggunakannya dalam *game* [BRA-08:1]. Sejak dikeluarkannya OpenCV alpha pada Januari 1999, OpenCV telah digunakan pada banyak aplikasi, produk, dan hasil riset. Salah satu contohnya yaitu sistem deteksi dan sistem keamanan. Struktur dasar OpenCV ditunjukkan pada Gambar 2.3.

Komponen CV berisi pemrosesan citra dasar dan level algoritma *computer vision* yang lebih tinggi. MLL adalah *library* mesin pembelajaran yang berisi beberapa *classifier* statistik dan *tools* untuk *clustering*. HighGUI berisi perintah untuk *input/output* dan fungsi untuk menyimpan dan membuka video dan citra. CXCORE berisi struktur data dasar [BRA-08:13].



Gambar 2.3 Struktur dasar OpenCV

Sumber : [BRA-08:13]

Pada Gambar 2.3 tidak termasuk CvAux, dimana CvAux berisi kedua area *defunct* (Pengenalan Wajah HMM *embedded*) dan algoritma yang bersifat eksperimental (segmentasi *background* atau *foreground*). CvAux tidak terdokumentasi dengan baik di dalam wiki, dan tidak terdokumentasi pada dokumentasi yang tersedia di *subdirectory ...opencv/docs*. CvAux meliputi [BRA-08:13]:

1. Objek Eigen, teknik pengenalan dengan komputasi yang efisien, secara esensial merupakan *template matching procedure*.
2. Hidden Markov Model 1D dan 2D, teknik pengenalan berbasis statistik yang diselesaikan dengan *dynamic programming*.
3. HMMs *embedded*.
4. *Gesture recognition* dari dukungan *stereo vision*.
5. *Stereo Vision*.
6. Pengenalan bentuk dengan *region contours*.
7. *Tracking* mata dan mulut.
8. *3D tracking*.
9. *Video Surveillance*.

2.3.1 Deteksi Wajah pada OpenCV

Menurut Bradski, deteksi wajah merupakan kebutuhan umum yang layak memiliki teknik dasar yang bekerja dengan baik, terkenal, sering digunakan dalam peningkatan statistik, dan juga umum digunakan. Faktanya, beberapa perusahaan telah merekayasa detektor wajah dalam OpenCV untuk mendeteksi bermacam-

macam objek (wajah, mobil, sepeda, tubuh manusia) dengan melakukan *training* detektor baru pada ribuan citra *training*. Teknik ini telah menjadi detektor yang disesuaikan dengan keadaan, walaupun dengan detektor berbeda-beda untuk setiap pose dari objek. Oleh karena itu, Haar *classifier* merupakan teknik yang berharga untuk melakukan tugas pengenalan [BRA-08:506].

OpenCV mengimplementasikan teknik deteksi wajah yang dikembangkan pertama kali oleh Paul Viola dan Michael Jones –secara umum sering disebut dengan detektor Viola-Jones- yang kemudian dikembangkan lagi oleh Rainer Lienhart dan Jochen Maydt dengan menggunakan fitur diagonal. OpenCV mengacu detektor ini sebagai “Haar *Classifier*” karena detektor tersebut menggunakan fitur Haar, atau lebih tepatnya *Haar-like wavelet* yang terdiri dari penjumlahan dan pengurangan *region* dari citra sebelum melakukan *thresholding* pada hasil citra [BRA-08:506].

OpenCV menyertakan objek pra-*training* untuk detektor ini yang tersimpan pada folder *haarcascade* dimana model terbaik untuk deteksi wajah adalah *haarcascade_frontalface_alt2.xml* [BRA-08:507]. Dengan menggunakan *library* Emgu CV sebagai pembungkus *library* OpenCV dalam .NET, untuk mendeteksi wajah menggunakan *haar classifier*, langkah-langkahnya adalah :

1. Membuat objek *CascadeClassifier* dengan parameter alamat file *haarcascade_frontalface_alt2.xml* untuk data *classifier* yang disediakan OpenCV.
2. Membuat objek *Capture* untuk mengambil citra dari *webcam*.
3. Membuat objek *Image* untuk mengambil *frame* dari citra yang tertangkap oleh *webcam*.
4. Dengan menggunakan objek *Image* yang telah berisi citra tiap *frame* yang ditangkap *webcam*, jalankan *method* “DetectMultiscale”. *Method* ini mengembalikan nilai berupa *array* objek wajah yang terdeteksi.
5. Untuk mengambil data tiap objek wajah yang terdeteksi, maka dapat dilakukan iterasi terhadap *array* objek wajah.

2.3.2 Pengenalan Wajah Pada OpenCV

Salah satu aplikasi utama dari PCA dalam *computer vision* adalah pengenalan wajah. PCA digunakan untuk mencari vektor-vektor dengan nilai terbaik pada distribusi citra wajah di dalam keseluruhan citra. Vektor-vektor ini mendefinisikan sub-ruang dalam citra wajah yang bisa disebut sebagai “ruang wajah”. Setiap vektor memiliki panjang N^2 atau citra dengan ukuran $N \times N$ dan berupa kombinasi linear dari citra wajah asli. Karena vektor-vektor ini adalah vektor *eigen* dari matriks kovarian berdasarkan citra wajah asli dan tampilan vektor-vektor ini menyerupai wajah, maka vektor-vektor ini disebut dengan *eigenfaces* [LAT-09:1].

Eigenfaces adalah sekumpulan vektor *eigen* yang digunakan dalam bidang *computer vision* untuk pengenalan wajah manusia. *Eigenfaces* mengacu pada pendekatan berbasis tampilan dalam melakukan pengenalan wajah yang berusaha menangkap variasi dari koleksi citra wajah dan menggunakannya untuk dibandingkan dengan citra wajah secara individu. Secara spesifik, *eigenfaces* merupakan komponen utama dari distribusi wajah, atau secara ekuivalen, *eigenfaces* adalah vektor-vektor *eigen* dari matriks kovarian sekumpulan citra wajah dimana citra dengan ukuran $N \times N$ *pixel* dianggap sebagai sebuah titik (atau vektor) dalam ruang dimensi N^2 [LAT-09:1].

Di dalam OpenCV, kelas yang digunakan untuk mengenali wajah adalah *EigenFaceRecognizer*. Kelas ini mengaplikasikan PCA pada setiap citra yang menghasilkan *array* dari nilai *eigen* sehingga dapat dipelajari dan dikenali menggunakan jaringan syaraf. Konstruktor *EigenFaceRecognizer* memiliki parameter berupa *array* dari citra abu-abu dan *array* label, dimana setiap citra ini harus memiliki ukuran yang sama dan dianjurkan telah dilakukan *histogram equalisation*.

2.4 Reuse-Oriented Model

Reuse-oriented merupakan rekayasa perangkat lunak yang menggunakan pendekatan berbasis tersedianya jumlah komponen *reusable* yang signifikan. Proses pengembangan sistem berfokus pada integrasi komponen-komponen ini ke

dalam sistem daripada mengembangkan sistem dari dasar [SOM-11:30]. Model proses secara umum dari pengembangan berbasis *reuse* dapat dilihat pada Gambar 2.4.

Walaupun inisialisasi spesifikasi kebutuhan dan validasi sama dengan proses pengembangan perangkat lunak yang lain, namun terdapat perbedaan pada pertengahan proses, dimana terdapat proses *reuse-oriented*. Proses ini antara lain:

1. Analisis komponen. Dari spesifikasi kebutuhan yang telah ditentukan, dicari komponen yang dapat mengimplementasi spesifikasi tersebut. Biasanya, komponen yang ada hanya beberapa fungsi yang digunakan dari komponen tersebut.



Gambar 2.4 *Reuse-Oriented Model*
Sumber : [SOM-11:35]

2. Pada proses modifikasi kebutuhan, kebutuhan dianalisa menggunakan informasi tentang komponen yang akan digunakan. Kebutuhan dimodifikasi untuk merefleksikan komponen yang tersedia. Tetapi jika kebutuhan yang dimodifikasi tidak mungkin untuk diterapkan, maka proses analisis komponen dapat dilakukan kembali untuk mencari solusi alternatif.
3. Desain sistem dengan *reuse*. Pada proses ini, *framework* dari sistem atau *framework* yang telah ada digunakan kembali (*reused*).
4. Pengembangan dan integrasi [SOM-11:35].

Terdapat tiga tipe komponen perangkat lunak yang dapat digunakan dalam proses *reuse-oriented*, yaitu :

1. *Web service* yang dikembangkan berdasarkan *service standard* dan tersedia untuk digunakan secara *remote*.

2. Koleksi *object* yang dikembangkan sebagai *package* untuk diintegrasikan dengan sebuah *framework* komponen seperti .NET dan J2EE.
3. Sistem perangkat lunak yang berdiri sendiri yang dapat dikonfigurasi untuk digunakan dalam lingkungan tertentu [SOM-11:36].

Tipe komponen perangkat lunak yang nantinya akan digunakan dalam skripsi ini yaitu dengan menggunakan koleksi *object* yang dikembangkan sebagai *package* dalam hal ini adalah .NET dan *library* OpenCV. Library OpenCV yang telah memenuhi komponen akan dibungkus (*wrap*) oleh Emgu CV. Rekayasa perangkat lunak *reuse-oriented* memiliki keuntungan jelas dalam mengurangi jumlah perangkat lunak yang dikembangkan dan mengurangi biaya dan resiko [SOM-11:36].

2.5 Pengujian

Pengembangan sistem perangkat lunak melibatkan serangkaian kegiatan produksi di mana peluang untuk keteledoran manusia sangat besar. Kesalahan dapat muncul pada awal proses dimana kemungkinan terjadi kekeliruan pada tujuan atau tujuan tidak terspesifikasi secara tepat. Karena ketidakmampuan manusia dalam membuat sesuatu yang sempurna, maka pengembangan perangkat lunak disertai dengan aktivitas penjaminan kualitas. Pengujian perangkat lunak merupakan elemen penting dari jaminan kualitas perangkat lunak dan merepresentasikan tinjauan utama spesifikasi, desain, dan pembuatan kode [PRE-09:483].

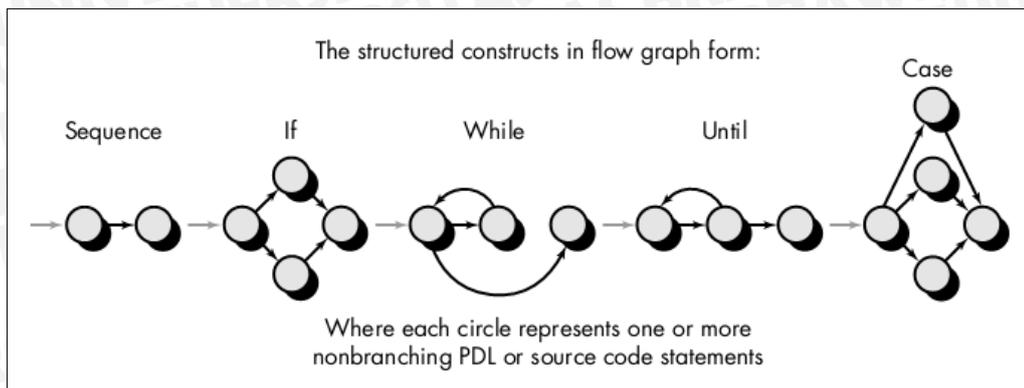
Setiap produk rekayasa dapat diuji dengan salah satu dari dua cara: (1) Dengan mengetahui spesifikasi fungsi dari produk yang dirancang, pengujian dapat dilakukan dengan menunjukkan bahwa setiap fungsi dapat beroperasi dengan baik sementara pada saat yang sama mencari kesalahan dari setiap fungsi, (2) dengan mengetahui cara kerja internal suatu produk, pengujian dapat dilakukan untuk memastikan bahwa setiap operasi internal berjalan sesuai dengan spesifikasi. Pendekatan pertama disebut pengujian *black-box*, dan pendekatan kedua disebut pengujian *white-box* [PRE-09:484].

2.5.1 Pengujian White-Box

Pengujian *white-box* atau pengujian *glass-box* merupakan sebuah metode perancangan kasus uji yang menggunakan struktur kontrol dari perancangan prosedural untuk memperoleh kasus uji [PRE-09:485]. Dengan *white-box* didapat kasus uji yang diperlukan untuk:

1. Menjamin bahwa semua alur independen di dalam modul telah dijalankan minimal sebanyak satu kali.
2. Menguji semua keputusan logis pada setiap percabangan.
3. Menguji semua perulangan pada setiap batas masing-masing perulangan.
4. Menguji struktur data internal untuk memastikan validitasnya.

Ada dua metode pengujian yang termasuk pengujian *white-box* yaitu pengujian *basis path* dan pengujian *control structure*. Dalam skripsi ini, metode pengujian *whitebox* yang digunakan adalah pengujian *basis path*. Metode *basis path* memungkinkan perancang kasus uji untuk memperoleh ukuran kompleksitas logis dari sebuah rancangan prosedural dan menggunakan ukuran ini sebagai pedoman untuk mendefinisikan *basis set* dari jalur eksekusi. Kasus uji dibuat untuk menjamin *basis set* mengeksekusi setiap baris statemen di dalam program minimal satu kali selama pengujian. Metode *basis path* dilakukan dengan membuat notasi yang merepresentasikan aliran kontrol yang disebut dengan *flow graph*. Sebenarnya metode *basis path* dapat dilakukan tanpa menggunakan *flow graph*, namun *flow graph* dapat dijadikan sebagai alat untuk memudahkan pemahaman aliran kontrol dan mengilustrasikan pendekatan metode ini [PRE-09:485]. Ilustrasi *flow graph* ditunjukkan pada Gambar 2.5.



Gambar 2.5 *Flow Graph*

Sumber: [PRE-09:486]

Setelah *flow graph* didefinisikan maka langkah selanjutnya adalah menentukan ukuran kompleksitas (*cyclomatic complexity*). Saat digunakan dalam konteks pengujian *basis path*, nilai untuk *cyclomatic complexity* mendefinisikan jumlah *independent path* dalam *basis set* program dan memberikan batas atas jumlah pengujian yang harus dilakukan untuk memastikan semua statemen telah dieksekusi minimal satu kali. *Independent path* adalah sembarang *path* di dalam program yang memberikan paling sedikit satu set statemen proses baru atau kondisi baru. *Complexity* dapat dihitung dengan beberapa cara, diantaranya [PRE-09:488]:

1. Jumlah *region* pada *flow graph* sesuai dengan *cyclomatic complexity*.
2. *Cyclomatic complexity* $V(G)$, untuk grafik G adalah $V(G) = E - N + 2$, dimana E adalah jumlah *edge*, dan N adalah jumlah *node*.
3. $V(G) = P + 1$, dimana P adalah jumlah *predicate node* yaitu *node* yang merupakan kondisi.

2.5.2 Pengujian Black-Box

Pengujian *black-box* atau pengujian *behavioral* berfokus pada persyaratan fungsional dari perangkat lunak [PRE-01:459]. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak memperoleh rangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian *black-box* bukan alternatif dari teknik *white-box*, tetapi

merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *white-box*.

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori berikut: (1) fungsi-fungsi yang tidak benar atau hilang. (2) kesalahan antarmuka, (3) kesalahan dalam struktur data atau akses *database eksternal*, (4) kesalahan kinerja, dan (5) inisialisasi dan kesalahan terminasi [PRE-09:495]. Tidak seperti pengujian *white-box*, yang dilakukan pada saat awal proses pengujian, pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black-box* memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi.

2.5.3 Pengujian Akurasi

Evaluasi performa dari sebuah model klasifikasi didasarkan pada perhitungan hasil pengujian yang prediksi secara benar dan salah oleh model klasifikasi tersebut. Perhitungan tersebut disusun dalam tabel yang disebut dengan *confusion matrix*. Tabel 2.1 menggambarkan *confusion matrix* dalam permasalahan klasifikasi biner. Setiap masukan f_{ij} dalam tabel ini menunjukkan jumlah hasil dari *class i* yang prediksi sebagai *class j*. Sebagai contoh, f_{01} adalah jumlah hasil dari *class 0* yang salah diprediksi sebagai *class 1*. Berdasarkan masukan dalam *confusion matrix*, jumlah prediksi yang benar adalah $(f_{11} + f_{00})$ dan jumlah prediksi angka yang salah yaitu $(f_{10} + f_{01})$.

Tabel 2.1 *Confusion matrix* untuk permasalahan 2 *class*

		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

Sumber: [TAN-06:149]

Walaupun *confusion matrix* memberikan informasi yang dibutuhkan untuk menentukan seberapa baik performa model klasifikasi, meringkas informasi menjadi sebuah angka akan membuat perbandingan dengan model yang berbeda menjadi lebih baik. Hal ini dapat dilakukan dengan menggunakan sebuah

performance metric seperti akurasi, yang didefinisikan sebagai berikut [TAN-06:149] :

$$\text{accuracy} = \frac{\text{number of correct predictions}}{\text{total number of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

Sama halnya dengan akurasi, performa klasifikasi dapat dilihat dari *error rate*, yang ditunjukkan pada persamaan berikut [TAN-06:149] :

$$\text{error rate} = \frac{\text{number of wrong predictions}}{\text{total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

2.5.4 Pengujian Usability

Pengujian *usability* adalah pengujian terhadap atribut kualitas yang menilai seberapa mudahnya *user interface* yang digunakan dalam pengembangan suatu perangkat lunak [JAK-03:1]. Selain itu, pengujian *usability* juga digunakan untuk mendapatkan *feedback* mengenai ketepatan tujuan dari pembuatan sistem. Teknik kuisisioner dilakukan untuk mendukung pengujian ini. Sampel yang digunakan dalam pengujian ini sebanyak 10 sampel. Pernyataan yang diajukan dalam kuisisioner ini menggunakan pernyataan-pernyataan dari IBM Computer Usability Satisfaction Questionnaires [LEW-93] dengan memodifikasi pilihan menjadi tiga pilihan, yaitu setuju, cukup, dan tidak setuju. Pernyataan-pernyataan tersebut diambil karena dirasa sudah sesuai dan relevan dengan parameter pengujian *usability* yang akan dilakukan dalam skripsi ini.

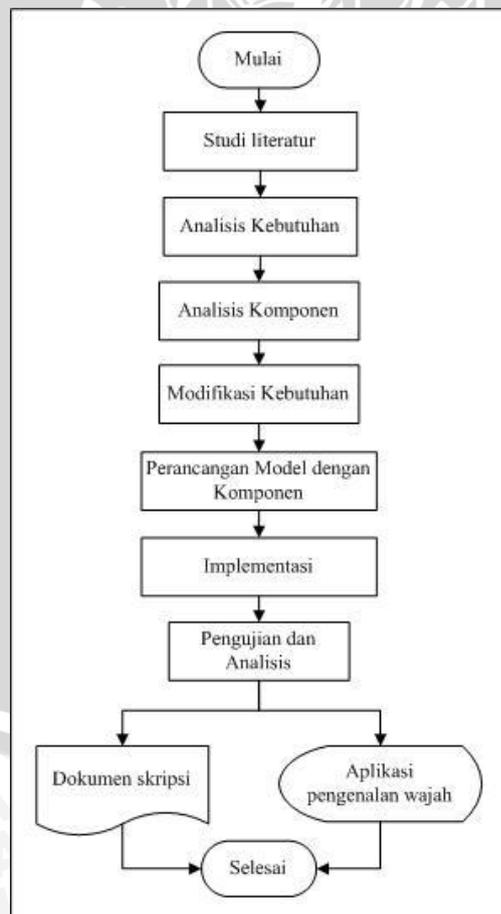
BAB III

METODE PENELITIAN DAN PERANCANGAN

Bab ini membahas mengenai metode penelitian skripsi dan perancangan Sistem Manajemen Pengunjung. Metode penelitian berisi langkah-langkah yang digunakan dalam penelitian skripsi yang terdiri dari studi literatur untuk dasar teori, metode yang digunakan dalam perancangan, serta pengujian dan analisis. Sedangkan dalam perancangan menjelaskan analisis kebutuhan perangkat lunak dan perancangan lunak.

3.1 Metode Penelitian

Berikut diagram alir runtutan pengerjaan skripsi secara umum dengan menggunakan *reuse-oriented model* seperti ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram alir runtutan pengerjaan skripsi secara umum (adaptasi dari [SOM-11:35])

Sumber : Perancangan

3.1.1 Studi Literatur

Studi literatur menjelaskan kajian pustaka dan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut meliputi:

- a. Pengenalan Wajah
- b. Emgu CV
 - OpenCV
- c. *Reuse-Oriented Model* SDLC
- d. Pengujian Perangkat Lunak
 - Pengujian *Black-box*
 - Pengujian *White-box*
 - Pengujian Akurasi

3.1.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan dengan menentukan kebutuhan apa saja yang dibutuhkan untuk membangun aplikasi Sistem Manajemen Pengunjung. Metode analisis yang digunakan adalah OOAD (*Object Oriented Analysis and Design*) dengan menggunakan bahasa pemodelan UML (*Unified Modelling Language*). Pada tahap analisis kebutuhan digunakan pemodelan dengan menggunakan diagram *use case*. Diagram *use case* digunakan untuk menggambarkan fungsionalitas perangkat lunak.

3.1.3 Analisis Komponen

Analisis komponen dilakukan dengan cara mencari komponen – komponen yang dapat mengimplemetasi kebutuhan yang sudah ditentukan. Setelah komponen-komponen tersebut ditemukan, komponen tersebut dipetakan sesuai dengan fungsi-fungsi yang sesuai dengan kebutuhan.

3.1.4 Modifikasi Kebutuhan

Pada proses modifikasi kebutuhan, dari komponen-komponen yang telah ditemukan dianalisis kesesuaiannya dengan spesifikasi kebutuhan. Kebutuhan dimodifikasi untuk merefleksikan komponen yang tersedia. Jika modifikasi

kebutuhan tidak sesuai untuk diterapkan pada aplikasi, maka proses analisis komponen dapat dilakukan kembali.

3.1.5 Perancangan Model dengan Komponen

Perancangan perangkat lunak dilakukan setelah menentukan kebutuhan untuk membangun sistem. Komponen yang digunakan untuk membangun sistem salah satunya yaitu dengan menggunakan aplikasi *framework* .NET yang merupakan komponen umum dalam sistem aplikasi ini. Selain itu, komponen Emgu CV (OpenCV) yang berperan sebagai *library* dalam proses pengenalan wajah. Perancangan perangkat lunak berdasarkan *object-oriented analysis* dan *object-oriented design* yaitu menggunakan pemodelan UML (*Unified Modeling Language*). Pada tahap perancangan digunakan pemodelan dengan menggunakan dua macam diagram, yaitu *class diagram* dan diagram sekuensial. *Class diagram* digunakan untuk mengidentifikasi kelas-kelas dan antarmuka yang dibutuhkan pada sistem. Sedangkan diagram sekuensial digunakan untuk menggambarkan hubungan interaksi antar elemen (objek) yang telah diidentifikasi dan dimodelkan pada sistem yang disusun dalam urutan waktu. Perancangan Sistem Manajemen Pengunjung ini secara garis besar meliputi:

- Pemodelan kelas (*class diagram*).
- Pemodelan sekuensial (*sequence diagram*).
- Perancangan basis data (*database*).
- Perancangan antarmuka pengguna (*user interface*).

3.1.6 Implementasi

Implementasi aplikasi dilakukan dengan mengacu pada perancangan aplikasi. Tahapan implementasi Sistem Manajemen Pengunjung yaitu:

1. Pembuatan *database* menggunakan Microsoft SQL Server 2008 *Database Management System*.
2. Pembuatan dan pengembangan sistem aplikasi menggunakan Microsoft Visual Studio 2010.
3. Bahasa pemrograman yang digunakan adalah bahasa C#.

4. Integrasi *visual control* yang disediakan Microsoft Visual Studio 2010 untuk membangun *user interface*.
5. Integrasi komponen Emgu CV ke dalam aplikasi untuk membangun fungsi deteksi dan pengenalan wajah.

3.1.7 Pengujian dan Analisis

Pengujian perangkat lunak pada skripsi ini dilakukan agar dapat menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan yang melandasinya. Dalam pengujian ini, strategi pengujian yang digunakan meliputi pengujian unit, pengujian integrasi dan pengujian validasi. Pengujian dilakukan dengan menggunakan metode pengujian *white box* yang diterapkan pada pengujian unit dan pengujian integrasi, sedangkan pengujian *black box* diterapkan pada pengujian validasi. Pengujian unit digunakan untuk mencari jumlah jalur pada method-method dasar yang di dalamnya tidak terdapat operasi memanggil method lain. Sedangkan pengujian integrasi digunakan untuk mencari jumlah jalur pada beberapa method yang lolos uji pada pengujian unit. Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan.

Selain itu untuk menguji performa pengenalan wajah dari sistem, maka dilakukan pengujian akurasi. Pengujian akurasi menguji tingkat akurasi dan *error rate* pengenalan wajah dengan menggunakan *confusion matrix* untuk menentukan performa model pengklasifikasian. Pengujian akurasi dilakukan dengan menguji 20 orang yang berbeda. Dari 20 orang tersebut, dilakukan pelatihan (*training*) terhadap 10 orang, kemudian diuji apakah 10 orang tersebut dapat dikenali dengan benar atau tidak oleh sistem. Sepuluh orang lainnya tidak dilakukan pelatihan terlebih dahulu, kemudian diuji apakah 10 orang tersebut berhasil disimpan dan dikenali sebagai *unknown* atau sebagai wajah yang telah dilatih sebelumnya oleh sistem.

Dari pengujian terhadap 20 orang tersebut dihitung akurasi dan *error rate* yang diperoleh. Pengambilan data latih dilakukan pada tempat yang sama dengan rata-rata tingkat keabuan lebih besar sama dengan 100 untuk skala 0 sampai 255. Setiap orang memiliki 7 data latih dengan 7 pose wajah yang berbeda. Gambar 3.2

merupakan contoh pose wajah yang akan digunakan sebagai data latih. Untuk proses pengujian, dilakukan sebanyak 1 kali dengan pose wajah bebas yang dapat dideteksi dan dikenali oleh sistem untuk setiap orang.



Gambar 3.2 Contoh Pose Wajah Data Latih
Sumber : [CUI-12:6]

Contoh pengujian akurasi yang akan dilakukan adalah sebagai berikut: diasumsikan terdapat 5 orang dengan citra wajah yang telah dilatih yaitu Andi, Didik, Ari, Tono, dan Jalu. Proses pengambilan data latih dilakukan dengan menyimpan citra wajah kelima orang tersebut ke dalam *database* dengan identitas masing-masing. Kelima identitas orang tersebut dalam pengujian akurasi disebut sebagai *class* atau kategori. Contoh identitas yang digunakan adalah nama masing-masing orang. Pengujian dilakukan sebanyak satu kali pada setiap orang. Masing-masing orang akan diprediksi identitasnya oleh sistem. Hasil prediksi sistem ditulis ke dalam *confusion matrix* yang ditunjukkan pada Tabel 3.1.

Tabel 3.1 Contoh hasil prediksi

		Kelas Prediksi				
		Andi	Didik	Ari	Tono	Jalu
Kelas Sebenarnya	Andi		1			
	Didik		1			
	Ari			1		
	Tono				1	
	Jalu					1

Dari data *confusion matrix*, dapat dilihat bahwa total pengujian adalah 5. Sistem memprediksi dengan benar untuk identitas dari Didik, Ari, Tono, dan Jalu, sedangkan Andi tidak diprediksi dengan benar karena sistem memprediksi Andi sebagai Didik. Kemudian dihitung akurasinya menggunakan rumus sebagai berikut:

$$accuracy = \frac{\text{number of correct predictions}}{\text{total number of predictions}} = \frac{4}{5} = 0,8 = 80\%$$

Error rate juga dihitung menggunakan rumus sebagai berikut:

$$\text{error rate} = \frac{\text{number of wrong predictions}}{\text{total number of predictions}} = \frac{1}{5} = 0,2 = 20\%$$

Dari hasil perhitungan akurasi dan *error rate* yang telah dilakukan dari contoh, dapat disimpulkan bahwa pada setiap proses pengenalan wajah oleh sistem terdapat 80% kemungkinan akan diprediksi dengan benar atau 20% kemungkinan akan diprediksi dengan salah.

Pengujian terakhir adalah pengujian *usability*. Pengujian *usability* menggunakan kuisioner dengan pernyataan-pernyataan didapatkan dari IBM Computer Satisfaction Questionnaires. Kuisioner akan diserahkan kepada target pengguna dari sistem ini, yaitu laboran dan orang-orang yang bekerja di dalam laboratorium.

3.2 Analisis Kebutuhan dan Perancangan Perangkat Lunak

Untuk mengembangkan perangkat lunak, tahap perancangan dilakukan dengan cara menganalisis kebutuhan perangkat lunak dan merancang perangkat lunak sesuai dengan kebutuhan.

3.2.1 Analisis Kebutuhan

Proses analisis kebutuhan diawali dengan penjabaran gambaran umum Sistem Manajemen Pengunjung, identifikasi aktor, analisis data yang akan disimpan, penjabaran tentang daftar kebutuhan yang kemudian memodelkannya ke dalam diagram *use case*, analisis komponen dan modifikasi kebutuhan. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan – kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

Kebutuhan yang digunakan dalam pembuatan aplikasi pengenalan wajah ini meliputi:

1. Kebutuhan Hardware, meliputi:
 - Komputer PC
 - Webcam

2. Kebutuhan Software, meliputi:
 - Microsoft Windows sebagai sistem operasi
 - Microsoft SQL Server 2008 *Database Management System*
 - Microsoft Visual Studio 2010 sebagai platform pengembangan
 - StarUML, Visual Paradigm dan ModelMaker11 C# Edition sebagai *tool* pembuatan diagram pemodelan sistem
3. Data yang dibutuhkan meliputi:
 - Data latih yaitu citra wajah pengunjung yang disimpan dalam *database* untuk pengenalan wajah.
 - Data masukan yaitu citra wajah pengunjung tertangkap oleh *webcam*.

3.2.1.1 Gambaran Umum Sistem

Pembahasan gambaran umum Sistem Manajemen Pengunjung Laboratorium PTIHK Berbasis Pengenalan Wajah terdiri atas dua bagian, yaitu deskripsi umum sistem dan cara kerja sistem.

1. Deskripsi Umum Sistem

Sistem perangkat lunak yang dikembangkan dalam skripsi ini yaitu Sistem Manajemen Pengunjung Laboratorium PTIHK Berbasis Pengenalan Wajah. Sistem ini berfungsi untuk mengembangkan sistem pemantau kejadian yang masih konvensional yang mana citra hasil tangkapan kamera tersimpan dalam bentuk rekaman. Selain memantau keadaan sekitar laboratorium, sistem ini juga dapat mengidentifikasi dan mencatat data pengunjung yang datang ke laboratorium, lalu menyimpannya ke dalam *database*. Sehingga, sistem perangkat lunak ini diharapkan dapat mengurangi kerja laboran atau pemantau dalam hal mencatat data pengunjung yang selalu mengisi *form* yang sudah disediakan ketika masuk ke dalam laboratorium. Sistem Manajemen Pengunjung ini memiliki beberapa fungsi, yaitu:

a. *Login*

Operasi ini bertujuan untuk memberi seleksi kepada pengguna sehingga terlihat perbedaan antara pemantau dan *administrator*.

b. Mengelola data pengunjung

Operasi ini bertujuan untuk melakukan manipulasi data yang disimpan pada tabel pengunjung pada *database*. Operasi ini terdiri atas beberapa operasi, yaitu:

1. Menambah data pengunjung.
2. Menghapus data pengunjung.
3. Melakukan *edit* data pengunjung.
4. Mencari data pengunjung.

c. Mengelola data wajah

Operasi ini bertujuan untuk melakukan manipulasi data yang disimpan pada tabel Wajah pada *database*. Operasi ini terdiri atas beberapa operasi, yaitu:

1. Menambah data wajah.
2. Menghapus data wajah.

d. Mengelola data pengguna

Operasi ini bertujuan untuk melakukan manipulasi data yang disimpan pada tabel Pengguna pada *database*. Operasi ini terdiri atas beberapa operasi, yaitu:

1. Menambah data pengguna.
2. Menghapus data pengguna.
3. Melakukan mengubah pengguna.

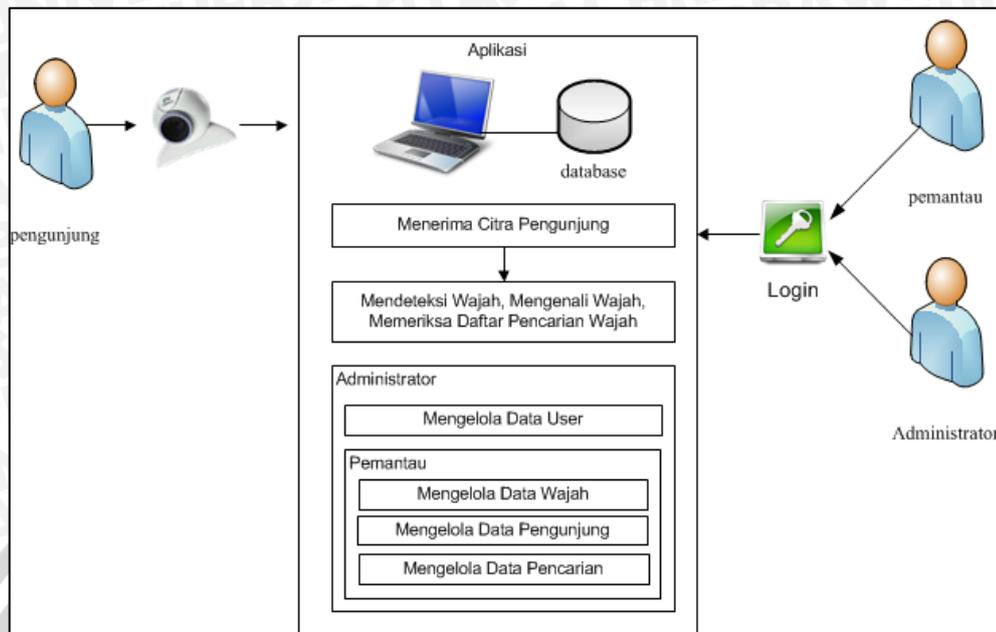
e. Mengelola data pencarian

Operasi ini bertujuan untuk melakukan manipulasi data yang disimpan pada tabel Pencarian pada *database*. Operasi ini terdiri atas beberapa operasi, yaitu:

1. Menambah pencarian pengunjung.
2. Menghapus pencarian pengunjung.

2. Cara Kerja Sistem

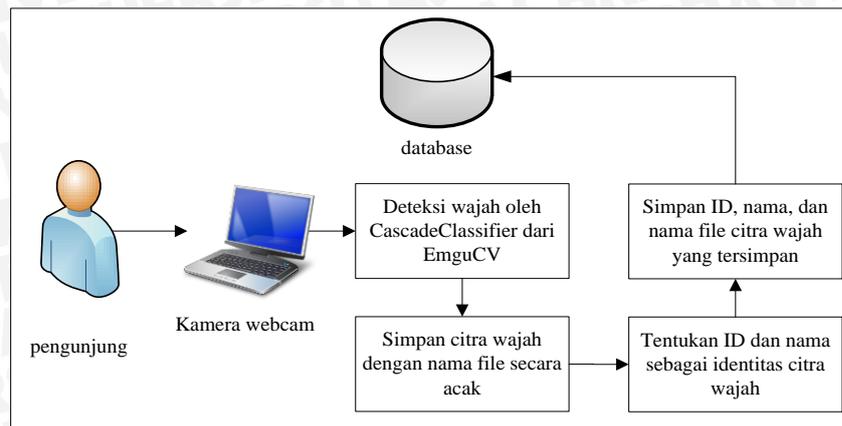
Cara kerja Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah secara umum ditunjukkan pada Gambar 3.3.



Gambar 3.3 Cara Kerja Sistem Secara Umum
Sumber: Perancangan

Penjelasan gambar cara kerja Sistem Manajemen Pengunjung yaitu sebagai berikut :

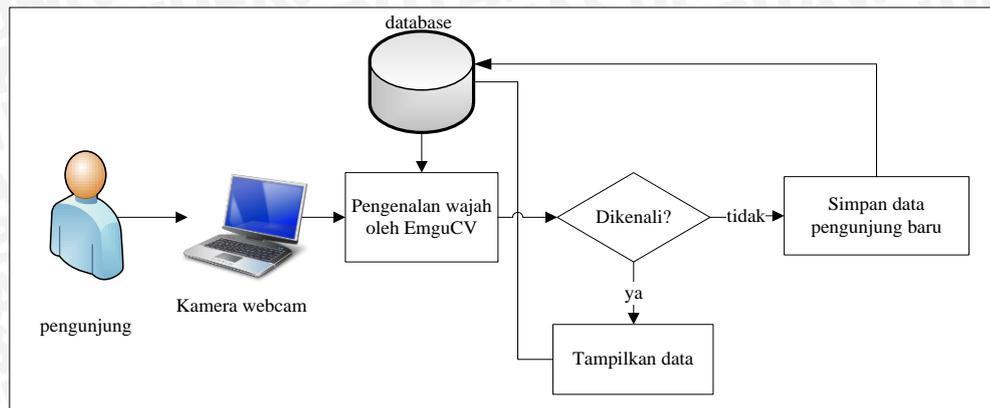
1. Pemantau atau administrator login ke dalam sistem.
2. Di dalam sistem, pemantau dan *administrator* dapat melakukan pengelolaan data wajah, data pengunjung, dan data pencarian. Pemantau dan *administrator* juga dapat mengubah preferensi aplikasi, melihat pengunjung-pengunjung yang masuk melalui *log*, dan menerima pemberitahuan dari sistem jika sistem menemukan pengunjung yang dicari. *Administrator* dapat mengelola pengguna untuk mengatur siapa saja pengguna yang bisa *login* ke dalam sistem.
3. Setelah pemantau atau administrator *login* ke dalam sistem, maka sistem siap untuk menangkap citra pengunjung dan menjalankan proses manajemen pengunjung.
4. Pengunjung masuk ke dalam laboratorium.
5. Kamera *webcam* menangkap citra wajah pengunjung.
6. Aplikasi sistem menerima citra pengunjung, kemudian melakukan deteksi wajah, kemudian melakukan pengenalan wajah. Jika wajah dikenali, maka sistem akan memeriksa daftar pencarian.



Gambar 3.4 Proses Pelatihan
Sumber: Perancangan

Proses di dalam aplikasi pengenalan wajah untuk pencatatan data pengunjung laboratorium PTIIK dibagi menjadi 2, yaitu proses pelatihan dan proses pengujian. Proses pelatihan adalah proses mengambil citra wajah baru sebagai data latih dan menyimpan data yang bersangkutan dengan citra wajah tersebut ke dalam *database*. Proses pengujian adalah cara kerja aplikasi dalam mengenali wajah melalui citra yang ditangkap oleh kamera. Proses pelatihan ditunjukkan pada Gambar 3.4, dan proses pengujian aplikasi ditunjukkan pada Gambar 3.5. Langkah-langkah proses pelatihan yaitu:

1. Pengunjung berada di depan kamera untuk pengambilan gambar.
2. Dari citra pengunjung yang tertangkap kamera, dilakukan deteksi wajah untuk mengambil citra wajah dari pengunjung.
3. Setelah citra wajah didapatkan, citra wajah tersebut disimpan dengan nama *file* yang berbeda secara acak sehingga tidak terjadi kesamaan nama *file* dengan *file* yang disimpan sebelumnya. File tersebut secara otomatis akan tersimpan dalam sebuah folder yang berada di dalam aplikasi, yaitu folder “wajah”.
4. Menentukan ID dan nama yang akan menjadi identitas citra wajah tersebut.
5. Menyimpan ID, nama, dan nama *file* citra wajah tersebut ke dalam *database*.



Gambar 3.5 Proses Pengujian
Sumber: Perancangan

Setelah dilakukan proses penambahan data latih, aplikasi akan melakukan pelatihan ulang untuk mengupdate *classifier*. Selanjutnya proses pengujian dilakukan jika terdapat data latih dan telah dilakukan proses pelatihan. Langkah-langkah proses pengujian yaitu:

1. Saat pengunjung berada di depan kamera, maka kamera akan menangkap citra wajah pengunjung. Kamera yang aktif ditentukan oleh pemakai aplikasi. Jika terdapat lebih dari satu kamera yang terpasang pada sistem komputer, pemakai dapat memilih kamera mana yang akan diaktifkan.
2. Citra wajah yang ditangkap kemudian dilakukan proses pengenalan wajah oleh EmguCV. Proses pengenalan wajah akan memberikan nilai sebuah ID jika dikenali. Dari ID yang didapat, kemudian dilakukan pencarian data pengunjung di dalam *database* berdasarkan ID tersebut.
3. Jika pengunjung dikenali, maka aplikasi akan menampilkan data pengunjung tersebut sesuai dengan data yang ada di dalam *database*. Aplikasi juga akan menampilkan data kejadian atau waktu kapan pengunjung tersebut datang ke laboratorium. Data kejadian tersebut secara otomatis akan tersimpan ke dalam tabel “log” yang ada di dalam *database*.
4. Jika pengunjung tidak dikenali, maka dilakukan penambahan data latih yaitu wajah pengunjung yang tidak dikenali dengan nama “unknown”.
5. Proses pengujian akan selalu berjalan saat aplikasi dalam keadaan aktif. Setiap citra yang ditangkap oleh kamera selalu diproses oleh aplikasi.

3.2.1.2 Identifikasi Aktor

Tahap ini merupakan tahap untuk melakukan identifikasi terhadap aktor - aktor yang akan berinteraksi dengan perangkat lunak Sistem Manajemen Pengunjung. Pada sistem ini terdapat dua aktor yang berinteraksi secara langsung terhadap sistem dan satu aktor yang secara tidak langsung berinteraksi dengan sistem. Aktor yang secara langsung berinteraksi dengan sistem yaitu *administrator* dan pemantau yang bertindak sebagai pemantau. Sedangkan aktor yang secara tidak langsung berinteraksi terhadap sistem yaitu pengunjung. Penjelasan masing-masing aktor ditunjukkan pada Tabel 3.2.

Tabel 3.2 Identifikasi Aktor

Aktor	Deskripsi Aktor
<i>Administrator</i>	<i>Administrator</i> merupakan pengguna aplikasi perangkat lunak Sistem Manajemen Pengunjung yang memiliki hak untuk mengakses semua modul yang terdapat di dalam sistem.
Pemantau	Pemantau merupakan pengguna aplikasi perangkat lunak Sistem Manajemen Pengunjung yang bertugas memantau pengunjung dan hanya bisa mengakses beberapa modul yang terdapat dalam sistem. Seperti mengelola data pengunjung, mengelola data wajah, dan mengelola data <i>log</i> .
Pengunjung	Pengunjung merupakan pengguna yang memberikan data latih terhadap sistem yang berupa citra wajah.

Sumber: Perancangan

3.2.1.3 Analisis Data

Analisis data bertujuan untuk mendapatkan struktur penyimpanan data yang dibutuhkan perangkat lunak Sistem Manajemen Pengunjung. Struktur penyimpanan data pada perangkat lunak Sistem Manajemen Pengunjung disusun berdasarkan analisis data sebagai berikut :

- Data pengunjung yang terdiri dari id, nama, alamat, telepon, email, dan jabatan.
- Data wajah yang terdiri atas id, label, dan *file*.
- Data pengguna yang terdiri dari *username*, *password* dan *class*.
- Data *log* yang terdiri atas id, waktu_masuk, id_pengunjung, dan *file_unknown*.
- Data pencarian yang terdiri dari id dan id_pengunjung.

3.2.1.4 Daftar Kebutuhan

Daftar kebutuhan terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional. Pada daftar kebutuhan fungsional akan dispesifikasikan menjadi tiga yaitu spesifikasi kebutuhan fungsional *administrator*, spesifikasi kebutuhan fungsional pemantau, dan spesifikasi kebutuhan fungsional pengunjung.

Administrator dan pemantau memiliki spesifikasi kebutuhan fungsional yang sama. Kebutuhan yang membedakan antara *administrator* dan pemantau adalah mengelola data pengguna. Hanya *administrator* yang memiliki hak untuk mengelola data pengguna. Tabel 3.3 menjelaskan spesifikasi kebutuhan pengguna yaitu spesifikasi kebutuhan yang dapat dilakukan oleh kedua pengguna yaitu *administrator* dan pemantau. Sedangkan Tabel 3.4 menjelaskan spesifikasi kebutuhan yang hanya dapat dilakukan oleh *administrator* dan Tabel 3.5 menjelaskan spesifikasi kebutuhan yang dilakukan oleh pengunjung.

Tabel 3.3 Spesifikasi Kebutuhan Pemantau dan *Administrator*

Nomor SRS	Kebutuhan	Use Case
SRS_001_01	Perangkat lunak harus menyediakan fasilitas untuk <i>login</i> sehingga terlihat perbedaan antara pemantau dan <i>administrator</i> .	<i>Login</i>
SRS_001_02	Perangkat lunak harus menyediakan fasilitas untuk mengelola data pengunjung.	Mengelola Data Pengunjung
SRS_001_03	Perangkat lunak harus menyediakan fasilitas untuk mengelola data wajah.	Mengelola Data Wajah
SRS_001_04	Perangkat lunak harus menyediakan fasilitas untuk melakukan pengolahan data daftar pencarian.	Mengelola Data Daftar Pencarian Pengunjung

Sumber: Perancangan

Tabel 3.4 Spesifikasi Kebutuhan *Administrator*

Nomor SRS	Kebutuhan	Use Case
SRS_002_01	Perangkat lunak harus menyediakan fasilitas untuk melakukan pengolahan data pengguna.	Mengelola Data pengguna

Sumber: Perancangan

Tabel 3.5 Spesifikasi Kebutuhan Pengunjung

Nomor SRS	Kebutuhan	Use Case
SRS_003_01	Perangkat lunak harus menyediakan fasilitas untuk menangkap dan mendeteksi citra wajah pengunjung.	Menghadap Kamera.

Sumber: Perancangan

Daftar kebutuhan non-fungsional Sistem Manajemen Pengunjung ditunjukkan pada Tabel 3.6.

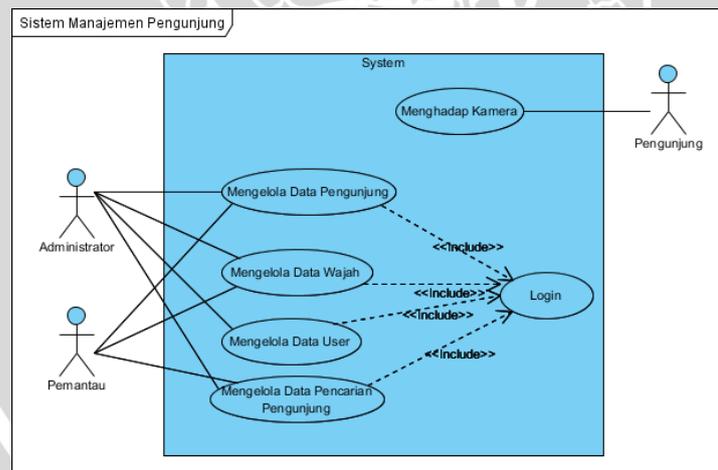
Tabel 3.6 Spesifikasi Kebutuhan Non-Fungsional

Parameter	Deskripsi Kebutuhan
<i>Performance</i>	Akurasi dari modul pengenalan wajah dari OpenCV yang digunakan sistem. Akurasi dari sistem setidaknya memiliki nilai 70% sebagai prototipe awal sistem dan dapat dikembangkan lagi untuk penelitian-penelitian selanjutnya.
<i>Usability</i>	Sistem harus dapat dengan mudah digunakan oleh pengguna khususnya pemantau. Selain itu, pengguna setidaknya dapat terbantuan dengan adanya sistem yang dikembangkan. Tingkat <i>usability</i> yang diperlukan kurang lebih 70% yang berkategori baik.

Sumber: Perancangan

3.2.1.5 Diagram Use Case

Diagram *use case* adalah salah satu diagram untuk memodelkan aspek perilaku sistem. Diagram *use case* menunjukkan sekumpulan *use case*, aktor, dan hubungannya. *Use case* merupakan fungsionalitas dari sistem yang diinisialisasi oleh aktor. Diagram *use case* yang menggambarkan fungsionalitas perangkat lunak Sistem Manajemen Pengunjung ditunjukkan pada Gambar 3.6.



Gambar 3.6 Diagram *Use Case* Sistem Manajemen Pengunjung

Sumber: Perancangan

1. Spesifikasi Use Case Login

Tabel 3.7 *Use Case Login*

Skenario Kasus Pada Sistem

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	SRS_001_01

Nama	<i>Login</i>
Deskripsi	<i>Use case</i> ini menjelaskan proses melakukan <i>login</i> .
Tujuan	Untuk menyeleksi <i>administrator</i> dan pemantau yang sah.
Aktor	<i>Administrator</i> dan pemantau.
Skenario Utama	
Kondisi Awal	Aplikasi Sistem Manajemen Pengunjung telah berjalan dan sistem menampilkan halaman <i>Login</i> .
Aksi Aktor	Reaksi Sistem
Aktor memasukkan <i>username</i> dan <i>password</i> , kemudian menekan tombol "Login".	<ol style="list-style-type: none"> 1. Sistem menerima data login dan kemudian sistem melakukan pengecekan terhadap data yang dimasukkan oleh aktor. Jika data benar, maka sistem akan menampilkan halaman utama aplikasi. 2. Jika pengguna adalah <i>administrator</i>, halaman utama menampilkan menu untuk <i>administrator</i>. Sedangkan jika pengguna adalah pemantau, halaman utama menampilkan menu untuk pemantau.
Skenario Alternatif 1	
Aktor memasukkan <i>username</i> atau <i>password</i> kosong, kemudian menekan tombol "Login".	Sistem menampilkan pesan peringatan "Username atau Password kosong".
Skenario Alternatif 2 : Jika <i>username</i> atau <i>password</i> salah	
Aktor memasukkan <i>username</i> atau <i>password</i> salah, kemudian menekan tombol "Login".	Sistem menampilkan pesan peringatan "Username atau Password salah".
Kondisi Akhir	Sistem menampilkan halaman utama sesuai dengan pengguna.

Sumber: Perancangan

2. Spesifikasi Use Case Mengelola Data Pengunjung

Tabel 3.8 *Use Case* Mengelola Data Pengunjung

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	SRS_001_02
Nama	Mengelola Data Pengunjung
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana pengguna dapat melakukan pengelolaan data pengunjung (menambah, mengubah dan menghapus).
Tujuan	Untuk melakukan pengelolaan data

	pengunjung laboratorium.
Aktor	<i>Administrator</i> dan pemantau
Kondisi Awal	Aplikasi Sistem Manajemen Pengunjung menampilkan halaman Data Pengunjung.
Skenario Menambah Data: Skenario Utama	
Aksi Aktor	Reaksi Sistem
Aktor menekan tombol “Data Baru”.	1. <i>Textbox</i> isian data menjadi kosong dan dapat di- <i>edit</i> . 2. Tampil tombol “Simpan” dan “Batal”.
Aktor mengisi data kemudian menekan tombol “Simpan”.	Data tersimpan dalam <i>database</i> dan tampil pada tampilan <i>list</i> .
Skenario Menambah Data: Skenario Alternatif 1	
Aktor menekan tombol “Simpan” dengan salah satu isian yang kosong	Sistem menampilkan pesan peringatan “Semua data harus diisi”.
Skenario Menambah Data: Skenario Alternatif 2	
Aktor menekan tombol “Batal”.	1. <i>Textbox</i> isian data tidak dapat di- <i>edit</i> . 2. Tombol “Simpan” dan “Batal” menghilang.
Skenario Edit Data: Skenario Utama	
Aktor memilih data yang sudah ada pada <i>list</i> .	1. Jika terdapat data yang dipilih, tombol “Edit Data” dan “Hapus Data” aktif. 2. <i>Textbox</i> isian data terisi detail dari data yang dipilih.
Aktor menekan tombol “Edit Data”.	1. <i>Textbox</i> isian data dapat di- <i>edit</i> 2. Tampil tombol “Simpan” dan “Batal”.
Aktor melakukan <i>edit</i> data kemudian menekan tombol “Simpan”	Data tersimpan dalam <i>database</i> dan tampil pada tampilan <i>list</i> .
Skenario Edit Data: Skenario Alternatif 1	
Aktor menekan tombol “Simpan” dengan mengubah salah satu isian <i>textbox</i> menjadi kosong.	Sistem menampilkan pesan peringatan “Semua data harus diisi”.
Skenario Edit Data: Skenario Alternatif 2	
Aktor menekan tombol “Batal”.	1. <i>Textbox</i> isian data tidak dapat di- <i>edit</i> . 2. Tombol “Simpan” dan “Batal” menghilang.
Skenario Menghapus Data: Skenario Utama	
Aktor memilih data yang sudah ada pada <i>list</i> .	1. Jika terdapat data yang dipilih, tombol “Edit Data” dan “Hapus Data” aktif. 2. <i>Textbox</i> isian data terisi detail dari data yang dipilih.
Aktor menekan tombol “Hapus Data”.	Sistem menampilkan pesan konfirmasi penghapusan data.
Aktor menekan tombol “Yes” pada pesan konfirmasi.	Data terhapus dari <i>database</i> dan hilang

	dari tampilan <i>list</i> .
Skenario Menghapus Data: Skenario Alternatif	
Aktor menekan tombol “No” pada pesan konfirmasi	Sistem akan kembali ke halaman data pengunjung.
Kondisi Akhir	Sistem akan menyimpan setiap perubahan yang dilakukan oleh pengguna.

Sumber: Perancangan

3. Spesifikasi Use Case Mengelola Data Wajah

Tabel 3.9 Use Case Mengelola Data Wajah

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	SRS_001_03
Nama	Mengelola Data Wajah
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana pengguna dapat melakukan pengelolaan data wajah pengunjung (menambah dan menghapus).
Tujuan	Untuk melakukan pengelolaan data wajah pengunjung laboratorium.
Aktor	<i>Administrator</i> dan pemantau
Kondisi Awal	Aplikasi Sistem Manajemen Pengunjung menampilkan halaman utama.
Skenario Menambah Data Wajah: Skenario Utama	
Aksi Aktor	Reaksi Sistem
Aktor membuka halaman data pengunjung.	Sistem menampilkan data pengunjung.
Aktor memilih data yang sudah ada pada <i>list</i> .	Jika terdapat data yang dipilih, tombol “Training wajah untuk data ini” aktif.
Aktor menekan tombol “Training wajah untuk data ini”.	Sistem menampilkan halaman <i>Training Wajah</i> dan terdapat layar yang sudah tersambung dengan <i>webcam</i> .
Aktor menekan tombol “Simpan” saat wajah terdeteksi pada layar.	Data wajah tersimpan atas nama data pengunjung yang dipilih sebelumnya.
Skenario Menambah Data Wajah: Skenario Alternatif 1	
Aktor menekan tombol “Simpan” saat wajah tidak terdeteksi pada layar.	Sistem menampilkan pesan “Tidak ada wajah yang terdeteksi”.
Skenario Menambah Data Wajah: Skenario Alternatif 2	
Aktor membuka halaman utama dan memilih salah satu data <i>unknown</i> .	Tombol “Tambahkan ke data training” aktif
Aktor menekan tombol “Tambahkan ke data training”	Tampil halaman <i>Training Data Unknown</i> .
Aktor memilih salah satu pengunjung	Data tersimpan dalam <i>database</i> .



yang terdaftar, kemudian menekan tombol “Simpan”	
Skenario Menghapus Data Wajah : Skenario Utama	
Aktor memilih data yang sudah ada pada <i>list</i> .	Jika terdapat data yang dipilih, tombol “Hapus Wajah” aktif.
Aktor menekan tombol “Hapus Wajah”.	Sistem menampilkan pesan konfirmasi penghapusan data.
Aktor menekan tombol “Yes” pada pesan konfirmasi.	Data terhapus dari <i>database</i> dan hilang dari tampilan <i>list</i> .
Skenario Menghapus Data Wajah: Skenario Alternatif	
Aktor menekan tombol “No” pada pesan konfirmasi.	Sistem akan kembali ke halaman data pengunjung.
Kondisi Akhir	Sistem akan menyimpan setiap perubahan yang dilakukan oleh pengguna.

Sumber: Perancangan

4. Spesifikasi Use Case Mengelola Data Pencarian Pengunjung

Tabel 3.10 Use Case Mengelola Data Pencarian Pengunjung

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	SRS_001_04
Nama	Mengelola Data Pencarian Pengunjung
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana pengguna dapat melakukan pengelolaan data pencarian pengunjung (menambah dan menghapus).
Tujuan	Untuk melakukan pengelolaan data pencarian pengunjung laboratorium.
Aktor	<i>Administrator</i> dan pemantau
Kondisi Awal	Aplikasi Sistem Manajemen Pengunjung menampilkan halaman Data Pencarian Pengunjung.
Skenario Menambah Data: Skenario Utama	
Aksi Aktor	Reaksi Sistem
Aktor menekan tombol “Pencarian Baru”.	Menampilkan <i>form</i> daftar pengunjung yang akan dicari.
Aktor memilih data pengunjung dan menekan tombol “OK”.	<ol style="list-style-type: none"> <i>Form</i> daftar pengunjung menutup dan kembali ke halaman daftar pencarian pengunjung. Data pencarian pengunjung tersimpan dalam <i>database</i> dan tampil pada <i>list</i>.
Skenario Menambah Data: Skenario Alternatif	
Aktor menekan tombol “Batal”.	Sistem kembali ke halaman daftar pencarian pengunjung.

Skenario Menghapus Data : Skenario Utama	
Aktor memilih data yang sudah ada pada <i>list</i> .	Jika terdapat data yang dipilih, tombol “Hapus Pencarian” aktif.
Aktor menekan tombol “Hapus Pencarian”.	Sistem menampilkan pesan konfirmasi penghapusan data.
Aktor menekan tombol “Yes” pada pesan konfirmasi.	Data terhapus dari <i>database</i> dan hilang dari tampilan <i>list</i> .
Skenario Menghapus Data: Skenario Alternatif	
Aktor menekan tombol “No” pada pesan konfirmasi.	Sistem akan kembali ke halaman daftar pencarian pengunjung.
Kondisi Akhir	Sistem akan menyimpan setiap perubahan yang dilakukan oleh pengguna.

Sumber: Perancangan

5. Spesifikasi Use Case Mengelola Data Pengguna

Tabel 3.11 Use Case Mengelola Data Pengguna

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_002_01
Nama	Mengelola Data Pengguna
Deskripsi	Use case ini menjelaskan bagaimana administrator dapat melakukan pengelolaan data pengguna (menambah, mengubah, dan menghapus).
Tujuan	Untuk melakukan pengelolaan data pengguna.
Aktor	Administrator
Kondisi Awal	Aplikasi Sistem Manajemen Pengunjung menampilkan halaman Daftar Pengguna.
Skenario Menambah Data Pengguna: Skenario Utama	
Aksi Aktor	Reaksi Sistem
Aktor menekan tombol “Data Baru”.	1. <i>Textbox</i> isian data menjadi kosong dan dapat diubah. 2. Tampil tombol “Simpan” dan “Batal”.
Aktor mengisi data kemudian menekan tombol “Simpan”.	Data tersimpan dalam <i>database</i> dan tampil pada tampilan <i>list</i> .
Skenario Menambah Data Pengguna: Skenario Alternatif 1	
Aktor menekan tombol “Simpan” dengan salah satu isian yang kosong.	Sistem menampilkan pesan peringatan “Semua data harus diisi”.
Skenario Menambah Data Pengguna: Skenario Alternatif 2	
Aktor menekan tombol “Batal”.	1. <i>Textbox</i> isian data tidak dapat di-edit. 2. Tombol “Simpan” dan “Batal” menghilang.

Skenario Edit Data Pengguna: Skenario Utama	
Aktor memilih data yang sudah ada pada <i>list</i> .	1. Jika terdapat data yang dipilih, tombol “Edit Data” dan “Hapus Data” aktif. 2. <i>Textbox</i> isian data terisi detail dari data yang dipilih.
Aktor menekan tombol “Edit Data”.	1. <i>Textbox</i> isian data dapat di- <i>edit</i> . 2. Tampil tombol “Simpan” dan “Batal”.
Aktor melakukan <i>edit</i> data kemudian menekan tombol “Simpan”.	Data tersimpan dalam <i>database</i> dan tampil pada tampilan <i>list</i> .
Skenario Edit Data Pengguna: Skenario Alternatif 1	
Aktor menekan tombol “Simpan” dengan mengubah salah satu isian <i>textbox</i> menjadi kosong.	Sistem menampilkan pesan peringatan “Semua data harus diisi”.
Skenario Edit Data Pengguna: Skenario Alternatif 2	
Aktor menekan tombol “Batal”.	1. <i>Textbox</i> isian data tidak dapat di- <i>edit</i> . 2. Tombol “Simpan” dan “Batal” menghilang.
Skenario Menghapus Data Pengguna: Skenario Utama	
Aktor memilih data yang sudah ada pada <i>list</i> .	1. Jika terdapat data yang dipilih, tombol “Edit Data” dan “Hapus Data” aktif. 2. <i>Textbox</i> isian data terisi detail dari data yang dipilih.
Aktor menekan tombol “Hapus Data”.	Sistem menampilkan pesan konfirmasi penghapusan data.
Aktor menekan tombol “Yes” pada pesan konfirmasi.	Data terhapus dari <i>database</i> dan hilang dari tampilan <i>list</i> .
Skenario Menghapus Data Pengguna: Skenario Alternatif	
Aktor menekan tombol “No” pada pesan konfirmasi.	Sistem akan kembali ke halaman Daftar Pengguna.
Kondisi Akhir	Sistem akan menyimpan setiap perubahan yang dilakukan oleh <i>administrator</i> .

Sumber: Perancangan

6. Deskripsi Use Case Menghadap Kamera

Tabel 3.12 Use Case Menghadap Kamera

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	SRS_003_01
Nama	Menghadap Kamera
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana pengunjung menghadap ke kamera yang ada dalam laboratorium.
Tujuan	Untuk mendapatkan hasil citra wajah pengunjung.
Aktor	Pengunjung

Kondisi Awal	Kamera aplikasi Sistem Manajemen Pengunjung dalam keadaan aktif.
Skenario Mengarahkan Wajah Kekamera: Skenario Utama	
Aksi Aktor	Reaksi Sistem
Aktor mengarahkan wajahnya ke kamera.	1. Sistem mendeteksi wajah pengunjung 2. Sistem mengenali wajah pengunjung.
Kondisi Akhir	Sistem akan menampilkan hasil deteksi atau pengenalan wajah

Sumber: Perancangan

3.2.1.6 Analisis Komponen

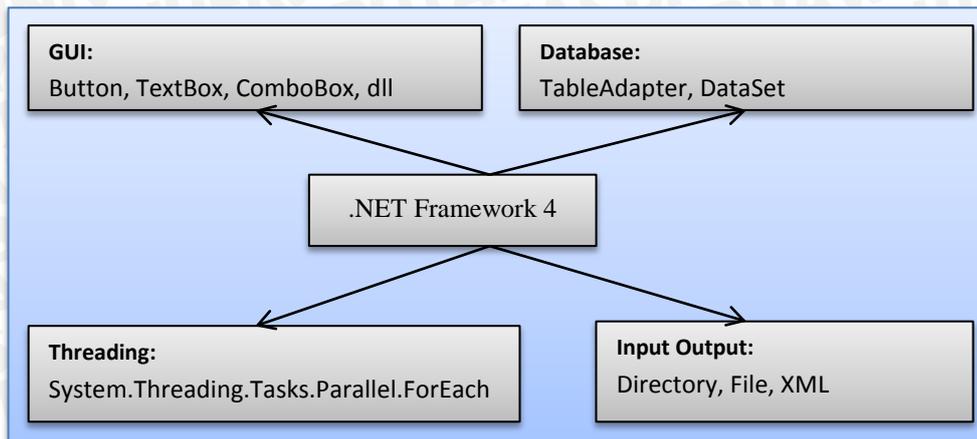
Analisis komponen dilakukan dengan mencari komponen yang dapat mengimplementasi kebutuhan yang telah ditentukan. Komponen-komponen yang dibutuhkan oleh Sistem Manajemen Pengunjung PTIIK Berbasis Pengenalan Wajah adalah *framework* untuk pengembangan aplikasi, komponen untuk pembuatan GUI, dan komponen untuk pengolahan citra. Tabel 3.13 adalah komponen yang dipilih untuk mengimplementasikan Sistem Manajemen Pengunjung PTIIK Berbasis Pengenalan Wajah.

Tabel 3.13 Daftar Komponen

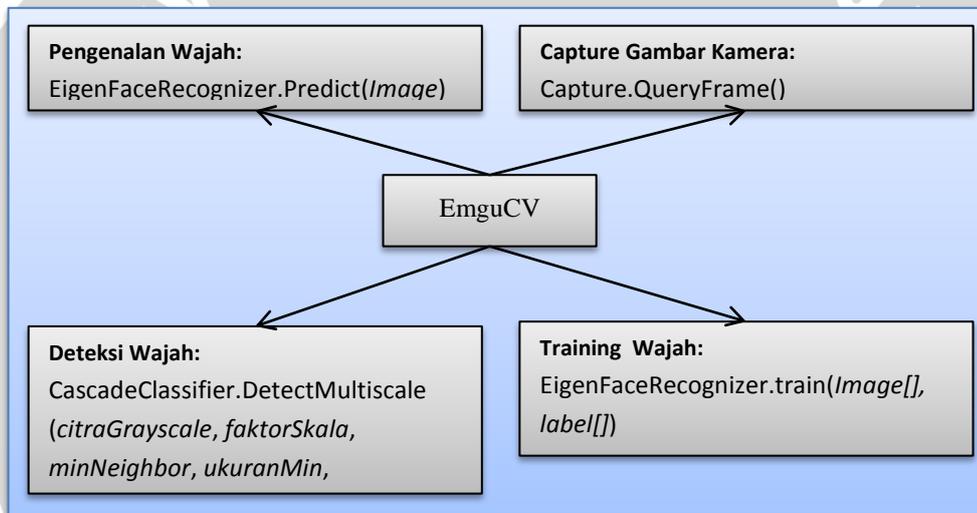
Komponen	Fungsi	Keterangan
.NET Framework 4	Komponen utama sebagai pembangun Sistem Manajemen Pengunjung.	SRS_001_01 - SRS_001_04, SRS_002_01
EmguCV (<i>wrapper</i> untuk OpenCV untuk .NET)	Komponen pengolah citra untuk pengenalan wajah.	SRS_001_03, SRS_003_01
AForge .NET	Komponen pengolah citra untuk mendeteksi gerakan (<i>motion detection</i>).	Fitur Aplikasi
DirectShowLib .NET	Komponen pengolah citra untuk mendeteksi kamera yang terhubung dengan komputer.	Fitur Aplikasi

Sumber: Perancangan

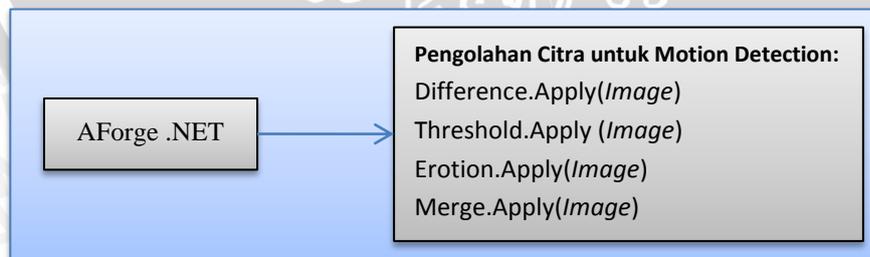
Gambar pemetaan fungsi masing-masing komponen ditunjukkan pada Gambar 3.7 untuk pemetaan fungsi komponen .NET Framework 4, Gambar 3.8 untuk pemetaan fungsi komponen EmguCV, Gambar 3.9 untuk pemetaan fungsi komponen AForge .NET, dan Gambar 3.10 untuk pemetaan fungsi komponen DirectShowLib .NET.



Gambar 3.7 Fungsi .NET Framework 4
Sumber: Perancangan



Gambar 3.8 Fungsi EmguCV
Sumber: Perancangan



Gambar 3.9 Fungsi Aforge .NET
Sumber: Perancangan



Gambar 3.10 Fungsi DirectShowLib .NET
Sumber: Perancangan

3.2.1.7 Modifikasi Kebutuhan

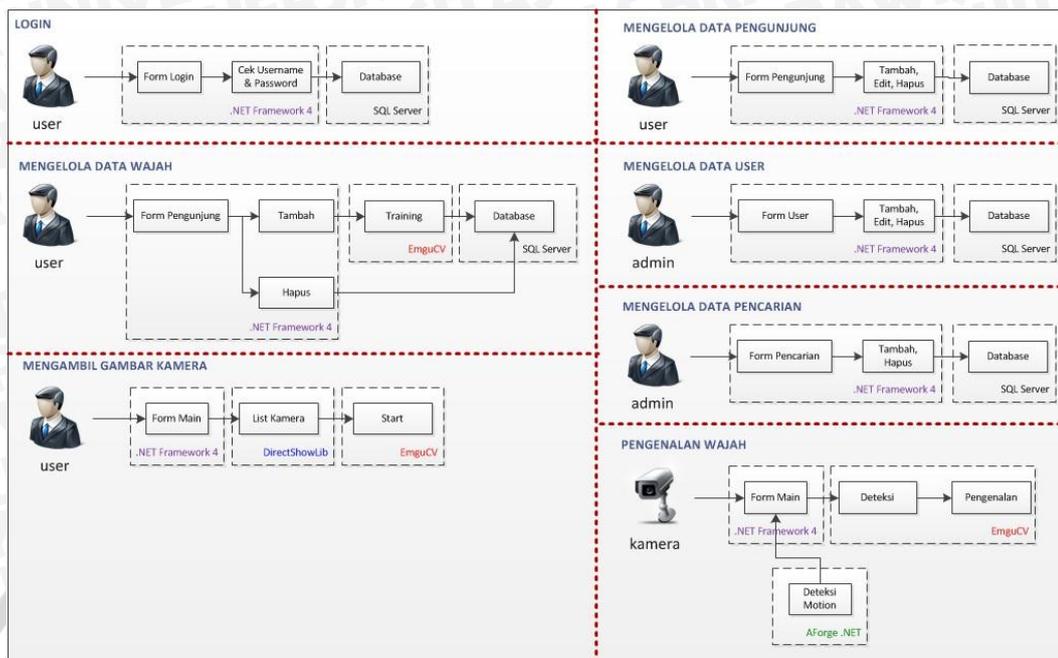
Semua komponen yang terpilih dari proses analisis komponen telah memenuhi kebutuhan yang telah dispesifikasikan. Oleh karena itu tidak perlu dilakukan modifikasi kebutuhan terhadap daftar kebutuhan.

3.2.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan dalam empat tahap, yaitu perancangan basis data, pemodelan kelas untuk menggambarkan perancangan struktur *class – class* yang menyusun perangkat lunak Sistem Manajemen Pengunjung, pemodelan sekuensial untuk menggambarkan interaksi antar objek atau *class* di dalam perangkat lunak Sistem Manajemen Pengunjung, dan perancangan antarmuka pengguna dari perangkat lunak Sistem Manajemen Pengunjung. Perancangan perangkat lunak pada skripsi ini menggunakan pendekatan desain berorientasi objek yang direpresentasikan dengan menggunakan UML (*Unified Modelling Language*).

3.2.2.1 Diagram Arsitektur Perangkat Lunak

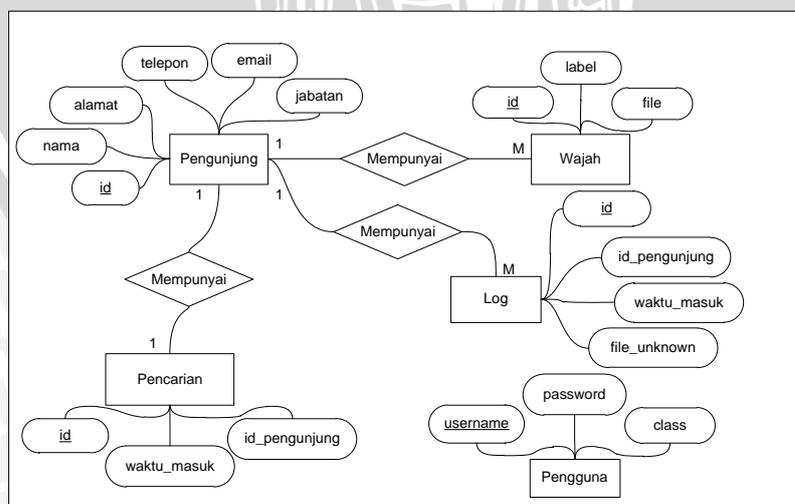
Diagram arsitektur perangkat lunak menggambarkan bagaimana elemen/komponen fungsional perangkat lunak disusun, diorganisasi dan distrukturkan sehingga hubungan antar elemen/komponen dapat dijelaskan. Diagram arsitektur Sistem Manajemen Pengunjung ditunjukkan pada Gambar 3.11.



Gambar 3.11 Diagram Arsitektur Sistem Manajemen Pengunjung
Sumber: Perancangan

3.2.2.2 Perancangan Basis Data

Basis data berfungsi sebagai tempat menyimpan data. Perancangan basis data digunakan untuk merancang basis data yang akan dibuat agar masukan dan keluaran program sesuai dengan apa yang diharapkan. Perancangan basis data mengambil acuan dari proses analisis data yang dilakukan pada tahap analisis kebutuhan. Arsitektur basis data yang akan dirancang dijelaskan pada Gambar 3.12.



Gambar 3.12 Diagram Entity Relationship Sistem Manajemen Pengunjung
Sumber: Perancangan



Berikut ini merupakan struktur tabel serta keterangan masing masing tabel dan *field* yang ada pada *database*.

1. Tabel Pengguna

Nama tabel : pengguna
 Jumlah *field* : 3
 Fungsi : Untuk menyimpan data pengguna

Tabel 3.14 Tabel Pengguna

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1.	<u>username</u>	Varchar	100	Nama pengguna
2.	password	varchar	100	<i>Password</i> pengguna
3.	Class	int	4	Untuk membedakan <i>administrator</i> dengan pemantau

Sumber: Perancangan

2. Tabel Pengunjung

Nama tabel : pengunjung
 Jumlah *field* : 6
 Fungsi : Untuk menyimpan data pengunjung

Tabel 3.15 Tabel Pengunjung

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1.	<u>Id</u>	varchar	100	Id pengunjung
2.	Nama	varchar	100	Nama pengunjung
3.	Alamat	varchar	100	Alamat pengunjung
4.	Telepon	varchar	100	Nomor telepon pengunjung
5.	Email	varchar	100	Alamat email pengunjung
6.	Jabatan	varchar	100	Jabatan pengunjung

Sumber: Perancangan

3. Tabel Wajah

Nama tabel : wajah
 Jumlah *field* : 4
 Fungsi : Untuk menyimpan data wajah pengunjung

Tabel 3.16 Tabel Wajah

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1.	<u>Id</u>	bigint	8	Id wajah
2.	label	varchar	100	Nama label wajah
3.	file	varchar	255	Nama <i>file</i> wajah

Sumber: Perancangan

4. Tabel Log

Nama tabel : log

Jumlah *field* : 3Fungsi : Untuk menyimpan data *log***Tabel 3.17** Tabel Log

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1.	<u>Id</u>	bigint	8	Id <i>log</i>
2.	waktu_masuk	datetime	8	Waktu masuk pengunjung
3.	id_pengunjung	varchar	100	Id pengunjung
4.	file_unknown	varchar	255	Nama <i>file</i> pada data <i>unknown</i>

Sumber: Perancangan

5. Tabel Pencarian

Nama tabel : pencarian

Jumlah *field* : 2

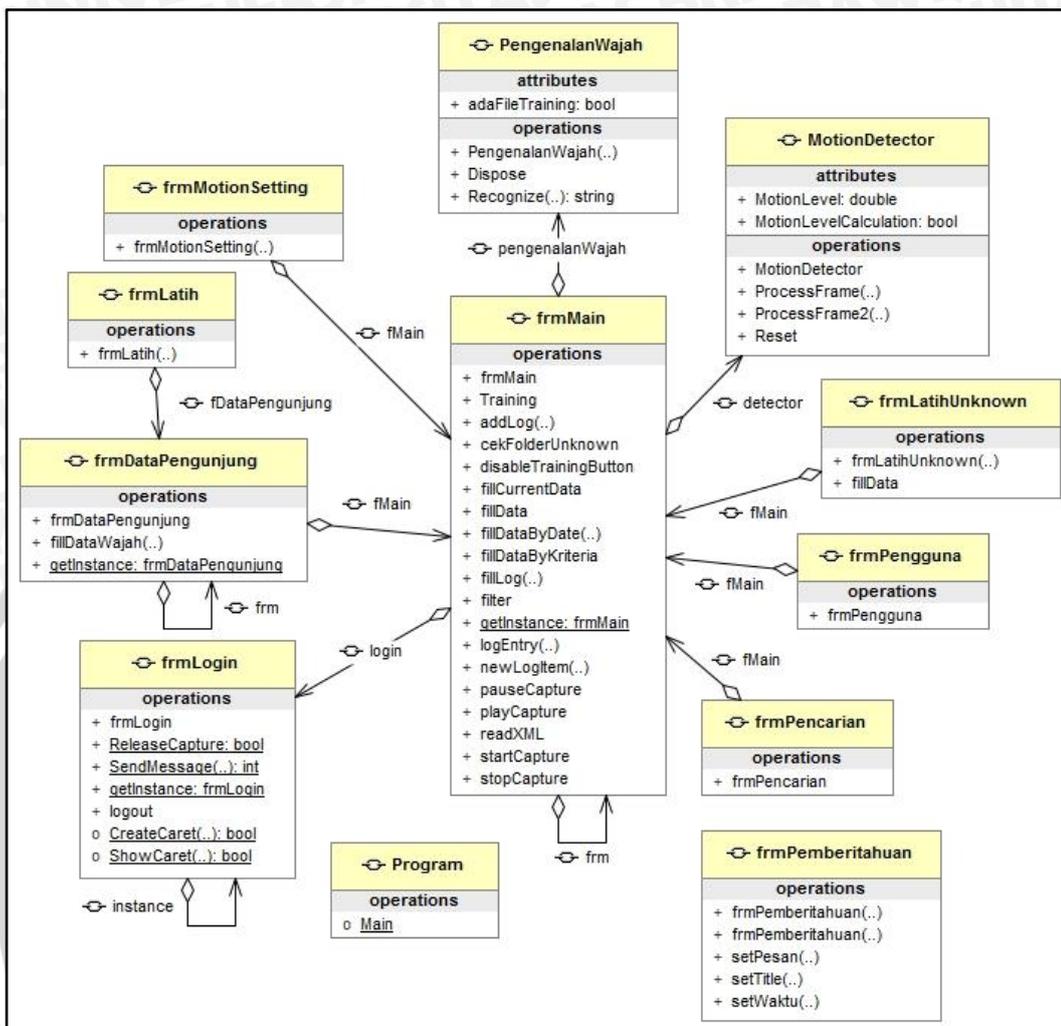
Fungsi : Untuk menyimpan data pencarian pengunjung

Tabel 3.18 Tabel Pencarian

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1.	<u>id</u>	Int	4	Id pencarian
2.	id_pengunjung	Varchar	100	Id pengunjung

Sumber: Perancangan**3.2.2.3 Class Diagram**

Class diagram memberikan gambaran pemodelan elemen-elemen kelas yang membentuk sebuah sistem. Kelas bisa didapatkan dengan menganalisis secara detail terhadap *use case* yang dimodelkan. Gambar 3.13 menunjukkan pemodelan kelas dengan menggunakan *class diagram* dari Sistem Manajemen Pengunjung yang akan dibuat. Penjelasan masing-masing class ditunjukkan oleh Tabel 3.19.



Gambar 3.13 Class Diagram Sistem Manajemen Pengunjung
Sumber: Perancangan

Tabel 3.19 Deskripsi Class

No.	Nama Class	Deskripsi
1.	Program	Class Program merupakan class yang pertama kali dipanggil oleh sistem. Class Program akan memanggil class frmLogin sebagai tampilan awal. Class Program adalah class yang secara otomatis dibuat oleh Visual Studio sebagai bagian dari framework .NET.
2.	frmLogin	Class frmLogin merupakan class yang menampilkan halaman Login. Class frmLogin berfungsi untuk menerima input username dan password dari pengguna kemudian memeriksa kesesuaiannya dengan database. Jika username dan password sesuai dengan database, maka frmLogin akan memanggil class frmMain.
3.	frmMain	Class frmMain merupakan class yang berfungsi sebagai halaman utama aplikasi yang berisi panel untuk menampilkan gambar dari kamera dan daftar log. Class frmMain juga dapat menampilkan data pengunjung



		secara detail yang dipilih dari data <i>log</i> . Selain itu, <i>class frmMain</i> juga berisi menu-menu untuk mengelola aplikasi Sistem Manajemen Pengunjung.
4.	<i>frmDataPengunjung</i>	<i>Class frmDataPengunjung</i> merupakan <i>class</i> yang berfungsi untuk mengelola data pengunjung. <i>Class frmDataPengunjung</i> dapat melakukan pelatihan wajah untuk data yang dipilih dengan memanggil <i>class frmTraining</i> .
5.	<i>frmLatih</i>	<i>Class frmLatih</i> merupakan <i>class</i> yang berfungsi untuk melakukan pelatihan data pengunjung yang diperoleh dari <i>class frmDataPengunjung</i> . <i>Class frmTraining</i> akan menampilkan gambar dari <i>webcam</i> , kemudian akan mendeteksi wajah yang ada pada gambar tersebut.
6.	<i>frmLatihUnknown</i>	<i>Class frmLatihUnknown</i> merupakan <i>class</i> yang berfungsi untuk melakukan pelatihan data pengunjung yang tidak dikenali oleh sistem. Data yang tidak dikenali tersebut diperoleh dari data <i>log</i> yang ditampilkan pada <i>class frmMain</i> .
7.	<i>frmPegguna</i>	<i>Class frmPegguna</i> merupakan <i>class</i> berfungsi untuk mengelola data pengguna.
8.	<i>frmPencarian</i>	<i>Class frmPencarian</i> merupakan <i>class</i> yang berfungsi untuk melakukan pencarian pengunjung yang terdaftar dalam sistem.
9.	<i>frmMotionSetting</i>	<i>Class frmMotionSetting</i> merupakan <i>class</i> yang berfungsi untuk mengatur <i>motion detection</i> (deteksi gerakan) pada gambar yang ditangkap oleh kamera. Pengaturan yang dapat dilakukan antara lain yaitu pengaturan sensitivitas <i>motion</i> , aktivasi/deaktivasi deteksi <i>motion</i> , dan aktivasi/deaktivasi tampilan <i>motion (motion trail)</i> .
10.	<i>frmPemberitahuan</i>	<i>Class frmPemberitahuan</i> merupakan <i>class</i> yang berfungsi untuk menampilkan pesan pemberitahuan saat pengunjung yang dicari telah ditemukan.
11.	PengenalanWajah	<i>Class PengenalanWajah</i> merupakan <i>class</i> yang berfungsi untuk melakukan proses pengenalan wajah terhadap citra dan proses pelatihan wajah dari <i>database</i> .
12.	MotionDetector	<i>Class MotionDetector</i> merupakan <i>class</i> yang berfungsi untuk melakukan proses deteksi <i>motion</i> dengan cara membandingkan citra yang ditangkap sebelumnya dengan citra yang ditangkap sesudahnya. <i>Motion</i> akan terdeteksi oleh <i>MotionDetector</i> jika terdapat perbedaan antara citra pertama dan kedua.

Sumber: Perancangan

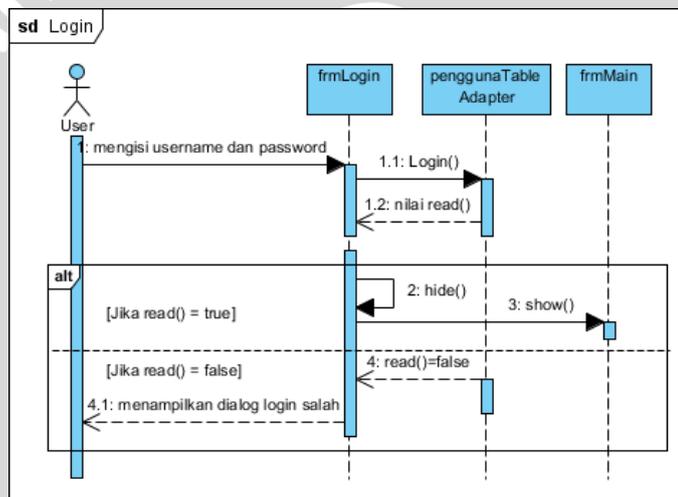
3.2.2.4 Diagram Sekuensial

Diagram sekuensial menunjukkan pemodelan aliran jalannya proses interaksi antar objek atau *class* yang disusun berdasarkan urutan waktu. Diagram

sekuensial digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Diagram sekuensial disusun dengan mengambil acuan pada *use case* dan *class - class* yang membentuk fungsionalitas yang digambarkan pada *use case* tersebut.

1. Diagram Sekuensial Login

Diagram sekuensial login menggambarkan bagaimana pengguna melakukan login untuk mengakses sistem. Diagram ini ditunjukkan pada Gambar 3.14.

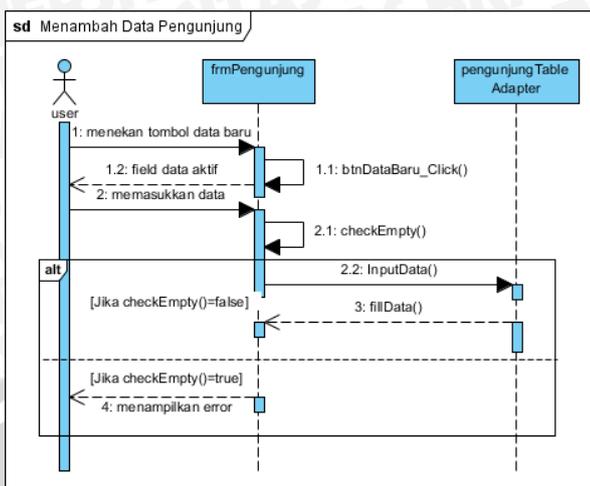


Gambar 3.14 Diagram Sekuensial Login

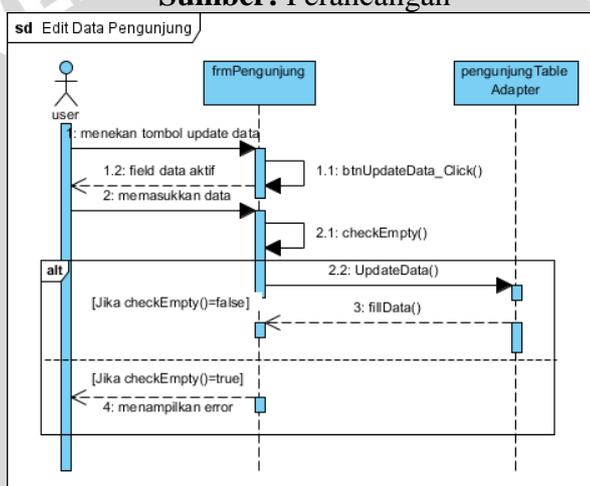
Sumber: Perancangan

2. Diagram Sekuensial Mengelola Data Pengunjung

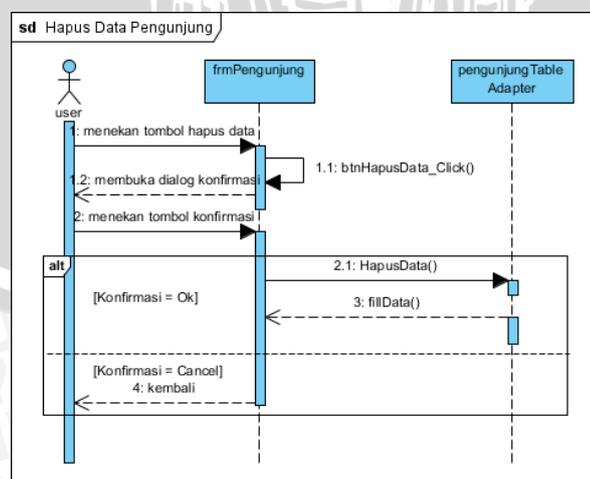
Diagram sekuensial mengelola data pengunjung meliputi operasi menambah data, mengedit data, dan menghapus data pengunjung. Diagram sekuensial untuk operasi menambah data pengunjung ditunjukkan pada Gambar 3.15. Diagram sekuensial untuk operasi meng-*edit* data pengunjung ditunjukkan pada Gambar 3.16. Diagram sekuensial untuk operasi menghapus data pengunjung ditunjukkan pada Gambar 3.17.



Gambar 3.15 Diagram Sekuensial Menambah Data Pengunjung
Sumber: Perancangan



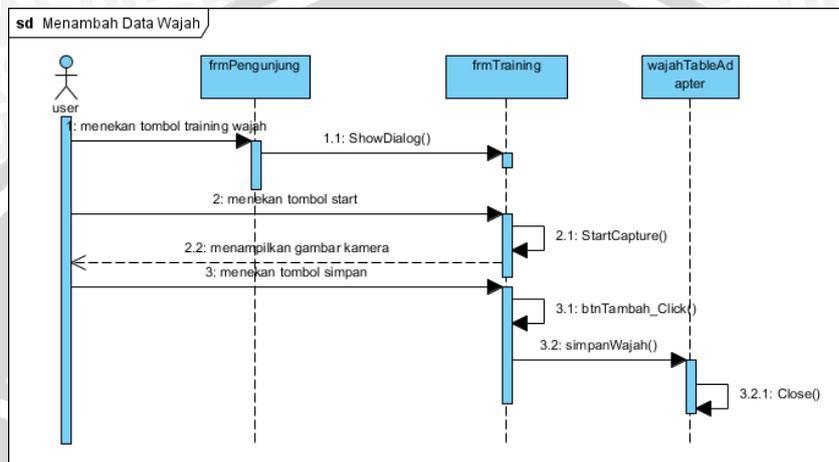
Gambar 3.16 Diagram Sekuensial *Edit* Data Pengunjung
Sumber: Perancangan



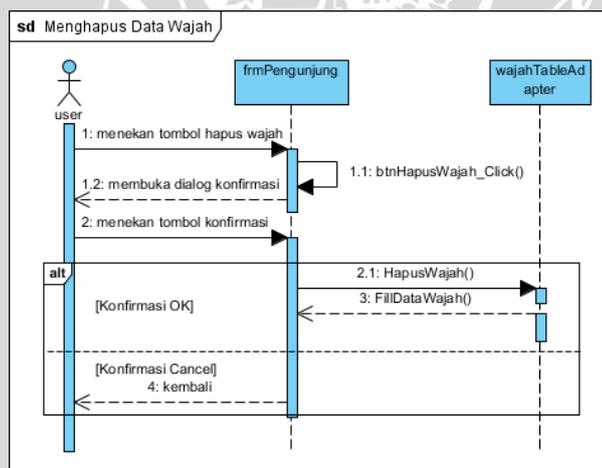
Gambar 3.17 Diagram Sekuensial Menghapus Data Pengunjung
Sumber: Perancangan

3. Diagram Sekuensial Mengelola Data Wajah

Diagram sekuensial mengelola data wajah meliputi operasi menambah data wajah dan menghapus data wajah. Diagram sekuensial untuk operasi menambah data wajah ditunjukkan pada Gambar 3.18. Diagram sekuensial untuk operasi menghapus data wajah ditunjukkan pada Gambar 3.19.



Gambar 3.18 Diagram Sekuensial Menambah Data Wajah
Sumber: Perancangan

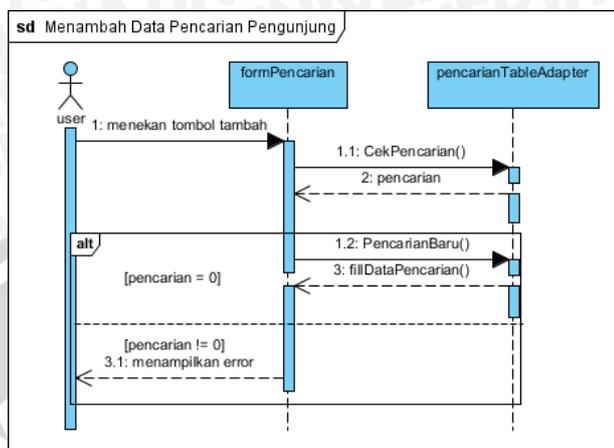


Gambar 3.19 Diagram Sekuensial Menghapus Data Wajah
Sumber: Perancangan

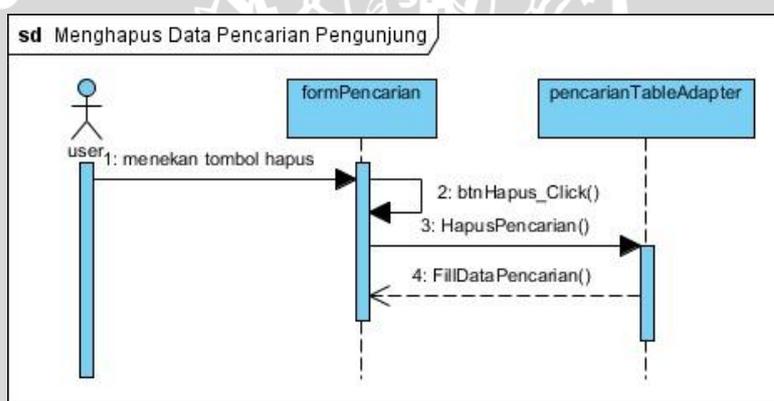
4. Diagram Sekuensial Mengelola Data Pencarian Pengunjung

Diagram sekuensial mengelola data pencarian pengunjung meliputi operasi menambah data dan menghapus data pengunjung. Diagram sekuensial untuk operasi menambah data pencarian pengunjung ditunjukkan pada Gambar 3.20.

Diagram sekuensial untuk operasi menghapus data pencarian pengunjung ditunjukkan pada Gambar 3.21.



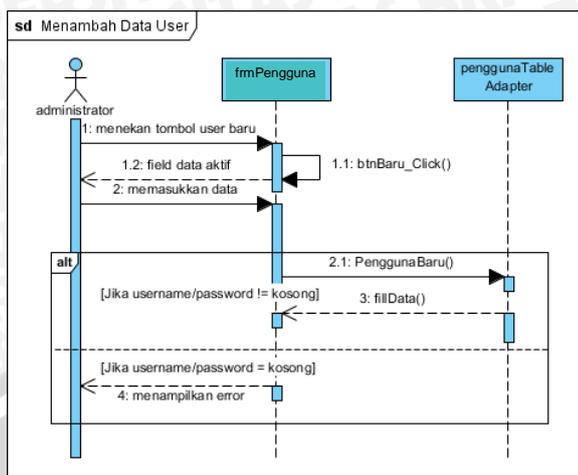
Gambar 3.20 Diagram Sekuensial Menambah Data Pencarian Pengunjung
Sumber: Perancangan



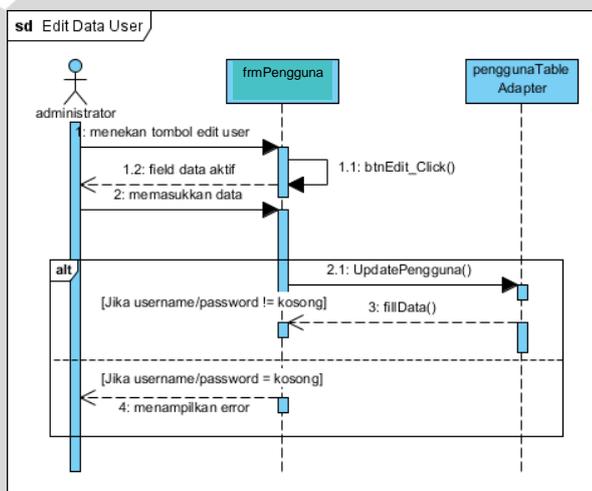
Gambar 3.21 Diagram Sekuensial Menghapus Data Pencarian Pengunjung
Sumber: Perancangan

5. Diagram Sekuensial Mengelola Data Pengguna

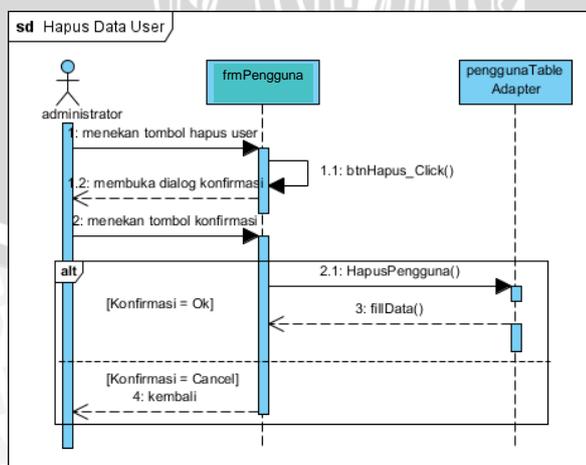
Diagram sekuensial mengelola data pengguna meliputi operasi menambah data, mengedit data, dan menghapus data pengguna. Diagram sekuensial untuk operasi menambah data pengguna ditunjukkan pada Gambar 3.22. Diagram sekuensial untuk operasi mengedit data pengguna ditunjukkan pada Gambar 3.23. Diagram sekuensial untuk operasi menghapus data pengguna ditunjukkan pada Gambar 3.24.



Gambar 3.22 Diagram Sekuensial Menambah Data Pengguna
Sumber: Perancangan



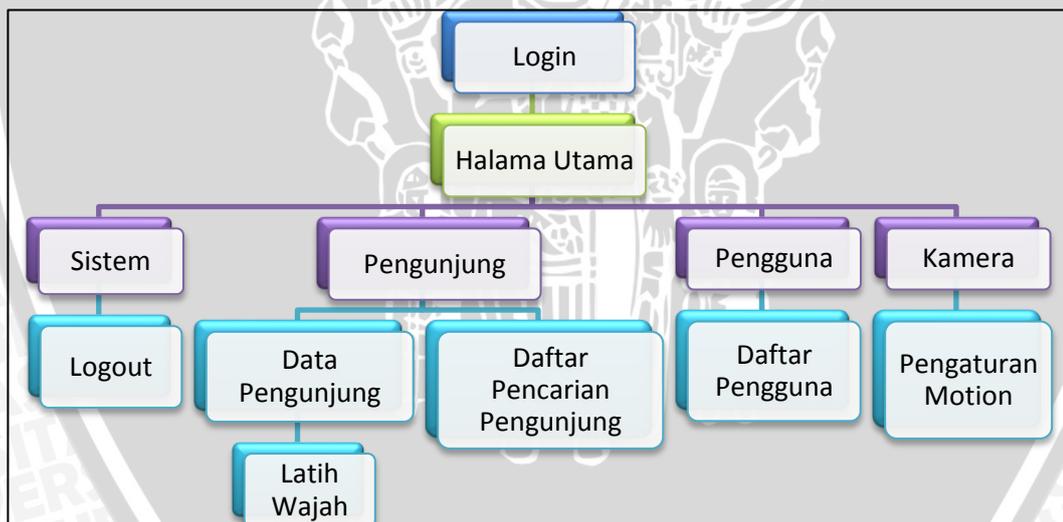
Gambar 3.23 Diagram Sekuensial *Edit* Data Pengguna
Sumber: Perancangan



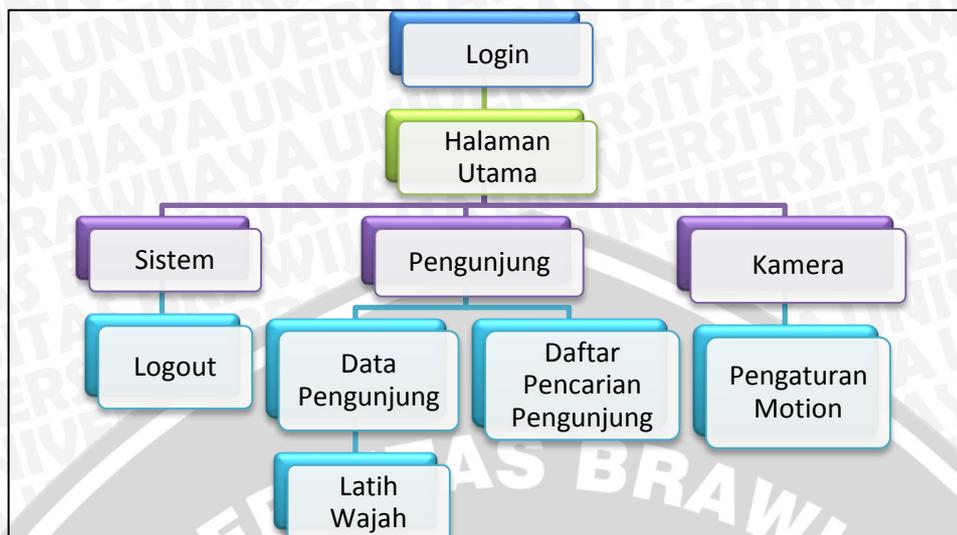
Gambar 3.24 Diagram Sekuensial Menghapus Data Pengguna
Sumber: Perancangan

3.2.2.5 Antarmuka Pengguna

Pada bagian ini akan dijelaskan tentang perancangan antarmuka Sistem Manajemen Pengunjung Berbasis Pengenalan Wajah. Antarmuka perangkat lunak ini akan digunakan oleh pengguna untuk berinteraksi dengan sistem. Antarmuka pengguna berupa halaman yang dapat digunakan oleh pengguna untuk mengelola data dan unsur – unsur lain yang diperlukan oleh perangkat lunak Sistem Manajemen Pengunjung. Antarmuka pengguna untuk *administrator* terdiri atas halaman Login, halaman Utama, halaman Data Pengunjung, halaman Latih Wajah, halaman Daftar Pencarian Pengunjung, halaman Daftar Pengguna, dan halaman Pengaturan Motion. *Sitemap* aplikasi *administrator* dari perangkat lunak manajemen pengunjung ditunjukkan oleh Gambar 3.25. Sedangkan antarmuka pengguna untuk pemantau sama dengan antarmuka pengguna *administrator* tetapi tanpa halaman daftar pengguna. *Sitemap* aplikasi pemantau dari perangkat lunak manajemen pengunjung ditunjukkan oleh Gambar 3.26.



Gambar 3.25 Struktur Hirarki Menu untuk *Administrator*
Sumber: Perancangan



Gambar 3.26 Struktur Hirarki Menu untuk Pemantau

Sumber: Perancangan

1. Halaman Login

Halaman login merupakan salah satu antarmuka pengguna untuk pengguna yang berfungsi untuk mempermudah pengguna dalam melakukan login. Gambar 3.27 akan menunjukkan perancangan tampilan antarmuka dari halaman login.

Gambar 3.27 Tampilan Antarmuka Halaman Login

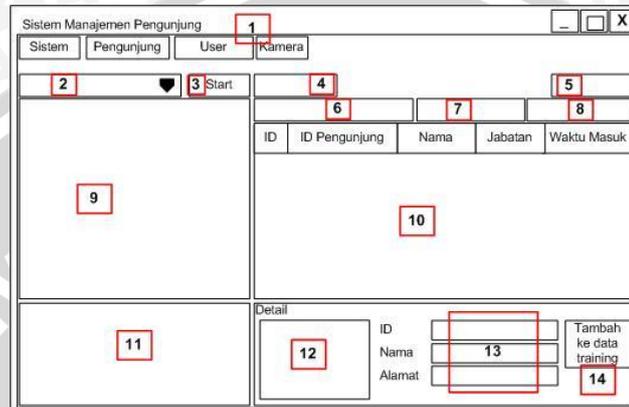
Sumber: Perancangan

Gambar 3.27 memiliki keterangan sebagai berikut :

1. *Field* untuk mengisi *username*.
2. *Field* untuk mengisi *password*.
3. Tombol untuk *login* masuk aplikasi.
4. Tombol untuk keluar dari aplikasi.

2. Halaman Utama

Halaman utama merupakan salah satu antarmuka pengguna untuk aplikasi pengguna yang berfungsi untuk menampilkan gambar dari kamera, *log*, dan menu untuk mengakses halaman-halaman lain. Gambar 3.28 akan menunjukkan perancangan tampilan antarmuka dari halaman utama.



Gambar 3.28 Tampilan Antarmuka Halaman Utama
Sumber: Perancangan

Gambar 3.28 memiliki keterangan sebagai berikut :

1. *Menu bar* untuk memilih halaman lain yang diinginkan.
2. *ComboBox* untuk memilih kamera yang terhubung dengan komputer.
3. Tombol untuk memulai menjalankan layar *webcam*.
4. Tanggal sekarang yang diperoleh dari sistem komputer.
5. Jam sekarang yang diperoleh dari sistem komputer.
6. *Field* untuk melakukan filter atau pencarian data *log*.
7. *DatePicker* untuk memilih tanggal awal pengunjung yang masuk ke dalam laboratorium sebagai kriteria untuk *filter* atau pencarian data *log*.
8. *DatePicker* untuk memilih tanggal akhir pengunjung yang masuk ke dalam laboratorium sebagai kriteria untuk *filter* atau pencarian data *log*.
9. Layar yang sudah tersambung dengan *webcam*.
10. Daftar *log* yang berisi data pengunjung yang masuk pada saat itu.
11. Kalender.
12. Gambar wajah dari data *log* yang dipilih.
13. *Field* detail data pengunjung yang akan terisi data *log* yang dipilih.
14. Tombol untuk menambah data latih dari data *log* yang dipilih.

3. Halaman Data Pengunjung

Halaman data pengunjung merupakan antarmuka yang digunakan untuk mengelola data pengunjung. Perancangan antarmuka ini mengacu pada spesifikasi kebutuhan SRS_001_02. Halaman ini akan ditampilkan setelah pengguna menekan *submenu* “Data Pengunjung” pada menu “Pengunjung” yang berada di *menu bar*. Gambar 3.29 akan menunjukkan perancangan tampilan antarmuka dari halaman “Data Pengunjung”.

Gambar 3.29 Tampilan Antarmuka Halaman Data Pengunjung

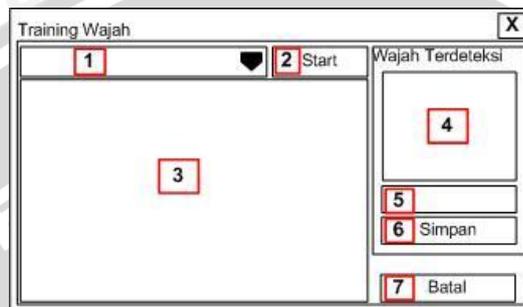
Sumber: Perancangan

Gambar 3.29 memiliki keterangan sebagai berikut :

1. *Field* untuk mengisi dan menampilkan data pengunjung.
2. Tombol untuk membatalkan pengisian data pengunjung.
3. Tombol untuk menyimpan data pengunjung.
4. *Field* untuk melakukan *filter* atau pencarian data pengunjung.
5. Tabel data pengunjung.
6. Tombol untuk menambah data pengunjung yang baru.
7. Tombol untuk mengubah data pengunjung.
8. Tombol untuk menghapus data pengunjung.
9. Gambar wajah pengunjung dari data pengunjung maupun data wajah yang dipilih.
10. Tombol untuk melatih wajah.
11. Tombol untuk menghapus data wajah.
12. Tabel data wajah.
13. Tombol untuk menutup halaman Data Pengunjung.

4. Halaman Latih Wajah

Halaman Latih Wajah merupakan antarmuka yang digunakan untuk mengelola data latih wajah pengunjung. Halaman ini akan ditampilkan setelah pengguna menekan tombol “Latih wajah untuk data ini” pada halaman “Data Pengunjung”. Gambar 3.30 akan menunjukkan perancangan tampilan antarmuka dari halaman “Latih Wajah”.



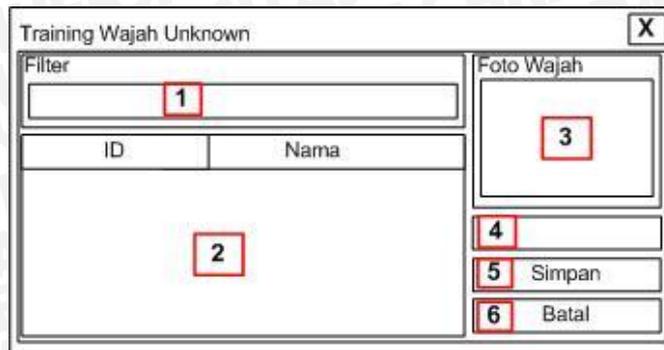
Gambar 3.30 Tampilan Antarmuka Halaman Latih Wajah dari Data Pengunjung
Sumber: Perancangan

Gambar 3.30 memiliki keterangan sebagai berikut :

1. *Combo box* untuk memilih kamera yang terhubung dengan komputer.
2. Tombol untuk memulai menjalankan layar *webcam* aplikasi.
3. Layar yang sudah tersambung dengan *webcam*.
4. Gambar wajah yang terdeteksi oleh kamera.
5. *Field* untuk mengisi ID pengunjung.
6. Tombol untuk menyimpan data latih.
7. Tombol untuk menutup halaman Latih Wajah.

5. Halaman Latih Wajah Unknown

Halaman Latih Wajah Unknown merupakan antarmuka yang digunakan untuk mengelola data latih wajah pengunjung yang belum dikenali oleh sistem aplikasi. Halaman ini akan ditampilkan setelah pengguna menekan tombol “Tambah ke data latih” pada halaman Utama. Gambar 3.31 akan menunjukkan perancangan tampilan antarmuka dari halaman “Latih Wajah Unknown”.



Gambar 3.31 Tampilan Antarmuka Halaman Latih Wajah *Unknown*

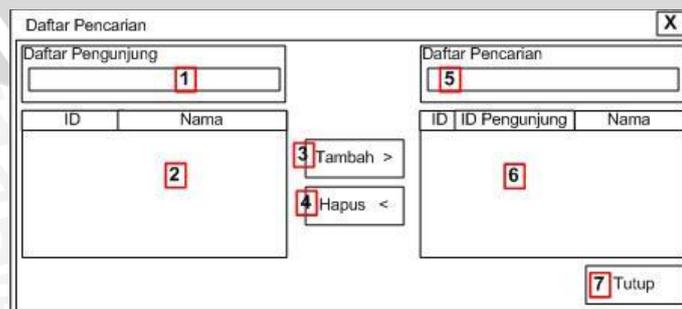
Sumber: Perancangan

Gambar 3.31 memiliki keterangan sebagai berikut :

1. *Field* untuk melakukan filter atau pencarian data pengunjung.
2. Tabel data pengunjung.
3. Gambar wajah yang terdeteksi oleh kamera.
4. *Field* yang berisi nama pengunjung.
5. Tombol untuk menyimpan data latih.
6. Tombol untuk menutup halaman Latih Wajah *Unknown*.

6. Halaman Daftar Pencarian Pengunjung

Halaman daftar pencarian pengunjung merupakan antarmuka yang digunakan untuk mengelola daftar pencarian pengunjung. Perancangan antarmuka ini mengacu pada spesifikasi kebutuhan SRS_001_04. Halaman ini akan ditampilkan setelah pengguna menekan *submenu* Daftar Pencarian Pengunjung pada menu Pengunjung yang berada di *menu bar*. Gambar 3.32 akan menunjukkan perancangan tampilan antarmuka dari halaman Daftar Pencarian Pengunjung.



Gambar 3.32 Tampilan Antarmuka Daftar Pencarian Pengunjung

Sumber: Perancangan

Gambar 3.32 memiliki keterangan sebagai berikut :

1. *Field* untuk melakukan filter atau pencarian data pengunjung.
2. Tabel data pengunjung.
3. Tombol untuk menambahkan pencarian.
4. Tombol untuk menghapus pencarian yang sudah dipilih.
5. *Field* untuk melakukan filter atau pencarian daftar pencarian.
6. Tabel daftar pencarian.
7. Tombol untuk menutup halaman Daftar Pencarian.

7. Halaman Daftar Pengguna

Halaman Daftar Pengguna merupakan antarmuka yang digunakan untuk mengelola data pengguna aplikasi Sistem Manajemen Pengunjung. Perancangan antarmuka ini mengacu pada spesifikasi kebutuhan SRS_002_05. Halaman ini akan ditampilkan setelah *administrator* menekan menu Pengguna yang berada di *menu bar*. Gambar 3.33 akan menunjukkan perancangan tampilan antarmuka dari halaman Daftar Pengguna.

Gambar 3.33 Tampilan Antarmuka Halaman Daftar Pengguna
Sumber: Perancangan

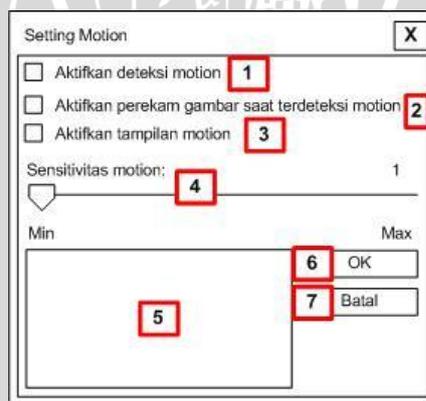
Gambar 3.33 memiliki keterangan sebagai berikut :

1. *Field* untuk mengisi *username*.
2. *Field* untuk mengisi *password*.

3. *Combo box* untuk memilih jenis pengguna yaitu *administrator* atau pemantau.
4. Tombol untuk membatalkan perubahan data pengguna.
5. Tombol untuk menyimpan data pengguna.
6. Tabel data pengguna.
7. Tombol untuk menambah pengguna baru.
8. Tombol untuk mengubah data pengguna.
9. Tombol untuk menghapus data pengguna.
10. Tombol untuk menutup halaman Daftar Pengguna.

8. Halaman Setting Motion

Halaman *setting motion* merupakan antarmuka yang digunakan untuk mengatur deteksi *motion*. Halaman ini akan ditampilkan setelah pengguna menekan *submenu* “Setting Motion” pada menu “Kamera” yang berada di *menu bar*. Gambar 3.34 akan menunjukkan perancangan tampilan antarmuka dari halaman “Setting Motion”.



Gambar 3.34 Tampilan Antarmuka Halaman *Setting Motion*
Sumber: Perancangan

Gambar 3.34 memiliki keterangan sebagai berikut :

1. *Check box* untuk mengaktifkan deteksi *motion*.
2. *Check box* untuk mengaktifkan gambar saat terdeteksi *motion*.
3. *Check box* untuk mengaktifkan tampilan *motion*.
4. *Slider* untuk menentukan nilai dari sensitivitas *motion*.
5. Label untuk menjelaskan tentang *setting motion*.

6. Tombol untuk menyimpan *setting motion*.
7. Tombol untuk menutup halaman *setting motion*.



BAB IV

IMPLEMENTASI

Bab ini membahas mengenai tahapan implementasi Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan sistem. Pembahasan terdiri atas penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi basis data, implementasi tiap kelas pada *file* program, implementasi *method*, dan implementasi antarmuka.

4.1 Spesifikasi Sistem

Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah dikembangkan dalam lingkungan implementasi yang terdiri dari perangkat keras dan perangkat lunak.

4.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang dipakai dalam proses pengembangan Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah dijelaskan pada Tabel 4.1.

Tabel 4.1 Spesifikasi Perangkat Keras Komputer

ASUS N43SL	
<i>Processor</i>	Intel Core i5
<i>Memory (RAM)</i>	4.00 GB
<i>Harddisk</i>	750 GB
<i>Graphic Card</i>	Nvidia GeForce GT540M 2GB
<i>Monitor</i>	14.1" High-definition widescreen
PC Camera	
<i>Model</i>	Logitech Webcam 101
<i>Image Resolution</i>	1,3 Mpx

Sumber: Implementasi

4.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang dipakai dalam proses pengembangan Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah dijelaskan pada Tabel 4.2.

Tabel 4.2 Spesifikasi Perangkat Lunak Komputer

ASUS N43SL	
<i>Operating System</i>	Microsoft Windows 7 Ultimate 64-bit
<i>Programming Language</i>	C#
<i>Database Management System</i>	SQL Server 2008
<i>Integrated Development Environment</i>	Microsoft Visual Studio 2010

Sumber: Implementasi

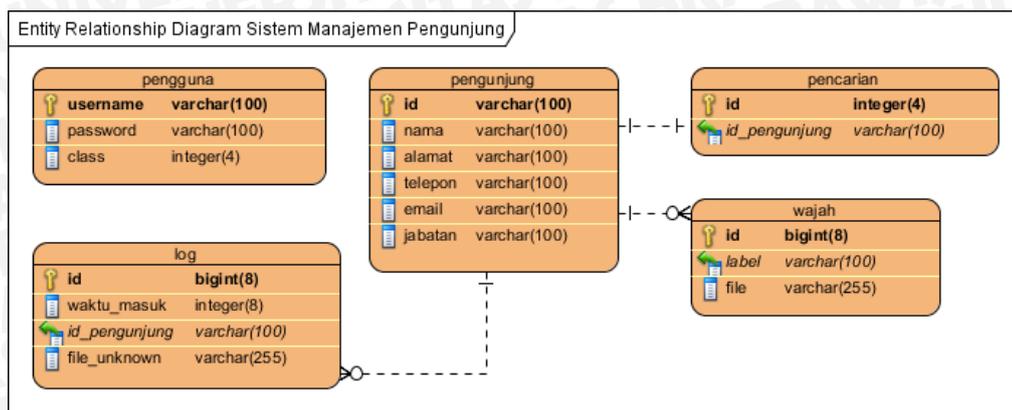
4.2 Batasan-batasan Implementasi

Beberapa batasan dalam mengimplementasikan Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah sebagai berikut :

1. Sistem Manajemen Pengunjung dirancang dan dijalankan dengan menggunakan *Visual C# 2010*.
2. *Database Management System* yang digunakan adalah Microsoft SQL Server 2008 sebagai *database server*.
3. *Library* grafis yang digunakan adalah EmguCV.
4. Sistem operasi yang digunakan adalah Windows 7.
5. Citra masukan wajah yang digunakan sebagai gambar latih memiliki format *.jpg*.
6. Pengambilan data latih dilakukan pada tempat yang sama dengan rata-rata tingkat keabuan lebih besar sama dengan 100 untuk skala 0 sampai 255.

4.3 Implementasi Basis Data

Implementasi penyimpanan data dilakukan dengan *database management system* Microsoft SQL Server 2008. Hasil implementasi SQL pada *database* ini dimodelkan dalam diagram konseptual *entity relationship*. Gambar 4.1 menggambarkan diagram konseptual *entitiy relationship* dari Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah.



Gambar 4.1 Diagram ER konseptual dari Sistem Manajemen Pengunjung

Sumber: Implementasi

4.4 Implementasi Class dan Antarmuka pada File Program

Setiap *class* yang telah dirancang pada proses perancangan direalisasikan pada sebuah *file* program dengan ekstensi *.cs. Tabel 4.3 menjelaskan mengenai pasangan antara *class* dengan *file* program yang digunakan untuk mengimplementasikannya.

Tabel 4.3 Implementasi *class* pada kode program

No.	Nama Class atau Antarmuka	Nama File Program
1	Program	Program.cs
2	frmLogin	frmLogin.cs
3	frmMain	frmMain.cs
4	frmDataPengunjung	frmDataPengunjung.cs
5	frmTraining	frmTraining.cs
6	frmTrainUnknown	frmTrainUnknown.cs
7	frmPegguna	frmPegguna.cs
8	frmPencarian	frmPencarian.cs
9	frmMotionSetting	frmMotionSetting.cs
10	frmPemberitahuan	frmPemberitahuan.cs
11	PengenalanWajah	PengenalanWajah.cs
12	MotionDetector	MotionDetector.cs

Sumber: Implementasi

4.5 Implementasi Method

Perangkat lunak Sistem Manajemen Pengunjung memiliki beberapa proses atau *method* yang terdapat pada beberapa *class*. Beberapa *method* yang akan dicantumkan dalam penulisan makalah skripsi ini hanya untuk algoritma dari beberapa proses (operasi) saja sehingga tidak semua *method* akan dicantumkan.



Implementasi algoritma ini akan direpresentasikan dalam bentuk *pseudocode* dari algoritma – algoritma tersebut.

4.5.1 Implementasi Method getLabel

Method getLabel merupakan *method* untuk mencari id pengunjung yang ada di dalam database dengan *input* id wajah yang didapatkan saat proses pengenalan wajah. Proses pencarian id pengunjung dilakukan oleh objek wajahTableAdapter yang merupakan objek yang berfungsi untuk melakukan manipulasi data tabel “wajah” di dalam database.

Tabel 4.4 Implementasi *method* getLabel

Elemen	Keterangan
Nama	getLabel
Deklarasi	<ul style="list-style-type: none"> wajahTableAdapter → adapter String → label
Procedure	<ul style="list-style-type: none"> Masukan: id Proses <ol style="list-style-type: none"> Dapatkan id pengunjung berdasarkan id input. label = adapter.getLabel(id) Jika label tidak kosong: label ≠ "" Hasil = label: Jika label kosong: label = "" Hasil = "unknown" Keluaran : String → Hasil

Sumber: Implementasi

4.5.2 Implementasi Method Recognize

Method Recognize merupakan *method* untuk melakukan proses pengenalan terhadap suatu citra wajah. Pengenalan wajah dilakukan untuk memanggil fungsi Predict pada objek EigenFaceRecognizer dari *library* EmguCV. Hasil pengenalan wajah kemudian disimpan ke dalam objek PredictionResult. Proses pengenalan wajah akan dilakukan jika AdaFileTraining bernilai *true*. Nilai dari AdaFileTraining diperoleh dari *method* BukaData yang dipanggil pada konstruktor *class* PengenalanWajah.

Tabel 4.5 Implementasi *method* Recognize

Elemen	Keterangan
Nama	Recognize
Deklarasi	<ul style="list-style-type: none"> EigenFaceRecognizer → recognizer

	<ul style="list-style-type: none"> • PredictionResult → PR • Boolean → AdaFileTraining
Procedure	<ul style="list-style-type: none"> • Masukan: Input_image • Proses <ol style="list-style-type: none"> 1. Jika AdaFileTraining = true <ol style="list-style-type: none"> a. Lakukan pengenalan wajah: PR = recognizer.recognize(Input_image) b. Dapatkan nilai dari hasil pengenalan: Hasil = getLabel(PR.Label) 2. Jika AdaFileTraining = false Hasil = "unknown" • Keluaran : String → Hasil

Sumber: Implementasi

4.5.3 Implementasi Method BukaData

Method BukaData merupakan *method* untuk melakukan proses pelatihan wajah berdasarkan data yang tersimpan pada tabel “wajah”. Proses pengambilan data dilakukan oleh object wajahTableAdapter yang berfungsi untuk melakukan manipulasi data pada tabel wajah. Pembacaan data dari objek wajahTableAdapter dilakukan oleh DataReader. Proses pelatihan dilakukan oleh objek EigenFaceRecognizer dari *library* EmguCV.

Tabel 4.6 Implementasi *method* BukaData

Elemen	Keterangan
Nama	BukaData
Deklarasi	<ul style="list-style-type: none"> • wajahTableAdapter → adapter • integer → num • IDataReader → reader • EigenFaceRecognizer → recognizer • List<integer> → nomorLabel • List<Image> → gambarTraining
Procedure	<ul style="list-style-type: none"> • Masukan: - • Proses <ol style="list-style-type: none"> 1. Buat data reader dari data yang diperoleh dari adapter: reader = adapter.GetData().CreateDataReader() 2. Lakukan pembacaan data tiap record: While (reader.Read()) <ol style="list-style-type: none"> a. Masukkan id ke dalam daftar nomorLabel: nomorLabel.Add((int) reader.GetInt64(2)) b. Masukkan nama file gambar ke dalam daftar gambarTraining: gambarTraining.Add(new Image<Gray, byte>(Application.StartupPath + "\\wajah\\" + reader.GetString(1))) c. Increment nilai num: num = num + 1 3. Jika num > 0 maka <ol style="list-style-type: none"> a. lakukan training data wajah:

```
recognizer.Train(gambarTraining.ToArray(),
    nomorLabel.ToArray())
b. hasil = true
4. Jika num = 0 maka hasil = false
• Keluaran : Boolean → hasil
```

Sumber: Implementasi

4.5.4 Implementasi Method simpanWajah

Method simpanWajah merupakan *method* untuk melakukan proses menyimpan gambar wajah dan menyimpan nama *file* gambar ke dalam *database*. Di dalam *method* simpanWajah terdapat beberapa operasi *file* yang digunakan untuk menyimpan gambar wajah. Nama *file* gambarWajah disimpan dengan format “wajah_idPengunjung_nomor acak”.

Tabel 4.7 Implementasi *method* simpanWajah

Elemen	Keterangan
Nama	simpanWajah
Deklarasi	<ul style="list-style-type: none"> • Random → rand • Boolean → filebaru • String → namafile • String → idpengunjung • wajahTableAdapter → adapter
Procedure	<ul style="list-style-type: none"> • Masukan: gambarWajah • Proses <ol style="list-style-type: none"> 1. Membuat objek Random: rand = new Random() 2. Memberi nilai filebaru = true untuk menandakan bahwa akan dibuat nama file yang baru 3. Memberi namafile sesuai dengan format: namafile = "wajah_" + txtNama.Text + "_" + rand.Next().ToString() + ".jpg" 4. Melakukan pemeriksaan namafile dengan file-file yang ada pada folder penyimpanan untuk menghindari duplikasi namafile: while (filebaru) <ol style="list-style-type: none"> a. Jika tidak terdapat file dengan nama yang sama dengan namafile: if (!File.Exists(namafile)) Berikan nilai filebaru = false b. Jika terdapat file dengan nama yang sama dengan namafile, maka beri nilai namafile dengan nomor acak yang baru: namafile = "wajah_" + txtNama.Text + "_" + rand.Next().ToString() + ".jpg" 5. Jika (Directory.exist("wajah")=false) Directory.CreateDirectory("wajah") 6. gambarWajah.Save(Application.StartupPath + "/wajah/" + namafile, ImageFormat.Jpeg)



	<p>7. wajahTableAdapter.simpanData(idpengunjung, namafile)</p> <ul style="list-style-type: none"> • Keluaran : -
--	---

Sumber: Implementasi

4.5.5 Implementasi Method ProsesCapture

Method ProsesCapture merupakan *method* untuk melakukan proses deteksi dan pengenalan wajah pada setiap citra yang ditangkap oleh kamera. Pada setiap gambar yang tertangkap oleh kamera, citra akan diubah menjadi *grayscale* dan dilakukan pemrosesan citra menggunakan fungsi fungsi yang ada pada EmguCV. Citra wajah yang terdeteksi kemudian akan dilakukan proses pengenalan wajah dan akan dimpan ke dalam log.

Tabel 4.8 Implementasi *method* ProsesCapture

Elemen	Keterangan
Nama	ProsesCapture
Deklarasi	<ul style="list-style-type: none"> • Image → gambarWarna • Image → gambarGrayscale • Capture → capturer • Rectangle[] → wajahTerdeteksi • String → name • Image → hasil • PengenalanWajah → pengenalanWajah • PictureBox → gambarKamera
Procedure	<ul style="list-style-type: none"> • Masukan: - • Proses <ol style="list-style-type: none"> 1. Mengambil citra dari kamera: gambarWarna = capturer.QueryFrame() 2. Mengubah gambarWarna menjadi grayscale: gambarGrayscale = gambarWarna.Convert<Gray, Byte>() 3. Melakukan deteksi wajah: wajahTerdeteksi = detektorWajah.DetectMultiScale() 4. Pada setiap wajah yang terdeteksi lakukan: ForEach(wajahTerdeteksi → wajah) <ol style="list-style-type: none"> a. Mengambil gambar wajah: hasil = gambarGrayscale.Copy(wajah) b. Melakukan proses histogram equalization terhadap hasil: hasil.EqualizeHist() c. Melakukan pengenalan wajah terhadap hasil: name = pengenalanWajah.Recognize(hasil) d. Jika nama ≠ null dan nama ≠ "unknown" <ul style="list-style-type: none"> - Gambar kotak terhadap wajah yang dikenali: gambarWarna.Draw(wajah, new Bgr(Color.Green), 2)

```

- Menampilkan id dari wajah yang dikenali:
gambarWarna.Draw(name)
e. Jika nama = null dan nama = "unknown"
- Gambar kotak terhadap wajah yang tidak
dikenali:
gambarWarna.Draw(wajah, new
Bgr(Color.Red), 1)
- Menampilkan tulisan "unknown":
gambarWarna.Draw("unknown")
5. Tampilkan gambarWarna pada PictureBox:
gambarKamera.Image = gambarWarna.ToBitmap()
• Keluaran : -

```

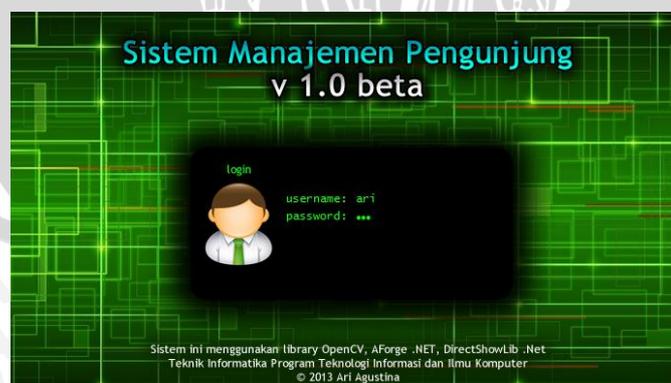
Sumber: Implementasi

4.6 Implementasi Antarmuka

Antarmuka aplikasi Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah digunakan oleh pengguna untuk berinteraksi dengan sistem perangkat lunak. Antarmuka pengguna untuk Sistem Manajemen Pengunjung terdiri dari halaman Login, halaman Utama, halaman Data Pengunjung, halaman Latih Wajah, halaman Latih Wajah *Unknown*, halaman Daftar Pencarian Pengunjung, halaman Daftar Pengguna, dan halaman Pengaturan *motion*.

4.6.1 Halaman Login

Halaman login merupakan halaman yang berfungsi untuk mempermudah pengguna dalam melakukan login dan untuk menyeleksi *administrator* dan pemantau. Gambar 4.2 menunjukkan implementasi antarmuka dari halaman login.

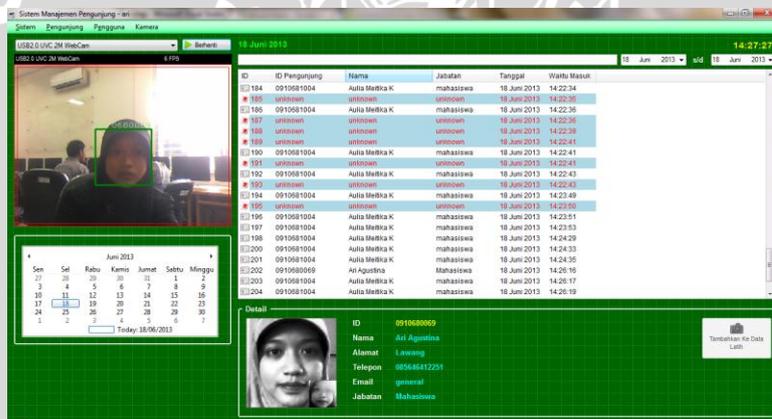


Gambar 4.2 Implementasi Antarmuka Halaman Login

Sumber: Implementasi

4.6.2 Halaman Utama

Halaman utama merupakan halaman yang pertama kali ditampilkan setelah pengguna melakukan *login*. Halaman utama berisi panel untuk menampilkan gambar dari kamera dan daftar *log*. Halaman ini juga dapat menampilkan data pengunjung secara detail yang dipilih dari data *log*. Selain itu, halaman ini juga berisi menu-menu untuk mengelola aplikasi Sistem Manajemen Pengunjung. Halaman utama aplikasi Sistem Manajemen Pengunjung ini terbagi menjadi dua, yaitu halaman utama untuk *administrator* dan halaman utama untuk pemantau. Perbedaan antara kedua halaman tersebut hanya pada menu Pengguna, dimana Pemantau tidak dapat mengelola data pengguna. Gambar 4.3 menunjukkan implementasi antarmuka pengguna dari halaman utama *administrator*. Sedangkan Gambar 4.4 menunjukkan implementasi antarmuka pengguna dari halaman utama pemantau.



Gambar 4.3 Implementasi Antarmuka Halaman Utama Administrator
Sumber: Implementasi

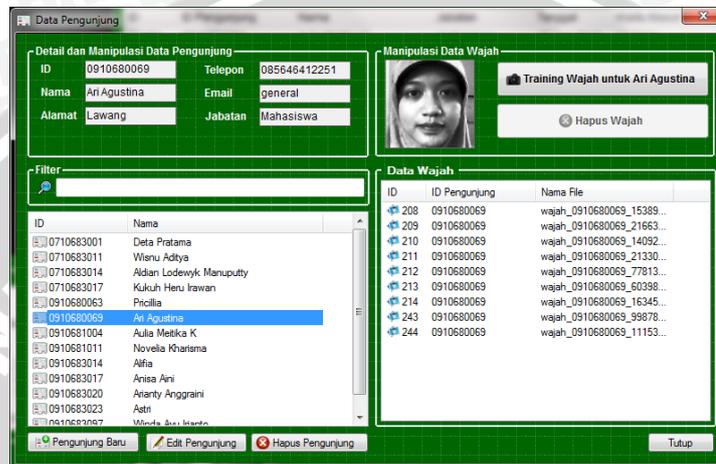


Gambar 4.4 Implementasi Antarmuka Halaman Utama Pemantau
Sumber: Implementasi



4.6.3 Halaman Data Pengunjung

Halaman Data Pengunjung merupakan halaman yang digunakan untuk mengelola data pengunjung. Halaman ini berisi data lengkap pengunjung dan data latih pengunjung. Gambar 4.5 menunjukkan implementasi antarmuka pengguna dari halaman Data Pengunjung.



Gambar 4.5 Implementasi Antarmuka Halaman Data Pengunjung
Sumber: Implementasi

4.6.4 Halaman Latih Wajah

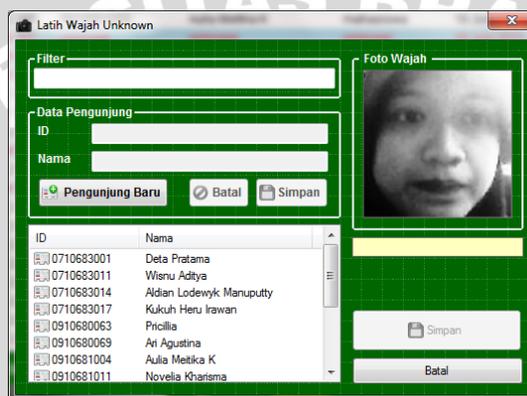
Halaman Latih Wajah merupakan halaman yang digunakan untuk mengelola data latih wajah pengunjung. Gambar 4.6 menunjukkan implementasi antarmuka pengguna dari halaman "Latih Wajah".



Gambar 4.6 Implementasi Antarmuka Halaman Latih Wajah
Sumber: Implementasi

4.6.5 Halaman Latih Wajah Unknown

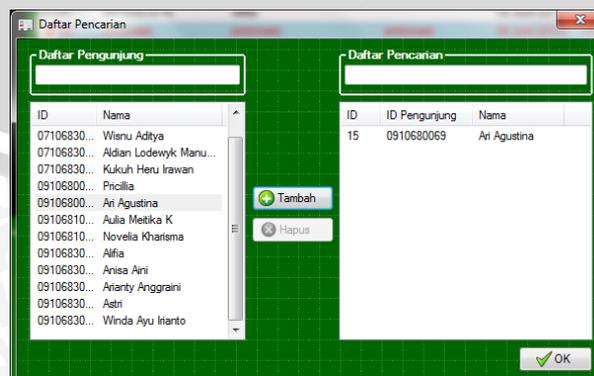
Halaman Latih Wajah Unknown merupakan halaman yang digunakan untuk mengelola data latih wajah pengunjung yang masih belum dikenali oleh sistem. Halaman ini berisi tabel data pengunjung yang berisi ID dan nama pengunjung. Data wajah *unknown* yang dipilih pada tabel Log di halaman Utama dapat ditambah ke dalam data latih pengunjung dengan cara memilih data pengunjung yang ada pada halaman Latih Wajah Unknown. Gambar 4.7 menunjukkan implementasi antarmuka pengguna dari halaman “Latih Wajah Unknown”.



Gambar 4.7 Implementasi Antarmuka Halaman Latih Wajah Unknown
Sumber: Implementasi

4.6.6 Halaman Daftar Pencarian Pengunjung

Halaman daftar pencarian pengunjung merupakan halaman yang digunakan untuk mengelola daftar pencarian pengunjung. Halaman ini dapat menambah atau menghapus data pengunjung yang akan dicari. Gambar 4.8 menunjukkan implementasi antarmuka pengguna dari halaman Daftar Pencarian Pengunjung.



Gambar 4.8 Implementasi Antarmuka Halaman Daftar Pencarian Pengunjung
Sumber: Implementasi

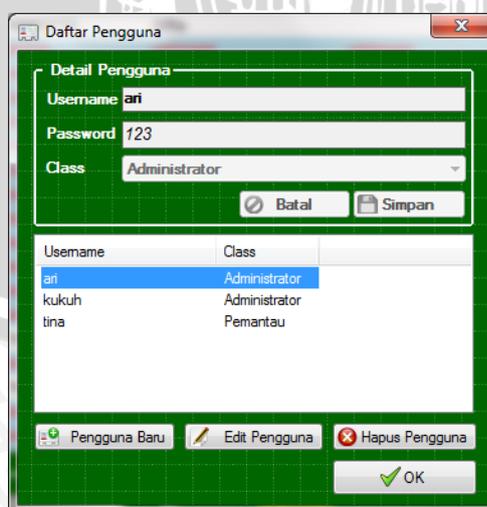
Setelah menambahkan data pengunjung yang ingin dicari, sistem akan memeriksa daftar pencarian pada setiap pengunjung yang dikenali. Jika pengunjung yang dikenali terdapat pada daftar pencarian pengunjung, maka sistem akan menampilkan pesan pemberitahuan. Pesan pemberitahuan ditunjukkan pada Gambar 4.9.



Gambar 4.9 Implementasi Antarmuka Tampilan Pesan Pemberitahuan
Sumber: Implementasi

4.6.7 Halaman Daftar Pengguna

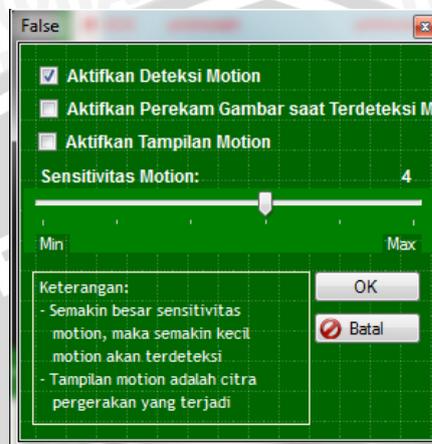
Halaman Daftar Pengguna merupakan halaman yang digunakan untuk mengelola data pengguna yaitu *administrator* dan pemantau pada aplikasi Sistem Manajemen Pengunjung. Gambar 4.10 menunjukkan implementasi antarmuka pengguna dari halaman “Daftar Pengguna”.



Gambar 4.10 Implementasi Antarmuka Halaman Daftar Pengguna
Sumber: Implementasi

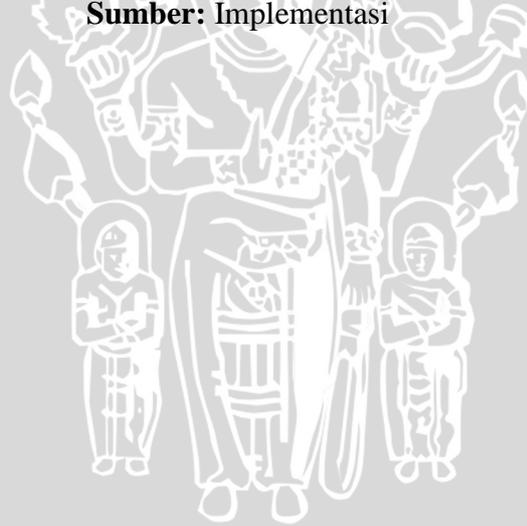
4.6.8 Halaman Setting Motion

Halaman *setting motion* merupakan halaman yang digunakan untuk mengatur deteksi *motion*. Gambar 4.11 menunjukkan implementasi antarmuka pengguna dari halaman “Setting Motion”.



Gambar 4.11 Implementasi Antarmuka Halaman Setting Motion

Sumber: Implementasi



BAB V

PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan proses pengujian dan analisis terhadap aplikasi Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah yang telah dikembangkan. Proses pengujian dilakukan melalui empat tahapan (strategi) yaitu pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian akurasi. Pada pengujian unit dan pengujian integrasi akan digunakan teknik pengujian *White Box (White Box Testing)*. Pada pengujian validasi digunakan teknik pengujian *Black Box (Black Box Testing)*. Pengujian akurasi digunakan untuk mengetahui prosentase ketepatan mengenali wajah dari aplikasi Sistem Manajemen Pengunjung.

5.1 Pengujian

Proses pengujian dilakukan melalui empat tahapan (strategi) yaitu pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian akurasi.

5.1.1 Pengujian Unit

Pada pengujian unit aplikasi Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah digunakan teknik *White-Box Testing* dengan teknik *Basis Path Testing*. Pada teknik *Basis Path Testing*, proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*, menentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*), menentukan sebuah basis set dari jalur independen dan memberikan kasus uji (*test case*) pada setiap basis set yang telah ditentukan. Pada penulisan laporan skripsi ini hanya dicantumkan hasil pengujian unit untuk algoritma dari beberapa metode (operasi) saja (tidak untuk keseluruhan metode).

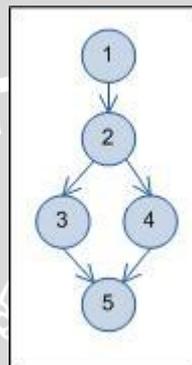
5.1.1.1 Pengujian Unit untuk Method getLabel

Tabel 5.1 Pengujian Unit untuk Method getLabel

Elemen	Keterangan	Node
Nama	getLabel	
Deklarasi	• wajahTableAdapter → adapter	

	<ul style="list-style-type: none"> String → label 	
Procedure	<ul style="list-style-type: none"> Masukan: id Proses <ol style="list-style-type: none"> Set label = adapter.getLabel(id) Jika label ≠ "" maka Hasil = label else Hasil = "unknown" Keluaran : String → Hasil 	

Sumber : Pengujian dan Analisis



Gambar 5.1 Flow Graph Method getLabel

Sumber : Pengujian dan Analisis

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap *method* `getLabel()` menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 5 - 3 + 2 \\
 &= 4
 \end{aligned}$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka didapatkan sebuah basis set dari jalur *independent*, yaitu :

Jalur 1 : 1 – 2 – 3 – 5

Jalur 2 : 1 – 2 – 4 – 5

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 5.2.



Tabel 5.2 Test Case untuk Pengujian Unit Method getLabel

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Jika getLabel dieksekusi dengan input label = "id pengunjung"	Hasil yang dikeluarkan = label.	Hasil yang dikeluarkan = label.
2	Jika getLabel dieksekusi dengan input label = "" (kosong)	Hasil yang dikeluarkan = unknown.	Hasil yang dikeluarkan = unknown.

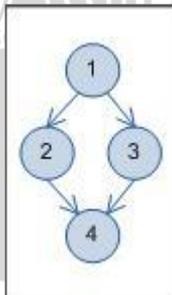
Sumber : Pengujian dan Analisis

5.1.1.2 Pengujian Unit untuk Method Recognize

Tabel 5.3 Pengujian Unit untuk Method Recognize

Elemen	Keterangan	Node
Nama	Recognize	
Deklarasi	<ul style="list-style-type: none"> EigenFaceRecognizer → recognizer PredictionResult → PR Boolean → AdaFileTraining 	
Procedure	<ul style="list-style-type: none"> Masukan: Input_image Proses <ol style="list-style-type: none"> Jika AdaFileTraining = true maka <ol style="list-style-type: none"> PR = recognizer.recognize(Input_image) Hasil = getLabel(PR.Label) else Hasil = "unknown" Keluaran : String → Hasil 	

Sumber : Pengujian dan Analisis

**Gambar 5.2** Flow Graph Method Recognize

Sumber : Pengujian dan Analisis

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap *method* Recognize() menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan

jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 4 - 4 + 2 \\ &= 2 \end{aligned}$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka didapatkan sebuah basis set dari jalur *independent*, yaitu :

Jalur 1 : 1 – 2 – 4

Jalur 2 : 1 – 3 – 4

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 5.4.

Tabel 5.4 Test Case untuk Pengujian Unit Method Recognize

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Jika ada <i>file</i> latih dalam <i>database</i>	Hasil yang dikeluarkan = label.	Hasil yang dikeluarkan = label.
2	Jika tidak ada <i>file</i> latih dalam <i>database</i>	Hasil yang dikeluarkan = unknown.	Hasil yang dikeluarkan = unknown.

Sumber : Pengujian dan Analisis

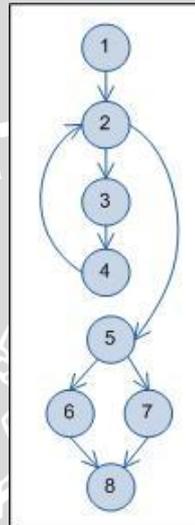
5.1.1.3 Pengujian Unit untuk Method BukaData

Tabel 5.5 Pengujian Unit untuk Method BukaData

Elemen	Keterangan	Node
Nama	BukaData	
Deklarasi	<ul style="list-style-type: none"> wajahTableAdapter → adapter integer → num IDataReader → reader EigenFaceRecognizer → recognizer List<integer> → nomorLabel List<Image> → gambarTraining 	
Procedure	<ul style="list-style-type: none"> Masukan: - Proses <ol style="list-style-type: none"> Set reader = adapter.GetData().CreateDataReader() Selama reader membaca: While (reader.Read()) <ol style="list-style-type: none"> nomorLabel.Add((int) reader.GetInt64(2)) gambarTraining.Add(new Image<Gray, byte>(Application.StartupPath + "\\wajah\\" + reader.GetString(1))) num = num + 1 End While 	<p>1</p> <p>2</p> <p>3</p> <p>4</p>

3. Jika num > 0 maka	5
a. recognizer.Train(gambarTraining.ToArray(), nomorLabel.ToArray())	6
b. hasil = true	
4. else	7
hasil = false	
• Keluaran : Boolean → hasil	8

Sumber : Pengujian dan Analisis



Gambar 5.3 Flow Graph Method BukaData

Sumber : Pengujian dan Analisis

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap *method* BukaData () menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 9 - 8 + 2 \\
 &= 3
 \end{aligned}$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka didapatkan sebuah basis set dari jalur *independent*, yaitu :

Jalur 1 : 1 – 2 – 3 – 4 – 2 – ...

Jalur 2 : 1 – 2 – 5 – 6 – 8

Jalur 3 : 1 – 2 – 5 – 7 – 8



Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 5.6.

Tabel 5.6 Test Case Pengujian Unit untuk Method BukaData

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Jika terdapat data dalam database : (while(reader.read()))	Menambah nomor label, gambar latih, increment number	Menambah nomor label, gambar latih, increment number
2	Jika number > 0	Melakukan pelatihan wajah. Hasil yang dikeluarkan = true.	Melakukan pelatihan wajah. Hasil yang dikeluarkan = true.
3	Jika number <= 0	Hasil = false	Hasil = false

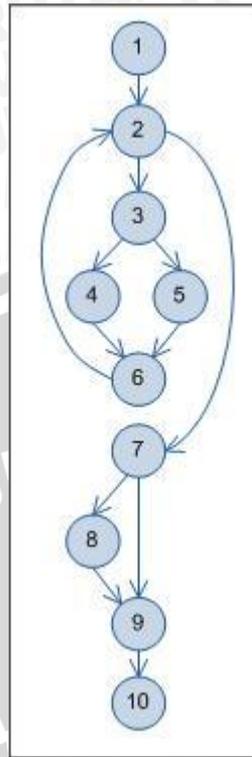
Sumber : Pengujian dan Analisis

5.1.1.4 Pengujian Unit untuk Method simpanWajah

Tabel 5.7 Pengujian Unit untuk Method simpanWajah

Elemen	Keterangan	Node
Nama	simpanWajah	
Deklarasi	<ul style="list-style-type: none"> Random → rand Boolean → filebaru String → namafile String → idpengunjung wajahTableAdapter → adapter 	
Procedure	<ul style="list-style-type: none"> Masukan: gambarWajah Proses <ol style="list-style-type: none"> rand = new Random() filebaru = true namafile = "wajah_" + txtNama.Text + "_" + rand.Next().ToString() + ".jpg" while (filebaru) <ol style="list-style-type: none"> Jika (!File.Exists(namafile)) filebaru = false else namafile = "wajah_" + txtNama.Text + "_" + rand.Next().ToString() + ".jpg" end while jika (Directory.exist("wajah")) Directory.CreateDirectory("wajah") gambarWajah.Save(Application.StartupPath + "/wajah/" + namafile, ImageFormat.Jpeg) wajahTableAdapter.simpanData(idpengunjung, namafile) Keluaran : - 	<p>10</p>

Sumber : Pengujian dan Analisis



Gambar 5.4 Flow Graph Method simpanWajah
Sumber : Pengujian dan Analisis

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap *method* simpanWajah () menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 12 - 10 + 2 \\
 &= 4
 \end{aligned}$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka didapatkan sebuah basis set dari jalur *independent*, yaitu :

Jalur 1 : 1 – 2 – 3 – 4 – 6 – 2 – ...

Jalur 2 : 1 – 2 – 3 – 5 – 6 – 2 – ...

Jalur 3 : 1 – 2 – 7 – 8 – 9 – 10

Jalur 4 : 1 – 2 – 7 – 9 – 10

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 5.8.

Tabel 5.8 Test Case untuk Pengujian Unit Method Simpan Wajah

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Selama filebaru = true dan tidak terdapat file yang sama	filebaru = false	filebaru = false
2	Selama filebaru = true dan terdapat file yang sama	filebaru = true	filebaru = true
3	Jika tidak terdapat direktori "wajah"	Buat direktori "wajah"	Buat direktori "wajah"
4	Jika terdapat direktori "wajah"	Simpan gambar ke direktori "wajah"	Simpan gambar ke direktori "wajah"

Sumber : Pengujian dan Analisis

5.1.2 Pengujian Integrasi

Pengujian integrasi diterapkan pada proses yang mengintegrasikan fungsionalitas dari beberapa *class* untuk melakukan sebuah operasi tertentu. Pada pengujian integrasi yang dijadikan sebagai objek uji adalah *class* inti yang menggabungkan kinerja dari *class – class* lain. Pengujian integrasi yang dilakukan terhadap *class* inti ini menggunakan strategi *bottom-up*, dimana modul - modul yang diintegrasikan masing - masing diuji terlebih dahulu dalam pengujian unit dan kemudian bergerak menuju ke pengujian modul - modul kontrol yang mengintegrasikannya.

5.1.2.1 Pengujian Integrasi untuk Method ProsesCapture

Tabel 5.9 Pengujian Integrasi untuk Method ProsesCapture

Elemen	Keterangan	Node
Nama	ProsesCapture	
Deklarasi	<ul style="list-style-type: none"> • Image → gambarWarna • Image → gambarGrayscale • Capture → capturer • Rectangle[] → wajahTerdeteksi • String → name • Image → hasil • PengenalanWajah → pengenalanWajah • PictureBox → gambarKamera 	
Procedure	<ul style="list-style-type: none"> • Masukan: - • Proses 	

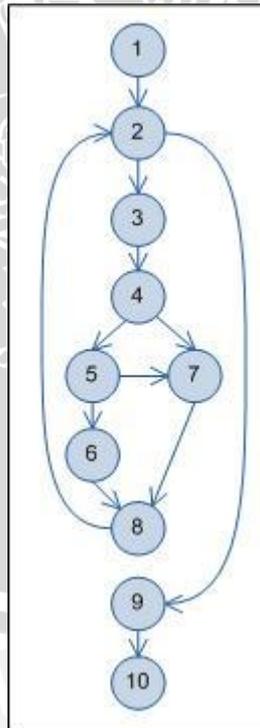
```

1. gambarWarna = capturer.QueryFrame()
2. gambarGrayscale =
   gambarWarna.Convert<Gray, Byte>()
3. wajahTerdeteksi =
   detektorWajah.DetectMultiScale()
4. ForEach (wajahTerdeteksi → wajah)
   a. hasil = gambarGrayscale.Copy(wajah)
   b. hasil._EqualizeHist()
   c. name =
      pengenalanWajah.Recognize(hasil)
   d. Jika nama ≠ null dan nama ≠
      "unknown"
      - gambarWarna.Draw(wajah, new
        Bgr(Color.Green), 2)
      - gambarWarna.Draw(name)
   e. else
      - gambarWarna.Draw(wajah, new
        Bgr(Color.Red), 1)
      - gambarWarna.Draw("unknown")
   end For
5. gambarKamera.Image =
   gambarWarna.ToBitmap()

```

• Keluaran : -

Sumber : Pengujian dan Analisis



Gambar 5.5 Flow Graph Method Proses Capture

Sumber : Pengujian dan Analisis

Pemodelan ke dalam *flow graph* yang telah dilakukan terhadap *method* `ProsesCapture()` menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 12 - 10 + 2 \\ &= 4 \end{aligned}$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka didapatkan sebuah basis set dari jalur *independent*, yaitu :

Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 8 – 2 – ...

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 7 – 8 – 2 – ...

Jalur 3 : 1 – 2 – 3 – 4 – 7 – 8 – 2 – ...

Jalur 4 : 1 – 2 – 9 – 10

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada Tabel 5.10.

Tabel 5.10 Test Case untuk Pengujian Integrasi Method `ProsesCapture`

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Selama wajah terdeteksi, jika nama \neq null dan nama \neq unknown	Hasil yang dikeluarkan = muncul kotak warna hijau dan id pengunjung.	Hasil yang dikeluarkan = muncul kotak warna hijau dan id pengunjung.
2	Selama wajah terdeteksi, jika nama \neq null dan nama = unknown	Hasil yang dikeluarkan = muncul kotak warna merah dan nama unknown.	Hasil yang dikeluarkan = muncul kotak warna merah dan nama unknown.
3	Selama wajah terdeteksi, jika nama = null dan nama = unknown	Hasil yang dikeluarkan = muncul kotak warna merah dan nama unknown.	Hasil yang dikeluarkan = muncul kotak warna merah dan nama unknown.
4	Selama wajah terdeteksi	Menampilkan gambar dari kamera.	Menampilkan gambar dari kamera.

Sumber : Pengujian dan Analisis

5.1.3 Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Item - item yang telah

dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *Black Box*, karena tidak diperlukan konsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan daftar kebutuhan. Pada skripsi ini dilakukan pengujian validasi terhadap Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah.

5.1.3.1 Kasus Uji

a. Kasus Uji untuk Login Sukses

Tabel 5.11 Kasus Uji Login

Nama Kasus Uji	Kasus Uji Login
Objek Uji	SRS_001_01
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas <i>login</i> bagi pengguna.
Prosedur Uji	1. Pengguna memasukkan <i>username</i> dan <i>password</i> di dalam <i>form login</i> . 2. Pengguna menekan tombol Login.
Hasil yang diharapkan	Aplikasi dapat melakukan penyeleksian kondisi <i>login</i> pada <i>database</i> berdasar data yang dimasukkan dan jika penyeleksian kondisi ini benar, maka pengguna dapat mengakses halaman utama.

Sumber : Pengujian dan Analisis

b. Kasus Uji Mengelola Data Pengunjung

Tabel 5.12 Kasus Uji Mengelola Data Pengunjung

Nama Kasus Uji	Kasus Uji Mengelola Data Pengunjung
Objek Uji	SRS_001_02
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas untuk mengelola data pengunjung bagi pengguna.
Prosedur Uji	1. Aplikasi dijalankan pada keadaan sudah <i>login</i> 2. Pengguna memilih menu Pengunjung pada halaman utama. 3. Pengguna memilih submenu Data Pengunjung. 4. <i>Form</i> data pengunjung tampil. 5. Pengguna menekan tombol Data Baru kemudian

	<p>mengisi <i>field</i> ID (nim), nama, alamat, telepon, email, dan jabatan. Selanjutnya pengguna menekan tombol Simpan.</p> <p>6. Pengguna memilih data pengunjung yang akan diubah pada tabel dan menekan tombol Update Data. Pengguna mengubah isi <i>field</i> data kemudian menekan tombol Simpan.</p> <p>7. Pengguna memilih data pengunjung yang akan dihapus pada tabel dan menekan tombol Hapus Data.</p>
Hasil yang diharapkan	Aplikasi menampilkan data pengunjung pada tabel, melakukan penambahan data, perubahan data serta penghapusan data pengunjung sesuai dengan operasi yang dipilih oleh Pengguna.

Sumber : Pengujian dan Analisis

c. Kasus Uji Mengelola Data Wajah

Tabel 5.13 Kasus Uji Mengelola Data Wajah

Nama Kasus Uji	Kasus Uji Mengelola Data Wajah
Objek Uji	SRS_001_03
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas untuk mengelola data wajah bagi pengguna.
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna memilih menu Pengunjung pada halaman utama. 2. Pengguna memilih submenu Data Pengunjung pada menu pengunjung. 3. <i>Form</i> data pengunjung tampil. 4. Pengguna memilih data pengunjung pada tabel. 5. Pengguna menekan tombol Training Wajah untuk menambah data wajah pengunjung. 6. Pengguna memilih data wajah yang akan dihapus dengan menekan tombol Hapus Waja.
Hasil yang diharapkan	Aplikasi menampilkan data wajah pada tabel, melakukan penambahan data serta penghapusan data wajah sesuai dengan operasi yang dipilih oleh pengguna.

Sumber : Pengujian dan Analisis

d. Kasus Uji Mengelola Data Daftar Pencarian Pengunjung

Tabel 5.14 Kasus Uji Mengelola Data Pencarian Pengunjung

Nama Kasus Uji	Kasus Uji Mengelola Data Pencarian Pengunjung
Objek Uji	SRS_001_04
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas untuk mengelola data daftar pencarian pengunjung bagi pengguna.
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna memilih menu Pengunjung pada halaman utama. 2. Pengguna memilih submenu Daftar Pencarian Pengunjung. 3. <i>Form</i> daftar pencarian pengunjung tampil. 4. Pengguna memilih data nama pengunjung lalu menekan tombol Tambah untuk menambah data daftar pengunjung yang akan dicari. 5. Pengguna memilih data nama pengunjung lalu menekan tombol Hapus untuk menghapus data daftar pengunjung.
Hasil yang diharapkan	Aplikasi menampilkan data daftar pencarian pengunjung pada tabel, melakukan penambahan data serta penghapusan data pencarian pengunjung sesuai dengan operasi yang dipilih oleh pengguna.

Sumber : Pengujian dan Analisis

e. Kasus Uji Mengelola Data Pengguna

Tabel 5.15 Kasus Uji Mengelola Data Pengguna

Nama Kasus Uji	Kasus Uji Mengelola Data Pengguna
Objek Uji	SRS_002_01
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas untuk mengelola data pengguna bagi <i>administrator</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Administrator</i> memilih menu Pengguna pada halaman utama. 2. <i>Form</i> data pengguna tampil. 3. <i>Administrator</i> menekan tombol Pengguna Baru kemudian mengisi field username, password, dan class. Selanjutnya pengguna menekan tombol Simpan. 4. <i>Administrator</i> memilih data pengguna yang akan diubah pada tabel dan menekan tombol Edit Pengguna. <i>Administrator</i> mengubah isi field data

	<p>kemudian menekan tombol Simpan.</p> <p>5. <i>Administrator</i> memilih data pengguna yang akan dihapus pada tabel dan menekan tombol Hapus Pengguna.</p>
Hasil yang diharapkan	Aplikasi menampilkan data pengguna pada tabel, melakukan penambahan data, pengubahan data serta penghapusan data pengguna sesuai dengan operasi yang dipilih oleh <i>administrator</i> .

Sumber : Pengujian dan Analisis

f. Kasus Uji Menghadap Kamera

Tabel 5.16 Kasus Uji Menghadap Kamera

Nama Kasus Uji	Kasus Uji Menghadap Kamera
Objek Uji	SRS_003_01
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam mendeteksi wajah dari pengunjung yang menghadap kamera.
Prosedur Uji	Pengunjung datang ke laboratorium dengan menghadap ke arah kamera.
Hasil yang diharapkan	Aplikasi menampilkan id pengunjung dan menampilkan data pengunjung pada log.

Sumber : Pengujian dan Analisis

5.1.3.2 Hasil Pengujian Validasi

Tabel 5.17 Hasil Pengujian Validasi

No.	Nama Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status Validitas
1	Kasus Uji untuk Login Sukses	Aplikasi dapat melakukan penyeleksian kondisi <i>login</i> pada <i>database</i> berdasar data yang dimasukkan dan jika penyeleksian kondisi ini benar, maka pengguna dapat mengakses halaman utama.	Aplikasi dapat melakukan penyeleksian kondisi <i>login</i> pada <i>database</i> berdasar data yang dimasukkan dan jika penyeleksian kondisi ini benar, maka pengguna dapat mengakses halaman utama.	Valid
2	Kasus Uji Mengelola Data Pengunjung	Aplikasi menampilkan data pengunjung pada tabel, melakukan penambahan data, pengubahan data serta penghapusan data pengunjung sesuai	Aplikasi menampilkan data pengunjung pada tabel, melakukan penambahan data, pengubahan data serta penghapusan data pengunjung sesuai	Valid

		dengan operasi yang dipilih oleh pengguna.	dengan operasi yang dipilih oleh pengguna.	
3	Kasus Uji Mengelola Data Wajah	Aplikasi menampilkan data wajah pada tabel, melakukan penambahan data serta penghapusan data wajah sesuai dengan operasi yang dipilih oleh pengguna.	Aplikasi menampilkan data wajah pada tabel, melakukan penambahan data serta penghapusan data wajah sesuai dengan operasi yang dipilih oleh pengguna.	Valid
4	Kasus Uji Mengelola Data Daftar Pencarian Pengunjung	Aplikasi menampilkan data daftar pencarian pengunjung pada tabel, melakukan penambahan data serta penghapusan data pencarian pengunjung sesuai dengan operasi yang dipilih oleh pengguna.	Aplikasi menampilkan data daftar pencarian pengunjung pada tabel, melakukan penambahan data serta penghapusan data pencarian pengunjung sesuai dengan operasi yang dipilih oleh pengguna.	Valid
5	Kasus Uji Mengelola Data Pengguna	Aplikasi menampilkan data pengguna pada tabel, melakukan penambahan data, pengubahan data serta penghapusan data pengguna sesuai dengan operasi yang dipilih oleh administrator.	Aplikasi menampilkan data pengguna pada tabel, melakukan penambahan data, pengubahan data serta penghapusan data pengguna sesuai dengan operasi yang dipilih oleh administrator.	Valid
6	Kasus Uji Menghadap Kamera	Aplikasi menampilkan id pengunjung dan menampilkan data pengunjung pada log.	Aplikasi menampilkan id pengunjung dan menampilkan data pengunjung pada log.	Valid

Sumber : Pengujian dan Analisis

5.1.4 Pengujian Akurasi

Untuk menguji performa pengenalan wajah dari sistem, maka dilakukan pengujian akurasi, sehingga pengenalan wajah dari sistem dikatakan memiliki performa tinggi apabila output yang dihasilkan sesuai dengan data yang telah disimpan pada *database*.

Untuk menentukan akurasi dari pengenalan wajah Sistem Manajemen Pengunjung ini, terlebih dahulu simpan citra wajah pengunjung yang akan digunakan sebagai pengenalan dalam direktori 'wajah'. Citra wajah yang disimpan terdiri dari 7 pose yang berbeda setiap orang. Untuk pengambilan gambar citra

wajah pengunjung yang akan dikenali pada saat pengunjung masuk ke dalam laboratorium, terdapat beberapa kondisi yang biasanya dilakukan, yaitu diantaranya:

1. Pose wajah standart yaitu citra wajah yang disimpan memiliki pose lurus ke depan, menghadap ke samping kanan dan samping kiri dengan ekspresi standart.
2. Cahaya saat pengambilan gambar citra berada pada rata-rata tingkat keabuan lebih besar sama dengan 100 untuk skala 0 sampai 255. Persentase
3. Jarak kamera dan wajah yang diambil dibuat konstan (± 2 meter).

Tabel 5.18 Data Latih Pengunjung

No	Nama	Citra Training
1	Deta Pratama	
2	Wisnu Aditya	
3	Aldian Lodewyk M.	
4	Kukuh Heru Irawan	
5	Ari Agustina	
6	Aulia Meitika K.	
7	Alifia	
8	Anisa Aini	
9	Arianty Anggraini	
10	Astri	

Sumber : Pengujian dan Analisis

5.1.4.1 Hasil Pengujian Akurasi

Untuk pengujian akurasi dilakukan percobaan dengan cara membandingkan citra uji yang telah terdeteksi dengan data latih yang terdiri dari 7 buah untuk masing-masing orang. Proses pengenalan wajah dilakukan pada halaman utama aplikasi. Hasil pengujian citra wajah pengunjung ditampilkan pada tabel berikut :

Tabel 5.19 Hasil Pengujian Akurasi

No	Nama	Citra Tertangkap	Prediksi	Waktu	Keterangan
1	Kukuh		Alifia 	15:46:45	Salah
			Kukuh 	15:46:45	Benar
			Kukuh 	15:46:47	Benar
			Kukuh 	15:46:49	Benar
2	Deta		Deta 	15:47:52	Benar
			Deta 	15:47:54	Benar
			Deta 	15:47:55	Benar
			Deta 	15:47:58	Benar
			Unknown	15:47:58	Salah
			Deta 	15:47:59	Benar
3	Alifia		Alifia 	15:49:20	Benar
			Alifia 	15:49:22	Benar
			Anisa 	15:49:23	Salah
			Anisa 	15:49:27	Salah



			Alifia 	15:49:27	Benar
4	Wisnu 		Wisnu 	15:50:07	Benar
			Wisnu 	15:50:10	Benar
			Wisnu 	15:50:11	Benar
			Wisnu 	15:50:11	Benar
5	Aldian 		Wisnu 	15:50:52	Salah
			Deta 	15:51:13	Salah
			Aldian 	15:51:13	Benar
			Aldian 	15:51:15	Benar
6	Ari 		Ari 	16:05:17	Benar
			Anisa 	16:05:19	Salah
			Ari 	16:05:21	Benar
			Ari 	16:05:22	Benar
			Ari 	16:05:23	Benar
			Ari 	16:05:25	Benar
			Ari 	16:05:25	Benar
7	Astri 		Unknown 	16:06:51	Salah
			Astri 	16:06:51	Benar
			Deta 	16:06:51	Salah
			Astri 	16:06:52	Benar
			Astri 	16:06:52	Benar
			Astri 	16:06:54	Benar
			Astri 	16:06:57	Benar
			Astri 	16:06:57	Benar

8			Arianty 	16:20:44	Benar
			Deta 	16:20:44	Salah
			Aldian 	16:20:46	Salah
			Arianty 	16:20:48	Benar
			Arianty 	16:20:48	Benar
			Arianty 	16:20:48	Benar
9			Aulia 	16:23:32	Benar
			Aulia 	16:23:34	Benar
			Aulia 	16:23:36	Benar
			Aulia 	16:23:37	Benar
			Ari 	16:28:38	Salah
			Aulia 	16:23:38	Benar
			Aulia 	16:23:38	Benar
10			Anisa 	16:26:24	Benar
			Anisa 	16:26:24	Benar
			Anisa 	16:26:25	Benar
			Aulia 	16:26:26	Salah
			Anisa 	16:26:29	Benar
			Anisa 	16:26:29	Benar
			Anisa 	16:26:30	Benar
			Arianty 	16:26:30	Salah
			Anisa 	16:26:30	Benar
			Anisa 	16:26:30	Benar

			Anisa 	16:27:17	Benar
			Anisa 	16:27:19	Benar
11	Nina : unknown 		Alifia 	11:13:33	Salah
			Unknown	11:13:36	Benar
			Unknown	11:13:36	Benar
			Unknown	11:13:36	Benar
			Unknown	11:13:39	Benar
			Unknown	11:13:41	Benar
			Unknown	11:13:41	Benar
			Unknown	11:13:42	Benar
			Unknown	11:13:42	Benar
		12	Hendro : unknown		Unknown
	Deta 			11:13:52	Salah
	Unknown			11:13:53	Benar
	Aldian 			11:13:54	Salah
	Unknown			11:13:54	Benar
	Unknown			11:13:55	Benar
	Unknown			11:13:57	Benar
	Unknown			11:13:57	Benar
	Deta 			11:13:57	Benar
13	Yugo: unknown		Unknown	11:28:06	Benar

			Unknown	11:28:06	Benar
			Unknown	11:28:06	Benar
			Unknown	11:28:08	Benar
			Unknown	11:28:09	Benar
			Unknown	11:28:11	Benar
			Unknown	11:28:13	Benar
			Wisnu 	11:28:15	Salah
			Unknown	11:28:15	Benar
14	Prima: unknown		Unknown	11:28:29	Benar
			Unknown	11:28:33	Benar
			Unknown	11:28:38	Benar
			Unknown	11:28:40	Benar
15	Ervin: unknown		Unknown	11:40:23	Benar
			Alifia 	11:40:24	Salah
			Unknown	11:40:24	Benar
			Unknown	11:40:24	Benar
			Unknown	11:40:25	Benar
			Deta 	11:40:25	Salah
16	Mayang: unknown		Deta 	12:30:57	Salah
			Unknown	12:30:58	Benar
			Unknown	12:31:00	Benar
			Unknown	12:31:40	Benar

			Unknown	12:31:42	Benar
			Deta 	12:31:44	Salah
			Unknown	12:31:44	Benar
17	Hoyi: unknown		Unknown	12:31:56	Benar
			Unknown	12:31:57	Benar
			Unknown	12:31:59	Benar
			Unknown	12:32:29	Benar
			Unknown	12:32:29	Benar
			Deta 	12:32:31	Salah
			Unknown	12:32:31	Benar
			Unknown	12:32:33	Benar
			Unknown	12:32:35	Benar
			Deta 	12:32:35	Salah
18	Margareth: unknown		Unknown	12:46:52	Benar
			Unknown	12:46:57	Benar
			Unknown	12:46:59	Benar
			Unknown	12:47:01	Benar
			Unknown	12:47:02	Benar
			Unknown	12:47:08	Benar
			Unknown	12:47:09	Benar
			Unknown	12:47:16	Benar
			Deta 	12:47:18	Salah

19	Heni: unknown		Unknown	12:47:29	Benar
			Unknown	12:47:30	Benar
			Deta 	12:47:31	Salah
			Unknown	12:47:51	Benar
20	Cahya: unknown		Unknown	12:48:08	Benar
			Unknown	12:48:16	Benar
			Unknown	12:48:23	Benar
			Unknown	12:48:24	Benar
			Unknown	12:48:44	Benar
			Unknown	12:48:44	Benar
			Unknown	12:48:45	Benar
			Unknown	12:48:46	Benar
			Unknown	12:48:50	Benar

Sumber : Pengujian dan Analisis

Hasil prediksi sistem disusun dalam *confusion matrix* yang ditunjukkan pada Tabel 5.20. Berdasarkan masukan dalam *confusion matrix*, jumlah prediksi yang benar adalah jumlah f_{ij} dimana $i=j$, dan jumlah prediksi yang salah yaitu jumlah f_{ij} dimana $i \neq j$.

Tabel 5.20 Confusion Matrix untuk hasil pengujian akurasi

		Prediksi										
		Kukuh	Deta	Alifia	Wisnu	Aldian	Ari	Astri	Arianty	Aulia	Anisa	unk
Aktual	Kukuh	3		1								
	Deta		5									1
	Alifia			4							1	
	Wisnu				3							
	Aldian		1		1	2						
	Ari						5				1	
	Astri		1					5				1
	Arianty		1			1			3			
	Aulia						1			5		
	Anisa								1		9	
unk		9	2	2	1	1					64	
		Jumlah Prediksi Benar										108
		Jumlah Prediksi Salah										26
		Jumlah Prediksi										134

Sumber : Pengujian dan Analisis

Nilai akurasi dan nilai *error rate* sistem dinilai dengan cara menghitung jumlah prediksi benar, jumlah prediksi salah, dan jumlah total prediksi.

$$accuracy = \frac{\text{number of correct predictions}}{\text{total number of predictions}} = \frac{108}{134} = 0,8059 = 80,59\%$$

$$error\ rate = \frac{\text{number of wrong predictions}}{\text{total number of predictions}} = \frac{26}{134} = 0,1940 = 19,40\%$$

5.1.5 Pengujian Usability

Pengujian ini ditujukan untuk mengetahui seberapa besar Sistem Manajemen Pengunjung ini dapat digunakan dengan baik oleh pengguna dan seberapa besar Sistem Manajemen Pengunjung ini dapat membantu pengguna. Pengujian ini dilakukan dengan memberikan kuisioner kepada laboran. Setiap pernyataan memiliki jawaban “Setuju”, “Cukup”, dan “Tidak Setuju” dengan bobot masing-masing yaitu 1, 0.5, dan 0. Jumlah pengguna yang akan diuji sebanyak 4 orang, yaitu laboran Lab. KCV, Multimedia, dan Komputer dasar, laboran Lab. RPL dan Jaringan Komputer, laboran Lab. Sistem Komputer dan Robotika, dan laboran Lab. Sistem Informasi. Pernyataan yang digunakan pada pengujian ini beserta hasil kuisioner ditunjukkan pada Tabel 5.21.

Tabel 5.21 Pernyataan pengujian *usability*

No	Pernyataan	Jawaban “Setuju”	Jawaban “Cukup”	Jawaban “Tidak Setuju”
1	Secara keseluruhan, Saya puas dengan kemudahan dalam menggunakan sistem ini.	3	1	0
2	Penggunaan sistem ini mudah.	2	2	0
3	Saya dapat melakukan pekerjaan menjadi lebih mudah dengan menggunakan sistem ini.	3	1	0
4	Saya dapat melakukan pekerjaan lebih cepat dengan menggunakan sistem ini.	3	1	0
5	Saya dapat menyelesaikan pekerjaan dengan lebih efisien menggunakan sistem ini.	1	3	0
6	Saya merasa nyaman menggunakan sistem ini.	0	4	0
7	Penggunaan sistem ini dapat dipelajari dengan mudah.	2	2	0
8	Saya percaya dapat menjadi lebih produktif menggunakan sistem ini.	3	1	0
9	Sistem memberikan pesan <i>error</i> dengan jelas yang memberitahukan saya	1	3	0

	bagaimana menyelesaikan masalah.			
10	Setiap kali saya melakukan kesalahan dengan sistem, saya dapat mengembalikan ke keadaan semula dengan mudah dan cepat.	1	3	0
11	Informasi (<i>on-screen message</i>).	2	2	0
12	Mudah dalam mencari informasi yang saya butuhkan.	2	2	0
13	Informasi yang disediakan sistem mudah untuk dimengerti.	4	0	0
14	Informasi yang diberikan dapat membantu saya secara efektif.	4	0	0
15	Organisasi informasi pada tampilan disajikan dengan jelas.	1	3	0
16	Antarmuka sistem <i>user-friendly</i> .	2	2	0
17	Saya suka dengan antarmuka sistem.	1	3	0
18	Sistem memiliki semua fungsi dan kemampuan yang saya inginkan.	1	3	0
19	Secara keseluruhan, saya puas dengan sistem ini.	2	2	0

Sumber : Pengujian dan Analisis

Dari pernyataan pada Tabel 5.21 di atas didapatkan 4 sampel dengan total 76 jawaban pengujian. Total dari hasil pengujian *usability* ini ditunjukkan pada Tabel 5.22.

Tabel 5.22 Pernyataan pengujian *usability*

No	Jawaban	Jumlah
1	Setuju	38
2	Cukup	38
3	Tidak Setuju	0
Total		76

Sumber : Pengujian dan Analisis

Dari hasil pengujian di atas didapatkan 38 jawaban “Setuju”, 38 jawaban “Cukup” dan tidak jawaban “Tidak Setuju”, dengan total keseluruhan jawaban 76. Maka dapat disimpulkan bahwa nilai persentase jawaban ”Setuju” sebesar 50% dan “Cukup” sebesar 50%. Pernyataan-pernyataan dari kuisioner dapat dibagi menjadi 4 komponen seperti yang ditunjukkan pada Tabel 5.23.

Tabel 5.23 Komponen Pernyataan

No	Komponen	Pernyataan	Jawaban "Setuju"	Jawaban "Cukup"	Jawaban "Tidak Setuju"	Total
1	Keseluruhan	1-19	38 (50%)	38 (50%)	0	76
2	Manfaat Sistem	1-8	17 (53.125%)	15 (46.875%)	0	32
3	Kualitas Informasi	9-15	15 (53.57%)	13 (46.42%)	0	28
4	Kualitas Antarmuka	16-18	4 (33.3%)	8 (66.6%)	0	12

Sumber : Pengujian dan Analisis

Berdasarkan bobot yang telah ditentukan yaitu 1 untuk jawaban "Setuju", 0.5 untuk jawaban "Cukup", dan 0 untuk jawaban "Tidak Setuju", maka tingkat *usability* untuk masing-masing komponen adalah sebagai berikut:

$$\text{Keseluruhan} = \frac{(\text{Setuju} \times 1) + (\text{Cukup} \times 0.5)}{\text{Total}} = \frac{(38 \times 1) + (38 \times 0.5)}{76} \times 100 = 75\%$$

$$\text{Manfaat Sistem} = \frac{(\text{Setuju} \times 1) + (\text{Cukup} \times 0.5)}{\text{Total}} = \frac{(17 \times 1) + (15 \times 0.5)}{32} \times 100 = 76,5\%$$

$$\text{Kualitas Informasi} = \frac{(\text{Setuju} \times 1) + (\text{Cukup} \times 0.5)}{\text{Total}} = \frac{(15 \times 1) + (13 \times 0.5)}{28} \times 100 = 76,78\%$$

$$\text{Kualitas Antarmuka} = \frac{(\text{Setuju} \times 1) + (\text{Cukup} \times 0.5)}{\text{Total}} = \frac{(4 \times 1) + (8 \times 0.5)}{12} \times 100 = 66,6\%$$

5.2 Analisis

Proses analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah yang telah dilakukan. Proses analisis mengacu pada dasar teori sesuai dengan hasil pengujian yang didapatkan. Analisis dilakukan terhadap hasil pengujian di setiap tahap pengujian. Proses analisis yang dilakukan meliputi analisis hasil pengujian unit, analisis hasil pengujian validasi dan analisis pengujian akurasi.

5.2.1 Analisis Hasil Pengujian Unit

Proses analisis terhadap hasil pengujian unit dilakukan dengan melihat kesesuaian fungsi dari implementasi unit modul yang diuji dengan hasil perancangan perangkat lunak yang telah dirancang sebelumnya. Dari seluruh pengujian unit yang dilakukan, jumlah jalur pada method-method dasar yang digambarkan dalam *flow graph* telah sesuai dengan hasil perhitungan dengan rumus *cyclomatic complexity*. Setiap kasus uji yang dibuat berdasarkan jumlah jalur tersebut telah diuji dan memberikan hasil sesuai dengan hasil yang diharapkan. Berdasarkan hal tersebut, maka dapat diambil kesimpulan bahwa unit modul dari program sudah memenuhi kebutuhan fungsional yang telah dirancang pada tahap perancangan.

5.2.2 Analisis Hasil Pengujian Integrasi

Proses analisis terhadap hasil pengujian integrasi dilakukan dengan melihat kesesuaian beberapa unit modul yang menyusun satu blok fungsi dalam Sistem Manajemen Pengunjung. Dari seluruh pengujian integrasi yang dilakukan, jumlah jalur pada logika dari beberapa method yang lolos uji pada pengujian unit yang digambarkan dalam *flow graph* telah sesuai dengan hasil perhitungan dengan rumus *cyclomatic complexity*. Setiap kasus uji yang dibuat berdasarkan jumlah jalur tersebut telah diuji dan memberikan hasil sesuai dengan hasil yang diharapkan. Berdasarkan hal tersebut, maka dapat diambil kesimpulan bahwa unit modul dari program sudah memenuhi kebutuhan fungsional yang telah dirancang pada tahap perancangan.

5.2.3 Analisis Hasil Pengujian Validasi

Proses analisis terhadap hasil pengujian validasi dilakukan dengan melihat kesesuaian antara hasil kinerja sistem dengan daftar kebutuhan. Berdasarkan hasil pengujian validasi dapat disimpulkan bahwa implementasi dan fungsionalitas Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan.

5.2.4 Analisis Hasil Pengujian Akurasi

Berdasarkan hasil pengujian performa menggunakan akurasi, dapat disimpulkan bahwa fitur pengenalan wajah Sistem Manajemen Pengunjung ini dalam melakukan pengenalan wajah saat pengunjung datang ke laboratorium dengan masukan pose wajah standar dengan berjalan menghadap ke kamera memiliki tingkat akurasi 80,59%, sedangkan *error rate* memiliki persentase 19,40% dengan tingkat keabuan lebih besar sama dengan 100 untuk skala 0 sampai 255.

Error yang terjadi pada pengenalan wajah disebabkan oleh kemiripan citra pengunjung yang diuji dengan citra training milik pengunjung lain dibandingkan dengan citra training milik pengunjung yang diuji tersebut. Hal ini dapat terjadi karena pada saat citra uji yang diperoleh tidak jelas atau samar. Tidak jelas atau samarnya citra uji disebabkan oleh gerakan pengunjung yang berjalan.

Tabel 5.24 Citra yang salah prediksi

No	Citra	Keterangan
1		Anisa
2		Deta
3		Alifia
4		Aldi
5		Aulia

Sumber : Pengujian dan Analisis

Tabel 5.24 menunjukkan beberapa citra yang gagal dikenali dengan benar. Dari beberapa citra tersebut, satu diantaranya merupakan citra yang jelas, yaitu citra milik Aulia. Pada citra yang samar seperti citra nomor 1-4, kemungkinan salah prediksi sangat besar, karena tidak terdapat data training yang mirip dengan citra-citra tersebut. Oleh karena itu, pengenalan wajah dari OpenCV akan mencari data training yang memiliki jarak terdekat dengan citra tersebut.

Pada kasus uji citra milik Aulia, citra yang terlihat jelas tersebut diprediksi sebagai Ari Agustina dengan citra training seperti yang ditunjukkan pada Gambar

5.6. Jika dilihat dari data training keduanya, tidak terdapat citra yang mirip. Oleh karena itu, prediksi dari pengenalan wajah akan memilih citra dengan jarak terdekat, sama dengan kasus uji gambar yang samar di atas.



Gambar 5.6 Citra yang salah prediksi
Sumber : Pengujian dan Analisis

5.2.5 Analisis Hasil Pengujian Usability

Proses analisis terhadap hasil pengujian dilakukan dengan melihat nilai persentase jawaban “Setuju” dan “Cukup”. Jawaban “Setuju” dan “Cukup” mewakili kepuasan pengguna yang bernilai positif. Komponen “Manfaat Sistem” memiliki nilai “Setuju” sebesar 17, nilai “Cukup” sebesar 15, dan tanpa nilai “Tidak Setuju” sehingga masing-masing jawaban memiliki persentase sebesar 53.125%, 46.875%, dan 0%. Tingkat *usability* keseluruhan dari komponen “Manfaat Sistem” adalah 76,5%. Komponen “Manfaat Sistem” adalah pernyataan-pernyataan tentang kegunaan sistem yang digunakan untuk menilai seberapa besar sistem dapat bermanfaat dan membantu bagi pengguna.

Komponen “Kualitas Informasi” memiliki nilai “Setuju” sebesar 15, nilai “Cukup” sebesar 13, dan tanpa nilai “Tidak Setuju” sehingga masing-masing jawaban memiliki persentase sebesar 53.57%, 46.42%, dan 0%. Tingkat *usability* keseluruhan dari komponen “Kualitas Informasi” adalah 76,78%. Komponen “Kualitas Informasi” adalah pernyataan-pernyataan tentang informasi yang disajikan oleh sistem yang digunakan untuk menilai seberapa besar informasi sistem dapat dimengerti dan bermanfaat.

Komponen “Kualitas Antarmuka” memiliki nilai “Setuju” sebesar 4, nilai “Cukup” sebesar 8, dan tanpa nilai “Tidak Setuju” sehingga masing-masing jawaban memiliki persentase sebesar 33.3%, 66.6%, dan 0%. Tingkat *usability*

keseluruhan dari komponen “Kualitas Antarmuka” adalah 66,6%. Komponen “Kualitas Antarmuka” adalah pernyataan-pernyataan tentang tampilan tatap muka sistem yang digunakan untuk menilai seberapa besar antarmuka sistem dapat dimengerti dan menyenangkan bagi pengguna.

Secara keseluruhan diperoleh nilai “Setuju” sebesar 38, nilai “Cukup” sebesar 38, dan tanpa nilai “Tidak Setuju” sehingga masing-masing jawaban memiliki persentase sebesar 50%, 50%, dan 0%. Tingkat *usability* keseluruhan sistem adalah 75%. Nilai 75% memiliki kategori dalam interval baik dan sangat baik, sehingga tingkat *usability* sistem adalah baik.



BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil pengamatan selama perancangan, implementasi, dan proses pengujian Sistem Manajemen Pengunjung laboratorium PTIIK berbasis pengenalan wajah, diambil kesimpulan sebagai berikut:

1. Aplikasi Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah dikembangkan melalui beberapa tahap. Tahap pertama yaitu perancangan sistem yang meliputi analisis kebutuhan dan perancangan perangkat lunak berdasarkan *Reuse-Oriented Model*. Tahap kedua, implementasi sistem menggunakan bahasa *c#* dan *library computer vision* meliputi EmguCV, Aforge .NET, serta DirectshowLib .NET. Tahap ketiga pengujian dan analisis sistem yang meliputi pengujian kebutuhan fungsional menggunakan *white-box* dan *black-box* serta pengujian kebutuhan non-fungsional menggunakan pengujian akurasi.
2. Hasil pengujian *white box* dan pengujian *black box* pada Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah telah valid. Hal ini telah dibuktikan dengan dilakukannya proses pengujian unit, pengujian integrasi, dan pengujian validasi.
3. Berdasarkan hasil pengujian akurasi, tingkat akurasi modul pengenalan wajah dari OpenCV yang digunakan dalam Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah dalam melakukan pengenalan wajah saat pengunjung datang ke laboratorium dengan pose wajah standar berjalan menghadap ke kamera memiliki tingkat akurasi 80,59%, sedangkan *error rate* memiliki persentase 19,40% pada tingkat keabuan lebih besar sama dengan 100 untuk skala 0 sampai 255.
4. Hasil pengujian *usability* secara keseluruhan memberikan nilai sebesar 75% berkategori baik. Hal ini menunjukkan bahwa pengguna atau

laboran merasa terbantu dengan Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan Sistem Manajemen Pengunjung Laboratorium PTIIK Berbasis Pengenalan Wajah lebih lanjut antara lain :

1. Untuk pengembangan lebih lanjut, aplikasi Sistem Manajemen Pengunjung ini dapat dikembangkan untuk menjadi aplikasi sistem manajemen pengunjung yang mendukung *IP camera* (CCTV).
2. Untuk pengembangan lebih lanjut, antarmuka aplikasi Sistem Manajemen Pengunjung ini dapat dikembangkan untuk menampilkan layar video lebih dari satu kamera pada satu layar komputer.
3. Untuk pengembangan lebih lanjut, hasil *log* dapat dikembangkan sehingga dapat diexport ke format *spreadsheet* atau Excell.
4. Untuk pengembangan lebih lanjut, pengambilan data training bisa otomatis (*autoburst*).

DAFTAR PUSTAKA

- [BAN-12] Banshal, A., Mehta, K., dan Arora, S. 2012. *Face Recognition Using PCA & LDA Algoritms*. Second International Conference on Advanced Computing & Communication Technologies
- [BRA-08] Bradski, Gary and Kaehler, Adrian. 2008. *Learning OpenCV*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
- [CUI-12] Cuil, Zhen., Shan, S., Zhang, H., Lao, S., Chen, X. 2012. *Image Sets Alignment for Video-based Face Recognition*. IEEE
- [EMG-12] Emgu CV Official Site. *Main Page – Emgu CV: OpenCV in .NET (C#, VB, C++ and more)*. Diakses dari http://www.emgu.com/wiki/index.php/Main_Page. Tanggal akses 29 Februari 2012.
- [INT-09] Introna, D Lucas and Nissenbaum, Helen. 2009. *Facial Recognition Technology. A Surver of Polcy and Implementation Issues*. Computer Science and Information Law Department, New York University.
- [JAI-12] Jain, A.K., Klare, B. dan Park, U. *Face Matching and Retrieval in Forensics Applications*. IEEE Multimedia, vol. 19, no. 1, pp. 20, 28, Jan.-Mar. 2012, doi:10.1109/MMUL.2012.4
- [JAK-03] Nielsen, Jakob. 2003. *Usability 101 :Introduction to Usability*. Aksesdari <http://faculty.wiu.edu/CB-Dilger/f07/480/readings/10-nielsen-usability-101.pdf>.

- [LEW-93] Lewis. James R. 1993. IBM Computer Usability Satisfaction Questionnaires : Psychometric Evaluation and Instructions. IBM Corporation.
- [PAT-10] Patil A.M., Kolhe S.R., dan Patil P.M. 2010. *2D Face Recognition Techniques: A Survey*. International Journal of Machine Intelligence, ISSN: 0975–2927, Volume 2, Issue 1, 2010, pp-74-83.
- [PRE-09] Pressman, Roger. 2009. *Software Engineering: A Practioner's Approach, 7th Edition*. Mc Graw-Hill.
- [SOM-11] Sommerville, Ian. 2011. *Software Engineering 9th Edition*. Pearson.
- [TAN-06] Tan, Pang-Ning., Steinbach, M., dan Kumar, Vipin. 2006. *Data Mining*. Pearson Education, Inc.

