

**PENGEMBANGAN SISTEM PENDATAAN PENGUNJUNG
PERPUSTAKAAN DENGAN SCAN WAJAH**

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh:

Nama : Muhammad Ferdi Firmansyah

NPM : 1062248

POLITEKNIK MANUFAKTUR NEGERI

BANGKA BELITUNG

TAHUN 2025

LEMBAR PENGESAHAN

JUDUL PROYEK AKHIR

SISTEM PENDATAAN PENGUNJUNG PERPUSTAKAAN DENGAN SCAN WAJAH

Oleh:

Muhammad Ferdi Firmansyah/1062248

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan Sarjana Terapan/Diploma IV Politeknik Manufaktur Negeri Bangka Belitung



Menyetujui,

Pembimbing 1

Pembimbing 2

Ahmat Josi, S.Kom., M.Kom
NIP 198908202019031015

Better Swengky ,M.Kom
NIP 199301222024061001

Penguji 1

Penguji 2

Linda Fujiyanti, S.T., M.T.I
NIP 198109262014042001

Putri Armilia Prayesy, M.Kom
NIP 199501042024062001

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa : Muhammad Ferdi Firmansyah NIM: 1062248
Dengan Judul : SISTEM PENDATAAN PENGUNJUNG DENGAN
SCAN WAJAH

Menyatakan bahwa laporan akhir ini adalah hasil kerja sendiri dan bukan merupakan plagiat. Pernyataan ini saya buat dengan sebenarnya dan bila ternyata di kemudian hari ternyata melanggar pernyataan ini, saya bersedia menerima sanksi yang berlaku.

Nama Mahasiswa

1. Muhammad Ferdi Firmansyah

Sungailiat, 18 Juli 2025

Tanda Tangan

()

ABSTRAK

Perpustakaan merupakan fasilitas vital dalam mendukung kegiatan akademik di lingkungan perguruan tinggi seperti Politeknik Manufaktur Negeri Bangka Belitung. Namun, proses pendataan pengunjung yang masih bergantung pada input Nomor Pokok Mahasiswa (NPM) secara langsung oleh pengguna dinilai kurang efisien dan rentan terhadap *human error* seperti kesalahan pengetikan. Untuk mengatasi permasalahan tersebut, dikembangkan "Sistem Pendataan Pengunjung Perpustakaan dengan Scan Wajah". Aplikasi berbasis web ini bertujuan untuk membantu mahasiswa, tamu, dan admin dalam mengelola data kunjungan secara digital, akurat, dan *real-time*. Fungsionalitas utamanya meliputi identifikasi wajah otomatis menggunakan *library* Face-API.js, pencatatan kunjungan, serta dasbor manajemen data untuk admin. Metode pengembangan yang digunakan adalah *Rapid Application Development* (RAD) agar proses perancangan dapat berlangsung secara cepat dan fokus pada kebutuhan pengguna. Diharapkan, dengan adanya sistem ini, proses pendataan pengunjung menjadi lebih efisien dan akurat, pengalaman pengguna meningkat, serta mendukung modernisasi administrasi di lingkungan kampus.

Kata kunci: Perpustakaan Digital, Scan Wajah, Sistem Informasi, Pengunjung, *Rapid Application Development* (RAD)

ABSTRACT

The library is a vital facility in supporting academic activities in a higher education environment such as the State Manufacturing Polytechnic of Bangka Belitung. However, the current visitor data collection process, which relies on the direct input of the Student Identification Number (NPM) by users, is considered less efficient and prone to human error, such as typos. To address these issues, the "Library Visitor Data Collection System with Face Scan" was developed. This web-based application aims to assist students, guests, and administrators in managing visit data digitally, accurately, and in real-time. Its main functionalities include automatic face identification using the Face-API.js library, visit recording, and a data management dashboard for administrators. The development method used is Rapid Application Development (RAD) to ensure the design process is fast and focused on user needs. It is expected that with this system, the visitor data collection process will become more efficient and accurate, the user experience will improve, and it will support the modernization of administration within the campus environment.

Keywords: Digital Library, Face Scan, Information System, Visitor, Rapid Application Development (RAD)

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah, segala puji penulis panjatkan ke hadirat Allah SWT atas segala nikmat, rahmat, dan hidayah-Nya, sehingga penulis dapat menyelesaikan laporan Proyek Akhir yang berjudul "*Sistem Pendataan Pengunjung Perpustakaan Dengan Scan Wajah*" ini dengan baik dan tepat waktu. Shalawat serta salam semoga senantiasa tercurah kepada junjungan Nabi Muhammad SAW, beserta keluarga dan para sahabatnya, sebagai teladan sepanjang masa.

Laporan ini disusun sebagai bagian dari pemenuhan syarat akademik dalam menyelesaikan studi di Politeknik Manufaktur Negeri Bangka Belitung. Selama proses penyusunan, penulis mendapatkan banyak pelajaran berharga, baik secara teknis maupun non-teknis. Meskipun begitu, penulis menyadari bahwa karya ini masih jauh dari kesempurnaan, mengingat keterbatasan pengetahuan dan pengalaman yang dimiliki.

Ucapan terima kasih penulis sampaikan kepada berbagai pihak yang telah memberikan dukungan dan kontribusi selama proses penyusunan laporan ini, antara lain:

1. Bapak Made Andik Setiawan, M.Eng., Ph.D., selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
2. Ibu Yang Agita Rindri, M. Eng. selaku Ketua Jurusan Informatika Dan Bisnis.
3. Bapak Ahmat Josi, S.Kom., M.Kom., selaku pembimbing I, atas bimbingan dan arahan yang sangat berarti.
4. Bapak Better Swengky, M.Kom., selaku pembimbing II, atas saran dan masukannya yang membangun.
5. Seluruh dosen dan staf Politeknik Manufaktur Negeri Bangka Belitung, atas ilmu dan dukungannya selama ini.
6. Rekan-rekan serta semua pihak yang telah memberikan bantuan, baik secara langsung maupun tidak langsung, yang tidak dapat penulis sebutkan satu per satu.
7. Kedua orang tua tercinta, atas doa, motivasi, dan kasih sayang yang tak pernah putus.

Akhir kata, penulis sangat mengharapkan kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat dan menjadi bahan referensi yang berguna dalam pengembangan sistem informasi.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Sungailiat, 24 Juli 2025

Hormat saya,



Muhammad Ferdi Firmansyah

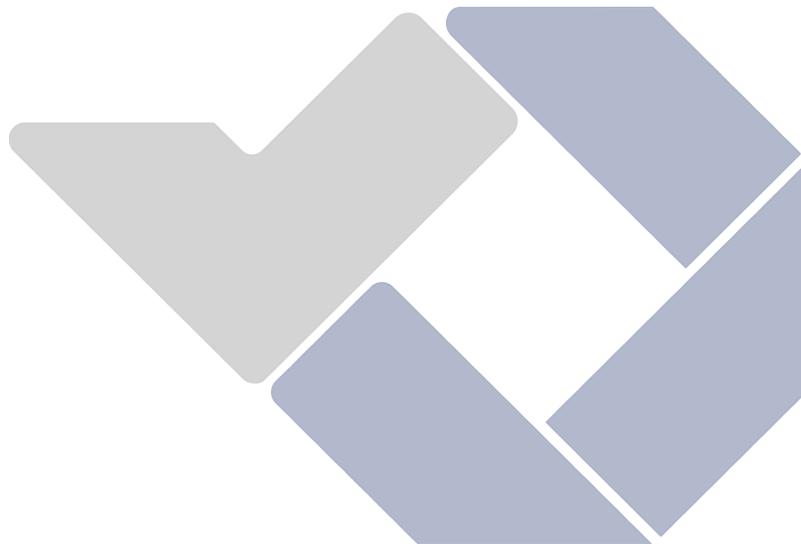


DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN BUKAN PLAGIAT	ii
ABSTRAK.....	iii
ABSTRACT.....	iv
KATA PENGANTAR	v
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Proyek Akhir	3
BAB II DASAR TEORI	4
2.1 Tinjauan Pustaka	4
2.3 Visual Studio Code	6
2.4 PHP (<i>Hypertext Preprocessor</i>).....	6
2.5 MySQL	6
2.6 <i>Face recognition</i>	7
2.7 Face API.js	7
2.8 <i>Convolutional Neural Network (CNN)</i>	7
2.9 PhpMyAdmin.....	8
2.10 Laragon	8
2.11 Laravel	8
2.12 <i>Metode Rapid Application Development (RAD)</i>	9
BAB III METODE PELAKSANAAN	11
3.1 Identifikasi Masalah.....	12
3.2 Studi Literatur	12
3.3 Analisis Kebutuhan.....	12
3.4 Desain Sistem.....	12
3.4.1 <i>Use case Diagram</i>	13
3.4.2 <i>Activity Diagram</i>	14

3.5	Sketsa Desain	18
3.6	Pengembangan Sistem	22
3.6.1	Sistem Manajemen Data (CRUD)	22
3.6.2	Sistem Pendeteksi dan Pencatatan	22
3.7	Uji Coba Sistem	26
3.8	Penerapan	26
BAB IV PEMBAHASAN.....		27
4.1	Inisiasi	27
4.2	Pra-Produksi.....	28
4.3.1	Hasil Kebutuhan Pengguna.....	32
4.3.2	Hasil Kebutuhan Fungsional.....	33
4.3.3	Hasil Kebutuhan Non-Fungsional.....	35
4.3	<i>Production</i>	37
4.3.1	Halaman Login Admin.....	37
4.3.2	Halaman <i>Dashboard</i>	38
4.3.3	Halaman Kelola Data Mahasiswa.....	39
4.3.4	Halaman Show Data Mahasiswa	40
4.3.5	Halaman Tambah Data Mahasiswa.....	41
4.3.6	Halaman Edit Data Mahasiswa.....	42
4.3.7	Halaman Kelola Data Tamu	43
4.3.8	Halaman Show Data Tamu	44
4.3.9	Halaman Edit Data Tamu	45
4.3.10	Halaman Kunjungan	46
4.3.11	Halaman Master Jurusan.....	47
4.3.12	Halaman Master Program Studi.....	48
4.3.13	Halaman Scan Wajah User.....	49
4.3.14	Halaman Pendaftaran Pengunjung Baru	50
4.3.15	Halaman Registrasi Mahasiswa	51
4.3.16	Halaman Registrasi Wajah Tamu	52
4.4	Pengujian.....	52
2.8.1	Analisis Kinerja Interaksi Pengguna.....	52
4.5	Implementasi.....	54
4.5.1	Pengujian Validasi Ahli Media.....	55

4.6	Hasil Kuesioner.....	63
4.7	Hasil Perhitungan User Acceptance Test (UAT).....	64
BAB V KESIMPULAN DAN SARAN.....		65
5.1	Kesimpulan	65
5.2	Saran	65
DAFTAR PUSTAKA.....		66
Lampiran.....		68



DAFTAR TABEL

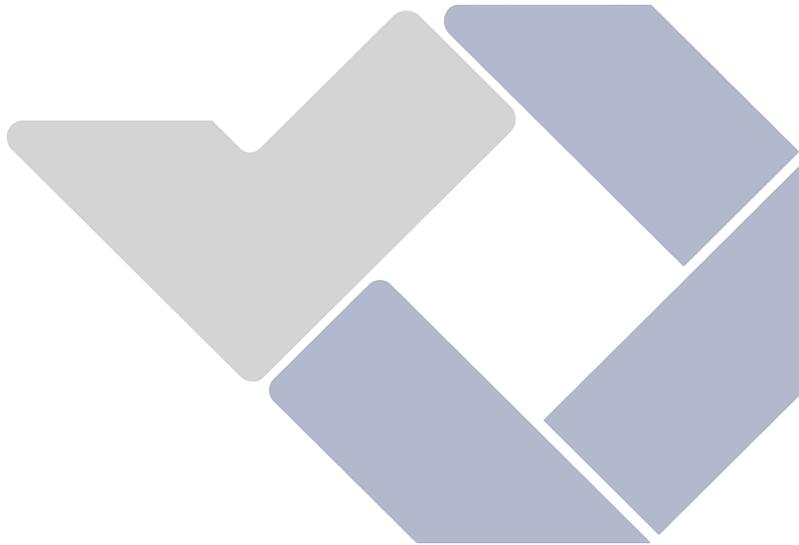
Tabel 2. 1 Tinjauan Pustaka	4
Table 4. 1 Konsep Aplikasi	28
Table 4. 2 Garis Waktu Pengerjaan Proyek.....	31
Table 4. 3 Pengujian Halaman <i>Login</i> Admin	55
Table 4. 4 Pengujian Halaman <i>Dashboard</i> Admin.....	56
Table 4. 5 Pengujian Halaman Kelola Data Mahasiswa	56
Table 4. 6 Pengujian Halaman <i>Scan</i> Wajah & Registrasi Pengunjung	57
Table 4. 7 Pengujian Halaman <i>Login</i> Admin	58
Table 4. 8 Pengujian Halaman <i>Dashboard</i> Admin.....	59
Table 4. 9 Pengujian Halaman Kelola Data Mahasiswa	60
Table 4. 10 Pengujian Halaman Kelola Data Tamu	61
Table 4. 10 Pengujian Halaman Kunjungan	62
Table 4. 8 Hasil Kuesioner Pengunjung Mahasiswa/Tamu.....	63

DAFTAR GAMBAR

Gambar 2. 1 Gambar Metode RAD	9
Gambar 3. 1 <i>Flowchart</i> Tahap Pelaksanaan Proyek Akhir	11
Gambar 3. 2 <i>UseCase</i> Diagram	13
Gambar 3. 3 <i>Activity</i> Diagram Tamu	15
Gambar 3. 4 <i>Activity</i> Diagram Mahasiswa	16
Gambar 3. 5 <i>Activity</i> Diagram Admin.....	17
Gambar 3. 6 Desain Halaman <i>Dashboard</i>	18
Gambar 3. 7 Desain Halaman <i>Login</i> Admin.....	18
Gambar 3. 8 Desain Halaman Kelola Data Mahasiswa dan Tamu	19
Gambar 3. 9 Desain Halaman Kelola Pengunjung	19
Gambar 3. 10 Desain Halaman <i>Scan</i> Wajah Pengunjung	20
Gambar 3. 11 Desain Halaman Opsi Pilihan <i>Role</i> Pengunjung.....	20
Gambar 3. 12 Desain Halaman Registrasi Pengunjung Tamu	21
Gambar 3. 13 Desain Halaman Registrasi Pengunjung Mahasiswa.....	21
Gambar 4. 1 Halaman <i>Login</i> Admin.....	37
Gambar 4. 2 Halaman <i>Dashboard</i>	38
Gambar 4. 3 Halaman Kelola Data Mahasiswa	39
Gambar 4. 4 Halaman <i>Show</i> Data Mahasiswa	40
Gambar 4. 5 Halaman Tambah Data Mahasiswa	41
Gambar 4. 6 Halaman Edit Data Mahasiswa	42
Gambar 4. 7 Halaman Kelola Data Tamu	43
Gambar 4. 8 Halaman <i>Show</i> Data Tamu	44
Gambar 4. 9 Halaman Edit Data Tamu	45
Gambar 4. 10 Halaman Kunjungan Tamu dan Mahasiswa.....	46
Gambar 4. 11 Halaman Master Jurusan	47
Gambar 4. 12 Halaman Master Program Studi	48
Gambar 4. 13 Halaman <i>Scan</i> Wajah User.....	49
Gambar 4. 14 Halaman Pendaftaran Pengunjung Baru	50
Gambar 4. 15 Halaman Registrasi Wajah Mahasiswa	51
Gambar 4. 16 Halaman Registrasi Wajah Tamu	52

DAFTAR LAMPIRAN

Lampiran 1: Lembar Validasi Ahli Penguji 1	68
Lampiran 2: Lembar Validasi Ahli Penguji 2	69
Lampiran 3 Tabel Hasil Kuesioner	70
Lampiran 4: Daftar Riwayat Hidup	72



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perpustakaan merupakan salah satu fasilitas penting di lingkungan kampus yang mendukung kegiatan akademik mahasiswa. (Endarti, 2022) menyebutkan di dalam jurnal (Tunardi, 2018) Perpustakaan adalah jembatan yang berfungsi sebagai penghubung antara sumber informasi dan ilmu pengetahuan. Di dalam pasal 1 ayat 1 Undang-Undang No.43 Tahun 2007 tentang perpustakaan menyebutkan bahwasannya perpustakaan adalah institusi pengelola karya tulis, karya cetak dan/karya rekam, secara profesional dengan sistem yang baku guna memenuhi kebutuhan pendidikan, penelitian, pelestarian, informasi dan rekreasi para pemustaka. Dalam upaya mewujudkan fungsi-fungsi tersebut, sebuah sistem pengelolaan yang baik menjadi hal yang penting bagi perpustakaan. Salah satu bentuk implementasinya dapat dilihat pada sistem pendataan pengunjung di Politeknik Manufaktur Negeri Bangka Belitung.

Di Politeknik Manufaktur Negeri Bangka Belitung, sistem pendataan mengharuskan setiap mahasiswa memasukkan Nomor Pokok Mahasiswa (NPM) ke komputer yang disediakan untuk mencatat kunjungan ke perpustakaan. Setelah NPM diinput, sistem akan menampilkan nama mahasiswa yang bersangkutan dan mencatat kehadirannya. Namun, proses ini memiliki kelemahan terkait efisiensi dan validitas data. Jika terjadi kesalahan pengetikan sehingga NPM yang dimasukkan tidak terdaftar di dalam basis data, sistem akan menampilkan notifikasi kegagalan. Lebih krusial lagi, jika seorang mahasiswa secara tidak sengaja memasukkan NPM milik mahasiswa lain yang valid, maka data kehadiran akan tercatat atas nama mahasiswa yang salah tersebut, bukan pengunjung yang sebenarnya. Kedua potensi kesalahan input ini menjadikan proses pendataan kurang efisien dan rentan terhadap ketidakakuratan data.

Sistem pendataan yang berjalan saat ini, meskipun sudah terkomputerisasi, masih mengandalkan input Nomor Pokok Mahasiswa (NPM) oleh setiap pengunjung. Metode ini memiliki potensi kendala, terutama terkait efisiensi dan pengalaman

pengguna. Ketergantungan pada input data ini rentan terhadap *human error*, seperti salah ketik, yang dapat menghambat alur proses dan memaksa pengguna untuk mengulang dari awal. Hal ini menunjukkan adanya ruang untuk peningkatan kecepatan dan kepraktisan, sehingga sistem yang ada saat ini dirasa kurang optimal.

Untuk mengatasi potensi kendala tersebut, diperlukan sebuah pembaruan pada sistem yang di pakai saat ini. Solusi yang diusulkan adalah mengimplementasikan teknologi pindai wajah sebagai fitur tambahan pada sistem pendataan pengunjung perpustakaan. Dengan pembaruan ini, mahasiswa tidak perlu lagi memasukkan NPM. Cukup dengan melakukan pindai wajah, sistem akan secara otomatis mengenali identitas mahasiswa dan mencatat kehadirannya pada hari itu, sehingga menyempurnakan proses yang sudah ada

Teknologi pengenalan wajah (*Facial recognition*) telah berkembang pesat dan dapat diintegrasikan ke dalam sistem pendataan modern. (Adindya Giovanni, 2023) menyebutkan “Pengenalan wajah (*Face Recognition*) adalah salah satu teknik identifikasi biometrik dengan menggunakan wajah yang bersangkutan sebagai parameter utamanya”. Dengan memanfaatkan pustaka Face API.js yang berbasis *Convolutional Neural Network* (CNN), sistem mampu mendeteksi dan mengenali wajah pengunjung secara akurat langsung melalui browser. Menurut (Herno Aji Sanjaya, 2024) Face-api.js adalah sebuah library JavaScript *opensource* yang dapat digunakan untuk deteksi wajah dan pengenalan ekspresi wajah melalui web browser. Librari Face-api.js ini dibangun di atas TensorFlow.js dan menyediakan banyak fitur yang dapat digunakan untuk aplikasi pengolahan gambar dan video, termasuk deteksi wajah, pengenalan ekspresi wajah, identifikasi wajah, dan lain sebagainya. Teknologi ini memungkinkan proses identifikasi dilakukan secara *real-time* menggunakan kamera perangkat pengguna tanpa memerlukan instalasi aplikasi tambahan.

Proyek ini bertujuan untuk menyempurnakan sistem pendataan yang telah ada dengan menambahkan fitur berbasis pindai wajah yang lebih praktis dan akurat. Diharapkan, dengan adanya sistem ini, proses pendataan pengunjung perpustakaan dapat berjalan lebih efisien dan meningkatkan pengalaman pengguna. Selain itu, sistem ini juga menjadi langkah penerapan teknologi canggih untuk mendukung digitalisasi administrasi di lingkungan kampus.

1.2 Rumusan Masalah

Berikut merupakan rumusan masalah yang menjadi dasar dalam pelaksanaan penelitian ini. Rumusan ini disusun untuk mengidentifikasi permasalahan utama yang dihadapi dalam sistem pendataan pengunjung perpustakaan saat ini, serta menjadi acuan dalam pengembangan solusi berbasis teknologi:

1. Bagaimana meningkatkan efisiensi dan kecepatan proses pendataan pengunjung perpustakaan agar lebih praktis?
2. Bagaimana sistem yang dikembangkan dapat meningkatkan pengalaman pengguna sekaligus mendukung upaya digitalisasi administrasi di lingkungan perpustakaan kampus?
3. Bagaimana mengurangi potensi kesalahan input data pengunjung yang disebabkan oleh ketergantungan pada proses entri Nomor Pokok Mahasiswa (NPM)?

1.3 Tujuan Proyek Akhir

Berikut ini adalah tujuan yang ingin dicapai dalam pelaksanaan proyek akhir ini, yang difokuskan untuk menjawab rumusan masalah yang telah diidentifikasi:

1. Mengembangkan sistem pendataan pengunjung perpustakaan dengan fitur pindai wajah untuk menciptakan proses yang lebih praktis dan efisien.
2. Meningkatkan pengalaman pengguna dalam proses pencatatan kunjungan sekaligus mendukung modernisasi administrasi perpustakaan di lingkungan kampus.
3. Mengurangi potensi kesalahan pendataan yang disebabkan oleh proses entri data dengan menghilangkan ketergantungan pada input Nomor Pokok Mahasiswa (NPM).

BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Dalam tinjauan pustaka ini penulis membandingkan kajian-kajian terdahulu yang sesuai dengan judul penelitian. Berikut adalah hasil dari kajian-kajian sebelumnya.

Tabel 2. 1 Tinjauan Pustaka

No.	Judul	Hasil
1.	Implementation of Face Recognition and Liveness Detection System Using TensorFlow.js (Muhammad Basurah, 2023)	Pada jurnal ini mengembangkan sistem pengenalan wajah berbasis web menggunakan Face API.js dan TensorFlow.js, serta mengintegrasikan <i>liveness detection</i> untuk mencegah <i>spoofing</i> . Sistem menggunakan metode <i>life sign</i> (gerakan wajah), serta <i>object detection</i> dengan ml5.js untuk mendeteksi perangkat yang digunakan untuk <i>spoofing</i> (misalnya video di smartphone). Akurasi sistem mencapai 85% untuk <i>face recognition</i> dan 82,5% untuk <i>liveness detection</i> .
2.	Pengenalan Wajah dengan Face-API.js Berbasis CNN dan Geolokasi Menggunakan Equirectangular Approximation (Bimo Akbar Fadli, 2023)	Jurnal ini mengembangkan sistem absensi berbasis web dan <i>mobile</i> dengan Face API.js dan TensorFlow.js menggunakan metode CNN (SSDMobileNetV1 + ResNet-34). Sistem melakukan deteksi wajah dan perhitungan lokasi menggunakan metode <i>equirectangular approximation</i> . Akurasi pengenalan wajah mencapai 93%–97% dengan rata-rata 95,2% dan waktu rata-rata proses 35,5 ms. Proses dilakukan sepenuhnya di sisi klien untuk mengurangi beban server.
3.	Sistem Pengendalian Akses Berbasis Face-Recognition dengan	Pada jurnal ini membahas tentang mengembangkan sistem pengendalian akses berbasis web dengan Face API.js dan

<p>Face-API.js dan Algoritma Manhattan Distance (Bambang Warsuta, 2024)</p>	<p>algoritma Manhattan Distance. Sistem menggunakan tiga model CNN dari Face API.js: SSDMobileNetV1 untuk deteksi wajah, landmark 68 titik wajah, dan deskriptor wajah. Akurasi pengenalan wajah mencapai 100% dan deteksi wajah mencapai 95,71%. Sistem menunjukkan eksekusi cepat dengan waktu rata-rata di bawah 0,7 ms.</p>
<p>4. Prototype Video Recognition Menggunakan Face API Untuk Keamanan Ruang Kritis Berbasis Web (Herno Aji Sanjaya, 2024)</p>	<p>Penelitian ini merancang sistem keamanan ruangan menggunakan Face API.js dan <i>framework</i> Laravel. Sistem dikembangkan menggunakan metode Research and Development (R&D) dan SDLC Spiral. Sistem mengenali wajah pengguna melalui kamera dan mencatat akses ke ruang kritis. Sistem dibangun menggunakan PHP, JavaScript, PostgreSQL, dan diproses melalui library Face API berbasis CNN. Hasilnya menunjukkan bahwa sistem bekerja akurat, dengan kemampuan mengenali wajah dalam berbagai sudut dan pencahayaan.</p>
<p>5. Aplikasi Absensi Siswa Menggunakan Face API Recognition (Tajuddin Abdillah, 2024)</p>	<p>Jurnal ini membahas rancangan sistem absensi siswa berbasis web di MTs Al-Mawaddah menggunakan Face API.js dan metode Agile. Sistem mendeteksi wajah siswa saat absensi, menampilkan notifikasi keterlambatan atau kehadiran, serta menghasilkan laporan rekap harian hingga tahunan. Pengujian <i>blackbox</i> menunjukkan sistem bekerja sesuai harapan. Sistem juga dilengkapi dengan fitur input izin dan verifikasi oleh wali kelas.</p>

2.2 Website

Website adalah kumpulan halaman informasi yang dapat diakses melalui internet menggunakan *browser*, berisi teks, gambar, video, atau elemen interaktif lainnya, dan biasanya memiliki alamat domain tertentu. Menurut (Fatimah Azzahro, 2024) Website adalah salah satu media yang sering digunakan untuk menginformasikan dan memberikan layanan.

2.3 Visual Studio Code

Visual Code Studio merupakan aplikasi *code editor* untuk membantu proses pengembangan sebuah aplikasi. *Software* ini dikembangkan oleh perusahaan teknologi raksasa ternama, Microsoft. Meskipun begitu, VSCODE mendukung untuk dioperasikan pada perangkat selain Windows, seperti Linux dan Mac OS. Teks editor ini secara langsung mendukung bahasa pemrograman Javascript, Typescript, dan Node. Js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace Visual Studio Code seperti: C++, C#, Python, Go, Java, PHP, dst (Kherina Surya Ningsih, 2022) menyebut di dalam jurnal (Ummy Gusti Salamah, 2021).

2.4 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *server-side* yang dirancang khusus untuk pengembangan web. Menurut (Woro Isti Rahayu, 2023) PHP menawarkan konektivitas yang baik dengan beberapa basis data antara lain Oracle, Sybase, MySQL, dan PostgreSQL. PHP digunakan untuk membuat halaman web dinamis dan dapat diintegrasikan dengan HTML serta berbagai basis data seperti MySQL. Bahasa ini berjalan di sisi server dan menghasilkan output yang dikirim ke browser pengguna.

2.5 MySQL

Menurut Wahana Komputer (2008,h.46) di dalam jurnal (Hartati, 2022) “MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan data dengan cepat dengan menggunakan perintah-perintah SQL”. MySQL adalah sistem manajemen basis data relasional (RDBMS) berbasis *open source* yang menggunakan bahasa SQL (*Structured Query Language*) untuk mengelola dan mengakses data. MySQL banyak digunakan dalam pengembangan aplikasi web untuk menyimpan, mengelola, dan mengambil data secara efisien, serta mendukung integrasi dengan berbagai bahasa pemrograman seperti PHP, Python, dan Java.

2.6 Face recognition

Face recognition adalah teknologi kecerdasan buatan yang digunakan untuk mengidentifikasi atau memverifikasi identitas seseorang berdasarkan fitur-fitur wajahnya. Menurut (Bambang Warsuta, 2024) *Face recognition* bertujuan mengidentifikasi atau memverifikasi identitas seseorang melalui gambar atau video wajah untuk kebutuhan otentikasi biometrik, keamanan finansial, pengendalian akses, dan pengawasan cerdas. Sistem ini bekerja dengan mendeteksi wajah dari gambar atau video, kemudian mencocokkannya dengan data wajah yang telah tersimpan sebelumnya dalam basis data. *Face recognition* umum digunakan dalam keamanan, absensi, dan sistem identifikasi otomatis.

2.7 Face API.js

Face API.js adalah pustaka JavaScript berbasis TensorFlow.js yang digunakan untuk mendeteksi dan mengenali wajah langsung di *browser* tanpa memerlukan server *backend*. Menurut (Bambang Warsuta, 2024) Face-api.js menyediakan sejumlah model *machine learning* yang berbasis JavaScript dan Tensorflow.js. Dengan *library* ini, kita dapat membangun aplikasi web yang mampu melakukan berbagai tugas pengolahan citra wajah, seperti deteksi wajah, pelacakan titik-titik wajah, dan pengenalan wajah. Pustaka ini mendukung fitur seperti deteksi wajah, pengenalan wajah, deteksi ekspresi wajah, serta pengenalan usia dan jenis kelamin, dan dapat bekerja secara *real-time* menggunakan kamera perangkat pengguna. Face API.js cocok digunakan untuk aplikasi web berbasis pengenalan wajah yang ringan dan fleksibel.

2.8 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah salah satu jenis arsitektur *Deep Learning* yang sangat efektif dalam tugas pengolahan data visual, seperti gambar dan video. Menurut (Gracia Yoel Christiawan, 2022) Convolutional Neural Network adalah salah satu metode machine learning dari pengembangan *Multi Layer Perceptron (MLP)* yang didesain untuk mengolah data dua dimensi. Arsitektur ini terinspirasi oleh cara kerja korteks visual pada otak manusia dan dirancang khusus untuk mengenali pola secara hierarkis, mulai dari pola sederhana seperti garis dan tepi hingga fitur yang lebih kompleks seperti bentuk objek atau wajah. Hal ini membuat CNN menjadi teknologi dasar di balik banyak aplikasi pengenalan gambar, termasuk deteksi dan identifikasi wajah. Kemampuan ini menjadikan CNN sebagai teknologi fundamental dalam

berbagai tugas visi komputer, termasuk klasifikasi gambar, deteksi objek, dan pengenalan wajah (*face recognition*). *Library* seperti Face-API.js dibangun di atas *framework* TensorFlow.js dan memanfaatkan model-model berbasis CNN untuk menjalankan fungsi deteksi dan pengenalan wajah secara efisien langsung di peramban web.

2.9 PhpMyAdmin

PhpMyAdmin adalah aplikasi berbasis web yang digunakan untuk mengelola basis data MySQL atau MariaDB melalui antarmuka grafis. Dengan phpMyAdmin, pengguna dapat melakukan berbagai operasi seperti membuat, mengedit, menghapus database dan tabel, menjalankan perintah SQL, serta mengelola pengguna dan hak akses tanpa harus menggunakan command line. (Hartati, 2022) menyebutkan PhpMyAdmin adalah perangkat lunak gratis (*freeware*) yang ditulis menggunakan bahasa pemrograman PHP, yang dimaksudkan untuk menangani administrasi *database* MySQL melalui *interface* Web. Aplikasi ini memudahkan pengelolaan *database*, terutama bagi pengembang web dan administrator sistem.

2.10 Laragon

Laragon adalah sebuah software bundling yang menyediakan lingkungan pengembangan lokal (*local development environment*) untuk aplikasi web, terutama berbasis PHP. Menurut (Roni Andarsyah, 2022) Laragon adalah perangkat lunak yang bersifat open source (terbuka) yang dapat mendukung banyak sekali sistem operasi dimana laragon bertugas sebagai server virtual atau sering disebut sebagai localhost. Laragon dilengkapi dengan server Apache/Nginx, *database* MySQL/MariaDB, dan dukungan untuk berbagai bahasa pemrograman seperti PHP, Node.js, dan Python. Laragon dikenal ringan, portabel, dan mudah digunakan, sehingga sangat cocok untuk membangun dan menguji aplikasi web secara *offline* di komputer *local*.

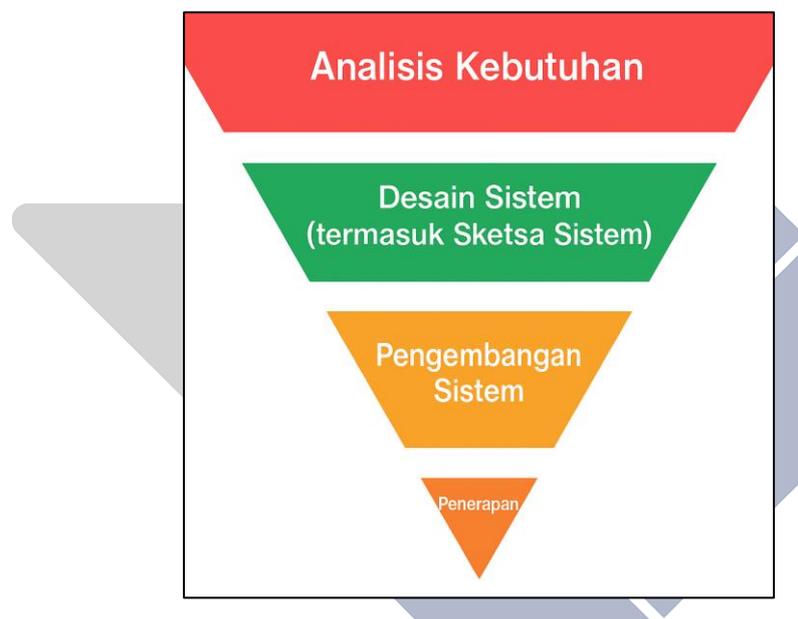
2.11 Laravel

Menurut (Rima Yuniarti, 2022) “Laravel merupakan salah satu *framework web* yang berbasis PHP dan dikembangkan secara *opensource*, laravel dikembangkan oleh Taylor Otwell dan digunakan untuk mengembangkan aplikasi berbasis web yang menerapkan sebuah pola yaitu MVC”. Laravel menyediakan fitur-fitur modern seperti *routing*, *middleware*, ORM (Eloquent), autentikasi, dan sistem *templating Blade*, serta

mendukung arsitektur yang rapi dan skalabel. Laravel banyak digunakan karena sintaksnya yang elegan, dokumentasi lengkap, dan komunitas yang aktif.

2.12 Metode *Rapid Application Development* (RAD)

Menurut (Nurman Hidayat, 2021) Rapid Application Development (RAD) merupakan model proses pengembangan perangkat lunak secara linear sequential yang menekankan pada siklus pengembangan yang sangat singkat. RAD dirancang untuk menghasilkan sistem yang dapat digunakan dalam waktu singkat dengan melibatkan pengguna secara langsung dalam setiap tahap pengembangan. Berikut gambar metode RAD:



Gambar 2. 1 Gambar Metode RAD

Proses RAD dimulai dengan *analisis kebutuhan*, di mana kebutuhan pengguna dikumpulkan secara cepat dan intensif. Tahap berikutnya adalah *desain sistem*, termasuk pembuatan sketsa sistem secara visual untuk memberikan gambaran awal kepada pengguna. Setelah desain disepakati, dilanjutkan dengan *pengembangan sistem* secara iteratif, di mana prototype diuji dan disempurnakan berdasarkan masukan pengguna. Tahap akhir adalah *penerapan*, yaitu implementasi sistem secara menyeluruh ke dalam lingkungan nyata. Menurut (Noertjahyana, 2002) RAD dapat dijadikan acuan untuk mengembangkan suatu sistem informasi yang unggul dalam hal kecepatan, ketepatan dan biaya yang lebih rendah

1. Analisis Kebutuhan

Pada tahap ini, pengembang sistem berfokus untuk memahami secara menyeluruh apa saja yang dibutuhkan oleh pengguna. Proses ini dilakukan melalui diskusi, observasi, atau wawancara langsung dengan pihak terkait. Tujuannya adalah untuk mengumpulkan informasi yang akan menjadi dasar dari pengembangan sistem, memastikan bahwa solusi yang dikembangkan nantinya benar-benar sesuai dengan permasalahan atau kebutuhan nyata di lapangan.

2. Desain Sistem (termasuk Sketsa Sistem)

Desain mencakup pembuatan diagram seperti *use case diagram* dan *activity diagram*, serta sketsa antarmuka pengguna (UI) sebagai gambaran awal tentang struktur dan tampilan sistem.

3. Pengembangan Sistem

Tahap pengembangan sistem merupakan proses implementasi dari desain yang telah ditentukan. Di mulai menulis kode program, mengintegrasikan antarmuka dengan basis data, serta menerapkan berbagai fitur sesuai kebutuhan. Dalam metode RAD, pengembangan dilakukan secara iteratif, jadi setiap fitur diuji dan diperbaiki secara cepat berdasarkan masukan dari pengguna.

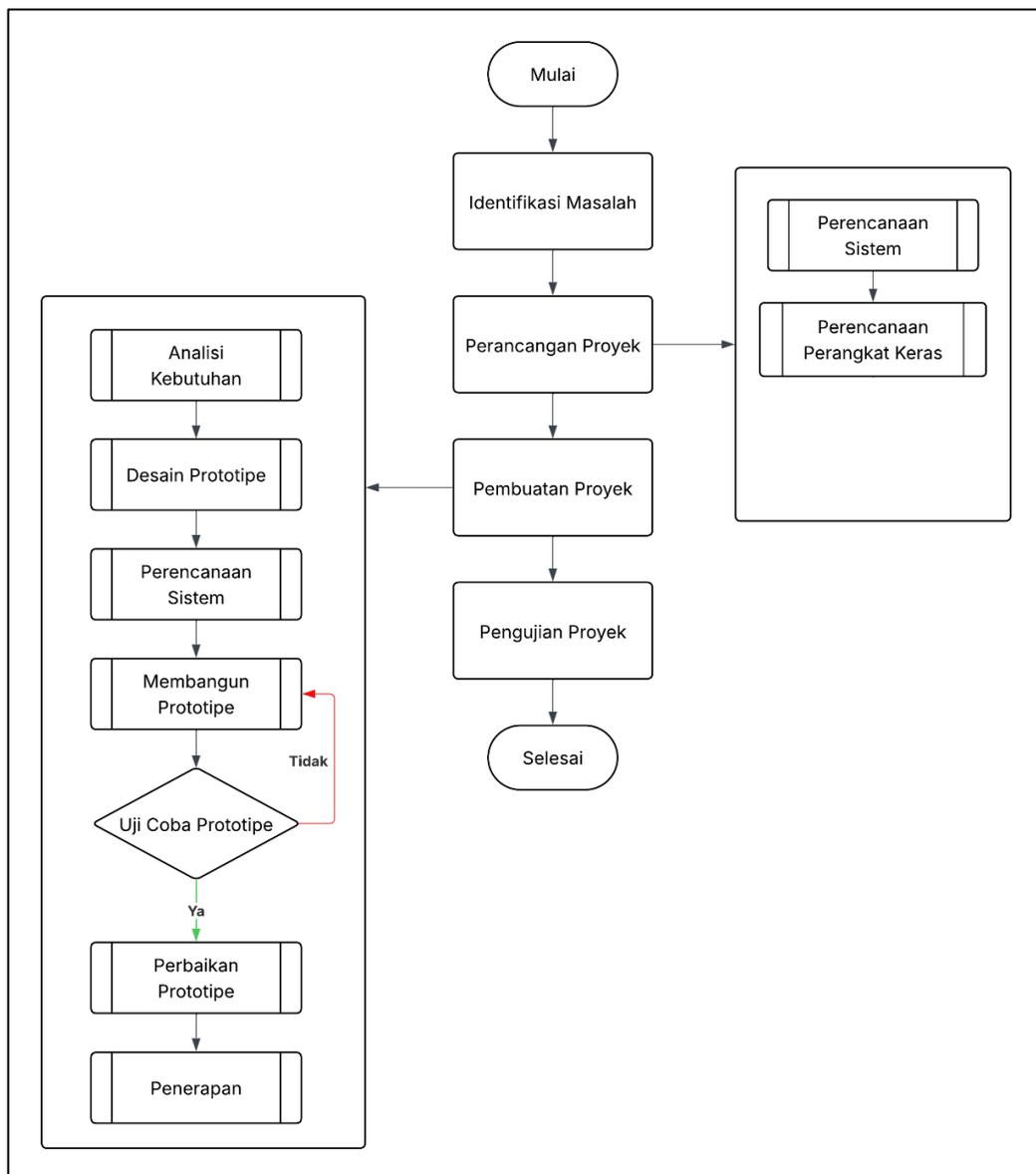
4. Penerapan

Tahap terakhir adalah penerapan sistem ke dalam lingkungan nyata. Pemantauan uji coba awal untuk memastikan bahwa sistem berjalan sesuai harapan. Evaluasi awal dari pengguna menjadi dasar untuk perbaikan lebih lanjut apabila ditemukan kendala saat digunakan secara langsung.

BAB III

METODE PELAKSANAAN

Judul yang diangkat dalam proyek akhir ini adalah "Sistem Pendataan Pengunjung Perpustakaan dengan Scan Wajah." Untuk memberikan gambaran yang jelas mengenai tahapan pelaksanaan, alur kerja proyek dirancang dan divisualisasikan menggunakan diagram alir (*flowchart*). Diagram berikut ini menguraikan keseluruhan proses pengembangan sistem secara sistematis.



Gambar 3. 1 Flowchart Tahap Pelaksanaan Proyek Akhir

3.1 Identifikasi Masalah

Tahap identifikasi masalah adalah langkah awal yang penting dalam sebuah proyek pengembangan sistem. Tujuannya adalah untuk memahami secara mendalam masalah pada proses yang ada, sehingga solusi yang diusulkan dapat menjawab kebutuhan secara tepat. Studi kasus pada proyek ini adalah sistem pencatatan pengunjung di perpustakaan kampus. Hasil pengamatan menunjukkan bahwa sistem yang berjalan saat ini mengharuskan setiap mahasiswa dan tamu untuk menginputkan data kunjungan melalui komputer yang disediakan di pintu masuk. Metode input data ini dinilai memiliki beberapa kelemahan, terutama dalam aspek efisiensi waktu dan validitas data. Oleh karena itu, penelitian ini dilatarbelakangi oleh kebutuhan untuk menyempurnakan sistem yang ada dengan fitur pencatatan otomatis yang mampu meningkatkan kecepatan layanan dan menjamin akurasi pencatatan.

3.2 Studi Literatur

Studi literatur dilakukan untuk memperoleh pemahaman mendalam mengenai konsep, metode, serta teknologi yang relevan dengan pengembangan sistem. Referensi-referensi tersebut bisa ditemukan di buku, jurnal, artikel, dan literatur lainnya. Adapun yang dikaji di proyek ini mencakup metode identifikasi, sistem otomatisasi, serta prinsip-prinsip dasar pengelolaan data pengunjung berbasis sistem informasi. Tahapan ini penting untuk membangun fondasi yang kuat dalam proses perancangan dan implementasi sistem.

3.3 Analisis Kebutuhan

Tahap ini bertujuan untuk mengidentifikasi dan merumuskan kebutuhan sistem yang akan dikembangkan. Kebutuhan tersebut terdiri dari dua aspek utama: kebutuhan fungsional yang mencakup fitur-fitur yang harus dimiliki oleh sistem, dan kebutuhan non-fungsional seperti kecepatan, keakuratan, dan keamanan. Hasil dari analisis ini akan menjadi pedoman dalam perancangan sistem secara keseluruhan.

3.4 Desain Sistem

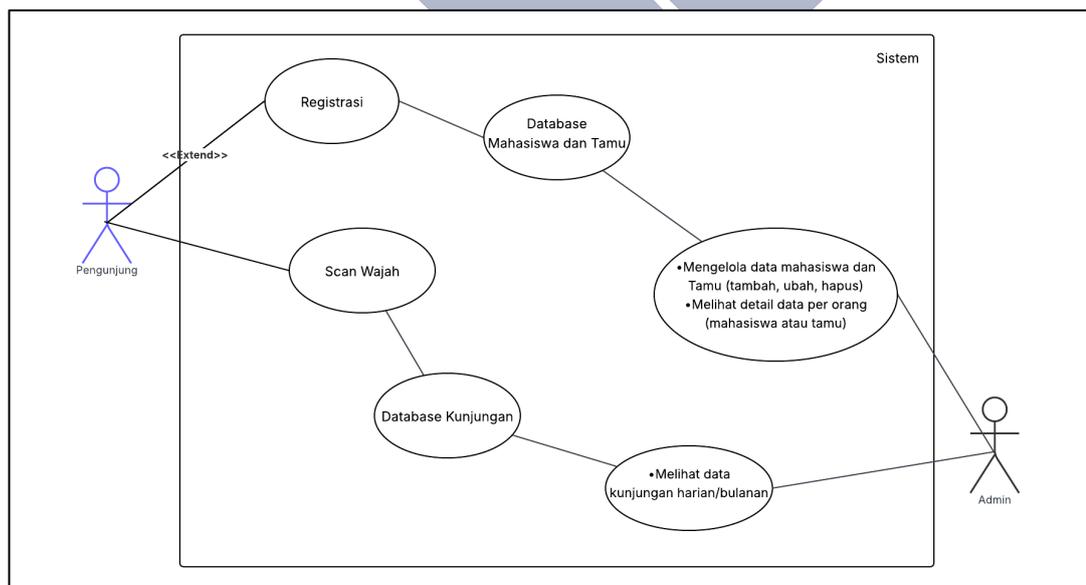
Desain sistem merupakan tahapan penting untuk menggambarkan alur dan struktur dari sistem yang akan dikembangkan. Dalam proses ini, digunakan beberapa model diagram untuk memperjelas rancangan, seperti:

3.4.1 Use case Diagram

Use case merupakan salah satu komponen penting dalam pemodelan sistem berbasis objek yang digunakan untuk menggambarkan bagaimana sistem berinteraksi dengan pengguna atau aktor eksternal untuk mencapai suatu tujuan tertentu. Use case menyajikan skenario atau alur fungsi-fungsi utama dari sistem yang dapat dilihat dari sudut pandang pengguna, bukan dari sisi teknis internal sistem. Dengan kata lain, use case menjelaskan "apa yang dilakukan sistem", bukan "bagaimana sistem melakukannya".

Dalam pemodelan *Unified Modeling Language (UML)*, *use case* biasanya divisualisasikan dalam bentuk *Use Case Diagram* yang terdiri dari elemen-elemen utama seperti aktor (pengguna, sistem lain atau perangkat eksternal), *use case* (fitur atau layanan sistem), dan hubungan antar elemen (asosiasi, <<include>>, <<extend>>). *Use case* diagram digunakan untuk memberikan gambaran menyeluruh mengenai fungsionalitas sistem secara menyeluruh, serta membantu pengembang system memahami bagaimana sistem akan digunakan dalam kehidupan nyata.

Use case juga berfungsi sebagai dasar untuk pengujian sistem, perancangan antarmuka pengguna (UI), serta dokumentasi teknis dan fungsional. Dengan mendefinisikan use case sejak awal, pengembang dapat memastikan bahwa sistem yang dibangun benar-benar menjawab kebutuhan pengguna akhir.



Gambar 3. 2 UseCase Diagram

3.4.2 *Activity Diagram*

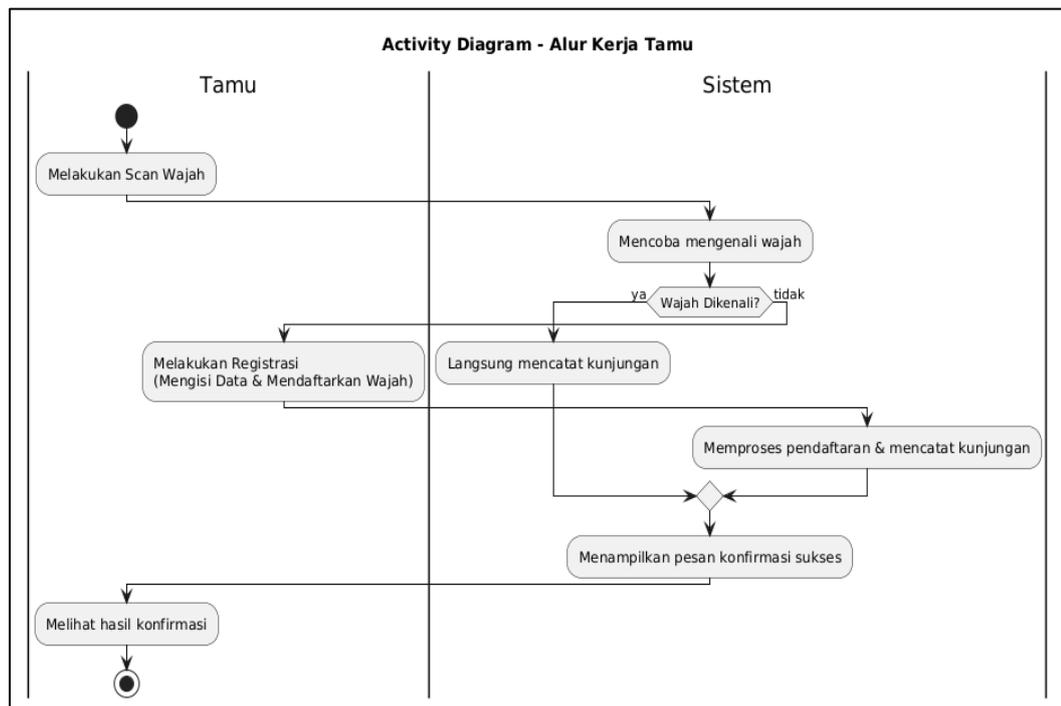
Activity Diagram adalah salah satu jenis diagram dalam pemodelan *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur kerja (*workflow*) atau aktivitas dari suatu proses dalam sistem. Diagram ini merepresentasikan urutan aktivitas, keputusan, percabangan, dan paralelisme dalam suatu sistem secara visual. Activity diagram sering digunakan untuk memodelkan alur logika sistem, proses bisnis, atau skenario dari sebuah *use case*.

Diagram ini menampilkan aktivitas dalam bentuk simbol persegi panjang dengan sudut tumpul (*rounded rectangles*), serta menunjukkan arah alur dengan panah, titik awal dengan simbol bulat hitam (*initial node*), dan akhir proses dengan lingkaran ganda (*final node*). Selain itu, dapat pula memuat elemen-elemen seperti *decision node* (percabangan), *merge node*, dan *swimlane* untuk membedakan aktivitas berdasarkan peran atau aktor.

Dalam konteks pengembangan perangkat lunak, activity diagram sangat berguna untuk:

1. Menganalisis proses prosedur yang kompleks,
2. Memahami dan merancang logika dari sistem,
3. Mengomunikasikan proses kepada tim pengembang dan pemangku kepentingan non-teknis,
4. Menentukan urutan implementasi modul program.

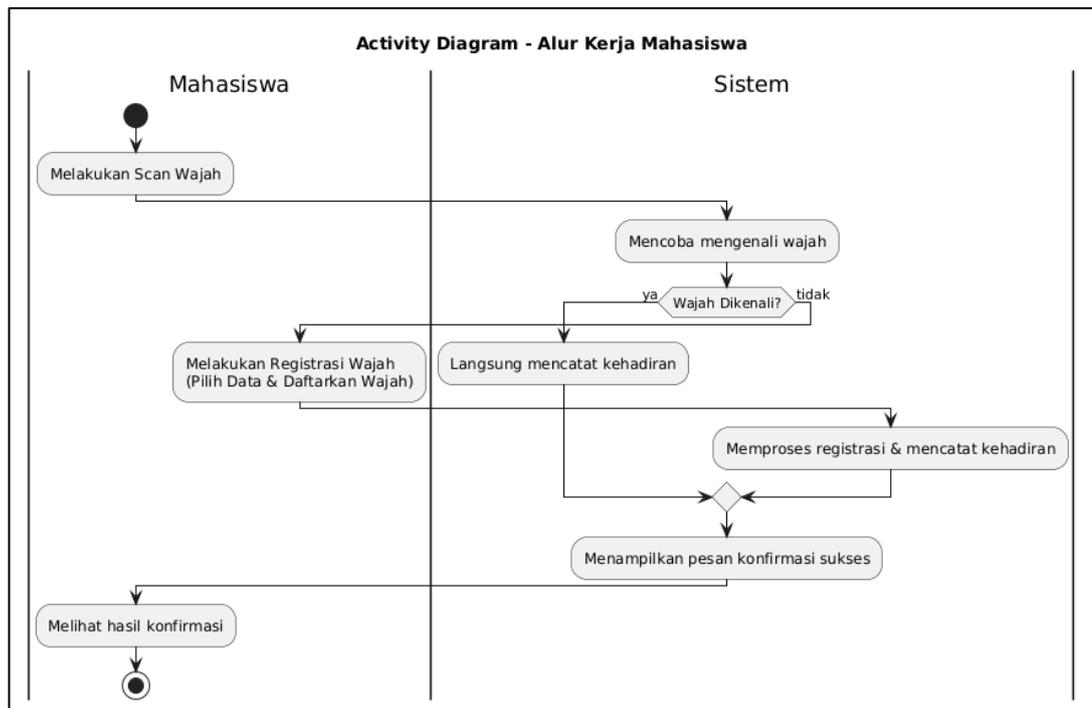
3.4.2.1 Activity Diagram Pengunjung Tamu



Gambar 3. 3 Activity Diagram Tamu

Alur kerja dalam diagram tamu dimulai ketika seorang tamu menginisiasi kunjungan dengan melakukan scan wajah. Selanjutnya, sistem melakukan verifikasi dengan membandingkan citra wajah yang diterima dengan data biometrik wajah yang telah tersimpan di database. Berdasarkan hasil verifikasi, alur akan bercabang ke dua skenario. Pada skenario pertama, jika wajah tamu dikenali, artinya tamu sudah pernah berkunjung dan datanya telah tersimpan, maka sistem langsung mencatat kunjungan tersebut. Pada skenario kedua, jika wajah tamu tidak dikenali, maka sistem akan mengarahkan tamu untuk melakukan proses registrasi, yang meliputi pengisian data biodata melalui formulir serta pendaftaran wajah melalui pemindaian. Setelah proses ini selesai, sistem akan menyimpan data tamu baru dan mencatat kunjungan pertamanya. Kedua jalur proses ini kemudian bergabung pada tahap akhir yang sama, di mana sistem menampilkan pesan konfirmasi keberhasilan sebagai bentuk umpan balik kepada tamu. Dengan demikian, sesi kunjungan dinyatakan selesai.

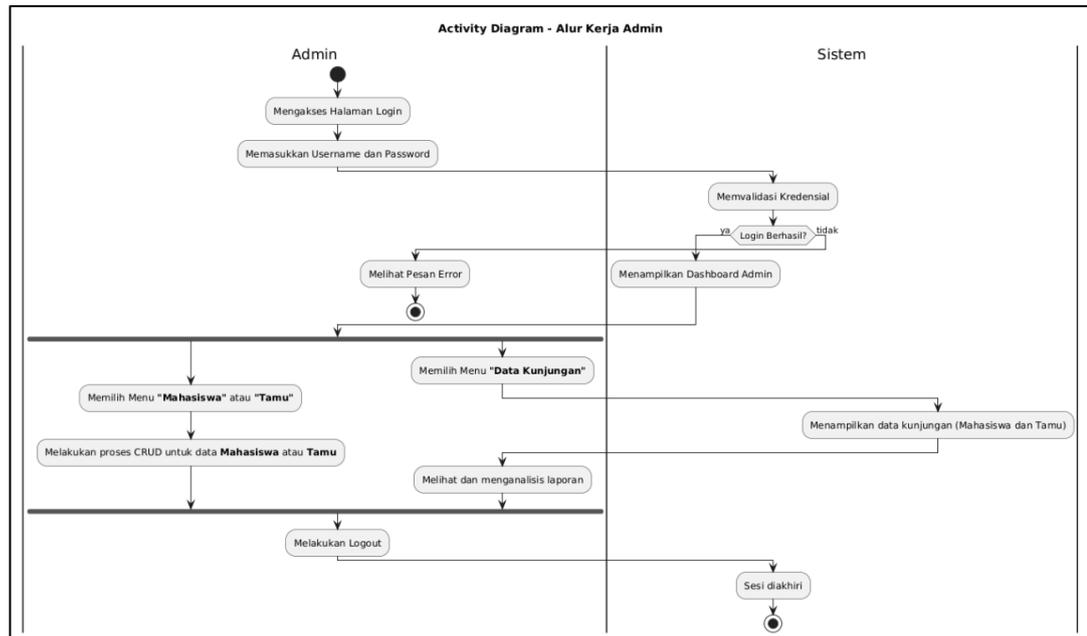
3.4.2.2 Activity Diagram Pengunjung Mahasiswa



Gambar 3. 4 Activity Diagram Mahasiswa

Alur kerja dalam diagram mahasiswa dimulai oleh mahasiswa yang memulai proses dengan melakukan scan wajah. Setelah menerima citra wajah tersebut, sistem akan melakukan validasi dengan mencoba mengenali wajah dan membandingkannya dengan data yang sudah tersimpan di dalam database. Berdasarkan hasil proses pengenalan ini, alur akan bercabang melalui sebuah *decision node* ke dua skenario. Pada skenario pertama, jika wajah mahasiswa dikenali, artinya mahasiswa sudah terdaftar, maka sistem akan langsung mencatat kehadiran. Pada skenario kedua, jika wajah tidak dikenali, mahasiswa akan diarahkan untuk melakukan registrasi wajah. Proses ini terdiri dari dua langkah, yaitu mahasiswa memilih data diri dari daftar yang telah diinput oleh admin untuk menjaga integritas dan privasi data, serta mendaftarkan citra wajah untuk pertama kali. Setelah itu, sistem akan memproses registrasi dan langsung mencatat kehadiran sebagai kehadiran pertama. Kedua aktivitas tersebut pada akhirnya akan bergabung dalam satu rangkaian proses akhir yang sama, yaitu sistem menampilkan pesan konfirmasi sukses sebagai bentuk umpan balik. Mahasiswa kemudian melihat hasil konfirmasi tersebut, dan alur kerja untuk sesi kehadiran dinyatakan selesai.

3.4.2.3 Activity Diagram Admin



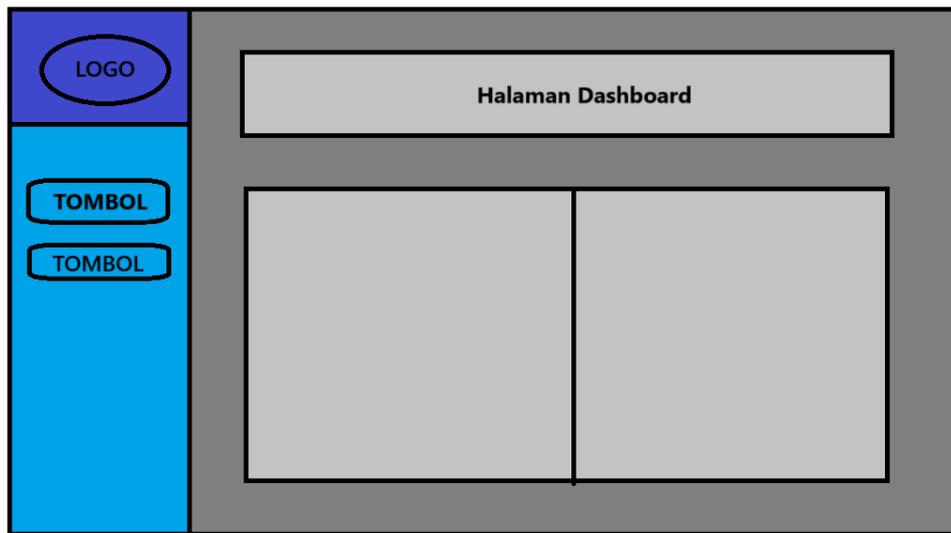
Gambar 3. 5 Activity Diagram Admin

Alur kerja dalam diagram Admin dimulai dengan proses autentikasi untuk memastikan keamanan akses. Admin mengakses halaman login dan memasukkan username serta password, lalu sistem menerima kredensial tersebut dan melakukan validasi. Berdasarkan hasil validasi, decision node akan menentukan alur selanjutnya: jika login berhasil, sistem menampilkan *Dashboard Admin*; jika gagal, sistem akan menampilkan pesan error dan proses sesi tersebut dihentikan. Setelah berhasil login, admin memperoleh akses ke berbagai fungsi utama sistem. Hal ini direpresentasikan dengan notasi fork dalam diagram, yang menandakan bahwa admin dapat memilih salah satu dari beberapa fungsi yang tersedia. Fungsi pertama adalah manajemen data pengguna, di mana admin dapat memilih menu "Mahasiswa" atau "Tamu" dan melakukan proses CRUD (*Create, Read, Update, Delete*) terhadap data pengguna. Fungsi kedua adalah pemantauan data kunjungan, di mana admin dapat memilih menu "Data Kunjungan" untuk melihat seluruh riwayat kunjungan dari mahasiswa maupun tamu. Data ini dapat dianalisis untuk keperluan pemantauan atau pelaporan. Setelah semua aktivitas selesai, admin dapat melakukan logout. Sistem akan mengakhiri sesi login, dan alur kerja pun berakhir pada Final Node.

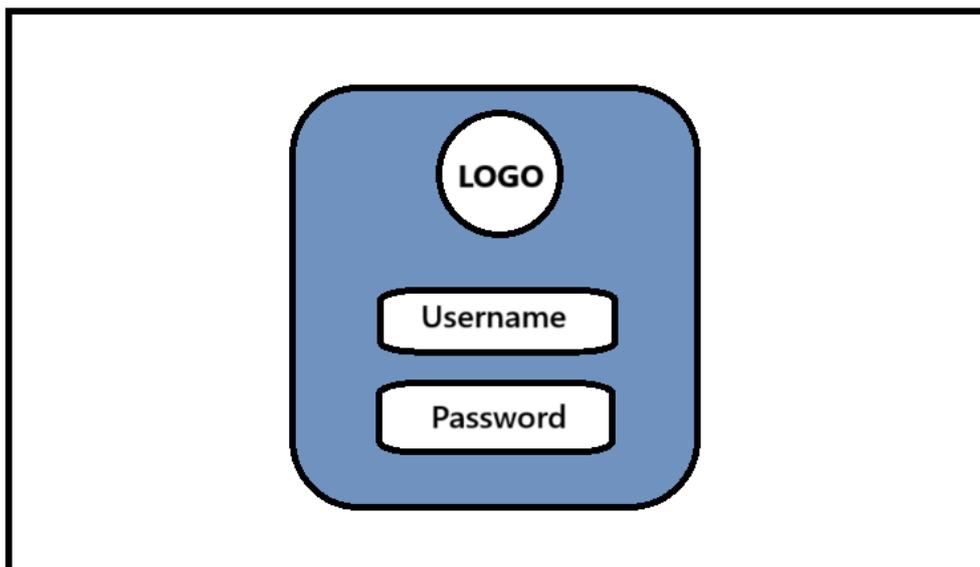
3.5 Sketsa Desain

Pada tahap ini, dibuat rancangan awal tampilan antarmuka sistem. Sketsa tersebut mencakup halaman utama, form interaksi pengguna, serta tampilan pengelolaan data. Tujuannya adalah memberikan gambaran visual awal sebelum proses pengembangan dimulai, sehingga memudahkan proses implementasi sistem ke depannya.

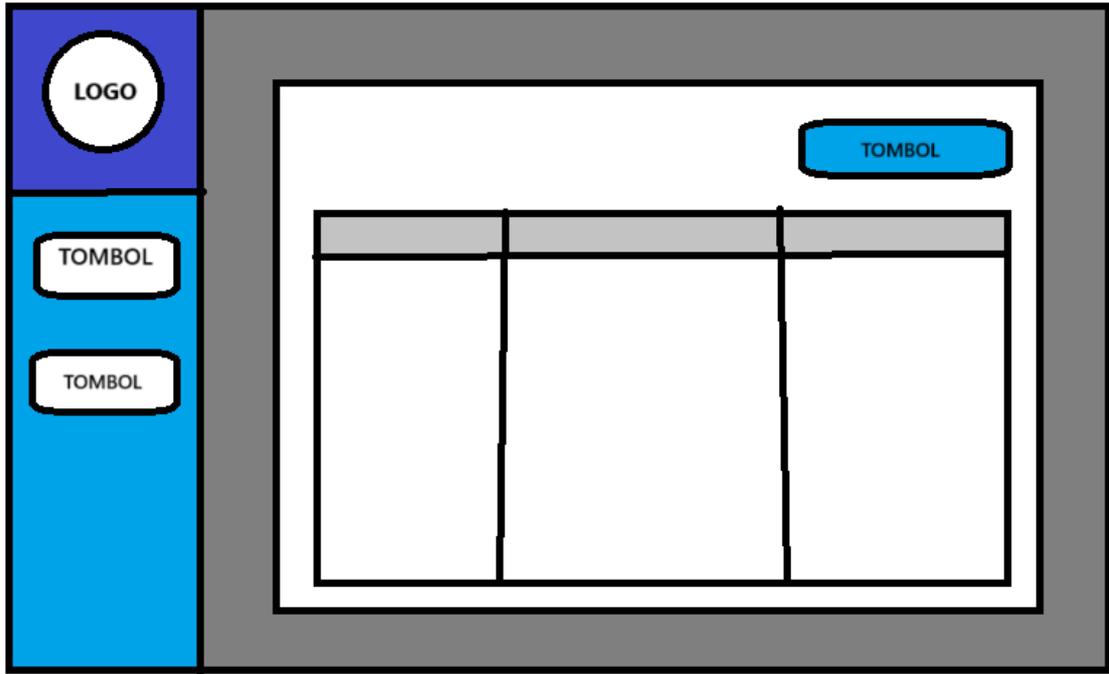
A. Desain Tampilan Admin



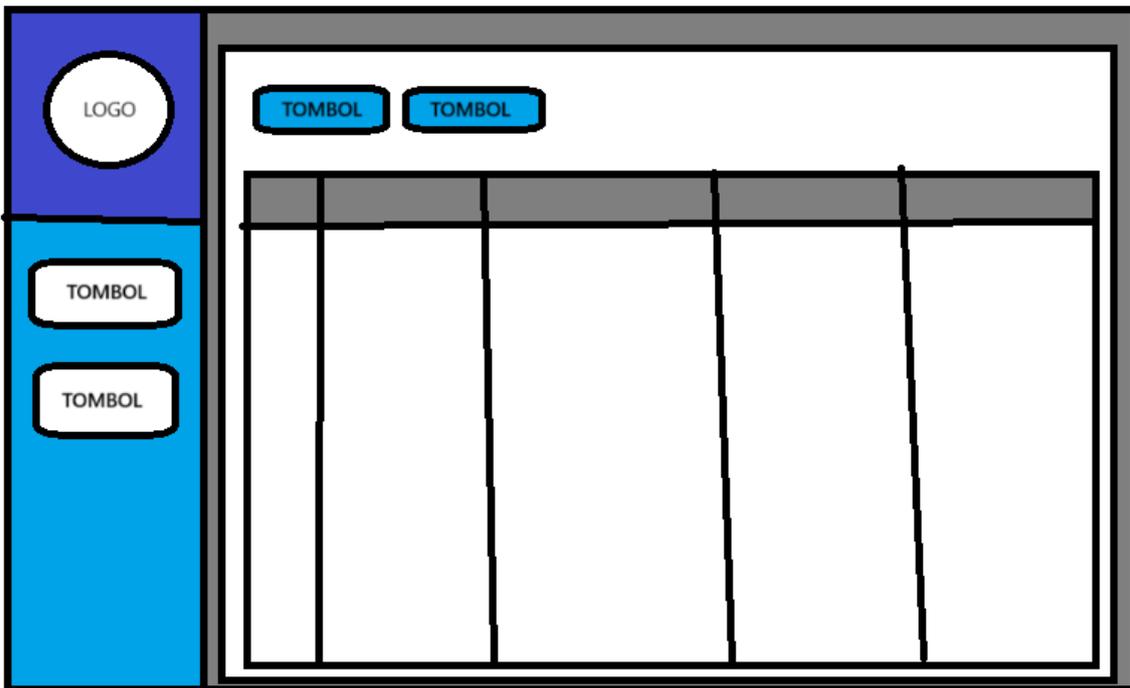
Gambar 3. 6 Desain Halaman Dashboard



Gambar 3. 7 Desain Halaman Login Admin

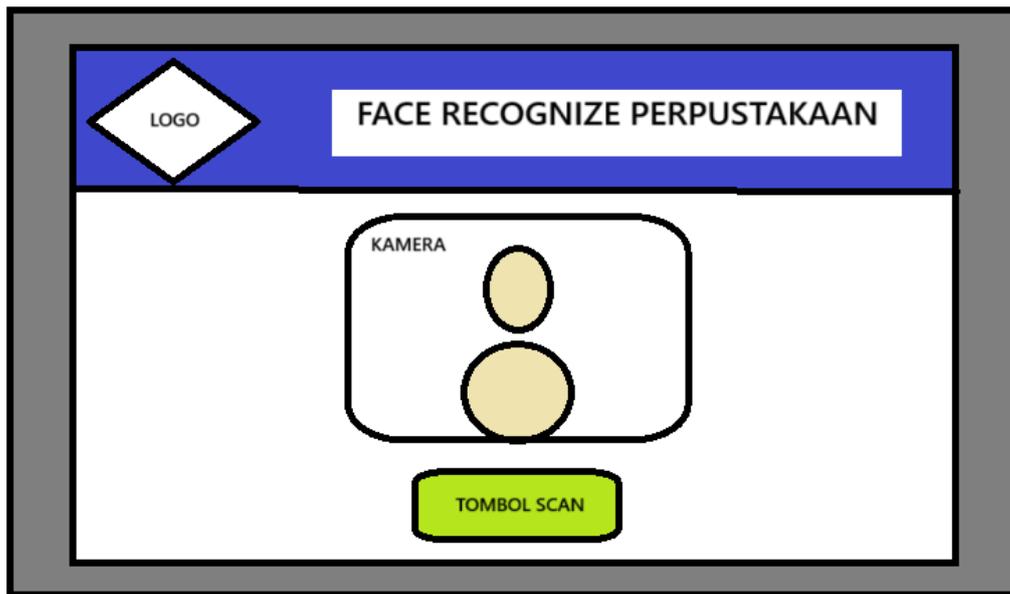


Gambar 3. 8 Desain Halaman Kelola Data Mahasiswa dan Tamu

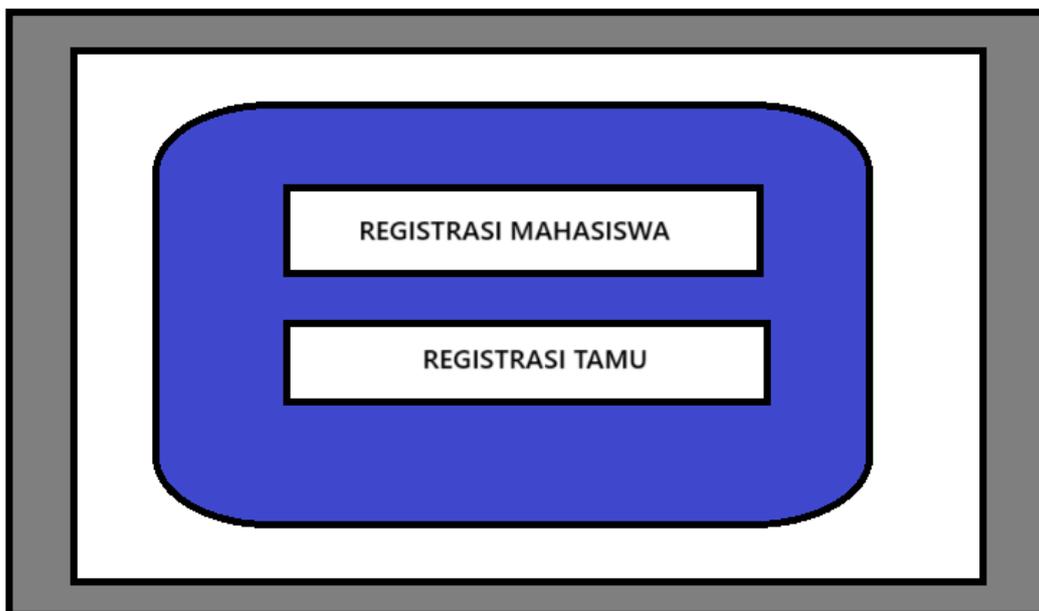


Gambar 3. 9 Desain Halaman Kelola Pengunjung

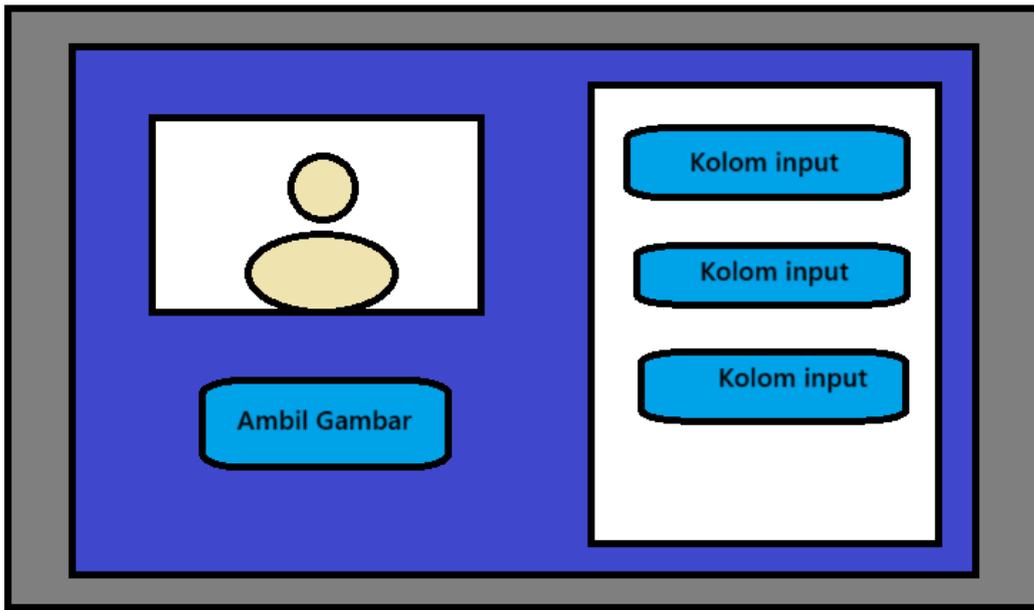
B. Desain Tampilan User



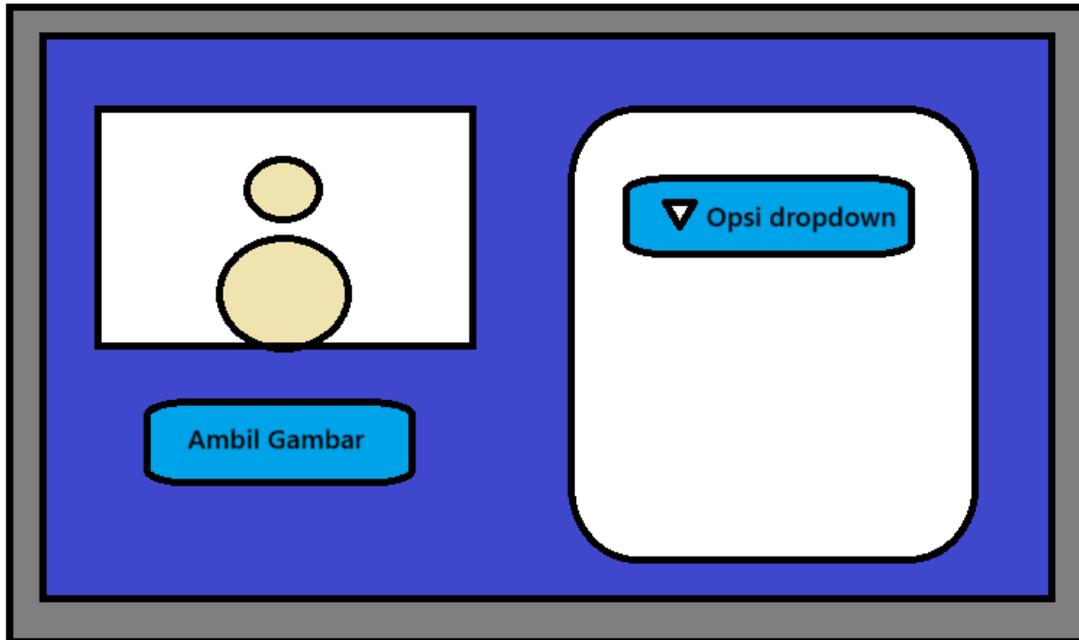
Gambar 3. 10 Desain Halaman Scan Wajah Pengunjung



Gambar 3. 11 Desain Halaman Opsi Pilihan Role Pengunjung



Gambar 3. 12 Desain Halaman Registrasi Pengunjung Tamu



Gambar 3. 13 Desain Halaman Registrasi Pengunjung Mahasiswa

3.6 Pengembangan Sistem

Setelah desain diselesaikan, pada tahap ini sistem mulai dikembangkan sesuai kebutuhan. Pengembangan dibagi menjadi dua bagian utama:

3.6.1 Sistem Manajemen Data (CRUD)

Sistem Manajemen Data merupakan fitur inti pada Admin, yang berfungsi untuk mengelola informasi pengguna (Mahasiswa dan Tamu) serta riwayat kunjungan. Fungsionalitas ini diimplementasikan melalui operasi CRUD, yang merupakan singkatan dari *Create*, *Read*, *Update*, dan *Delete*. Admin memiliki wewenang penuh untuk melakukan keempat operasi ini guna memastikan data dalam sistem tetap akurat, relevan, dan terorganisir.

Berikut adalah rincian dari setiap operasi CRUD yang dilakukan oleh Admin:

1. *Create* (Membuat Data Baru)

Operasi *Create* memungkinkan Admin untuk menambahkan data baru ke dalam database sistem.

- a. Pada Data Mahasiswa: Admin dapat mendaftarkan data mahasiswa baru secara manual. Proses ini dilakukan untuk menambahkan mahasiswa angkatan baru atau mahasiswa yang datanya belum ada di sistem. Informasi yang dimasukkan mencakup NIM (Nomor Induk Mahasiswa), nama lengkap, program studi dan jurusan. Data master inilah yang nantinya akan digunakan oleh mahasiswa saat melakukan registrasi wajah.
- b. Pada Data Tamu: Meskipun tamu melakukan registrasi mandiri, fitur ini memungkinkan admin untuk membuat data tamu secara manual jika diperlukan dalam kondisi tertentu dan Admin juga bisa mengedit data Tamu jika ada kesalahan.

3.6.2 Sistem Pendeteksi dan Pencatatan

Sistem Pendeteksi dan Pencatatan adalah komponen inti dari proyek ini yang berfungsi sebagai antarmuka utama bagi pengunjung (Mahasiswa dan Tamu). Sistem ini bertanggung jawab untuk melakukan proses identifikasi dan pendataan pengunjung secara otomatis. Untuk mencapai fungsionalitas ini, penelitian ini mengimplementasikan Face-API.js, sebuah pustaka (*library*)

JavaScript yang berjalan langsung di peramban (*browser*) untuk melakukan deteksi dan pengenalan wajah secara *real-time*.

Tujuan utamanya adalah menyempurnakan proses pencatatan yang ada dengan menyediakan alur yang lebih cepat, akurat, dan terverifikasi, dengan memanfaatkan model *Deep Learning* yang telah dioptimalkan oleh Face-API.

Proses kerja sistem ini dapat diuraikan ke dalam beberapa tahapan logis sebagai berikut:

1. Akuisisi Citra (*Image Acquisition*)

Tahap pertama dalam sistem ini adalah akuisisi atau penangkapan citra wajah pengguna. Proses ini terjadi ketika kamera yang terpasang di area masuk secara otomatis menangkap gambar (*frame*) dari video secara *real-time* saat mendeteksi wajah seorang pengunjung yang berada di depannya. Hasil dari tahap ini adalah sebuah citra digital wajah pengunjung yang akan diproses lebih lanjut, di mana kualitas gambar seperti pencahayaan dan resolusi menjadi faktor penting untuk menunjang akurasi pada tahap berikutnya.

2. Deteksi Wajah (*Face Detection*)

Setelah gambar berhasil ditangkap, tahap selanjutnya adalah deteksi wajah, di mana sistem harus menemukan lokasi wajah di dalam citra tersebut. Proses ini dilakukan oleh algoritma deteksi wajah yang akan menganalisis citra untuk mengidentifikasi dan melokalisasi keberadaan wajah manusia, sekaligus memisahkannya dari latar belakang atau objek lain. Hasil akhir dari tahap deteksi ini adalah output berupa koordinat atau sebuah bounding box yang menandai area spesifik di mana wajah terdeteksi.

3. Ekstraksi Fitur (*Feature Extraction*)

Tahap selanjutnya adalah ekstraksi fitur, yang merupakan inti dari proses pengenalan wajah. Pada tahap ini, sistem menganalisis area wajah yang telah dideteksi untuk mengekstrak berbagai fitur biometrik yang unik, seperti jarak antar mata, bentuk hidung, dan kontur rahang. Fitur-fitur tersebut kemudian diubah menjadi sebuah representasi matematis yang unik, sering disebut sebagai faceprint atau vektor. Hasil akhir dari proses ini adalah sebuah vektor numerik unik yang berfungsi sebagai "tanda tangan" digital dari wajah tersebut dan siap untuk dibandingkan dengan data di dalam basis data.

4. Pencocokan Data (Data Matching)

Tahap selanjutnya adalah pencocokan data, di mana vektor fitur yang baru saja dibuat akan dibandingkan dengan yang sudah ada di dalam basis data. Dalam proses ini, sistem membandingkan vektor fitur wajah saat ini dengan seluruh vektor fitur yang telah tersimpan sebelumnya, yang merupakan milik mahasiswa dan tamu yang sudah terdaftar. Hasil dari perbandingan ini adalah sebuah output berupa skor kemiripan (*similarity score*) yang akan menjadi dasar untuk tahap pengambilan keputusan.

5. Pengambilan Keputusan dan Pencatatan

Tahap terakhir adalah pengambilan keputusan, di mana sistem akan bertindak berdasarkan skor kemiripan dari tahap pencocokan. Apabila skor kemiripan melebihi ambang batas kesalahan (*threshold*) yang telah ditentukan, sistem akan mengidentifikasi pengunjung tersebut sebagai "Cocok" (*Match Found*) dan secara otomatis mencatat kunjungannya ke dalam basis data, lengkap dengan ID pengunjung serta tanggal dan waktu saat itu. Sebaliknya, jika tidak ada skor kemiripan yang mencapai ambang batas, sistem akan menganggap pengunjung tersebut "Tidak Cocok" (*No Match*) dan akan mengarahkannya ke alur registrasi sebagai pengguna baru.

Dalam implementasinya pada proyek ini, beberapa model CNN spesifik dari *library* Face-API.js digunakan secara bersamaan untuk membentuk sebuah alur kerja pengenalan wajah yang lengkap. Model-model yang dimuat dan digunakan dalam sistem ini meliputi:

1. *Tiny Face Detector*

Model ini adalah komponen pertama dalam alur kerja yang berfungsi untuk deteksi wajah (*face detection*). *Tiny Face Detector* merupakan implementasi dari arsitektur *Tiny YOLO v2* yang telah dimodifikasi dan dioptimalkan untuk kecepatan. Dengan menggunakan *depthwise separable convolutions* sebuah teknik yang lebih efisien daripada konvolusi standar model ini memiliki ukuran yang sangat kecil (sekitar 190 KB) dan mampu mendeteksi wajah secara *real-time* bahkan pada perangkat dengan sumber daya terbatas. Tugasnya adalah secara cepat menemukan lokasi wajah dalam gambar dan menghasilkan kotak

pembatas (*bounding box*) di sekelilingnya. Meskipun sangat cepat, model ini memiliki sedikit kelemahan dalam mendeteksi wajah yang berukuran sangat kecil dibandingkan model yang lebih besar.

2. Face Landmark 68 Net

Setelah wajah berhasil dideteksi, model Face Landmark 68 Net akan bekerja untuk melakukan deteksi titik wajah (*face landmark detection*). Model CNN ini dilatih secara khusus untuk mengidentifikasi dan memetakan 68 titik koordinat (x, y) pada fitur-fitur wajah yang spesifik, seperti kontur mata, bentuk alis, garis hidung, kontur bibir, dan garis rahang. Informasi dari 68 titik digunakan untuk proses *face alignment*, yaitu menyejajarkan atau menormalkan posisi wajah ke pose standar (misalnya, lurus ke depan). Proses *alignment* ini memastikan bahwa wajah yang akan dianalisis oleh model pengenalan berada dalam orientasi yang konsisten, sehingga meningkatkan akurasi identifikasi secara signifikan.

3. Face Recognition Net

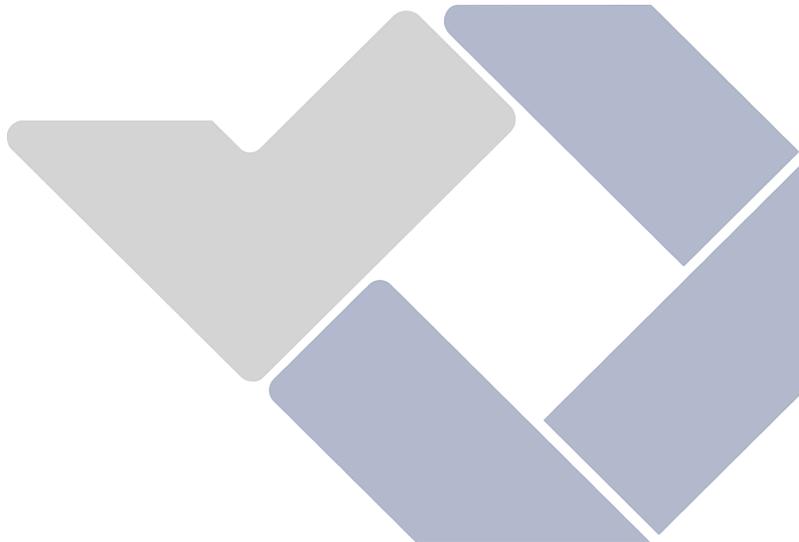
Ini adalah model inti yang melakukan pengenalan wajah (*face recognition*). Model ini dibangun di atas arsitektur yang menyerupai ResNet-34 (*Residual Network*). Arsitektur ResNet memungkinkan jaringan syaraf tiruan menjadi sangat dalam, sehingga mampu mempelajari fitur-fitur wajah yang kompleks. Tugas model ini adalah mengambil citra wajah yang sudah terdeteksi dan disejajarkan, lalu mengubahnya menjadi sebuah representasi matematis yang disebut deskriptor wajah (*face descriptor*). Deskriptor ini adalah sebuah vektor unik berdimensi 128 yang berfungsi sebagai "sidik jari" digital dari wajah tersebut. Vektor inilah yang kemudian digunakan dalam tahap akhir untuk dibandingkan dengan data di dalam basis data guna proses verifikasi dan identifikasi pengunjung.

3.7 Uji Coba Sistem

Sistem yang telah selesai dikembangkan akan melalui tahap pengujian untuk memastikan semua fitur berjalan dengan baik. Pengujian dilakukan terhadap seluruh fungsi utama sistem. Jika ditemukan kesalahan atau ketidaksesuaian, dilakukan perbaikan sebelum sistem dinyatakan siap untuk diterapkan.

3.8 Penerapan

Setelah pengujian berhasil dan sistem berfungsi sesuai harapan, sistem kemudian diterapkan pada lingkungan yang telah ditentukan. Penerapan mencakup proses instalasi, penggunaan secara langsung, serta pemantauan awal untuk memastikan sistem berjalan dengan stabil.



BAB IV

PEMBAHASAN

4.1 Inisiasi

Sistem pencatatan pengunjung perpustakaan ini dikembangkan sebagai sebuah platform digital terintegrasi yang melayani tiga pengguna utama: Mahasiswa, tamu, dan Admin. Platform ini memanfaatkan teknologi pengenalan wajah untuk menyempurnakan proses yang ada menjadi lebih efisien dan akurat. Fungsionalitas utamanya mencakup otentikasi biometrik wajah, pencatatan otomatis, pendaftaran pengguna, serta manajemen data terpusat untuk mendukung operasional perpustakaan.

Bagi Mahasiswa, sistem ini memperbarui metode pencatatan kunjungan. Melalui perangkat pemindai wajah yang ditempatkan di pintu masuk, kehadiran mereka tercatat secara otomatis, sehingga menghilangkan ketergantungan kebutuhan untuk memasukkan Nomor Pokok Mahasiswa (NPM). Untuk penggunaan awal, setiap mahasiswa diwajibkan melakukan aktivasi satu kali dengan menautkan data diri mereka yang sudah ada di sistem dengan data biometrik wajahnya. Proses ini tidak hanya menghemat waktu tetapi juga memastikan bahwa setiap catatan kehadiran valid dan tidak dapat diwakilkan

Untuk pengunjung Tamu perpustakaan. Pengunjung Tamu baru dapat mendaftarkan diri secara mandiri melalui fitur registrasi yang meliputi pengisian biodata dan pemindaian wajah. Setelah terdaftar, kunjungan mereka selanjutnya akan diproses secara otomatis, sama seperti mahasiswa.

Dari sisi manajemen, Admin memiliki akses ke sebuah dasbor administratif berbasis web untuk pengawasan dan pengelolaan sistem. Melalui halaman Admin, admin dapat mengelola data mahasiswa (meliputi penambahan, pembacaan, pembaruan, dan penghapusan data), mengelola daftar tamu, dan menganalisis seluruh data kunjungan yang terekam.

Table 4. 1 Konsep Aplikasi

Keterangan	Deskripsi
Nama Sistem	Sistem Pendataan Pengunjung Perpustakaan Dengan Scan Wajah
Target Pengguna	Mahasiswa Politeknik Manufaktur Negeri Bangka Belitung dan Tamu (pengunjung perpustakaan), serta Admin sebagai pengelola sistem.
Jenis Sistem	Sistem Informasi Otomatisasi Pendataan Berbasis Web dengan Teknologi Face Recognition.
Fitur Utama	Pemindaian wajah (face recognition), registrasi wajah, pencatatan kunjungan otomatis, manajemen data pengguna (mahasiswa dan tamu), dan <i>dashboard</i> admin berbasis web.
Tujuan Pengembang	Mengembangkan sistem pendataan pengunjung perpustakaan yang praktis dan efisien melalui teknologi pemindaian wajah secara otomatis.
Manfaat	Mempermudah pencatatan kehadiran pengunjung, mengurangi kesalahan input, mempercepat proses masuk ke perpustakaan.
Nilai Tambah	Pendataan lebih cepat dan akurat, efisiensi antrian, teknologi modern berbasis browser.

4.2 Pra-Produksi

Fase persiapan ini direncanakan secara sistematis untuk memastikan proses pengembangan nantinya berjalan lancar dan efisien sesuai tujuan utamanya, yaitu menyederhanakan pencatatan kunjungan perpustakaan secara otomatis melalui pemindaian wajah. Sistem ini akan melayani mahasiswa, tamu, dan admin. Kegiatan tersebut mencakup perancangan alur kerja sistem, perencanaan struktur teknis untuk sisi frontend dan backend, penentuan komponen inti sistem seperti modul registrasi wajah dan pencatatan kunjungan, serta perancangan antarmuka web untuk admin. Seluruh perencanaan ini bertujuan untuk memastikan bahwa sistem yang dibangun nantinya mampu menjawab kebutuhan pengguna secara efektif serta meningkatkan efisiensi administrasi di lingkungan perpustakaan kampus.

1. Perancangan Alur Sistem.
 - a. Menentukan interaksi antara pengguna berdasarkan peran: mahasiswa, tamu, dan admin.
 - b. Membuat alur penggunaan sistem mulai dari login admin, scan wajah mahasiswa/tamu, hingga pencatatan kunjungan.

- c. Merancang logika alur registrasi wajah bagi pengguna baru dan verifikasi data pengguna eksisting.

2. Perencanaan Struktur Proyek.

- a. Menyusun struktur direktori dan file untuk frontend (HTML/JS) dan backend (Laravel + Face API.js).
- b. Pembagian direktori sebagai berikut:

Frontend (User):

- 1) `\resources\views\user\scan` – halaman scan wajah untuk mahasiswa dan tamu.
- 2) `\resources\views\user\register` – halaman registrasi wajah mahasiswa dan registrasi data tamu.
- 3) `\resources\views\user\layouts` – struktur dasar dan elemen-elemen umum.

Frontend (Admin):

- 1) `\resources\views\admin\auth` – halaman login.
- 2) `\resources\views\admin` – halaman *dashboard*.
- 3) `\resources\views\admin\mahasiswa` – halaman kelola data mahasiswa.
- 4) `\views\admin\tamu` – halaman Kelola data tamu.
- 5) `\views\admin\kunjungan` – halaman data kunjungan.
- 6) `\resources\views\admin\prodi` – halaman master kelola program studi.
- 7) `\resources\views\admin\jurusan` – halaman master Kelola jurusan.
- 8) `\resources\views\admin\layouts` – struktur dasar dan elemen-elemen umum.

Backend Laravel (Web Admin & API):

- 1) `\routes\web.php` – mendefinisikan endpoint sistem (login, *dashboard*, data kunjungan, dsb).
- 2) `\app\Http\Controllers` – berisi controller logika pengelolaan pengguna, kunjungan, dan autentikasi.
- 3) `\app\Models` – model data seperti Mahasiswa, Tamu, Kunjungan, Admin, DekriptorWajah, Jurusan, MasterJurusan, MasterProdi, User.
- 4) `\database\Migrations` – Menampung semua file untuk mendefinisikan skema database.

5) \routes\api.php – endpoint API Face API.js.

3. Penentuan Komponen Inti Aplikasi.

- a. Mahasiswa: pengguna yang kehadirannya tercatat otomatis melalui scan wajah.
- b. Tamu: pengunjung yang dapat mendaftar mandiri dan terekam datanya.
- c. Admin: petugas yang mengelola data pengguna dan monitoring kunjungan melalui *dashboard*.
- d. Face Recognition: fitur utama yang mengidentifikasi wajah dan mencatat kehadiran.
- e. Registrasi Wajah: proses input biometrik wajah bagi pengguna baru.
- f. Validasi Kunjungan: sistem verifikasi kecocokan wajah terhadap database sebelum pencatatan kunjungan.

4. Perancangan halaman admin

- a. Merancang antarmuka admin untuk mengelola data mahasiswa, tamu, dan kunjungan.
- b. Mendesain UI/UX halaman admin yang akan menjadi pusat kontrol utama yang digunakan oleh petugas perpustakaan.
- c. Menentukan fitur utama yang dibutuhkan dalam sistem admin, meliputi:
 - 1) Autentikasi dan hak akses admin
 - 2) Manajemen data mahasiswa
 - 3) Manajemen data tamu
 - 4) Monitoring riwayat kunjungan
 - 5) Statistik kunjungan dan pencarian data

5. Penyusunan Dokumen & Desain Awal Sistem

- a. Menyusun flowchart proses kunjungan perpustakaan, termasuk alur scan wajah, registrasi wajah baru, dan pencatatan kehadiran
- b. Merancang wireframe untuk halaman scan wajah (mahasiswa dan tamu), halaman registrasi, halaman login admin, dan *dashboard* admin

- c. Mendesain antarmuka pengguna (UI) menggunakan tools desain seperti Figma atau Canva.
 - d. Menyusun dokumen teknis proyek seperti analisis kebutuhan fungsional dan non-fungsional, use case diagram, activity diagram.
6. Garis Waktu Pengembangan Sistem

Berikut adalah rincian alokasi waktu yang direncanakan untuk setiap tahapan dalam proyek ini, mulai dari fase persiapan hingga pengembangan sistem.

Table 4. 2 Garis Waktu Pengerjaan Proyek

Tanggal	Kegiatan
1–14 Januari 2025	Identifikasi kebutuhan sistem dan penentuan peran pengguna (Mahasiswa, Tamu, Admin)
15–28 Januari 2025	Penyusunan konsep alur sistem: scan wajah, registrasi wajah baru, dan pencatatan kunjungan
29 Januari – 11 Februari 2025	Perancangan struktur folder dan file proyek (Laravel backend dan frontend)
12–25 Februari 2025	Sketsa awal tampilan antarmuka halaman scan, login admin, registrasi, dan <i>dashboard</i>
26 Februari – 11 Maret 2025	Desain UI/UX halaman admin dan halaman pengguna berbasis web
12–18 Maret 2025	Pengembangan halaman Admin: login, <i>dashboard</i> , manajemen data
19–25 Maret 2025	Pengembangan halaman User: scan wajah, registrasi.
26 Maret – 8 April 2025	Pengembangan fitur CRUD untuk Admin: data mahasiswa, tamu, dan kunjungan
9–15 April 2025	Integrasi Face API.js untuk proses deteksi dan pencocokan wajah secara real-time
16–22 April 2025	Finalisasi desain antarmuka dan skenario penggunaan sistem
23 April – 6 Mei 2025	Penyusunan flowchart, use case diagram, dan activity diagram untuk setiap peran pengguna
7–20 Mei 2025	Testing sistem secara keseluruhan

4.3.1 Hasil Kebutuhan Pengguna

Dalam pengembangan "Sistem Pendataan Pengunjung Perpustakaan dengan Scan Wajah", analisis kebutuhan pengguna dilakukan untuk memastikan bahwa fitur-fitur yang dikembangkan dapat menjawab permasalahan secara efektif dan sesuai dengan karakteristik pengguna utama, yaitu Mahasiswa, Tamu, dan Admin. Kebutuhan ini mencakup aspek fungsionalitas, efisiensi, akurasi, dan pengalaman pengguna untuk menyempurnakan sistem yang berjalan saat ini.

Berikut adalah hasil identifikasi kebutuhan pengguna:

1. Admin membutuhkan dashboard administratif berbasis web yang terpusat untuk mengelola keseluruhan sistem. Fitur ini harus mencakup kemampuan untuk mengelola data mahasiswa dan tamu melalui operasi CRUD (*Create, Read, Update, Delete*) serta memantau dan melihat riwayat data kunjungan secara harian maupun bulanan.
2. Mahasiswa membutuhkan proses pencatatan kehadiran yang cepat dan otomatis hanya dengan melakukan pemindaian wajah. Sistem ini harus menghilangkan keharusan untuk menginput Nomor Pokok Mahasiswa (NPM) secara langsung, sehingga mengurangi potensi *human error*. Bagi mahasiswa yang belum terdaftar, diperlukan fitur registrasi wajah satu kali yang terhubung dengan data master yang sudah ada.
3. Tamu (pengunjung umum) membutuhkan fitur registrasi mandiri yang mudah digunakan. Proses ini mencakup pengisian data diri singkat dan pendaftaran wajah melalui pemindaian. Setelah terdaftar, proses kunjungan berikutnya harus berjalan otomatis seperti halnya mahasiswa.
4. Sistem harus memiliki kemampuan untuk mendeteksi dan mengenali wajah secara akurat dan real-time melalui peramban web. Teknologi yang digunakan, seperti Face API.js, harus mampu membedakan wajah yang sudah terdaftar dan yang belum untuk kemudian diarahkan ke alur yang sesuai (pencatatan kunjungan atau registrasi).

5. Sistem harus dirancang untuk meningkatkan efisiensi dan kecepatan proses pendataan. Tujuannya adalah untuk meminimalkan waktu yang dibutuhkan setiap pengunjung untuk mencatatkan kehadirannya, sehingga secara efektif meningkatkan efisiensi waktu.
6. Tampilan antarmuka (UI), baik untuk halaman pemindaian wajah maupun dasbor admin, harus di buat user friendly, responsif, dan mudah digunakan. Pengalaman pengguna (UX) yang baik akan memastikan sistem dapat diadopsi dengan mudah oleh semua kalangan pengguna.
7. Fitur pencatatan riwayat kunjungan menjadi penting bagi Admin untuk melakukan pemantauan, analisis, dan pelaporan.

4.3.2 Hasil Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan fitur-fitur dan interaksi spesifik yang harus tersedia dalam "Sistem Pendataan Pengunjung Perpustakaan dengan Scan Wajah" untuk mendukung kebutuhan semua peran pengguna (Admin, Mahasiswa, dan Tamu).

1. Admin

Fungsionalitas untuk Admin berpusat pada pengelolaan, pemantauan, dan pemeliharaan data sistem melalui dasbor berbasis web.

- a. Login Sistem: Admin memulai sesi dengan masuk ke sistem melalui halaman login menggunakan *username* dan *password* untuk otentikasi.
- b. *Dashboard* Utama: Setelah berhasil login, sistem menampilkan dasbor utama yang menjadi pusat kontrol untuk mengelola seluruh fitur administratif.
- c. Manajemen Data Pengguna (CRUD): Admin dapat mengakses dan mengelola data Mahasiswa. Fungsionalitas ini mencakup menambah data baru (*Create*), melihat detail data (*Read*), memperbarui data (*Update*), dan menghapus data (*Delete*).

- d. *Manajemen Data Master*: Admin dapat mengelola data pendukung seperti Program Studi dan Jurusan, yang digunakan untuk kelengkapan data mahasiswa.
- e. *Pemantauan Data Kunjungan*: Admin dapat melihat dan menganalisis seluruh riwayat kunjungan dari mahasiswa maupun tamu untuk keperluan pemantauan atau pelaporan.

2. Mahasiswa

Fitur untuk Mahasiswa berfokus pada proses pencatatan kehadiran yang cepat dan otomatis menggunakan identifikasi biometrik.

- a. *Pencatatan Kunjungan Otomatis*: Mahasiswa cukup melakukan scan wajah di depan kamera, dan sistem akan secara otomatis mendeteksi, mengenali, dan mencatat kehadirannya sebagai pengunjung perpustakaan pada hari itu.
- b. *Registrasi Wajah (Pengguna Baru)*: Jika wajah mahasiswa tidak dikenali (pengguna pertama kali), sistem akan mengarahkannya untuk melakukan proses registrasi wajah.
- c. *Pemilihan Data Diri*: Selama proses registrasi, mahasiswa memilih data dirinya (seperti nama dan NPM) dari daftar yang telah diinput oleh Admin, kemudian menautkannya dengan data wajahnya.
- d. *Konfirmasi Sistem*: Setelah proses pemindaian atau registrasi berhasil, sistem akan menampilkan pesan konfirmasi keberhasilan sebagai umpan balik kepada mahasiswa.

3. Tamu

Fitur untuk Tamu dirancang untuk memberikan kemudahan bagi pengunjung non-mahasiswa untuk mendaftar dan mencatatkan kunjungannya.

- a. *Pencatatan Kunjungan*: Tamu melakukan scan wajah untuk mencatat kunjungan. Jika data wajah sudah pernah terdaftar, sistem akan langsung mencatat kunjungan tersebut.

- b. Registrasi Mandiri (Pengguna Baru): Jika wajah tamu tidak dikenali, sistem akan mengarahkannya untuk melakukan proses registrasi mandiri.
- c. Input Biodata dan Pendaftaran Wajah: Pada proses registrasi, tamu akan mengisi formulir biodata serta melakukan pendaftaran wajah melalui pemindaian.
- d. Penyimpanan Data Otomatis: Setelah registrasi selesai, sistem akan menyimpan data tamu baru dan tamu diarahkan Kembali untuk scan lalu sistem langsung mencatat kunjungan pertamanya.
- e. Konfirmasi Sistem: Sistem akan menampilkan pesan konfirmasi keberhasilan setelah proses pencatatan kunjungan atau registrasi selesai.

4.3.3 Hasil Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan spesifikasi perangkat lunak dan perangkat keras yang diperlukan untuk mendukung proses pengembangan dan implementasi "Sistem Pendataan Pengunjung Perpustakaan dengan Scan Wajah". Sistem ini dirancang sebagai platform berbasis web yang dapat diakses oleh pengguna (Mahasiswa dan Tamu) dan dikelola oleh Admin. Berikut ini adalah rincian kebutuhannya:

1. Kebutuhan Perangkat Lunak (Software)

Berikut adalah perangkat lunak yang digunakan dalam pengembangan sistem ini:

- a. Laravel Framework Digunakan sebagai framework PHP di sisi server (backend) untuk membangun fungsionalitas utama sistem, termasuk dasbor admin, manajemen data pengguna, dan pengelolaan riwayat kunjungan.
- b. Face API.js Merupakan pustaka (library) JavaScript inti yang digunakan untuk fitur deteksi dan pengenalan wajah di sisi klien (client-side). Pustaka ini berjalan di atas TensorFlow.js dan dieksekusi langsung di browser.

- c. Visual Studio Code Berfungsi sebagai code editor utama yang digunakan untuk menulis dan mengembangkan kode backend (PHP, Laravel) dan frontend (JavaScript, HTML).
- d. Laragon Digunakan sebagai lingkungan pengembangan lokal (local development environment) yang menyediakan server Apache/Nginx, database MySQL, dan PHP dalam satu paket untuk memudahkan proses pengembangan secara offline.
- e. MySQL dan PhpMyAdmin MySQL digunakan sebagai sistem manajemen basis data relasional (RDBMS) untuk menyimpan semua data terkait pengguna dan kunjungan. Sementara itu, PhpMyAdmin digunakan sebagai antarmuka grafis berbasis web untuk mengelola basis data MySQL tersebut.
- f. Web Browser Diperlukan peramban web modern (misalnya, Google Chrome, Firefox) untuk menjalankan aplikasi, baik dari sisi admin maupun dari sisi pengguna yang melakukan pemindaian wajah.
- g. Figma / Canva Digunakan pada tahap perancangan untuk membuat desain antarmuka pengguna (UI/UX), wireframe, dan elemen visual lainnya sebelum diimplementasikan ke dalam kode.

2. Kebutuhan Perangkat Keras (Hardware)

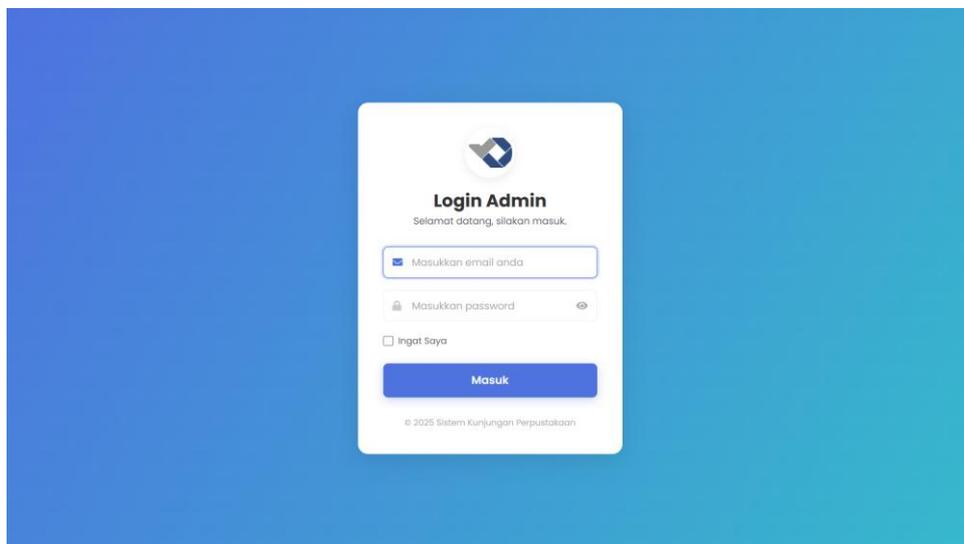
Perangkat keras yang diperlukan untuk pengembangan dan penerapan sistem ini antara lain:

- a. Laptop/PC (Untuk Pengembangan) Dibutuhkan komputer dengan spesifikasi yang memadai untuk menjalankan semua perangkat lunak pengembangan dengan lancar. Spesifikasi yang disarankan meliputi prosesor setara Core i3/Ryzen 3, RAM minimal 8 GB, dan penyimpanan SSD untuk performa yang lebih cepat.
- b. Kamera (Webcam) Merupakan perangkat keras penting yang terhubung ke komputer. Kamera ini berfungsi untuk menangkap citra wajah pengunjung secara real-time untuk diproses oleh sistem.

4.3 Production

Production adalah tahap pengembangan inti di mana sistem diwujudkan menjadi aplikasi fungsional menggunakan framework Laravel untuk backend admin dan user dan pustaka Face API.js untuk fitur pemindaian wajah di frontend. Proses ini mencakup pengembangan fungsionalitas CRUD untuk admin, pembuatan antarmuka pemindaian dan registrasi bagi pengunjung, serta integrasi logika pengenalan wajah secara real-time. Seluruh pengembangan kemudian diakhiri dengan tahap pengujian sistem secara menyeluruh untuk memastikan semua fitur berjalan sesuai harapan sebelum diterapkan.

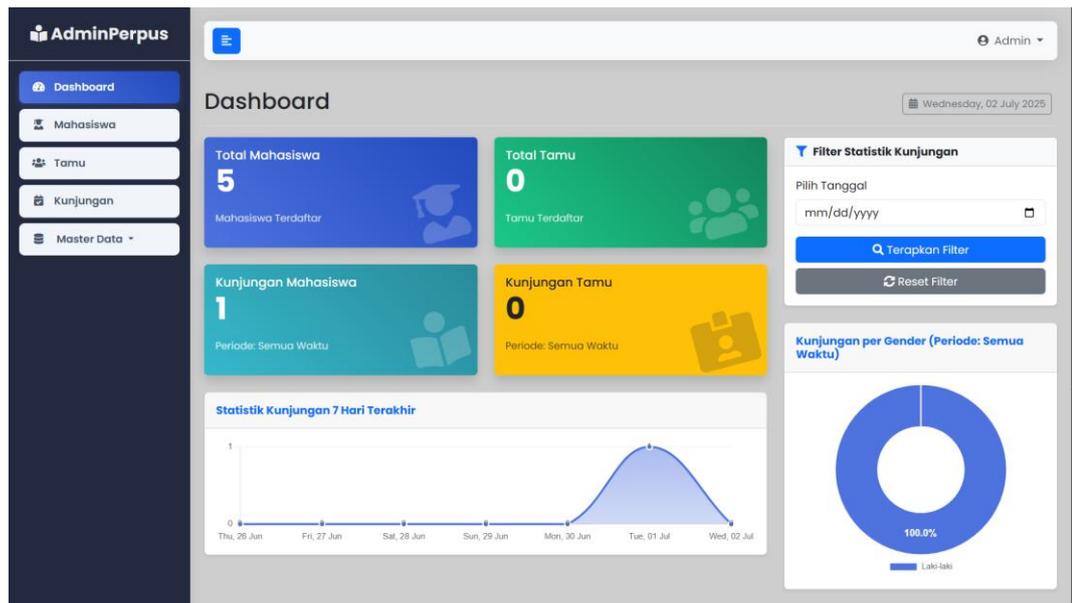
4.3.1 Halaman Login Admin



Gambar 4.1 Halaman Login Admin

Halaman Login Admin ini merupakan gerbang akses dan lapisan keamanan utama bagi administrator yang mengelola "Sistem Pendataan Pengunjung Perpustakaan dengan Scan Wajah". Admin harus memasukkan kredensial berupa email dan kata sandi pada kolom *input*-an yang tersedia. Setelah menekan tombol "Masuk", sistem akan memvalidasi data tersebut. Jika berhasil, admin akan diarahkan ke dasbor administratif utama.

4.3.2 Halaman *Dashboard*



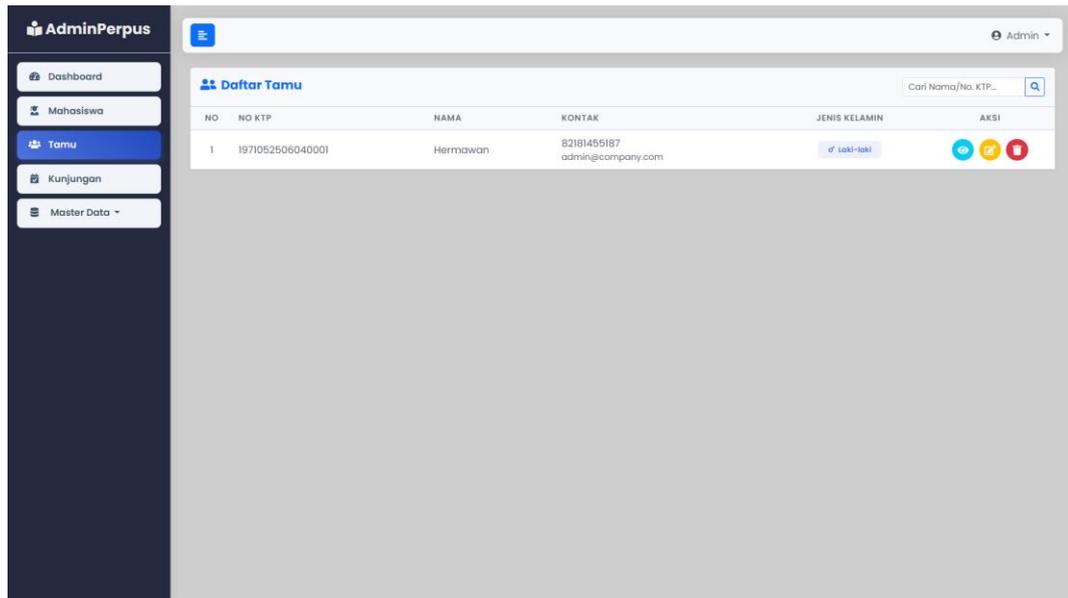
Gambar 4. 2 Halaman *Dashboard*

Halaman *Dashboard* merupakan pusat kontrol utama yang dapat diakses oleh admin setelah berhasil melakukan proses login. Pada halaman ini, ditampilkan ringkasan visual aktivitas pengunjung perpustakaan. Admin dapat memantau berbagai statistik penting, seperti total jumlah mahasiswa dan tamu yang terdaftar, serta jumlah kunjungan berdasarkan kategori pengunjung.

Dashboard ini juga dilengkapi dengan grafik garis yang menampilkan visual kunjungan selama tujuh hari terakhir, serta grafik pie yang menggambarkan distribusi statistik pengunjung. Terdapat fitur filter berdasarkan tanggal yang mendukung analisis data secara lebih spesifik.

Menu navigasi yang terletak di sisi kiri halaman menyediakan akses cepat menuju berbagai modul pengelolaan data, seperti manajemen data mahasiswa, data tamu, riwayat kunjungan, serta data master lainnya, sesuai dengan hak akses yang dimiliki oleh masing-masing pengguna.

4.3.3 Halaman Kelola Data Mahasiswa

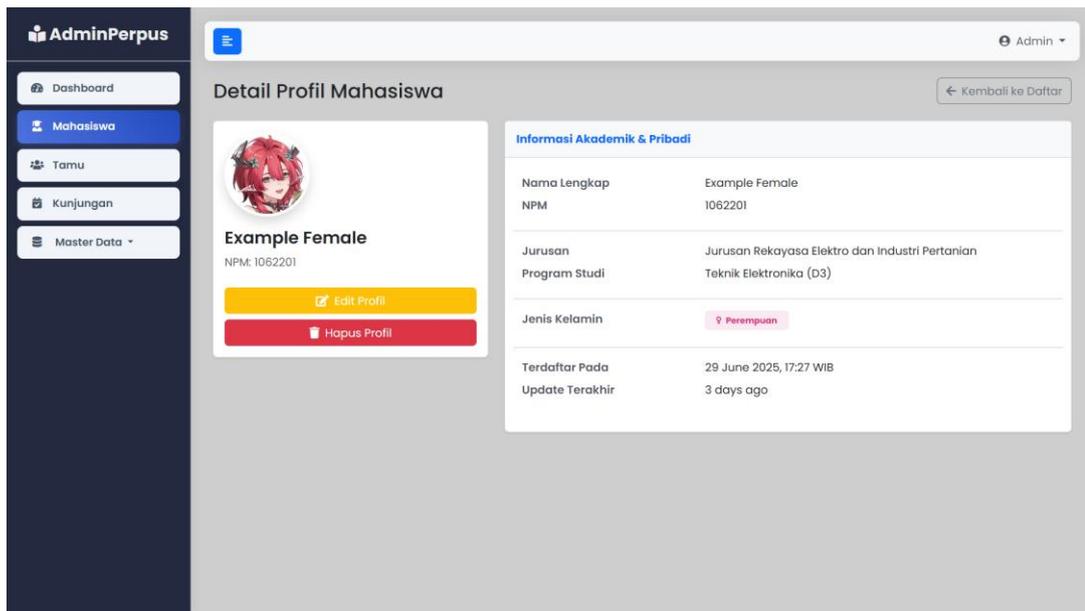


Gambar 4. 3 Halaman Kelola Data Mahasiswa

Halaman *Daftar Mahasiswa* adalah halaman utama yang digunakan oleh admin untuk mengelola seluruh data mahasiswa yang terdaftar dalam sistem. Halaman ini mengimplementasikan CRUD (*Create, Read, Update, Delete*), yang memungkinkan admin untuk menambahkan mahasiswa baru melalui tombol + *Tambah Mahasiswa Baru*, serta melihat detail, mengubah, atau menghapus data mahasiswa yang telah ada melalui tombol aksi yang tersedia pada setiap baris tabel.

Data mahasiswa yang ditampilkan mencakup informasi seperti nama, NPM, jenis kelamin, program studi, dan jurusan, yang disajikan dalam format table. Terdapat fitur pencarian dan filter berdasarkan kata kunci (misalnya nama atau NPM) serta berdasarkan program studi.

4.3.4 Halaman Show Data Mahasiswa

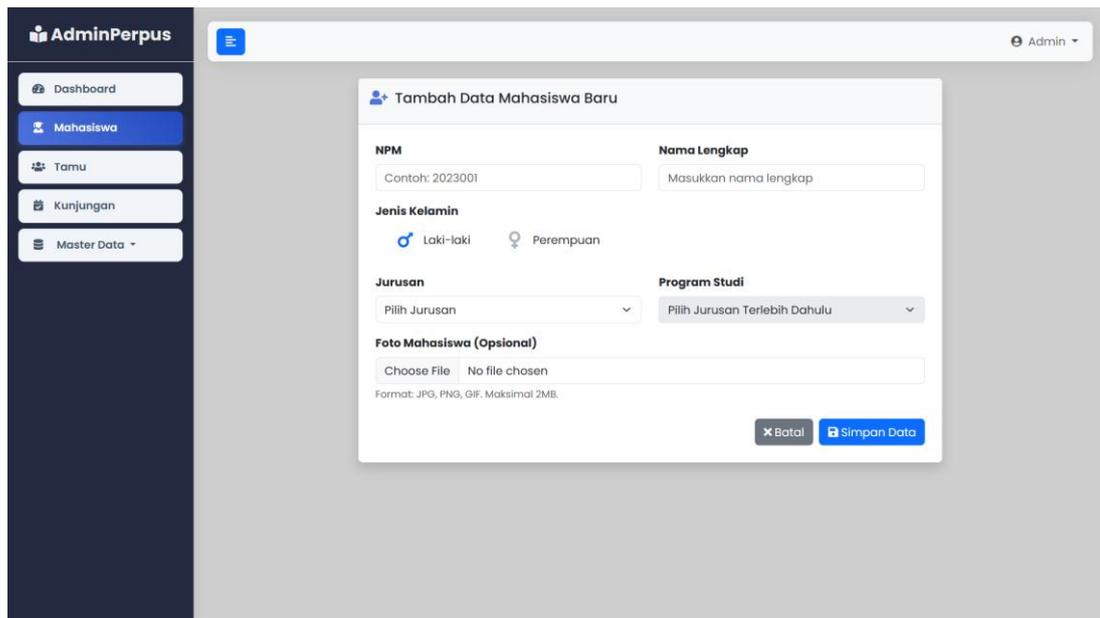


Gambar 4. 4 Halaman Show Data Mahasiswa

Halaman *Detail Profil Mahasiswa* merupakan halaman yang merepresentasikan fungsi *Read* atau *Show* dalam siklus manajemen data CRUD. Halaman ini diakses oleh admin setelah memilih salah satu data mahasiswa dari halaman *Daftar Mahasiswa*, dan berfungsi untuk menampilkan informasi lengkap mengenai mahasiswa yang bersangkutan.

Tampilan halaman terbagi menjadi dua panel utama. Panel kiri menyajikan ringkasan profil visual yang mencakup foto mahasiswa, nama, dan NPM, serta dilengkapi dengan tombol aksi langsung untuk *Edit Profil* (Update) dan *Hapus Profil* (Delete). Sementara itu, panel kanan menampilkan informasi akademik dan data pribadi secara lebih rinci, meliputi nama lengkap, NPM, jurusan, program studi, jenis kelamin, serta metadata sistem seperti tanggal pembuatan dan pembaruan terakhir data.

4.3.5 Halaman Tambah Data Mahasiswa



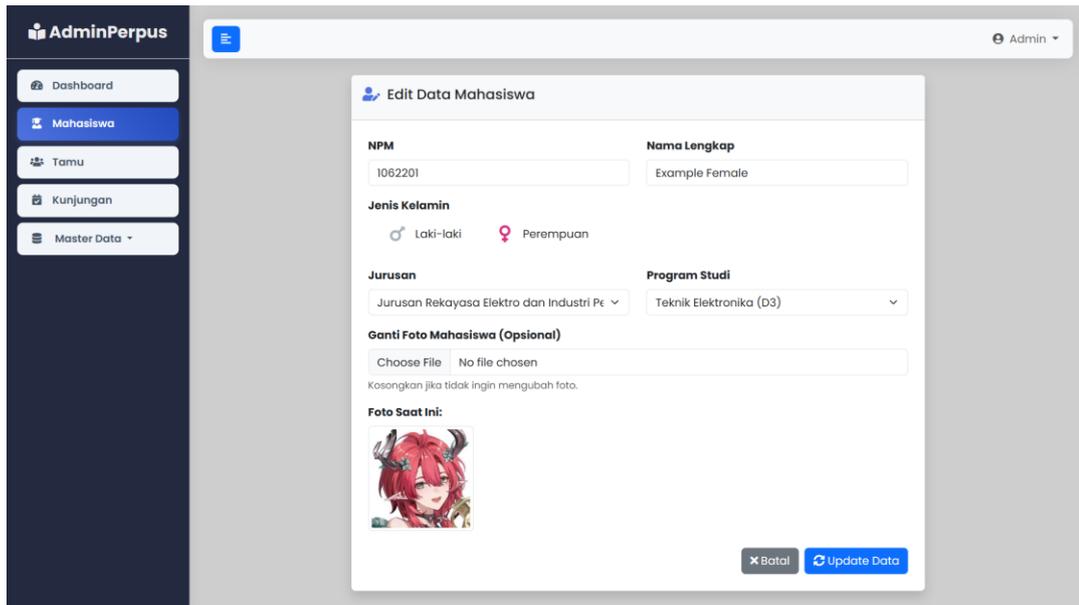
Gambar 4. 5 Halaman Tambah Data Mahasiswa

Halaman *Tambah Data Mahasiswa Baru* merupakan implementasi dari fungsi *Create* dalam siklus manajemen data CRUD dan dapat diakses oleh admin melalui tombol tambah pada halaman *Daftar Mahasiswa*. Di halaman ini terdapat formulir kosong yang digunakan untuk mendaftarkan mahasiswa, khususnya bagi mahasiswa yang datanya belum tersedia dalam sistem.

Admin diwajibkan mengisi data penting seperti NPM, nama lengkap, jenis kelamin, serta memilih jurusan dan program studi melalui menu dropdown yang tersedia. Tombol *Upload Foto* berfungsi untuk mengunggah foto formal mahasiswa sesuai dengan ketentuan format dan ukuran file yang telah ditetapkan.

Setelah seluruh informasi diisi dengan benar, admin dapat menyimpan data ke dalam basis data dengan menekan tombol *Simpan Data*, atau membatalkan proses dengan memilih tombol *Batal*.

4.3.6 Halaman Edit Data Mahasiswa



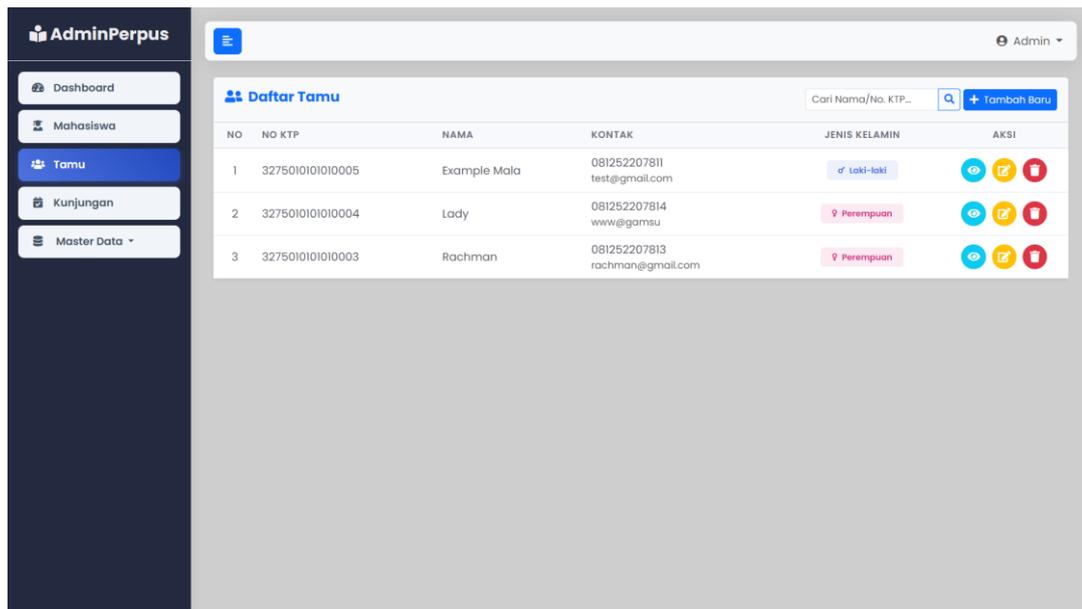
Gambar 4. 6 Halaman Edit Data Mahasiswa

Halaman *Edit Data Mahasiswa* adalah halaman yang menjalankan fungsi *Update* dalam siklus manajemen data CRUD. Halaman ini diakses oleh admin setelah menekan tombol *Edit Profil* pada halaman detail mahasiswa, dan digunakan untuk memperbarui informasi data mahasiswa yang telah terdaftar.

Halaman ini menampilkan formulir yang telah terisi dengan data mahasiswa sesuai dengan data di dalam *database*, sehingga memudahkan admin dalam melakukan perubahan. Informasi yang dapat diedit mencakup nama lengkap, jenis kelamin, jurusan, dan program studi, melalui input teks dan menu *dropdown* Jurusan dan Prodi. Selain itu, tersedia pula opsi untuk mengganti foto profil mahasiswa apabila diperlukan.

Setelah proses perubahan selesai, admin dapat menyimpan data yang diperbarui dengan menekan tombol *Update Data*, atau membatalkan proses dengan memilih tombol *Batal*.

4.3.7 Halaman Kelola Data Tamu

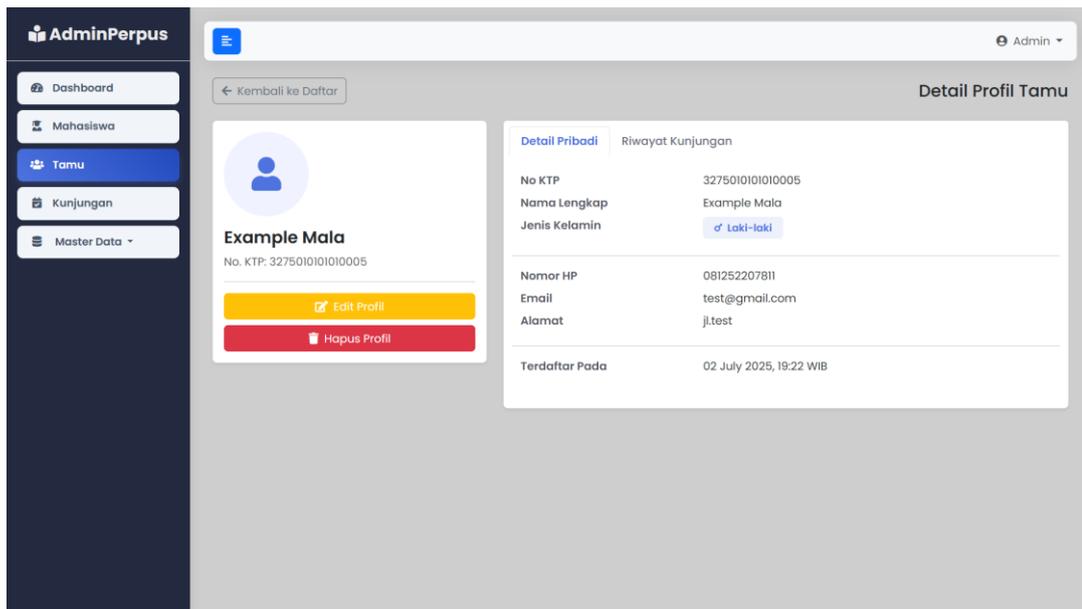


Gambar 4. 7 Halaman Kelola Data Tamu

Halaman *Daftar Tamu* merupakan halaman yang dapat diakses oleh admin melalui menu navigasi *Tamu*, dan berfungsi untuk menampilkan data tamu dan mengelola seluruh data pengunjung tamu yang terdaftar dalam sistem. Halaman ini mendukung implementasi penuh fungsi CRUD (*Create, Read, Update, Delete*), memungkinkan admin untuk menambahkan, melihat, memperbarui, dan menghapus data tamu sesuai kebutuhan.

Admin dapat menambahkan tamu baru secara manual melalui tombol + *Tambah Baru*, serta mengakses fitur detail, edit, atau hapus pada setiap data melalui ikon aksi yang tersedia di masing-masing baris tabel. Informasi tamu, seperti Nomor KTP, nama lengkap, kontak, dan jenis kelamin,

4.3.8 Halaman Show Data Tamu



Gambar 4. 8 Halaman Show Data Tamu

Halaman *Detail Profil Tamu* merupakan halaman yang merepresentasikan fungsi *Read* atau *Show* dalam siklus CRUD, dan dapat diakses oleh admin dengan menekan tombol tamu di samping pada halaman daftar. Halaman ini menampilkan data pengunjung tamu, dengan tampilan ringkasan profil di sisi kiri dan informasi mendetail di sisi kanan.

Pada tab *Detail Pribadi*, admin dapat melihat seluruh informasi personal tamu, termasuk Nomor KTP, nama lengkap, jenis kelamin, serta kontak yang dapat dihubungi. Selain itu, tersedia tab *Riwayat Kunjungan* yang menampilkan daftar kunjungan yang telah dilakukan oleh tamu tersebut, sehingga memudahkan proses pelacakan aktivitas secara historis.

Admin bisa mengedit atau menghapus data tamu dari sistem dengan menggunakan tombol *Edit Profile* atau *Hapus Profil*.

4.3.9 Halaman Edit Data Tamu

The screenshot displays the 'AdminPerpus' interface. On the left is a dark sidebar with navigation options: Dashboard, Mahasiswa, Tamu (highlighted), Kunjungan, and Master Data. The main content area is titled 'Edit Data Tamu' and contains a form with the following sections:

- Informasi Pribadi**
 - No KTP (Tidak bisa diubah)**: 3275010101010005
 - Nama Lengkap**: Example Mala
- Jenis Kelamin**: Radio buttons for Laki-laki (selected) and Perempuan.
- No HP**: 081252207811
- Email (Opsional)**: test@gmail.com
- Alamat**: jl.test

At the bottom right of the form are two buttons: 'Batal' (cancel) and 'Update Data' (save).

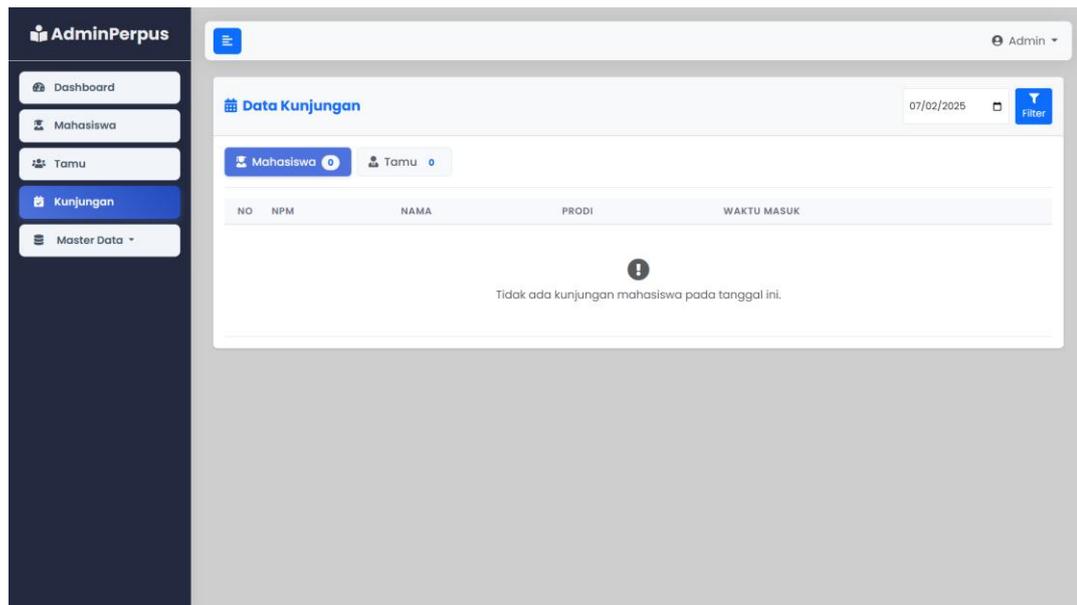
Gambar 4. 9 Halaman Edit Data Tamu

Halaman *Edit Data Tamu* merupakan halaman yang mengimplementasikan fungsi *Update* dalam siklus CRUD dan dapat diakses oleh admin melalui halaman detail profil tamu. Halaman ini digunakan untuk memperbarui informasi pengunjung tamu yang telah terdaftar dalam sistem.

Pada halaman ini, tersedia formulir *Informasi Pribadi* yang telah terisi dengan data tamu yang dipilih, memungkinkan admin untuk melakukan perubahan terhadap sejumlah data, seperti nama lengkap, jenis kelamin, nomor HP, email, dan alamat. Untuk menjaga integritas data, Nomor KTP sebagai identitas utama ditampilkan dalam keadaan tidak dapat diedit.

Setelah melakukan penyesuaian, admin dapat menyimpan perubahan dengan menekan tombol *Update Data*, atau membatalkan proses pengeditan dengan memilih tombol *Batal*.

4.3.10 Halaman Kunjungan

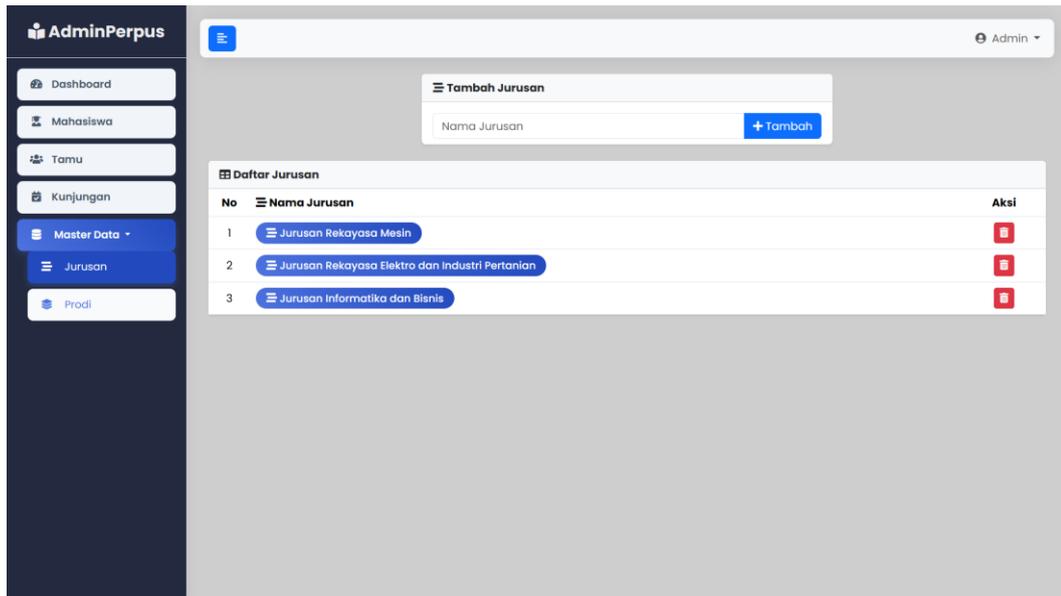


Gambar 4. 10 Halaman Kunjungan Tamu dan Mahasiswa

Halaman *Data Kunjungan* merupakan halaman yang berfungsi untuk memantau riwayat kehadiran pengunjung perpustakaan secara terperinci. Terdapat dua tombol utama, yaitu *Mahasiswa* dan *Tamu*, yang berfungsi sebagai filter untuk mengelompokkan data kunjungan berdasarkan kategori pengunjung.

Apabila admin memilih tombol *Mahasiswa*, sistem akan menampilkan tabel yang memuat daftar kunjungan mahasiswa pada tanggal yang dipilih, lengkap dengan informasi seperti NPM, nama, program studi, dan waktu kedatangan. Sebaliknya, jika tombol *Tamu* dipilih, tabel akan diperbarui untuk menampilkan data kunjungan dari pengunjung tamu. Terdapat tombol filter pemilihan tanggal di pojok kanan atas yang memungkinkan admin untuk menelusuri data kunjungan pada hari-hari sebelumnya.

4.3.11 Halaman Master Jurusan

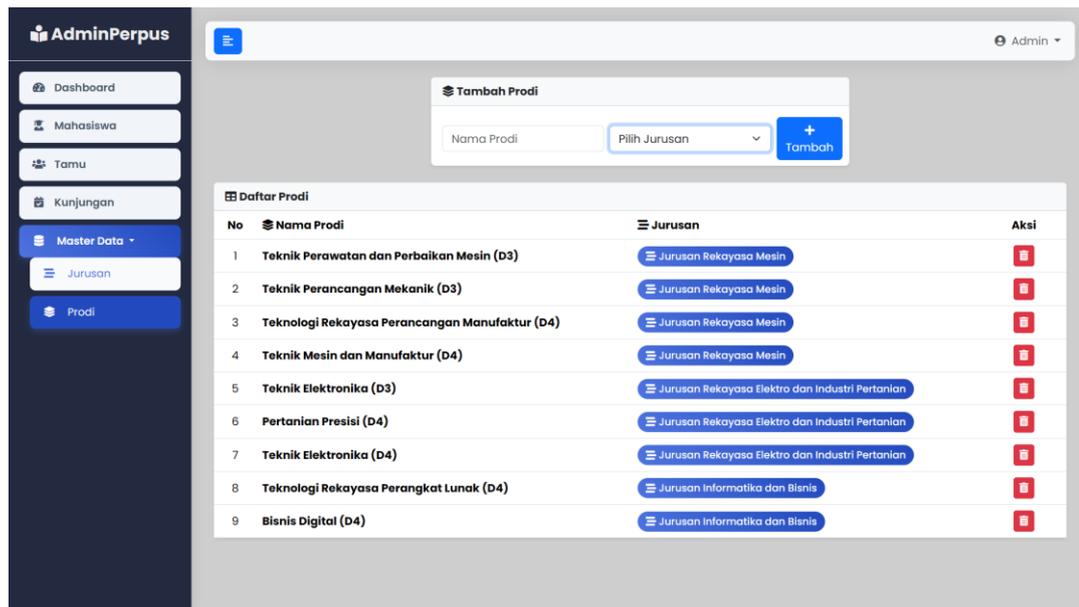


Gambar 4. 11 Halaman Master Jurusan

Halaman pengelolaan *Jurusan* merupakan bagian dari menu *Master Data* yang dapat diakses oleh admin dan berfungsi sebagai pusat manajemen daftar jurusan yang tersedia. Admin dapat menambahkan jurusan baru dengan mengisi nama jurusan pada formulir *Tambah Jurusan*.

Di bagian bawah, terdapat *Daftar Jurusan* yang menampilkan seluruh jurusan yang telah terdaftar dalam sistem. Setiap baris data terdapat ikon hapus pada kolom *Aksi* yang memungkinkan admin untuk menghapus data jurusan yang tidak lagi digunakan.

4.3.12 Halaman Master Program Studi

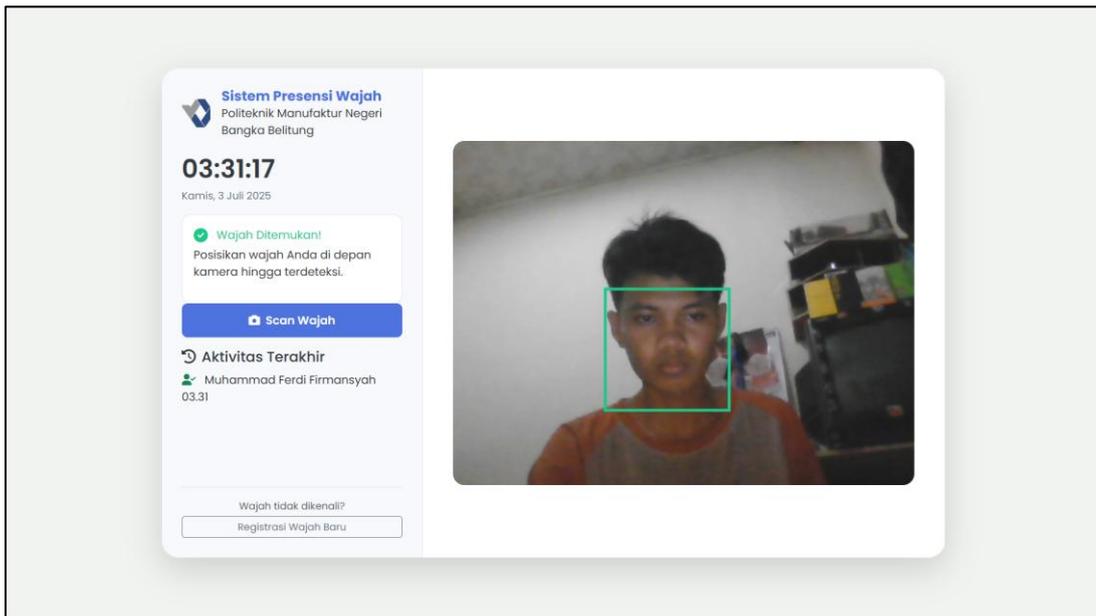


Gambar 4. 12 Halaman Master Program Studi

Halaman pengelolaan *Prodi* (Program Studi) merupakan bagian dari menu *Master Data* yang dapat diakses oleh admin, dan berfungsi untuk mengatur daftar program studi yang tersedia dalam sistem. Pada halaman ini, admin dapat menambahkan program studi baru melalui formulir *Tambah Prodi* dengan mengisi nama program studi dan memilih jurusan induk melalui menu dropdown, yang mengacu pada relasi data antara jurusan dan program studi.

Di bagian bawah, terdapat *Daftar Prodi* yang menampilkan seluruh program studi yang telah terdaftar beserta jurusan terkait, disajikan dalam format tabel. Setiap tabel terdapat tombol ikon aksi yang memungkinkan admin untuk menghapus data yang tidak diperlukan.

4.3.13 Halaman Scan Wajah User



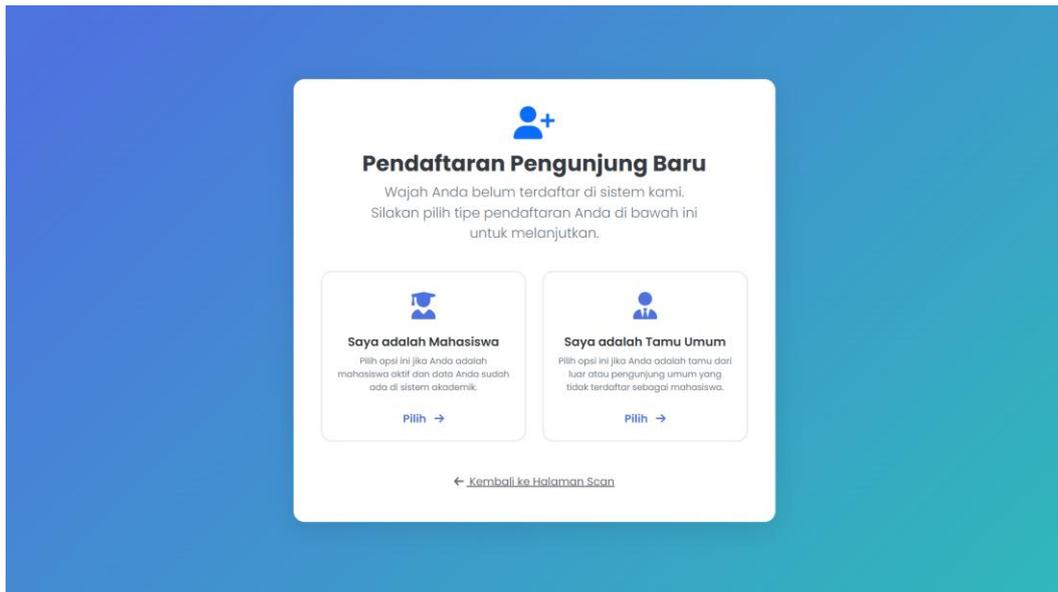
Gambar 4. 13 Halaman Scan Wajah User

Halaman *Sistem Presensi Wajah* merupakan halaman utama yang diakses oleh seluruh pengunjung perpustakaan, baik mahasiswa maupun tamu, dan menjadi komponen inti dari sistem yang dikembangkan. Halaman ini menampilkan umpan video langsung dari kamera pada sisi kanan layar, di mana teknologi pendeteksi wajah (*face detection*) bekerja secara real-time untuk mengenali wajah pengunjung dan menampilkannya dalam kotak pembatas.

Pada sisi kiri halaman, ditampilkan informasi waktu dan tanggal terkini, serta tombol utama *Scan Wajah* yang digunakan untuk memulai proses identifikasi. Apabila sistem berhasil mengenali wajah pengunjung, bagian *Aktivitas Terakhir* akan secara otomatis menampilkan nama pengunjung yang telah tercatat dalam sistem.

Terdapat tombol *Registrasi Wajah Baru* untuk pengunjung mendaftar di bagian bawah halaman yang berfungsi untuk mengarahkan pengguna ke proses pendaftaran jika data mereka belum ada di dalam *database*.

4.3.14 Halaman Pendaftaran Pengunjung Baru

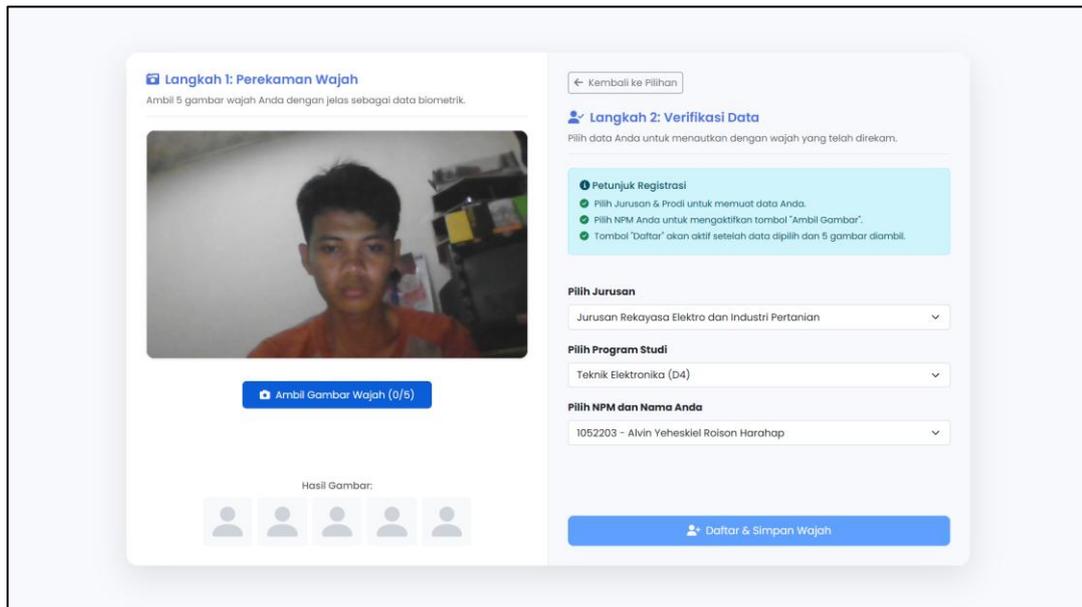


Gambar 4. 14 Halaman Pendaftaran Pengunjung Baru

Halaman *Pendaftaran Pengunjung Baru* ditampilkan ketika sistem tidak berhasil mengenali wajah pengunjung pada tahap pemindaian, dan pengunjung menekan tombol registrasi. Halaman ini berfungsi sebagai titik pemisah alur (*branching point*) yang mengarahkan pengguna untuk memilih jenis pendaftaran sesuai dengan status mereka. Terdapat dua opsi yang tersedia, yaitu *Saya adalah Mahasiswa* bagi pengunjung yang merupakan mahasiswa dan telah memiliki data di dalam sistem, serta *Saya adalah Tamu Umum* bagi pengunjung dengan status tamu.

Pilihan yang dipilih akan menentukan halaman registrasi selanjutnya. Mahasiswa akan diarahkan ke halaman untuk menautkan data akademik yang telah diinput oleh admin dengan data biometrik wajah, sedangkan tamu akan diarahkan ke halaman pengisian formulir biodata secara mandiri. Selain itu, tersedia pula tombol *Kembali ke Halaman Scan* bagi pengguna yang memutuskan untuk membatalkan proses pendaftaran.

4.3.15 Halaman Registrasi Mahasiswa



Gambar 4. 15 Halaman Registrasi Wajah Mahasiswa

Halaman registrasi wajah mahasiswa merupakan tahapan di mana mahasiswa mendaftarkan wajahnya untuk ditautkan dengan data akademik yang telah diinput oleh admin. Proses registrasi ini terdiri dari dua langkah, yaitu perekaman wajah dan verifikasi data. Pada langkah pertama, mahasiswa diminta untuk mengambil lima sampel gambar wajah sebagai data biometrik. Selanjutnya, pada langkah kedua, mahasiswa memilih jurusan dan program studi melalui menu *dropdown*. Opsi *dropdown* tersebut berfungsi sebagai filter untuk menampilkan data nama dan NPM yang sesuai dengan Mahasiswa tersebut. Setelah seluruh tahapan selesai, tombol “Daftar dan Simpan Wajah” akan aktif dan dapat digunakan untuk menyelesaikan proses registrasi.

4.3.16 Halaman Registrasi Wajah Tamu

The screenshot displays a user registration interface for a guest. It is titled 'Langkah 1: Perekaman Wajah' and 'Langkah 2: Isi Data Diri'. The first step involves capturing five face images for biometric data, with a camera view and a button 'Ambil Gambar Wajah (0/5)'. The second step is a manual data entry form including fields for 'Nomor KTP' (with a hint '16 digit NIK'), 'Nama Lengkap', 'Jenis Kelamin' (radio buttons for 'Laki-laki' and 'Perempuan'), 'Nomor HP', 'Email (Opsional)', and 'Alamat'. A 'Daftar & Simpan Wajah' button is located at the bottom right of the form.

Gambar 4. 16 Halaman Registrasi Wajah Tamu

Halaman registrasi ini ditujukan bagi tamu umum yang mendaftar secara mandiri setelah memilih opsi sebagai tamu. Proses registrasi terdiri dari dua tahap, yaitu perekaman lima sampel wajah sebagai data biometrik dan pengisian data diri secara manual yang mencakup nomor KTP, nama lengkap, jenis kelamin, dan kontak. Berbeda dengan mahasiswa yang memilih data yang telah tersedia, tamu harus mengisi seluruh informasi secara langsung. Setelah kedua tahap selesai, tombol daftar dan simpan wajah akan aktif untuk menyelesaikan proses pendaftaran.

4.4 Pengujian

Setelah tahap pengembangan selesai, sistem yang dibangun akan memasuki fase pengujian menyeluruh untuk proses verifikasi dan validasi. Tujuan utama dari tahap ini adalah untuk memastikan bahwa perangkat lunak tidak hanya berfungsi sesuai dengan spesifikasi teknis yang telah ditetapkan, tetapi juga dapat diterima dan menjawab kebutuhan dari para pengguna akhir secara efektif. Untuk mencapai hal ini, serangkaian metode evaluasi dan pengujian yang komprehensif diterapkan guna menjamin kualitas dan keandalan sistem sebelum diimplementasikan sepenuhnya.

2.8.1 Analisis Kinerja Interaksi Pengguna

Salah satu aspek yang diuji adalah perbandingan kecepatan interaksi pada sistem lama dan sistem baru dari perspektif pengguna. Pengukuran ini difokuskan

pada durasi yang dibutuhkan dari aksi awal pengguna hingga sistem memberikan konfirmasi bahwa data telah berhasil dicatat.

Metodologi pengukuran kecepatan difokuskan pada total durasi interaksi untuk setiap sistem, yang dihitung dengan cara berbeda sesuai alur kerjanya. Untuk sistem lama, waktu dihitung mulai dari saat pengunjung mengetikkan Nomor Pokok Mahasiswa (NPM) mereka, menekan tombol konfirmasi, hingga sistem menampilkan respon bahwa data kunjungan telah terekam. Sedangkan untuk sistem baru, waktu dihitung mulai dari saat pengunjung menekan tombol "Scan Wajah", dilanjutkan dengan proses identifikasi wajah oleh sistem, hingga sistem menampilkan respon konfirmasi bahwa data kunjungan telah berhasil direkam. Berdasarkan pengujian sampel, waktu rata-rata yang dibutuhkan pada sistem lama adalah:

Jumlah Pengujian: 3

Pengujian pertama: 3.88 detik

Pengujian kedua: 5.72 detik

Pengujian ketiga: 4.87 detik

Rata-rata Waktu Sistem Lama = $(3.88 + 5.72 + 4.87) \div 3 = 4.82$ detik

Waktu ini kemudian dibandingkan dengan sistem baru yang dalam pengujian membutuhkan waktu 5.90 detik. Untuk memahami implikasi perbedaan waktu ini dalam skala yang lebih besar, dilakukan estimasi waktu total untuk melayani 66 pengunjung sesuai data kunjungan perpustakaan pada 17-18 Juli 2025.

Estimasi Waktu Total Sistem Lama:

$4.82 \text{ detik/pengunjung} \times 66 \text{ pengunjung} = 318.12 \text{ detik} \approx 5.30 \text{ menit}$

Estimasi Waktu Total Sistem Baru:

$5.90 \text{ detik/pengunjung} \times 66 \text{ pengunjung} = 389.40 \text{ detik} \approx 6.49 \text{ menit}$

Berdasarkan perhitungan, terlihat bahwa proses pindai wajah memang membutuhkan waktu sedikit lebih lama dibandingkan dengan proses input NPM. Namun, keunggulan sistem *scan* wajah ini bukanlah pada kecepatan pertransaksinya, melainkan pada kemampuannya untuk menjamin keakuratan data dan validitas data.

Kecepatan dan akurasi sistem pindai wajah sangat bergantung pada beberapa kondisi utama, yaitu:

- a. Pencahayaan: Cahaya di area pemindaian harus cukup terang dan tidak menyilaukan agar wajah dapat terdeteksi dengan baik.
- b. Posisi Wajah: Wajah pengguna harus terlihat jelas, menghadap ke depan, dan tidak terhalang oleh objek seperti masker atau tangan.
- c. Koneksi Internet: Diperlukan koneksi yang stabil saat pertama kali membuka halaman untuk memuat model-model AI yang dibutuhkan sistem.
- d. Kualitas Kamera: Kejelasan gambar yang dihasilkan kamera sangat memengaruhi kecepatan dan akurasi deteksi. Pengujian sistem ini menggunakan webcam standar beresolusi 720p.

4.5 Implementasi

Pada tahapan implementasi, "Sistem Pendataan Pengunjung Perpustakaan dengan Scan Wajah" melalui proses pengujian menyeluruh untuk memastikan bahwa seluruh fitur dan alur kerja sistem berjalan dengan baik sesuai dengan tujuannya. Pengujian ini difokuskan pada verifikasi fungsionalitas dari sudut pandang setiap pengguna (Admin, Mahasiswa, dan Tamu) untuk menjamin sistem dapat bekerja secara optimal di lingkungan nyata. Metode pengujian yang digunakan black-box testing, di mana pengujian dilakukan untuk memeriksa apakah setiap fungsi memberikan hasil yang diharapkan tanpa perlu melihat struktur internal kodenya.

Pengujian ini mencakup validasi seluruh alur kerja sistem. Untuk peran Admin, pengujian meliputi proses autentikasi saat login, fungsionalitas dasbor dalam menampilkan data statistik, serta cara kerja *Create, Read, Update, Delete* (CRUD) pada halaman pengelolaan data mahasiswa, tamu, dan master data (jurusan dan prodi). Untuk peran Pengunjung (Mahasiswa dan Tamu), pengujian difokuskan pada fungsi inti sistem, yaitu keakuratan proses pemindaian wajah untuk mencatat kunjungan, serta kelancaran alur registrasi pengguna baru, baik untuk mahasiswa yang menautkan data maupun tamu yang melakukan pendaftaran mandiri.

Tujuan utama dari pengujian ini adalah untuk memvalidasi bahwa sistem telah memenuhi semua kebutuhan fungsional, berjalan dengan stabil, dan memberikan

pengalaman pengguna yang efisien dan mudah digunakan. Hasil dari setiap skenario pengujian yang dilakukan kemudian didokumentasikan dalam lembar validasi untuk menjadi bukti kesiapan sistem sebelum diterapkan sepenuhnya. Berikut hasil dari pengujian yang telah dilakukan.

4.5.1 Pengujian Validasi Ahli Media

Validasi ahli media dilakukan untuk mengevaluasi kualitas sistem dari aspek antarmuka (UI). Hasil dari pengujian yang telah dilakukan tersebut adalah sebagai berikut.

1. Praktisi Penguji Pertama

a. Pengujian Halaman Login Admin

Table 4. 3 Pengujian Halaman Login Admin

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Login dengan email dan password yang benar.	Admin berhasil masuk dan diarahkan ke halaman dasbor utama.		[<input checked="" type="checkbox"/>]Diterima [<input type="checkbox"/>]Ditolak
2.	Login dengan email yang salah/tidak terdaftar.	Sistem menampilkan pesan error bahwa email tidak terdaftar.		[<input checked="" type="checkbox"/>]Diterima [<input type="checkbox"/>]Ditolak
3.	Login dengan password yang salah.	Sistem menampilkan pesan error bahwa password salah.		[<input checked="" type="checkbox"/>]Diterima [<input type="checkbox"/>]Ditolak

b. Pengujian Halaman *Dashboard* Admin

Table 4. 4 Pengujian Halaman Dashboard Admin

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Berhasil login dan masuk ke dasbor.	Halaman dasbor tampil dengan semua komponen, termasuk kartu statistik, grafik, dan menu navigasi.		[√]Diterima []Ditolak
2.	Memeriksa kartu statistik ringkas.	Kartu menampilkan data jumlah mahasiswa, tamu, dan total kunjungan dengan benar.		[√]Diterima []Ditolak
3.	Memeriksa grafik kunjungan.	Grafik menampilkan data tren kunjungan dengan akurat.		[√]Diterima []Ditolak
4.	Menggunakan filter tanggal.	Data pada kartu statistik dan grafik berubah sesuai dengan rentang tanggal yang dipilih.		[√]Diterima []Ditolak

c. Pengujian Halaman Kelola Data Mahasiswa (CRUD)

Table 4. 5 Pengujian Halaman Kelola Data Mahasiswa

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Mengakses menu "Mahasiswa".	Halaman menampilkan daftar mahasiswa dalam format tabel beserta		[√]Diterima []Ditolak

		tombol aksi (Lihat, Edit, Hapus).	
2.	Menekan tombol "Tambah Mahasiswa Baru".	Halaman formulir tambah data mahasiswa baru muncul.	[√]Diterima []Ditolak
3.	Mengisi dan menyimpan data mahasiswa baru yang valid.	Data mahasiswa baru berhasil tersimpan dan muncul di tabel daftar mahasiswa.	[√]Diterima []Ditolak
4.	Menekan tombol "Edit" pada salah satu data.	Halaman edit muncul dengan formulir yang sudah terisi data mahasiswa tersebut.	[√]Diterima []Ditolak
5.	Mengubah dan menyimpan data pada halaman edit.	Data mahasiswa pada tabel berhasil diperbarui.	[√]Diterima []Ditolak
6.	Menekan tombol "Hapus" pada salah satu data.	Setelah konfirmasi, data mahasiswa terhapus dari daftar.	[√]Diterima []Ditolak

d. Pengujian Halaman Scan Wajah & Registrasi Pengunjung

Table 4. 6 Pengujian Halaman Scan Wajah & Registrasi Pengunjung

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Pengguna (mahasiswa/tamu) terdaftar melakukan scan wajah.	Wajah berhasil terdeteksi, sistem mengenali pengguna, dan wajah.	[√]Diterima []Ditolak	

		mencatat kunjungan.	
2.	Pengguna baru (tidak terdaftar) melakukan scan wajah.	Wajah terdeteksi namun tidak dikenali, sistem mengarahkan ke halaman registrasi.	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3.	Registrasi Mahasiswa: Memilih data diri dari dropdown dan mengambil 5 gambar wajah.	Mahasiswa berhasil menautkan data akademik dengan data wajah dan menyelesaikan registrasi.	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4.	Registrasi Tamu: Mengisi form biodata dan mengambil 5 gambar wajah.	Tamu berhasil membuat akun baru dan menyelesaikan registrasi.	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
5.	Pengguna yang baru mendaftar melakukan scan wajah kembali.	Sistem berhasil mengenali dan mencatat kunjungan pertama.	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

2. Praktisi Penguji Kedua

a. Pengujian Halaman Login Admin

Table 4. 7 Pengujian Halaman Login Admin

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
----	-------------------	-----------------------	-----------------	------------

1.	Login dengan email dan password yang benar.	Admin berhasil masuk dan diarahkan ke halaman dasbor utama.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2.	Login dengan email yang salah/tidak terdaftar.	Sistem menampilkan pesan error bahwa email tidak terdaftar.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3.	Login dengan password yang salah.	Sistem menampilkan pesan error bahwa password salah.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

b. Pengujian Halaman Dashboard Admin

Table 4. 8 Pengujian Halaman Dashboard Admin

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Berhasil login dan masuk ke dasbor.	Halaman dasbor tampil dengan semua komponen, termasuk kartu statistik, grafik, dan menu navigasi.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2.	Memeriksa kartu statistik ringkas.	Kartu menampilkan data jumlah mahasiswa, tamu, dan total kunjungan dengan benar.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

3.	Memeriksa grafik kunjungan.	Memeriksa grafik kunjungan.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4.	Menggunakan filter tanggal.	Data pada kartu statistik dan grafik berubah sesuai dengan rentang tanggal yang dipilih.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

c. Pengujian Halaman Kelola Data Mahasiswa (CRUD)

Table 4. 9 Pengujian Halaman Kelola Data Mahasiswa

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Mengakses menu "Mahasiswa".	Halaman menampilkan daftar mahasiswa dalam format tabel beserta tombol aksi (Lihat, Edit, Hapus).	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2.	Menekan tombol "Tambah Mahasiswa Baru".	Halaman formulir tambah data mahasiswa baru muncul.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3.	Mengisi dan menyimpan data mahasiswa baru yang valid.	Data mahasiswa baru berhasil tersimpan dan muncul di tabel daftar mahasiswa.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4.	Menekan tombol "Edit" pada salah satu data.	Halaman edit muncul dengan formulir yang sudah terisi data	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

		mahasiswa tersebut		
5.	Mengubah dan menyimpan data pada halaman edit.	Data mahasiswa pada tabel berhasil diperbarui.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
6.	Menekan tombol "Hapus" pada salah satu data.	Setelah konfirmasi, data mahasiswa terhapus dari daftar.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
7.	Mencari data menggunakan fitur pencarian dan filter	Menampilkan data sesuai yang di cari	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

d. Pengujian Halaman Kelola Data Tamu

Table 4. 10 Pengujian Halaman Kelola Data Tamu

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Mengakses menu "Tamu".	Halaman menampilkan daftar Tamu dalam format tabel beserta tombol aksi (Lihat, Edit, Hapus).	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2.	Menekan tombol "Edit" pada salah satu data.	Halaman edit muncul dengan formulir yang sudah terisi data mahasiswa tersebut.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

3.	Mengubah dan menyimpan data pada halaman edit.	Data mahasiswa pada tabel berhasil diperbarui.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4.	Menekan tombol "Hapus" pada salah satu data.	Setelah konfirmasi, data mahasiswa terhapus dari daftar.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
5.	Mencari data menggunakan fitur pencarian dan filter	Menampilkan data sesuai yang di cari	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

e. Pengujian Halaman Kunjungan

Table 4. 11 Pengujian Halaman Kunjungan

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Mengakses menu "Kunjungan"	Halaman menampilkan daftar Kunjungan Tamu dan Mahasiswa	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2.	Menekan tombol "Lihat Detail" pada salah satu data.	Menampilkan Halaman Detail Kunjungan.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3.	Menekan tombol "Cetak".	Menghasilkan data yang bisa di print dalam bentuk PFD.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Berdasarkan hasil validasi oleh ahli media, disimpulkan bahwa seluruh fungsionalitas sistem telah berjalan sesuai rancangan. Semua alur kerja utama mulai dari manajemen data oleh Admin hingga proses pemindaian dan registrasi wajah oleh Pengunjung berfungsi dengan benar, responsif, dan tanpa kendala.

4.6 Hasil Kuesioner

Hasil Kuisisioner Pengunjung Mahasiswa/Tamu

Table 4. 12 Hasil Kuesioner Pengunjung Mahasiswa/Tamu

No	Nama	Pertanyaan										Skor
		1	2	3	4	5	6	7	8	9	10	
1	Aulia Sabila	4	5	4	5	4	5	4	5	4	5	45
2	Nafisa Silmi Zirconi	5	5	5	5	5	5	5	5	5	5	50
3	Atha Said Fajri	4	5	5	4	5	3	5	5	5	5	46
4	Alvin Yeheskiel Roison Harahap	5	5	5	4	5	5	4	4	4	4	45
5	Muhammad Raihan Pasha	5	5	4	5	4	4	5	5	5	5	47
6	Septiandi Hermawan	5	5	4	5	4	3	5	4	4	5	44
7	Deswita Syahran	4	4	4	4	4	5	4	4	4	5	42
8	Dicky Rahmat Rifai	4	4	4	4	4	4	4	4	4	4	40
9	Ana Maharani	4	3	4	4	4	3	4	4	4	4	38
10	Jimmi Rianto	5	5	4	5	4	4	4	5	4	5	45
11	Rizqi Pratama	5	5	4	5	4	5	4	5	5	5	47
12	Taqiy Jayya Muhammad	4	4	4	4	5	3	4	4	5	4	41
13	Zadin Fatan Zirconi	5	5	4	4	5	3	4	4	5	5	44
TOTAL SKOR											574	

4.7 Hasil Perhitungan User Acceptance Test (UAT)

Hasil Perhitungan Pengunjung Mahasiswa/Tamu

Berikut ini merupakan hasil perhitungan dari pengujian pada pengunjung Mahasiswa/Tamu yang sudah dilakukan:

Rumus Perhitungan:

Skor Maksimal = (Maksimal Poin \times Total Pertanyaan \times Total Penguji)

Hasil UAT = (Total skor \div Skor maksimal) \times 100%

Perhitungan:

Skor Maksimal = (5 \times 10 \times 13) = 650

Hasil UAT = (574 \div 650) \times 100%

= 88,31%



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengembangan dan pengujian, dapat disimpulkan bahwa "Sistem Pendataan Pengunjung Perpustakaan dengan Scan Wajah" ini berhasil dibangun dan mencapai tujuannya. Intinya, sistem ini berhasil mengatasi masalah utama yang ada, yaitu potensi kesalahan saat pengguna mengetik Nomor Pokok Mahasiswa (NPM). Melalui penambahan fitur pindai wajah, sistem secara efektif menghilangkan risiko salah ketik atau keliru memasukkan data orang lain, sehingga data kunjungan yang tercatat menjadi jauh lebih akurat.

Meskipun hasil pengujian menunjukkan satu kali proses pindai wajah (5,90 detik) membutuhkan waktu sedikit lebih lama dibandingkan rata-rata input NPM yang tanpa kesalahan (4,82 detik), sistem ini terbukti lebih unggul dalam hal pengalaman pengguna dan efisiensi proses secara keseluruhan. Hal ini dibuktikan dengan hasil UAT (*User Acceptance Test*) yang mencapai skor 88,31%, menunjukkan bahwa pengguna merasa sistem ini sangat baik dan mudah digunakan. Proses yang lebih lancar tanpa perlu coba-ulang akibat salah ketik.

5.2 Saran

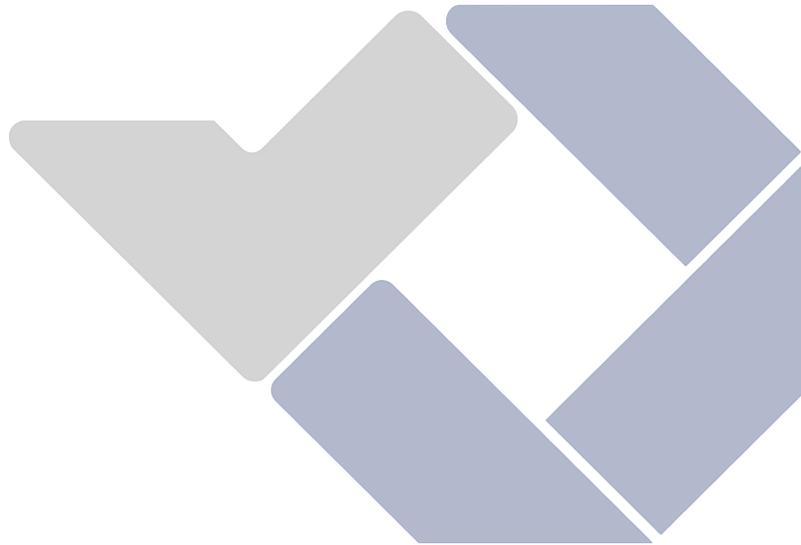
Berdasarkan hasil pengembangan dan pengujian sistem yang telah dilakukan, terdapat beberapa saran untuk pengembangan di masa mendatang agar sistem dapat menjadi lebih optimal, aman, dan fungsional:

1. Penggunaan Kamera dengan Spesifikasi Lebih Tinggi Untuk meningkatkan akurasi deteksi wajah dalam berbagai kondisi pencahayaan yang kurang ideal, pengembangan di masa depan dapat mempertimbangkan penggunaan perangkat keras kamera yang lebih canggih, seperti kamera inframerah (IR), yang tidak terlalu terpengaruh oleh perubahan cahaya lingkungan.

DAFTAR PUSTAKA

- Adindya Giovanni, W. I. (2023). Pendeteksi Wajah sebagai Sebuah Sistem Keamanan Ruangan. *Prosiding Seminar Nasional Fisika (E-Journal)*, XI.
- Bambang Warsuta, R. E. (2024). Sistem Pengendalian Akses Berbasis Face-Recognition dengan Face-API.js dan Algoritma Manhattan Distance. *JURNAL MULTINETICS*, 113-120.
- Bimo Akbar Fadli, E. W. (2023). Pengenalan Wajah Dengan Face-Api.js Berbasis CNN dan Geolokasi Menggunakan Equirectangular Approximation. *Jurnal Ilmiah Komputer*, 935-944.
- Endarti, S. (2022). Perpustakaan sebagai Tempat Rekreasi Informasi. *ABDI PUSTAKA - Jurnal Perpustakaan dan Kearsipan*, 23-28.
- Fatimah Azzahro, N. M. (2024). Pelatihan Promosi Website Informasi RW13 Cipinang Melayu. *GENDIS Jurnal Pengabdian Masyarakat*, 29–33.
- Gracia Yoel Christiawan, R. A. (2022). Penerapan Metode Convolutional Neural Network (CNN) Dalam Mengklasifikasikan Penyakit Daun Tanaman Padi. *J-INTECH (Journal of Information and Technology)*, 294-306.
- Hartati, E. (2022). Sistem Informasi Transaksi Gudang Berbasis Website pada CV. Asyura. *Klik - Jurnal Ilmu Komputer*, 12-18.
- Herno Aji Sanjaya, A. B. (2024). Prototype Video Recognition Menggunakan Face API Untuk Keamanan Ruang Kritis Berbasis Web. *Digital Transformation Technology (Digitech)*, 876-886.
- Herno Aji Sanjaya, A. B. (2024). Prototype Video Recognition Menggunakan Face API Untuk Keamanan Ruang Kritis Berbasis Web. *Digital Transformation Technology (Digitech)*, 876-886.
- Kherina Surya Ningsih, N. J. (2022). Aplikasi Buku Tamu Menggunakan Fitur Kamera dan AJAX Berbasis Website pada Kantor Dispora Kota Medan. *SITek: Jurnal Sains, Informatika, dan Teknologi*, 95-99.
- Lilis Setyowati, V. S. (2022). Pemanfaatan Face Recognition pada Sistem Pencatatan Kehadiran. *UG JURNAL*.
- Muhammad Basurah, W. S. (2023). Implementation of Face Recognition and Liveness Detection System Using TensorFlow.js. *JIP (Jurnal Informatika Polinema)*, 510-516.
- Noertjahyana, A. (2002). Studi Analisis Rapid Application Development sebagai Salah Satu Alternatif Metode Pengembangan Perangkat Lunak. *JURNAL INFORMATIKA*, 74-79.

- Nurman Hidayat, K. H. (2021). Penerapan Metode Rapid Application Development (RAD) dalam Rancang Bangun Sistem Informasi Rapor Online (SIRALINE). *JURNAL SISTEM INFORMASI STMIK ANTAR BANGSA* , 8-17.
- Rima Yuniarti, ,. I. (2022). Perancangan Aplikasi Point of Sale untuk Manajemen Pemesanan Bahan Pangan Berbasis Framework Laravel. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 67-74.
- Roni Andarsyah, C. Y. (2022). Implementasi Code Coverage pada Chatbot Telegram sebagai Media Alternatif Sistem Informasi. *Jurnal Teknik Informatik*, 14, 112-117.
- Tajuddin Abdillah, S. S. (2024). Aplikasi Absensi Siswa Menggunakan Face API Recognition. *Digital Transformation Technology (Digitech)*, 497-503.
- Woro Isti Rahayu, J. M. (2023). Implementasi Framework Laravel pada Perancangan Aplikasi Sistem Pendaftaran Programming Course Roblox. *Jurnal Teknik Informatika*, 15, 18-25.



Lampiran

Lampiran 1: Lembar Validasi Ahli Penguji 1

Nama Penguji: *Berkles Almandin Wisasa, S.Nom., M.Nom.*
 Pendidikan Terakhir: *S2*
 Tanggal Pengujian: *03 Juli 2025*

Pengujian Halaman Login Admin

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Login dengan email dan password yang benar.	Admin berhasil masuk dan diarahkan ke halaman dasbor utama.		<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2.	Login dengan email yang salah/idak terdaftar.	Sistem menampilkan pesan error bahwa email tidak terdaftar.		<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4.	Login dengan password yang salah.	Sistem menampilkan pesan error bahwa password salah.		<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Pengujian Halaman Dashboard Admin

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Berhasil login dan masuk ke dasbor.	Halaman dasbor tampil dengan semua komponen, termasuk kartu statistik, grafik, dan menu navigasi.		<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2.	Memeriksa kartu statistik ringkas.	Kartu menampilkan data jumlah mahasiswa, tamu, dan total kunjungan dengan benar.		<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

6	Menekan tombol "Hapus" pada salah satu data.	Setelah konfirmasi, data mahasiswa terhapus dari daftar.	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
---	--	--	--

Sungaliat, 03 Juli 2025
Berkles Almandin Wisasa, S.Nom., M.Nom.

3	Memeriksa grafik kunjungan.	Grafik menampilkan data tren kunjungan dengan akurat.	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4	Menggunakan filter tanggal.	Data pada kartu statistik dan grafik berubah sesuai dengan rentang tanggal yang dipilih.	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Pengujian Halaman Kelola Data Mahasiswa (CRUD)

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Mengakses menu "Mahasiswa".	Halaman menampilkan daftar mahasiswa dalam format tabel beserta tombol aksi (Lihat, Edit, Hapus).		<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2.	Menekan tombol "Tambah Mahasiswa Baru".	Halaman formulir tambah data mahasiswa baru muncul.		<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3.	Mengisi dan menyimpan data mahasiswa baru yang valid.	Data mahasiswa baru berhasil tersimpan dan muncul di tabel daftar mahasiswa.		<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4.	Menekan tombol "Edit" pada salah satu data.	Halaman edit muncul dengan formulir yang sudah terisi data mahasiswa tersebut.		<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
5.	Mengubah dan menyimpan data pada halaman edit.	Data mahasiswa pada tabel berhasil diperbarui.		<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Lampiran 2: Lembar Validasi Ahli Penguji 2

Nama Penguji : Ongki Ade Saputra
 Pendidikan Terakhir : S1 Teknik Informatika
 Tanggal Pengujian : 23 Juli 2025

Pengujian Halaman Login Admin

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Login dengan email dan password yang benar.	Admin berhasil masuk dan diarahkan ke halaman dasbor utama.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	Login dengan email yang salah/tidak terdaftar.	Sistem menampilkan pesan error bahwa email tidak terdaftar.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	Login dengan password yang salah.	Sistem menampilkan pesan error bahwa password salah.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Pengujian Halaman Dashboard

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Berhasil login dan masuk ke dasbor.	Halaman dasbor tampil dengan semua komponen, termasuk kartu statistik, grafik, dan menu navigasi.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	Memeriksa kartu statistik ringkas.	Kartu menampilkan data jumlah mahasiswa, tamu, dan total kunjungan dengan benar.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

3	Memeriksa grafik kunjungan.	Grafik menampilkan data tren kunjungan dengan akurat.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4	Menggunakan filter tanggal.	Data pada kartu statistik dan grafik berubah sesuai dengan rentang tanggal yang dipilih.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Pengujian Halaman Kelola Data Mahasiswa

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Mengakses menu "Mahasiswa".	Halaman menampilkan daftar mahasiswa dalam format tabel beserta tombol aksi (Lihat, Edit, Hapus).	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	Menekan tombol "Tambah Mahasiswa Baru".	Halaman formulir tambah data mahasiswa baru muncul.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	Mengisi dan menyimpan data mahasiswa baru yang valid.	Data mahasiswa baru berhasil tersimpan dan muncul di tabel daftar mahasiswa.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4	Menekan tombol "Edit" pada salah satu data.	Halaman edit muncul dengan formulir yang sudah terisi data mahasiswa tersebut.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
5	Mengubah dan menyimpan data pada halaman edit.	Data mahasiswa pada tabel berhasil diperbarui.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
6	Menekan tombol "Hapus" pada salah satu data.	Setelah konfirmasi, data mahasiswa terhapus dari daftar.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

7	Mencari data menggunakan fitur pencarian dan filter	Menampilkan data sesuai yang di cari	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
---	---	--------------------------------------	-----------	--

Pengujian Halaman Kelola Data Tamu

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Mengakses menu "Tamu".	Halaman menampilkan daftar Tamu dalam format tabel beserta tombol aksi (Lihat, Edit, Hapus).	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	Menekan tombol "Edit" pada salah satu data.	Halaman edit muncul dengan formulir yang sudah terisi data mahasiswa tersebut.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	Mengubah dan menyimpan data pada halaman edit.	Data mahasiswa pada tabel berhasil diperbarui.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4	Menekan tombol "Hapus" pada salah satu data.	Setelah konfirmasi, data mahasiswa terhapus dari daftar.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
5	Mencari data menggunakan fitur pencarian dan filter	Menampilkan data sesuai yang di cari	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Pengujian Halaman Kunjungan

No	Aktivitas Penguji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Mengakses menu "Kunjungan"	Halaman menampilkan daftar Kunjungan Tamu dan Mahasiswa	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

2	Menekan tombol "Lihat Detail" pada salah satu data.	Menampilkan Halaman Detail Kunjungan.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	Menekan tombol "Cetak".	Menghasilkan data yang bisa di print dalam bentuk PDF.	BERFUNGSI	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Air Duren, 23 Juli 2025



(Ongki Ade Saputra)

Lampiran 3 Tabel Hasil Kuesioner

No	Nama	Pertanyaan										Skor
		1	2	3	4	5	6	7	8	9	10	
1	Aulia Sabila	4	5	4	5	4	5	4	5	4	5	45
2	Nafisa Silmi Zirroni	5	5	5	5	5	5	5	5	5	5	50
3	Atha Said Fajri	4	5	5	4	5	3	5	5	5	5	46
4	Alvin Yeheskiel Roison Harahap	5	5	5	4	5	5	4	4	4	4	45
5	Muhammad Raihan Pasha	5	5	4	5	4	4	5	5	5	5	47
6	Septiandi Hermawan	5	5	4	5	4	3	5	4	4	5	44
7	Deswita Syahran	4	4	4	4	4	5	4	4	4	5	42
8	Dicky Rahmat Rifai	4	4	4	4	4	4	4	4	4	4	40
9	Ana Maharani	4	3	4	4	4	3	4	4	4	4	38
10	Jimmi Rianto	5	5	4	5	4	4	4	5	4	5	45
11	Rizqi Pratama	5	5	4	5	4	5	4	5	5	5	47
12	Taqry Jayya Muhammad	4	4	4	4	5	3	4	4	5	4	41
13	Zadin Fatan Zirroni	5	5	4	4	5	3	4	4	5	5	44
TOTAL SKOR											574	

Lampiran 4 Surat Pernyataan

SURAT PERNYATAAN

Saya/Kami yang bertandatangan dibawah ini telah menyelesaikan Proyek Akhir yang berjudul:

Sistem Pendataan Pengunjung Perpustakaan dengan Scan Wajah

Oleh :

1. Muhammad Ferdi Firmansyah / NPM 1062248

Dengan ini menyatakan bahwa isi laporan akhir proyek akhir sama dengan *hardcopy*.
Demikian surat pernyataan ini dibuat dengan sebenar-benarnya.

Sungailiat,September 2025

1. Muhammad Ferdi Firmansyah ()

Mengetahui,

Pembimbing 1.



Ahmat Josi, S.Kom., M.Kom.
NIP. 198908202019031015

Pembimbing 2



Better Swengky, M.Kom.
NIP. 199301222024061001

Lampiran 4: Daftar Riwayat Hidup

DAFTAR RIWAYAT

4.4.1 Data Pribadi

Nama Lengkap : Muhammad Ferdi Firmansyah
Tempat & Tanggal Lahir : Lubuk Linggau, 22 Februari 2004
Alamat Rumah : Jl. Selendang, RW.06, Desa Sempan
No. Handphone : +62 878 9389 0356
Email : budiwardo2@gmail.com
Jenis Kelamin : Laki-laki
Agama : Islam



4.4.2 Riwayat Pendidikan

TK Amanah Bunda	Lulus tahun 2010
SD Negeri 4 Pemali	Lulus tahun 2016
SMP Negeri 3 Pemali	Lulus tahun 2019
SMA Negeri 1 Pemali	Lulus tahun 2022

4.4.3 Pendidikan Non Formal

—

Sungailiat, 18 Juli 2025

Muhammad Ferdi Firmansyah