

**LAPORAN PROYEK**  
**APLIKASI PENDETEKSI PENYAKIT PADA TANAMAN CABAI**



Oleh:

Vicha Farazahla	2100018400,
Reza Sefiyanti	2100018429,
Salih Arya Gumilang	2100018438,
Dea Amelia Setyorini	2100018445,
Yudit Aditya Nugraha	2100018464,

**PROGRAM STUDI S1 INFORMATIKA**  
**FAKULTAS TEKNOLOGI INDUSTRI**  
**UNIVERSITAS AHMAD DAHLAN**

2024

## HALAMAN PENGESAHAN

### APLIKASI PENDETEKSI PENYAKIT PADA TANAMAN CABAI



Yogyakarta, 3 Agustus 2024

Kaprodi S1 Informatika

Dr. Murinto, S.Si., M. Kom

NIPM: 19730710 200409 111 0951298

## Kata Pengantar

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, atas rahmat dan karuniaNya kami dapat menyelesaikan Laporan Manajemen Proyek Teknologi Informasi dengan judul "Aplikasi Pendekripsi Penyakit pada Tanaman Cabai" di Klaten, Jawa Tengah. Laporan ini merupakan salah satu persyaratan untuk pengajuan seminar dalam rangka memenuhi tugas akhir di Program Studi S1 Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan.

Proyek ini bertujuan untuk mengembangkan aplikasi *Android* yang mampu mendekripsi penyakit pada tanaman cabai menggunakan teknologi pengenalan citra, khususnya *TensorFlow Lite*. Aplikasi ini diharapkan dapat membantu petani dalam mengidentifikasi penyakit pada tanaman cabai secara cepat dan akurat, sehingga dapat mengurangi kerugian dan meningkatkan produktivitas.

Kami mengucapkan terima kasih yang sebesar-besarnya kepada Ibu Endri Biwinarni, petani cabai di Gantiwarno, Klaten, yang telah bersedia menjadi mitra dan memberikan dukungan serta data yang diperlukan selama proses proyek ini. Ucapan terima kasih juga kami sampaikan kepada dosen pembimbing, Bapak Taufiq Ismail, S.T., M.CS. yang telah memberikan bimbingan dan arahan dalam penyusunan laporan ini.

Kami menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun untuk perbaikan di masa yang akan datang.

Yogyakarta, 3 Agustus 2024

Hormat kami,



Penulis

## Daftar Isi

LAPORAN PROYEK .....	i
HALAMAN PENGESAHAN .....	ii
Kata Pengantar .....	iii
Daftar Isi .....	iv
Daftar Gambar .....	v
Daftar Tabel .....	vi
Daftar Lampiran .....	vii
BAB I Pendahuluan .....	1
A. Latar Belakang .....	1
1. Proses Mendapatkan Proyek .....	2
B. Project Charter .....	3
1. Tujuan .....	3
2. Ruang Lingkup .....	3
3. Stakeholder .....	4
BAB II Perencanaan Proyek .....	6
A. Analisis Kelayakan .....	6
1. <i>Strength</i> (Kekuatan) .....	6
2. <i>Weakness</i> (Kelemahan) .....	6
3. <i>Opportunities</i> (Peluang) .....	6
4. <i>Threat</i> (Ancaman) .....	6
B. Work Breakdown Structure .....	7
C. Kebutuhan Sumber Daya .....	7
1. Sumber Daya Manusia .....	7
2. Sumber Daya Fisik .....	10
D. Rencana Jadwal Pelaksanaan Proyek .....	11
E. Rencana Nilai Proyek .....	12
BAB III Pelaksanaan Proyek .....	15
A. Realisasi Jadwal Pelaksanaan .....	15
B. Realisasi Hasil Pekerjaan .....	16
1. Implementasi <i>Design UI/UX</i> .....	18
C. Penjaminan Kualitas Proyek .....	33
1. Pengujian SUS .....	33
2. Pengujian <i>BlackBox</i> .....	35

D. Keberlanjutan Proyek.....	40
BAB IV Penutup .....	41
A. Kesimpulan.....	41
B. Saran .....	41
DAFTAR PUSTAKA .....	42
LAMPIRAN.....	43

## Daftar Gambar

Gambar 3. 1 Palet Warna .....	19
Gambar 3. 2 Font Shrikhard .....	21
Gambar 3. 3 Font Kadwa .....	21
Gambar 3. 4 Font Inter .....	21
Gambar 3. 5 Logo Aplikasi .....	22
Gambar 3. 6 Greetings Wireframe .....	23
Gambar 3. 7 Login Wireframe .....	23
Gambar 3. 8 Register Wireframe .....	24
Gambar 3. 9 Dashboard Wireframe .....	24
Gambar 3. 10 Kamera Wireframe .....	25
Gambar 3. 11 Upload Gambar dari Galeri Wireframe .....	25
Gambar 3. 12 Analisis Penyakit Wireframe .....	26
Gambar 3. 13 Rekomendasi Penanganan Wireframe .....	26
Gambar 3. 14 Riwayat Pemeriksaan Wireframe .....	27
Gambar 3. 15 Profil Wireframe .....	27
Gambar 3. 16 UI/UX Greetings .....	28
Gambar 3. 17 UI/UX Login .....	28
Gambar 3. 18 UI/UX Register .....	29
Gambar 3. 19 UI/UX Dashboard .....	29
Gambar 3. 20 UI/UX Kamera .....	30
Gambar 3. 21 UI/UX Upload Gambar .....	30
Gambar 3. 22 UI/UX Analisis Penyakit .....	31
Gambar 3. 23 UI/UX Rekomendasi Penanganan .....	31
Gambar 3. 24 UI/UX Riwayat Pemeriksaan .....	32
Gambar 3. 25 UI/UX Profil .....	32

## **Daftar Tabel**

Tabel 2. 1 Tugas Tim Manajemen Proyek.....	8
Tabel 2. 2 Gantt Chart Jadwal Pelaksanaan Proyek.....	11
Tabel 2. 3 Rencana Biaya .....	12
Tabel 3. 1 Realisasi Jadwal Pelaksanaan.....	15
Tabel 3. 2 Realisasi Hasil Pekerjaan .....	16
Tabel 3. 3 Deskripsi warna design .....	19
Tabel 3. 4 Hasil Inputan Responden .....	33
Tabel 3. 5 Hasil Perhitungan System Usability Scale (SUS).....	34
Tabel 3. 6 Hasil Pengujian BlackBox .....	35

## **Daftar Lampiran**

<b>Lampiran 1 Proposal Proyek .....</b>	<b>43</b>
<b>Lampiran 2 Surat Perintah Kerja / Kontrak Kerja dengan Mitra / Client .....</b>	<b>55</b>
<b>Lampiran 3 Log Book Kelompok sudah terisi minimal 7 .....</b>	<b>60</b>
<b>Lampiran 4 Log Book Individu sudah terisi minimal 7x.....</b>	<b>62</b>
<b>Lampiran 5 Foto Dokumentasi Kegiatan Proyek .....</b>	<b>67</b>
<b>Lampiran 6 Bukti Serah Terima Proyek .....</b>	<b>76</b>
<b>Lampiran 7 Bukti Pembayaran Proyek.....</b>	<b>77</b>
<b>Lampiran 8 Tools: source code &amp; user manual.....</b>	<b>77</b>
<b>Lampiran 9 Link Video Profil Produk Luaran Proyek .....</b>	<b>149</b>
<b>Lampiran 10 Poster Produk Luaran Proyek.....</b>	<b>149</b>
<b>Lampiran 11 Slide presentasi proyek.....</b>	<b>150</b>

## BAB I Pendahuluan

### A. Latar Belakang

Budidaya cabai merupakan sektor penting dalam perekonomian negara. Menurut data dari Direktorat Jenderal Hortikultura Kementerian Pertanian, produksi cabai di Indonesia terus meningkat [1]. Cabai memiliki peran signifikan dalam inflasi di Indonesia. Kenaikan harga cabai dapat berdampak pada inflasi, mengurangi daya beli masyarakat, dan menghambat pertumbuhan ekonomi nasional [2]. Pemerintah telah melakukan berbagai upaya untuk menstabilkan pasokan dan mengendalikan kenaikan harga cabai [1].

Namun, petani cabai masih menghadapi tantangan dalam mengatasi penyakit pada tanaman cabai. Salah satu masalah utama adalah kurangnya pengetahuan tentang jenis-jenis penyakit yang menyerang tanaman cabai dan cara pengendaliannya [3]. Penyakit pada tanaman cabai dapat disebabkan oleh serangan virus, bakteri, dan jamur. Faktor lingkungan seperti suhu, pH tanah, dan kelembaban juga mempengaruhi kesehatan tanaman cabai. Kurangnya pemahaman tentang faktor-faktor ini membuat petani kesulitan dalam menerapkan tindakan pencegahan yang efektif. Jika penyakit pada tanaman cabai tidak ditangani dengan baik, hal ini akan berdampak negatif pada produktivitas dan kualitas hasil panen. Penyebaran penyakit dapat menyebabkan kerugian besar bagi petani dan bahkan mengancam kelangsungan usaha pertanian cabai.

Permasalahan yang dipaparkan di atas dapat dideteksi dini menggunakan aplikasi *Android* yang dilengkapi dengan teknologi pengenalan citra, seperti menggunakan *TensorFlow Lite*. *Android* adalah sistem operasi populer untuk smartphone dan tablet, dengan lebih dari 2.5 miliar pengguna aktif per bulan. Dikenal karena fitur canggih dan tampilan menarik, *Android* menjadi pilihan utama bagi berbagai produsen perangkat. Keunggulannya terletak pada sifat *open-source* yang mendukung pengembangan produk, serta *Google Play*

*Store* sebagai *platform* distribusi aplikasi dan konten digital seperti musik, buku, dan film. *Android Software Development Kit (SDK)* menyediakan tools penting untuk pengembangan aplikasi, menggunakan Java sebagai bahasa utama dan Kotlin sebagai bahasa tambahan resmi [4]. *TensorFlow Lite* adalah library untuk implementasi model *machine learning* pada perangkat seluler, dengan fitur konversi model *TensorFlow* menjadi *file flat buffer* yang lebih kecil ukurannya. *TensorFlow Lite* mendukung berbagai fitur seperti identifikasi objek, deteksi objek, dan penggunaan model bahasa alami [5].

Berdasarkan pemaparan diatas, proyek ini bertujuan untuk membangun aplikasi *Android* yang mampu mengidentifikasi penyakit pada tanaman cabai berdasarkan citra daun ataupun buah. Proyek ini diharapkan dapat menghasilkan solusi yang efektif dalam mendeteksi serta mengklasifikasi penyakit pada tanaman cabai sehingga dapat meminimalisir kerusakan tanaman yang cepat menyebar ke tanaman lain. Hama yang juga merupakan ancaman serius bagi tanaman cabai tidak akan dibahas dalam lingkup proyek ini.

## 1. Proses Mendapatkan Proyek

Sebelum proyek pembuatan Aplikasi Pendekripsi Penyakit pada Tanaman Cabai ini didapatkan, Tim FundEase telah merencanakan pembuatan aplikasi penggalangan dana, namun karena tidak adanya kecocokan antara tim dan klien akhirnya pembuatan proyek tersebut pun dibatalkan. Setelah pembatalan tersebut, salah satu anggota tim mengusulkan untuk membuat aplikasi pendekripsi penyakit pada tanaman cabai, tim pun melakukan diskusi mengenai usulan tersebut. setelah selesai melakukan diskusi, tim pun setuju untuk membuat Aplikasi Pendekripsi Penyakit pada Tanaman Cabai. Langkah pertama yang kami lakukan adalah membagi *job desk* atau tugas dalam pembuatan proyek ini, serta tahapan-tahapan apa saja yang akan dilakukan dari awal hingga akhir proses pembuatan proyek.

Selanjutnya kami juga mulai menghubungi calon Klien / Mitra yang telah diusulkan oleh salah satu anggota tim sebelumnya untuk mendiskusikan terkait proyek yang akan dikerjakan. Dimana Klien ini adalah petani cabai yang ada di daerah Klaten, Jawa Tengah. Setelah diskusi lebih lanjut, Klien pun sepakat untuk menerima tawaran kerja sama pembuatan Aplikasi Pendekripsi Penyakit pada Tanaman Cabai.

## B. Project Charter

### 1. Tujuan

- a) Membantu petani untuk mendekripsi penyakit pada tanaman cabai secara dini dan akurat.
- b) Mengurangi kerugian akibat penyakit pada tanaman cabai. Aplikasi ini dapat meminimalkan kerugian akibat penyakit.

### 2. Ruang Lingkup

#### a) Analisis kebutuhan

- (1) Analisis Kebutuhan Pengguna
  - (a) Memungkinkan pengguna menggunakan kamera untuk mendekripsi penyakit secara *real-time*.
  - (b) Memungkinkan pengguna mengunggah gambar melalui galeri.
  - (c) Menyimpan riwayat penyakit cabai yang pernah terdeteksi pada tanaman cabai sebelumnya.
  - (d) Menampilkan rekomendasi penanganan.
  - (e) Aplikasi harus mudah digunakan.
  - (f) Aplikasi harus akurat dan terpercaya.
  - (g) Aplikasi dapat diakses dengan mudah.

#### (2) Analisis Kebutuhan Sistem

- (a) Gambar daun cabai yang terkena penyakit.
- (b) Informasi tentang penyakit cabai seperti gejala yang terjadi pada cabai.

(c) Data pengguna aplikasi, seperti nama, alamat, dan nomor telepon.

(3) Analisis Kebutuhan Data

(a) Data yang dibutuhkan

(i) Data gambar tanaman cabai sehat dan berpenyakit didapatkan dari website *Roboflow Universe*

(ii) Data teks rekomendasi penanganan

(iii) Data pengguna dikumpulkan saat pengguna mendaftar dan menggunakan aplikasi

(iv) Data disimpan dalam *cloud firebase database*

3. Stakeholder

a) Ibu Endri Biniwarni - Pemilik dan Pengelola Utama

Tanggung Jawab:

(1) Mengelola keseluruhan usaha pertanian.

(2) Membuat keputusan strategis terkait budidaya dan pemasaran.

(3) Mengawasi seluruh kegiatan pertanian.

(4) Melakukan pencatatan dan pelaporan hasil panen dan keuangan.

b) Visi dan Misi

Visi: Menjadi petani cabai yang sukses dan berkelanjutan, menghasilkan cabai berkualitas tinggi untuk memenuhi kebutuhan pasar lokal.

Misi:

(1) Menanam cabai yang sehat dan berkualitas tinggi menggunakan teknik budidaya terbaik.

(2) Menerapkan praktik pertanian yang ramah lingkungan untuk menjaga kesuburan tanah dan kelestarian alam.

(3) Meningkatkan hasil panen melalui penggunaan teknologi pertanian yang tepat dan efisien.

- (4) Memastikan kepuasan pelanggan dengan menyediakan cabai yang segar dan berkualitas secara konsisten.
- (5) Berkolaborasi dengan petani lain dan mengikuti pelatihan untuk terus belajar dan meningkatkan keterampilan bertani.

## BAB II Perencanaan Proyek

### A. Analisis Kelayakan

#### 1. *Strength* (Kekuatan)

- a) Banyaknya data pertanian yang tersedia, mulai dari data pertanian tradisional hingga data *modern*.
- b) Cabai merupakan salah satu komoditas tanaman pangan utama di Indonesia, sehingga ada dorongan untuk menjaga dan melindungi tanaman dari penyakit.
- c) Kesadaran petani dalam mengurangi kerugian dan meningkatkan produktivitas.

#### 2. *Weakness* (Kelemahan)

- a) Data yang tersedia bisa saja tidak lengkap dan tidak terbaru sehingga menghambat pengembangan dan akurasi model deteksi penyakit.
- b) Banyaknya petani di Indonesia yang mungkin tidak memiliki akses penuh terhadap teknologi.
- c) Penyakit pada tanaman cabai bisa sangat beragam yang dapat menambah kompleksitas dalam mendiagnosis penyakit secara akurat dan tepat waktu.

#### 3. *Opportunities* (Peluang)

- a) Mengembangkan alat deteksi yang lebih canggih dan akurat.
- b) Kesempatan bekerja sama dengan ahli pertanian untuk mengembangkan solusi yang tepat.
- c) Potensi pasar yang besar.

#### 4. *Threat* (Ancaman)

- a) Meningkatnya persaingan teknologi baru yang terus berkembang.
- b) Keterbatasan infrastruktur pada beberapa daerah.
- c) Perubahan iklim yang dapat mempengaruhi penyebaran penyakit pada tanaman.

B. Work Breakdown Structure

1. Analisis dan Perencanaan

1.1. Identifikasi kebutuhan pengguna

1.2. Identifikasi kebutuhan Sistem

1.3. Identifikasi kebutuhan Data

2. Desain

2.1. Desain UX/UI

2.1.1. Kerangka Kerja (*Wireframes*)

2.2. Desain Basis Data

2.2.1. Diagram Aliran Data

2.3. Desain Arsitektur Sistem

2.3.1. Tentukan Komponen Sistem

2.3.2. Pilih Teknologi

2.4. Pembuatan UI/UX

3. Membangun Konstruksi Aplikasi

3.1. Implementasi Desain UI/UX

3.2. Pengembangan Model AI

3.3. Implementasi AI

3.4. Pembangunan Database Firebase

4. Pengujian

4.1. SUS

4.2. *BlackBox*

5. Pengembangan Sistem

5.1. Evaluasi

5.2. Pemeliharaan sistem

C. Kebutuhan Sumber Daya

1. Sumber Daya Manusia

Tim Manajemen Proyek “Aplikasi pendekripsi penyakit pada Tanaman Cabai” beranggotakan 5 orang yang memiliki tugas masing-masing seperti pada tabel 2.1

Tabel 2. 1 Tugas Tim Manajemen Proyek

Nama	NIM	Jabatan
Reza Sefiyanti	2100018429	<i>Project Manager, Sekretaris</i>
Vicha Farazahla	2100018400	<i>Data Analyst, Bendahara</i>
Dea Amelia Setyorini	2100018445	<i>UI/UX Designer</i>
Salih Arya Gumlilang	2100018438	<i>Mobile Developer</i>
Yudit Aditya Nugraha	2100018464	<i>Quality Assurance</i>

Berikut deskripsi pembagian tugas dari anggota Tim FundEase:

- a) Jabatan : *Project Manager, Sekretaris*

Nama : Reza Sefiyanti

NIM : 2100018429

Deskripsi Tugas:

- (1) Perencanaan keseluruhan proyek.
- (2) Melakukan pengawasan dan koordinasi tim.
- (3) Memastikan bahwa proyek berjalan sesuai rencana .
- (4) Membuat laporan penggerjaan proyek.
- (5) Membuat Proposal.
- (6) Membuat *Logbook* kelompok.
- (7) Membuat MOU.

- b) Jabatan : *Data Analyst, Bendahara*

Nama : Vicha Farazahla

NIM : 2100018400

Deskripsi Tugas:

- (1) Menganalisis data terkait dengan pendekripsi penyakit pada tanaman cabai.
- (2) Mengumpulkan dan menyajikan data relevan untuk membantu dalam pengambilan keputusan proyek.
- (3) Menyusun anggaran terhadap penggunaan dana.

(4) Berkoordinasi dengan tim terkait untuk memastikan terhadap anggaran yang telah disetujui.

c) Jabatan : *UI/UX Designer*

Nama : Dea Amelia Setyorini

NIM : 2100018445

Deskripsi Tugas:

- (1) Membuat *wireframe*.
- (2) Merancang antarmuka pengguna (UI) dan pengalaman pengguna (UX).
- (3) Memastikan aplikasi memiliki tampilan dan interaksi yang menarik serta mudah digunakan.
- (4) Membuat poster aplikasi.

d) Jabatan : *Mobile Developer*

Nama : Salih Arya Gumilang

NIM : 2100018438

Deskripsi Tugas:

- (1) Menerjemahkan konsep dan desain ke dalam kode computer.
- (2) Membangun model AI.
- (3) Mengintegrasikan model AI ke aplikasi *Android*.

e) Jabatan : *Quality Assurance*

Nama : Yudit Aditya Nugraha

NIM : 2100018464

Deskripsi Tugas:

- (1) Menguji aplikasi cek cabai guna memastikan bahwa tidak ada *bug* atau masalah yang signifikan.
- (2) Melakukan pengujian fungsionalitas.
- (3) Mengidentifikasi masalah yang perlu diperbaiki sebelum peluncuran.

## 2. Sumber Daya Fisik

Berikut adalah alat dan bahan yang digunakan dalam pembuatan aplikasi pendeteksi penyakit pada tanaman cabai:

### a) Hardware

*Hardware* yang digunakan untuk pengembangan dan pengujian aplikasi adalah sebuah laptop dengan prosesor Intel i3 generasi ke-11. Laptop ini memiliki performa yang cukup untuk pengembangan aplikasi.

### b) Database

*Database* yang digunakan untuk menyimpan data aplikasi adalah Firebase. Firebase menyediakan solusi penyimpanan data yang aman dan scalable, serta mendukung integrasi yang mudah dengan aplikasi *mobile*.

### c) Aplikasi *Mobile*

*Platform* pengembangan aplikasi *mobile* yang digunakan adalah *Android Studio*. *Android Studio* adalah *Integrated Development Environment* (IDE) resmi yang didukung oleh Google untuk pengembangan aplikasi *Android*. Dengan menggunakan *Android Studio*, pengembang dapat membuat, menguji, dan memublikasikan aplikasi secara efisien.

### d) Kamera

Kamera *smartphone* digunakan untuk mengambil gambar tanaman cabai yang akan dianalisis oleh aplikasi. Gambar-gambar ini kemudian diproses untuk mendeteksi adanya penyakit pada tanaman.

D. Rencana Jadwal Pelaksanaan Proyek

Tabel 2. 2 Gantt Chart Jadwal Pelaksanaan Proyek

No	Kegiatan	Bulan Pelaksanaan				
		Maret	April	Mei	Juni	Juli
<b>Analisis dan Perencanaan</b>						
1.1	Identifikasi kebutuhan pengguna					
1.2	Identifikasi kebutuhan Sistem					
1.3	Identifikasi kebutuhan Data					
<b>Desain</b>						
2.1	Desain UI/UX					
2.1.1	Kerangka Kerja ( <i>wireframes</i> )					
2.2	Desain Basis Data					
2.2.1	Diagram Aliran Data					
2.3	Desain Arsitektur Sistem					
2.3.1	Tentukan komponen sistem					
2.3.2	Pilih teknologi					
2.4	Pembuatan UI/UX					
<b>Membangun Konstruksi Aplikasi</b>						
3.1	Implementasi Desain UI/UX					
3.2	Pengembangan Model AI					
3.3	Integrasi AI ke dalam Aplikasi					
3.4	Pembangunan <i>Database</i>					
<b>Pengujian</b>						
4.2	<i>Unit testing</i>					
4.2	<i>Integration Testing</i>					
4.3	<i>System Testing</i>					
<b>Pengembangan Sistem</b>						
5.1	Evaluasi					

5.2	Pemeliharaan sistem						
-----	---------------------	--	--	--	--	--	--

E. Rencana Nilai Proyek

Tabel 2. 3 Rencana Biaya

Kegiatan	Bulan					
	Maret	April	Mei	Juni	Juli	Total
<b>1. Analisis dan Perencanaan</b>						
1.1 Identifikasi Kebutuhan Pengguna	35.000					35.000
1.2 Identifikasi Kebutuhan Sistem	35.000					35.000
1.3 Identifikasi Kebutuhan Data	35.000					35.000
<b>2. Desain</b>						
2.1 Desain UI/UX		100.000				100.000
2.1.1 Kerangka Kerja		40.000				40.000
2.2 Desain Basis Data		30.000				30.000
2.2.1 Diagram Aliran Data		25.000				25.000
2.3 Desain Arsitektur Sistem			45.000			45.000

2.3.1 Tentukan Komponen Sistem			20.000			20.000
2.3.2 Pilih Teknologi			20.000			20.000
2.4 Pembuatan UI/UX				100.000		100.000
<b>3. Membangun Aplikasi <i>Android</i></b>						
3.1 Implementasi UI/UX				100.000		100.000
3.2 Implementasi AI				100.000		100.000
3.3 Pembangunan <i>Database</i>				70.000		70.000
<b>4. Pengujian</b>						
4.1 <i>Unit testing</i>					35.000	35.000
4.2 <i>Integration Testing</i>					35.000	35.000
4.3 <i>System Testing</i>					45.000	45.000
<b>5. Pengembangan Sistem</b>						
5.1 Evaluasi					40.000	40.000

5.2 Pemeliharaan Sistem					40.000	40.000
<b>Total</b>	105.000	195.000	85.000	300.000	195.000	850.000

### BAB III Pelaksanaan Proyek

#### A. Realisasi Jadwal Pelaksanaan

Tabel 3. 1 Realisasi Jadwal Pelaksanaan

No	Kegiatan	Bulan Pelaksanaan						Realisasi
		Maret	April	Mei	Juni	Juli		
<b>1. Analisis dan Perencanaan</b>								
1.1	Identifikasi kebutuhan pengguna							Ya
1.2	Identifikasi kebutuhan Sistem							Ya
1.3	Identifikasi kebutuhan Data							Ya
<b>2. Desain</b>								
2.1	Desain UI/UX							Ya
2.1.1	Kerangka Kerja (wireframes)							Ya
2.2	Desain Basis Data							Ya
2.2.1	Diagram Aliran Data							Ya
2.3	Desain Arsitektur Sistem							Ya
2.3.1	Tentukan komponen sistem							Ya

2.3.2	Pilih teknologi												Ya
2.4	Pembuatan UI/UX												Ya
<b>3. Membangun Konstruksi Aplikasi</b>													
3.1	Implementasi Desain UI/UX												Ya
3.2	Pengembangan Model AI												Ya
3.3	Integrasi AI ke dalam Aplikasi												Ya
3.4	Pembangunan Database												Ya
<b>4. Pengujian</b>													
4.2	<i>Unit testing</i>												Ya
4.2	<i>Integration Testing</i>												Ya
4.3	<i>System Testing</i>												Ya
<b>5. Pengembangan Sistem</b>													
5.1	Evaluasi												Ya
5.2	Pemeliharaan sistem												Ya

## B. Realisasi Hasil Pekerjaan

Tabel 3. 2 Realisasi Hasil Pekerjaan

Nama	Jabatan	Realisasi Hasil Pekerjaan
Reza Sefiyanti	<i>Project Manager, Sekretaris</i>	<ol style="list-style-type: none"> <li>1. Susunan Rencana Kerja.</li> <li>2. Melakukan jadwal pertemuan.</li> </ol>

		<ol style="list-style-type: none"> <li>3. Memonitoring progress dan melakukan evaluasi.</li> <li>4. Membuat Laporan Bulanan dan Laporan akhir.</li> <li>5. Menyusun dan merevisi Proposal.</li> <li>6. Penyusunan <i>Logbook</i> Kelompok.</li> <li>7. Menyusun MOU kesepakatan antara pihak-pihak terkait.</li> </ol>
Vicha Farazahla	Data Analyst, Bendahara	<ol style="list-style-type: none"> <li>1. Menganalisis data terkait dengan pendekripsi penyakit pada tanaman cabai.</li> <li>2. Mengumpulkan dan menyajikan data relevan untuk membantu dalam pengambilan keputusan proyek.</li> <li>3. Menyusun anggaran terhadap penggunaan dana.</li> <li>4. Berkoordinasi dengan tim terkait untuk memastikan terhadap anggaran yang telah disetujui.</li> </ol>
Dea Amelia Setyorini	UI/UX Designer	<ol style="list-style-type: none"> <li>1. Membuat <i>wireframe</i></li> <li>2. Merancang antarmuka pengguna (UI) dan</li> </ol>

		<p>pengalaman pengguna (UX).</p> <ol style="list-style-type: none"> <li>3. Memastikan aplikasi memiliki tampilan dan interaksi yang menarik serta mudah digunakan.</li> <li>4. Membuat poster aplikasi.</li> </ol>
Salih Arya Gumlilang	<i>Mobile Developer</i>	<ol style="list-style-type: none"> <li>1. Menerapkan desain UI/UX ke dalam <i>code kotlin Android</i>.</li> <li>2. Membangun model AI pendekripsi penyakit tanaman cabai berdasarkan citra daun dan buah.</li> <li>3. mengintegrasikan model AI ke dalam aplikasi <i>Android</i>.</li> </ol>
Yudit Aditya Nugraha	<i>Quality Assurance</i>	<ol style="list-style-type: none"> <li>1. Mencari responden untuk pengisian SUS.</li> <li>2. Membuat <i>BlackBox</i> test.</li> <li>3. Memastikan aplikasi berjalan dengan semestinya.</li> </ol>

1. Implementasi *Design UI/UX*

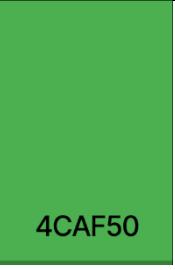
a) Pemilihan Warna



Gambar 3. 1 Palet Warna

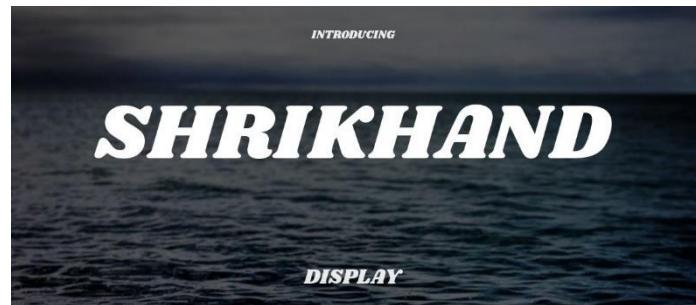
Tabel 3. 3 Deskripsi warna design

No	Kode	Deskripsi	Warna
1	F5F2F2 (Abu Muda)	Warna abu-abu muda yang lembut dan netral. Warna ini digunakan sebagai latar belakang untuk elemen-elemen yang tidak ingin mencolok, seperti latar belakang <i>scan</i> kamera dan formulir. Abu-abu muda memberikan kontras yang lembut tanpa mengganggu fokus pengguna.	F5F2F2
2	A5F4AD (Hijau Pucat)	Warna hijau pucat yang lembut dan menenangkan. Warna ini digunakan sebagai campuran gradasi halaman <i>greetings</i> . Hijau pucat memberikan kesan yang segar dan alami, sesuai dengan tema aplikasi yang berkaitan dengan tanaman.	A5F4AD

3	2DD01F (Hijau Terang)	Warna hijau terang yang cerah dan energik. Warna ini digunakan sebagai campuran gradasi halaman <i>greetings</i> .	 2DD01F
4	FF0000 (Merah)	Warna merah terang yang sangat mencolok. Warna ini digunakan untuk tombol keluar ( <i>logout</i> ), sehingga mudah dikenali dan mencolok. Warna merah menandakan tindakan penting dan memastikan pengguna memperhatikan tombol ini sebelum mereka benar-benar keluar dari aplikasi.	 FF0000
5	4CAF50 (Hijau Medium)	Warna hijau sedang yang menenangkan. Warna ini digunakan untuk elemen-elemen utama seperti <i>header</i> , <i>footer</i> , dan tombol. Hijau sedang memberikan tampilan yang menenangkan.	 4CAF50
6	368139 (Hijau Tua)	Warna hijau tua yang mendalam dan alami. Warna ini digunakan untuk elemen-elemen yang membutuhkan penekanan kuat namun tetap ingin mempertahankan nuansa alami, seperti tombol dan navigasi.	 368139

b) Pemilihan *Font*

Shrikhand



Gambar 3. 2 Font Shrikhard

Font Shrikhand digunakan untuk nama aplikasi. Font ini menarik perhatian, memberikan kesan berani dan ceria.

### Kadwa



Gambar 3. 3 Font Kadwa

Font Kadwa digunakan untuk *greetings*, *register*, *login*, dan profil. Font ini cocok untuk formulir pendaftaran dan *login*, serta profil pengguna.

### Inter



Gambar 3. 4 Font Inter

*Font* Inter digunakan untuk elemen yang lainnya. *Font* ini mudah dibaca, ideal untuk teks umum, formulir, tombol, dan konten utama.

c) Logo Aplikasi



Gambar 3. 5 Logo Aplikasi

(1) Deskripsi Visual

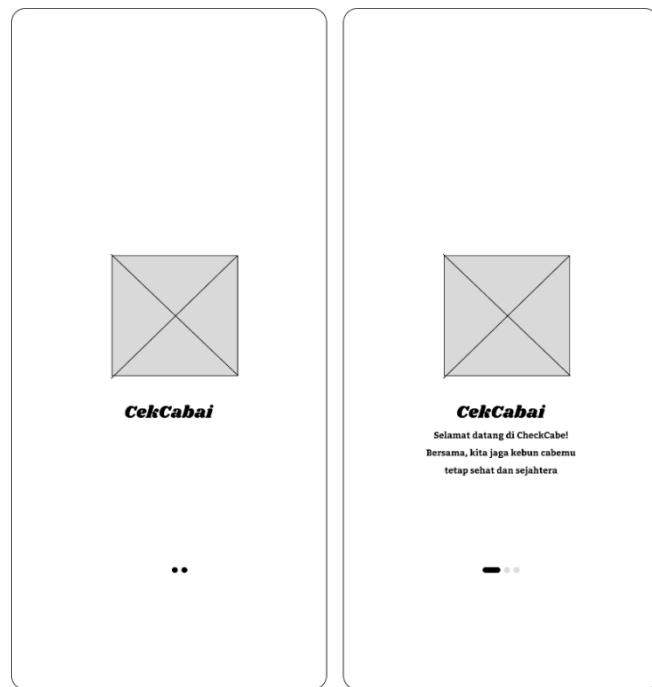
- (i) Logo aplikasi terdiri dari satu daun hijau yang tampak segar dan alami.
- (ii) Warna hijau yang cerah dan segar digunakan untuk daun.
- (iii) Daun tersebut memiliki bentuk yang sederhana dan elegan, dengan kurva yang lembut untuk menciptakan kesan segar dan sehat.

(2) Makna dan Filosofi Logo

- (i) Daun hijau melambangkan kesehatan, pertumbuhan, dan kehidupan. Ini sangat relevan dengan aplikasi yang berfokus pada deteksi penyakit tanaman, karena tujuan utamanya adalah menjaga tanaman cabai tetap sehat.
- (ii) Warna hijau dan bentuk daun yang segar memberikan kesan alami, menekankan bahwa aplikasi ini berkaitan dengan alam dan pertanian.
- (iii) Desain yang sederhana namun elegan mencerminkan kemudahan penggunaan dan kejelasan dalam aplikasi. Pengguna bisa merasa nyaman dan percaya bahwa aplikasi ini mudah digunakan dan efektif.

d) *Wireframe* Aplikasi

(1) *Greetings*



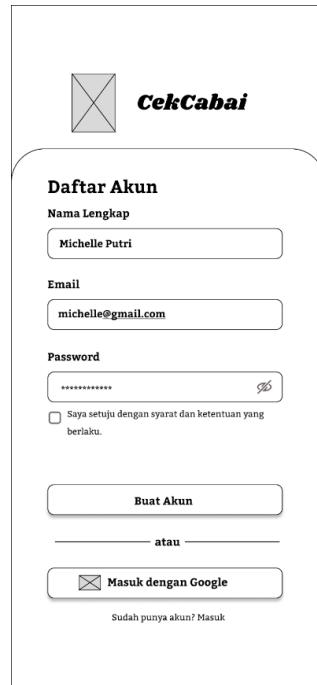
Gambar 3. 6 *Greetings Wireframe*

(2) *Login*

The wireframe shows a central account creation section above a login form. The account section has "CekCabai" text and a "Masuk dengan Google" button. The login form is titled "Akun" and contains fields for "Email/Username" (with placeholder "michelle@gmail.com") and "Password" (with placeholder "\*\*\*\*\*"). It features a "Masuk" button and a "Belum punya akun? Daftar" link.

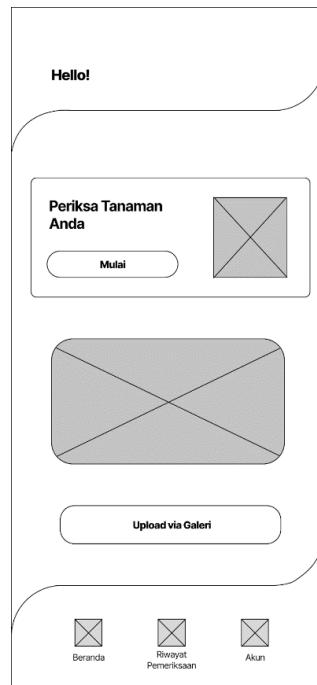
Gambar 3. 7 *Login Wireframe*

(3) Register



Gambar 3. 8 Register Wireframe

(4) Dashboard



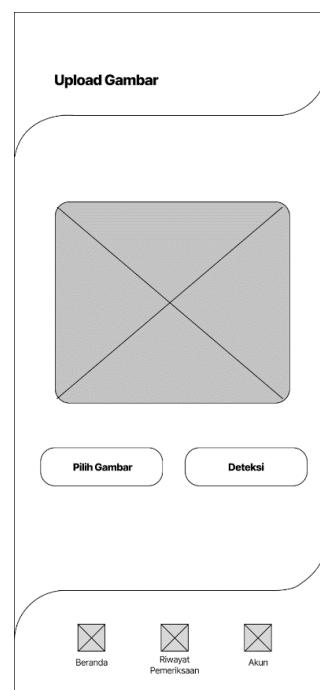
Gambar 3. 9 Dashboard Wireframe

(5) Kamera



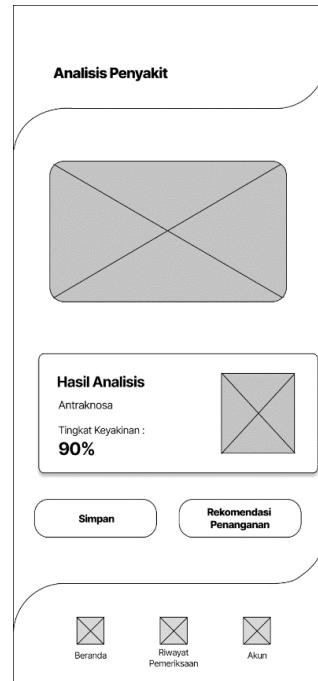
Gambar 3. 10 Kamera *Wireframe*

(6) *Upload* Gambar dari Galeri



Gambar 3. 11 *Upload* Gambar dari Galeri *Wireframe*

## (7) Analisis Penyakit



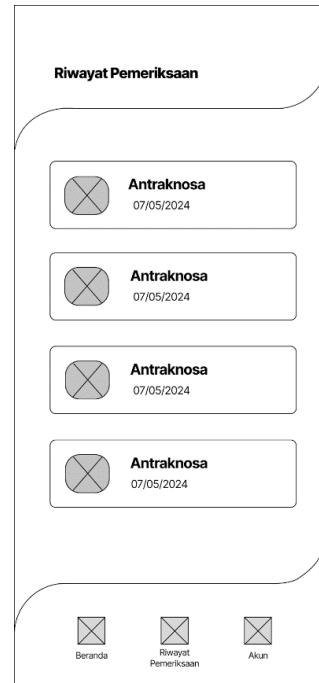
Gambar 3. 12 Analisis Penyakit *Wireframe*

## (8) Rekomendasi Penanganan



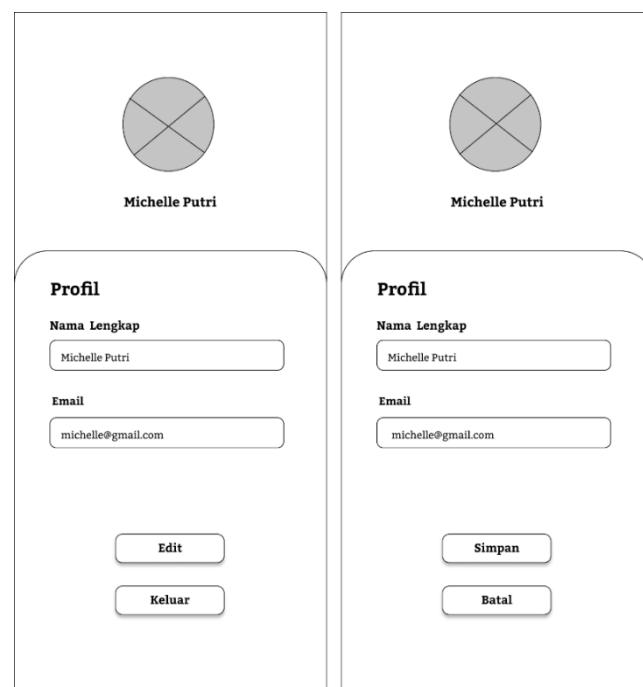
Gambar 3. 13 Rekomendasi Penanganan *Wireframe*

(9) Riwayat Pemeriksaan



Gambar 3. 14 Riwayat Pemeriksaan *Wireframe*

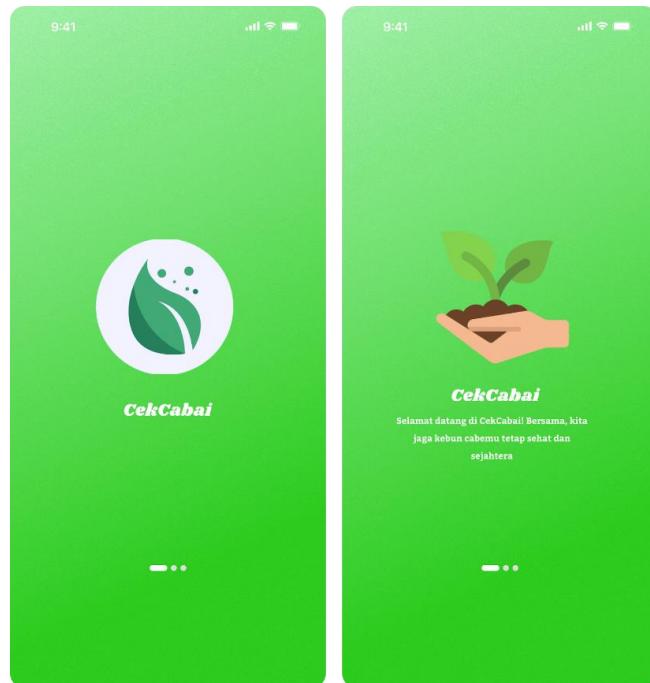
(10) Profil



Gambar 3. 15 Profil *Wireframe*

e) UI/UX

(1) *Greetings*



Gambar 3. 16 UI/UX *Greetings*

(2) *Login*



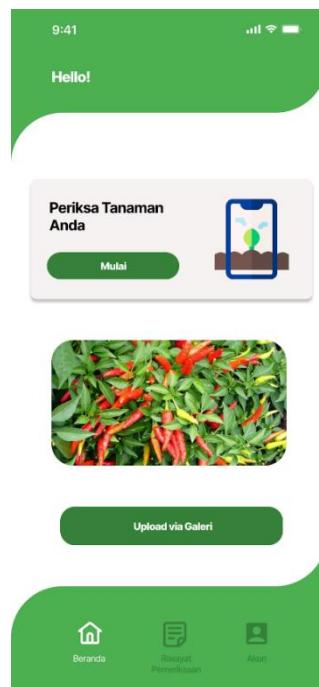
Gambar 3. 17 UI/UX *Login*

### (3) Register



Gambar 3. 18 UI/UX *Register*

### (4) Dashboard



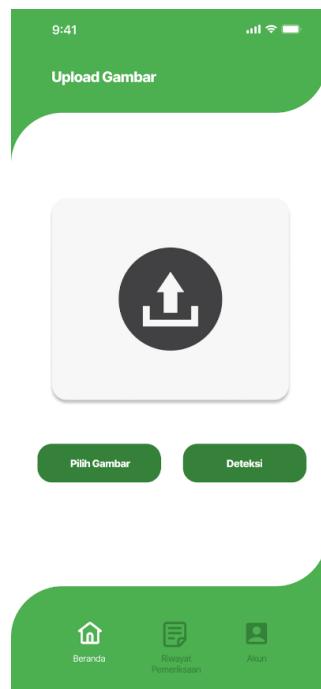
Gambar 3. 19 UI/UX *Dashboard*

(5) Kamera



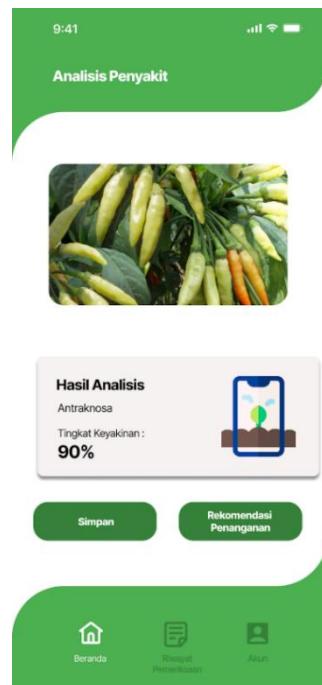
Gambar 3. 20 UI/UX Kamera

(6) Upload Gambar dari Galeri



Gambar 3. 21 UI/UX Upload Gambar

## (7) Analisis Penyakit



Gambar 3. 22 UI/UX Analisis Penyakit

## (8) Rekomendasi Penanganan



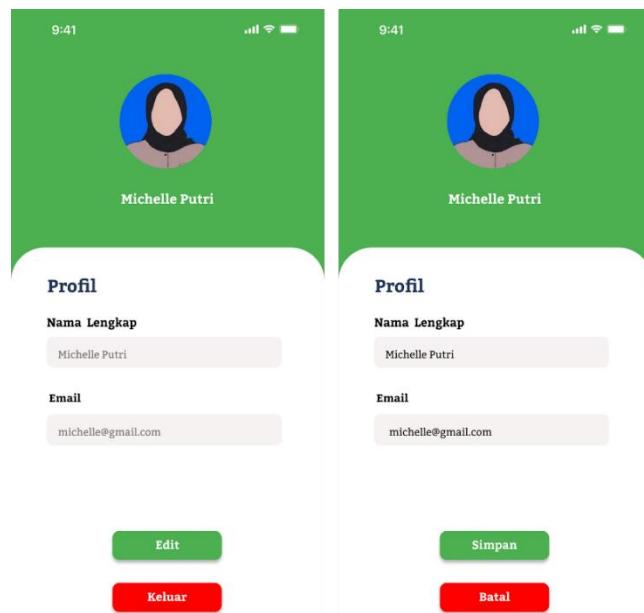
Gambar 3. 23 UI/UX Rekomendasi Penanganan

(9) Riwayat Pemeriksaan



Gambar 3. 24 UI/UX Riwayat Pemeriksaan

(10) Profil



Gambar 3. 25 UI/UX Profil

C. Penjaminan Kualitas Proyek

- a) Proses pengujian telah melibatkan *stakeholder* yang relevan, termasuk tim pengembang dan pengguna akhir. Setiap bug dan masalah signifikan telah diidentifikasi dan diperbaiki sebelum peluncuran.
- b) Proses analisis data telah melewati tahap verifikasi dan validasi untuk memastikan keakuratan dan keandalannya.
- c) Desain UI/UX telah melalui proses pengujian untuk memastikan kesesuaian dengan kebutuhan pengguna dan fungsionalitas aplikasi.
- d) Hasil pengujian SUS dan *BlackBox*.

1. Pengujian SUS

Tabel 3. 4 Hasil Inputan Responden

No	Responden	Skor asli									
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Resp 1	5	2	5	1	4	2	5	2	4	2
2	Resp 2	4	3	3	2	4	2	5	3	4	1
3	Resp 3	4	3	5	1	4	2	4	2	5	2
4	Resp 4	5	2	5	1	5	2	4	1	5	1
5	Resp 5	5	3	3	2	4	3	4	2	4	1
6	Resp 6	4	1	5	1	4	3	5	3	4	1
7	Resp 7	5	2	4	1	4	3	5	3	4	2
8	Resp 8	5	3	3	2	5	1	5	2	5	2
9	Resp 9	5	2	5	1	4	2	4	2	5	2
10	Resp 10	4	3	3	1	4	2	5	2	5	2
11	Resp 11	5	3	3	1	4	3	5	3	4	2
12	Resp 12	4	1	5	1	4	1	5	3	5	1
13	Resp 13	5	2	5	1	4	3	4	2	5	1
14	Resp 14	5	1	5	2	4	2	5	2	5	2
15	Resp 15	5	2	5	1	4	2	4	3	5	2
16	Resp 16	5	1	5	1	4	3	4	2	5	1

17	Resp 17	4	3	3	2	4	2	5	2	5	1
18	Resp 18	4	2	5	1	5	2	4	3	4	1
19	Resp 19	5	3	3	1	4	3	5	2	5	2
20	Resp 20	5	1	5	1	5	1	5	2	4	1
21	Resp 21	4	3	4	1	4	3	5	1	4	1
22	Resp 22	4	3	3	3	4	3	4	3	3	4
23	Resp 23	5	1	5	2	4	2	5	2	5	1
24	Resp 24	5	1	5	1	4	1	5	2	4	1

Tabel 3. 5 Hasil Perhitungan *System Usability Scale (SUS)*

Skor Hasil Hitung										Jumlah	Nilai (Jumlah x 2.5)
Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10		
4	3	4	4	3	3	4	3	3	3	34	85
3	2	2	3	3	3	4	2	3	4	29	73
3	2	4	4	3	3	3	3	4	3	32	80
4	3	4	4	4	3	3	4	4	4	37	93
4	2	2	3	3	2	3	3	3	4	29	73
3	4	4	4	3	2	4	2	3	4	33	83
4	3	3	4	3	2	4	2	3	3	31	78
4	2	2	3	4	4	4	3	4	3	33	83
4	3	4	4	3	3	3	3	4	3	34	85
3	2	2	4	3	3	4	3	4	3	31	78
4	2	2	4	3	2	4	2	3	3	29	73
3	4	4	4	3	4	4	2	4	4	36	90
4	3	4	4	3	2	3	3	4	4	34	85
4	4	4	3	3	3	4	3	4	3	35	88
4	3	4	4	3	3	3	2	4	3	33	83
4	4	4	4	3	2	3	3	4	4	35	88
3	2	2	3	3	3	4	3	4	4	31	78

3	3	4	4	4	3	3	2	3	4	33	83
4	2	2	4	3	2	4	3	4	3	31	78
4	4	4	4	4	4	4	3	3	4	38	95
3	2	3	4	3	2	4	4	3	4	32	82
3	2	2	2	3	2	3	2	2	1	22	82
4	4	4	3	3	3	4	3	4	4	36	83
4	4	4	4	3	4	4	3	3	4	37	83
<b>Skor Rata-rata</b>											<b>82.3</b>

Dari hasil SUS yang telah dilakukan, aplikasi CekCabai memiliki Rata-rata Skor SUS sebesar 82.3 yang berarti aplikasi sudah diterima oleh pengguna dengan tingkatan *acceptability ranges* yaitu *acceptable*, dengan tingkatan *grade scale* yaitu A, dan *Adjective rating* yaitu *excellent*.

## 2. Pengujian *BlackBox*

Tabel 3. 6 Hasil Pengujian *BlackBox*

No	Fitur yang Diuji	Skenario dan Hasil Uji			
		Test case	Hasil Diharapkan	Hasil Uji	Kesimpulan
1	<i>Login</i>	Klik <i>login</i> menggunakan Google	Aplikasi menampilkan <i>fragment Google login</i>	Aplikasi menampilkan <i>fragment Google login</i>	Berhasil
		Klik Daftar	Aplikasi mengarahkan ke halaman <i>register</i>	Aplikasi mengarahkan ke	Berhasil

				halaman <i>register</i>	
		<i>Login</i> menggunakan <i>email</i> dan <i>password</i>	Aplikasi dapat menerima input <i>email</i> dan <i>password</i> , dan <i>login</i> menggunakan <i>email</i> dan <i>password</i>	Aplikasi dapat menerima input <i>email</i> dan <i>password</i> , dan bisa <i>login</i> menggunakan <i>email</i> dan <i>password</i>	Berhasil
		Saat <i>login</i> berhasil	Aplikasi mengarahkan pengguna ke <i>dashboard</i> secara langsung	Aplikasi mengarahkan pengguna ke <i>dashboard</i> secara langsung	Berhasil
		Saat <i>login</i> Gagal	Aplikasi menampilkan <i>warning login</i> gagal sesuai dengan kasus	Aplikasi menampilkan <i>warning login</i> gagal sesuai dengan kasus	Berhasil
2	<i>Register</i>	Mengisi formulir pendaftaran	Aplikasi dapat menerima input <i>email</i> dan <i>password</i> , dan <i>login</i> menggunakan <i>email</i> dan <i>password</i>	Aplikasi dapat menerima input <i>email</i> dan <i>password</i> , tetapi tidak bisa <i>login</i> menggunakan <i>email</i> dan <i>password</i>	Berhasil

		Klik Masuk	Aplikasi mengarahkan ke halaman <i>login</i>	Aplikasi mengarahkan ke halaman <i>login</i>	Berhasil
3	<i>Dashboard</i>	Menekan tombol Mulai	Aplikasi mengarahkan ke kamera	Aplikasi mengarahkan ke kamera	Berhasil
		Menekan tombol <i>Upload via Galeri</i>	Aplikasi mengarahkan ke halaman <i>upload foto</i>	Aplikasi mengarahkan ke halaman <i>upload foto</i>	Berhasil
4	Kamera Scan Tanaman	Kamera mendeteksi penyakit tanaman cabai	Aplikasi dapat menampilkan penyakit saat <i>scan tanaman cabai</i>	Aplikasi dapat menampilkan penyakit saat <i>scan tanaman cabai</i>	Berhasil
		Menekan tombol Ambil Gambar	Aplikasi dapat mengarahkan ke halaman analisis penyakit	Aplikasi dapat mengarahkan ke halaman analisis penyakit	Berhasil
5	<i>Upload via Galeri</i>	Menekan tombol <i>Upload Foto</i>	Aplikasi dapat mengarah ke galeri untuk <i>upload foto tanaman cabai</i>	Aplikasi dapat mengarah ke galeri untuk <i>upload foto tanaman cabai</i>	Berhasil
		Foto tanaman cabai	Aplikasi dapat menampilkan foto yang sudah dipilih	Aplikasi dapat menampilkan foto yang sudah dipilih	Berhasil

			dipilih pengguna untuk dideteksi	pengguna untuk dideteksi	
		Menekan tombol Deteksi	Aplikasi dapat mendeteksi foto yang telah diunggah	Aplikasi tidak dapat mendeteksi foto yang telah diunggah	Tidak Berhasil
6	Analisis Penyakit	Analisis penyakit	Aplikasi dapat menampilkan hasil analisis penyakit	Aplikasi dapat menampilkan hasil analisis penyakit	Berhasil
		Menekan tombol Simpan	Aplikasi dapat menyimpan hasil analisis penyakit	Aplikasi dapat menyimpan hasil analisis penyakit	Berhasil
		Menekan tombol Rekomendasi Penanganan	Aplikasi dapat mengarah ke halaman rekomendasi penanganan	Aplikasi dapat mengarah ke halaman rekomendasi penanganan	Berhasil
7	Rekomendasi Penanganan	Rekomendasi penanganan penyakit	Aplikasi menampilkan rekomendasi penanganan sesuai diagnosis penyakit pada tanaman cabai	Aplikasi menampilkan rekomendasi penanganan sesuai diagnosis penyakit pada tanaman cabai	Berhasil

8	Riwayat Pemeriksaan	Riwayat pemeriksaan yang telah disimpan	Aplikasi dapat menampilkan riwayat pemeriksaan yang dilakukan oleh pengguna	Aplikasi dapat menampilkan riwayat pemeriksaan yang dilakukan oleh pengguna	Berhasil
9	Profil Pengguna	Menekan tombol Edit	Aplikasi dapat melakukan edit terhadap akun	Aplikasi dapat melakukan edit terhadap akun	Berhasil
		Menekan tombol Simpan	Aplikasi dapat menyimpan hasil edit akun	Aplikasi dapat menyimpan hasil edit akun	Berhasil
		Menekan tombol Keluar	Aplikasi dapat <i>logout</i> akun pengguna	Aplikasi dapat <i>logout</i> akun pengguna	Berhasil
10	Navigasi	Menekan tombol Beranda	Aplikasi dapat mengarah ke <i>dashboard</i>	Aplikasi dapat mengarah ke <i>dashboard</i>	Berhasil
		Menekan tombol Riwayat	Aplikasi dapat mengarah ke riwayat pemeriksaan	Aplikasi dapat mengarah ke riwayat pemeriksaan	Berhasil
		Menekan tombol Akun	Aplikasi dapat mengarah ke profil pengguna	Aplikasi dapat mengarah ke profil pengguna	Berhasil

Dari pengujian yang telah dilakukan, maka didapat kesimpulan bahwa aplikasi dapat berjalan sesuai harapan, dimana fitur maupun

fungsi dari setiap menu maupun objek yang ada berfungsi dengan baik dan sesuai dengan tujuan perancangan.

#### D. Keberlanjutan Proyek

Setelah proyek selesai, tim akan melanjutkan dengan tahapan-tahapan berikut:

1. Evaluasi performa aplikasi setelah peluncuran.
2. Memberikan dukungan teknis dan perbaikan jika diperlukan.
3. Menerima umpan balik dari pengguna untuk perbaikan dan pengembangan selanjutnya.
4. Proses pengujian telah dilakukan secara menyeluruh untuk memastikan kualitas dan kinerja aplikasi sebelum peluncuran.
5. Proses pengujian telah memastikan bahwa aplikasi siap diluncurkan dengan kualitas yang tinggi dan minim *bug*.

## **BAB IV Penutup**

### A. Kesimpulan

Aplikasi ini telah berhasil dalam mengembangkan aplikasi *Android* yang memanfaatkan teknologi pengenalan gambar, yaitu *TensorFlow Lite*, untuk mendeteksi penyakit pada tanaman cabai. Aplikasi ini dirancang dengan tujuan untuk membantu petani dalam mengidentifikasi penyakit pada cabai dengan cepat dan akurat, sehingga dapat mengambil tindakan pencegahan dan pengobatan yang tepat waktu. Hasil pengujian *System Usability Scale* (SUS) menunjukkan skor sebesar 82.5, yang mengindikasikan bahwa aplikasi ini memiliki tingkat kegunaan yang sangat baik. Dengan aplikasi ini, diharapkan petani dapat mengurangi kerugian akibat penyakit pada cabai dan meningkatkan kualitas serta kuantitas hasil panen.

### B. Saran

Untuk pengembangan aplikasi lebih lanjut, disarankan agar aplikasi ini tidak hanya memberikan hasil deteksi penyakit tetapi juga dilengkapi dengan fitur-fitur tambahan seperti:

1. Dapat mencakup informasi yang lebih dalam mengenai cara pencegahan dan pengobatan penyakit yang terdeteksi pada tanaman cabai.
2. Memberikan panduan tentang penggunaan pestisida yang tepat dan metode budidaya cabai sehingga dapat mengurangi risiko penyakit.
3. Tindakan perawatan dan pemulihan tanaman yang terkena penyakit.

Dengan demikian, aplikasi ini tidak hanya berfungsi sebagai alat diagnostik, tetapi juga sebagai sumber pengetahuan bagi petani untuk mengelola kesehatan tanaman mereka secara lebih efektif.

## DAFTAR PUSTAKA

- [1] *TensorFlow*, “*TensorFlow Lite for Android*.” Accessed: Apr. 05, 2024. [Online]. Available: <https://www.TensorFlow.org/Lite/Android>
- [2] A. R. Respati and E. Djumena, “Cabai Jadi Pendorong Inflasi, BPS: Faktor El Nino dan Distribusi.” Accessed: Apr. 05, 2024. [Online]. Available: <https://money.kompas.com/read/2023/12/01/161500126/cabai-jadi-pendorong-inflasi-bps-faktor-el-nino-dan-distribusi>
- [3] hortikultura.pertanian.go.id, “Meski Harga Cabai Masih Tinggi, Kementan Tegaskan Tak Perlu Impor.”
- [4] Insan Nurzaman Bangga Adi Pratama, Issa Arwani, and Handoko, “Pengembangan Aplikasi *Mobile* Pendekripsi Penyakit pada Tanaman Cabai dengan menggunakan Teknologi Clarifai,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 5, pp. 1472–1479, May 2020, Accessed: Apr. 25, 2024. [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/7253/3501/51162>
- [5] Indra Wijaya, “IMPLEMENTASI *TENSORFLOW LITE* UNTUK MENGETAHUI JENIS CACAT PADA BIJI KOPI ROBUSTA BERBASIS *ANDROID*,” Cilegon, 2023. Accessed: Apr. 24, 2024. [Online]. Available: [https://eprints.untirta.ac.id/31531/6/INDRAWIJAYA\\_3332180011\\_Fulltext.pdf](https://eprints.untirta.ac.id/31531/6/INDRAWIJAYA_3332180011_Fulltext.pdf)
- [6] Refika Uswatun Muhamaroh, “Analisis Ekonomi Pertanian Komoditas Cabai Merah untuk Mewujudkan Pembangunan Pertanian Berkelanjutan.” Accessed: Mar. 01, 2024. [Online]. Available: <https://www.kompasiana.com/refikauswatun/5eec9703097f365af3617072/analisis-ekonomi-pertanian-komoditas-cabai-merah-untuk-mewujudkan-pembangunan-pertanian-berkelanjutan>

## LAMPIRAN

**Lampiran 1 Proposal Proyek**

### **PROPOSAL PROYEK**

#### **APLIKASI PENDETEKSI PENYAKIT PADA TANAMAN CABAI (CekCabai)**

**Klaten, Jawa Tengah**



Oleh:

Vicha Farazahla 2100018400

Reza Sefiyanti 2100018429

Salih Arya Gumilang 2100018438

Dea Amelia Setyorini 2100018445

Yudit Aditya Nugraha 2100018464

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UNIVERSITAS AHMAD DAHLAN**

**TAHUN 2024**

## **1. Latar Belakang dan Permasalahan**

### A. Latar Belakang

Budidaya cabai merupakan sektor penting dalam perekonomian negara. Menurut data dari Direktorat Jenderal *Hortikultura* Kementerian Pertanian, produksi cabai di Indonesia terus meningkat. Cabai memiliki peran signifikan dalam inflasi di Indonesia. Kenaikan harga cabai dapat berdampak pada inflasi, mengurangi daya beli masyarakat, dan menghambat pertumbuhan ekonomi nasional. Pemerintah telah melakukan berbagai upaya untuk menstabilkan pasokan dan mengendalikan kenaikan harga cabai.

Namun, petani cabai masih menghadapi tantangan dalam mengatasi penyakit pada tanaman cabai. Salah satu masalah utama adalah kurangnya pengetahuan tentang jenis-jenis penyakit yang menyerang tanaman cabai dan cara pengendaliannya. Penyakit pada tanaman cabai dapat disebabkan oleh serangan virus, bakteri, dan jamur. Faktor lingkungan seperti suhu, pH tanah, dan kelembaban juga mempengaruhi kesehatan tanaman cabai. Kurangnya pemahaman tentang faktor-faktor ini membuat petani kesulitan dalam menerapkan tindakan pencegahan yang efektif. Jika penyakit pada tanaman cabai tidak ditangani dengan baik, hal ini akan berdampak negatif pada produktivitas dan kualitas hasil panen. Penyebaran penyakit dapat menyebabkan kerugian besar bagi petani dan bahkan mengancam kelangsungan usaha pertanian cabai.

Permasalahan yang dipaparkan di atas dapat dideteksi dini menggunakan aplikasi *Android* yang dilengkapi dengan teknologi pengenalan citra, seperti menggunakan *TensorFlow Lite*. *Android* adalah sistem operasi populer untuk *smartphone* dan tablet, dengan lebih dari 2.5 miliar pengguna aktif per bulan. Dikenal karena fitur canggih dan tampilan menarik, *Android* menjadi pilihan utama bagi berbagai produsen perangkat. Keunggulannya terletak pada *sifat open-source* yang mendukung pengembangan produk, serta *Google Play Store* sebagai *platform* distribusi aplikasi dan konten digital seperti musik, buku, dan film. *Android Software Development Kit (SDK)* menyediakan *tools* penting untuk pengembangan aplikasi, menggunakan Java sebagai bahasa utama dan Kotlin sebagai bahasa tambahan resmi. *TensorFlow Lite* adalah *library* untuk implementasi model *machine learning* pada perangkat seluler, dengan fitur konversi model *TensorFlow*

menjadi *file flat buffer* yang lebih kecil ukurannya. *TensorFlow Lite* mendukung berbagai fitur seperti identifikasi objek, deteksi objek, dan penggunaan model bahasa alami.

Berdasarkan pemaparan diatas, proyek ini bertujuan untuk membangun aplikasi *Android* yang mampu mengidentifikasi penyakit pada tanaman cabai berdasarkan citra daun ataupun buah. Proyek ini diharapkan dapat menghasilkan solusi yang efektif dalam mendeteksi serta mengklasifikasi penyakit pada tanaman cabai sehingga dapat meminimalisir kerusakan tanaman yang cepat menyebar ke tanaman lain. Hama yang juga merupakan ancaman serius bagi tanaman cabai tidak akan dibahas dalam lingkup proyek ini.

B. Permasalahan yang ingin diselesaikan

1. Pemahaman tentang faktor yang mempengaruhi kesehatan tanaman cabai.
2. Pengetahuan tentang jenis-jenis penyakit yang menyerang tanaman cabai dan cara pengendaliannya.
3. Keterbatasan teknologi yang digunakan untuk mendeteksi penyakit pada tanaman cabai.

## 2. Tujuan Proyek

A. Kebutuhan Bisnis

1. Meningkatkan kualitas cabai. Aplikasi ini dapat membantu petani untuk menghasilkan cabai yang lebih berkualitas.
2. Meningkatkan efisiensi dan efektivitas pengelolaan tanaman cabai. Dengan membantu para petani untuk mendeteksi penyakit pada tanaman cabai mereka, aplikasi ini dapat membantu petani untuk mengelola tanaman cabai mereka dengan lebih efisien dan efektif.
3. Meningkatkan hasil panen cabai. Dapat membantu petani untuk mendeteksi penyakit pada tanaman cabai secara dini dan akurat.
4. Mengurangi kerugian akibat penyakit pada tanaman cabai. Aplikasi ini dapat meminimalkan kerugian akibat penyakit.

## B. Peluang Bisnis

1. Pasar yang besar. Menurut data dari Badan Pusat Statistik (BPS), pada tahun 2022, produksi cabai di Indonesia mencapai 28,86 juta ton. Sehingga menunjukkan bahwa terdapat pasar yang besar untuk aplikasi ini.
2. Permintaan yang tinggi. Para petani di Indonesia membutuhkan solusi yang dapat membantu mereka untuk meningkatkan hasil panen cabai mereka dan mengurangi kerugian akibat penyakit.
3. Persaingan rendah. Terdapat peluang bisnis yang besar untuk aplikasi pendekripsi penyakit tanaman cabai di Indonesia.

## 3. Produk Luaran yang dihasilkan

Luaran yang diharapkan dari proyek:

- a. Aplikasi dapat mencatat riwayat penyakit tanaman cabai di berbagai lokasi, yang dapat membantu petani memantau pola dan tren penyakit seiring waktu.
- b. Aplikasi bisa memberikan pemantauan real-time terhadap kondisi tanaman cabai, mendekripsi adanya penyakit secara dini sehingga petani dapat mengambil tindakan pencegahan dengan cepat.
- c. Aplikasi dapat menganalisis data yang dikumpulkan dari berbagai sumber, seperti citra tanaman, data cuaca, dan pola penyakit historis, untuk memberikan wawasan yang lebih dalam tentang faktor-faktor yang mempengaruhi kesehatan tanaman cabai.
- d. Aplikasi dapat menyediakan sumber daya pendidikan dan pelatihan bagi petani tentang cara mengidentifikasi penyakit, tindakan pencegahan, dan strategi manajemen penyakit.

## 4. Analisis Kebutuhan Pengguna, Sistem, dan Data.

### A. Analisis kebutuhan pengguna

1. Kebutuhan Fungsionalitas Pengguna
  - a. Memungkinkan pengguna berbagi gambar.
  - b. Menyimpan riwayat penyakit cabai yang pernah terdeteksi pada tanaman cabai sebelumnya.

- c. Memprediksi risiko serangan penyakit cabai berdasarkan data
2. Kebutuhan Non-Fungsionalitas Pengguna
  - a. Aplikasi harus mudah digunakan.
  - b. Aplikasi harus akurat dan terpercaya.
  - c. Aplikasi dapat diakses dengan mudah.

## B. Analisis Kebutuhan Sistem

1. Yang dibutuhkan sistem untuk mendeteksi adalah:
  - a. Gambar daun cabai yang terkena penyakit.
  - b. Informasi tentang penyakit cabai seperti gejala yang terjadi pada cabai.
  - c. Data pengguna aplikasi, seperti nama, alamat, dan nomor telepon.
2. Hardware dan Software yang dibutuhkan sistem:
  - a. *Hardware*
  - b. *Database*
  - c. Aplikasi *mobile*
  - d. Kamera

## C. Analisis kebutuhan Data

1. Data yang dibutuhkan
  - a. Data gambar
  - b. Data teks
  - c. Data Numerik
  - d. Pemeliharaan data
2. Analisis kebutuhan Data
  - a. Data akan dikumpulkan dari berbagai sumber
  - b. Data pengguna dikumpulkan saat pengguna mendaftar dan menggunakan aplikasi
  - c. Data disimpan dalam *database*
  - d. Data disimpan di *server* yang aman

## 5. Asumsi, Kendala, dan Ketergantungan

### A. Asumsi

1. Pengguna memahami cara menggunakan aplikasi.
2. Pengguna memiliki ponsel dengan kamera yang berfungsi dengan baik.
3. Pengguna memiliki koneksi internet yang stabil.
4. Pengguna dapat mengambil gambar daun cabai yang sakit dengan jelas tanpa distorsi.

### B. Kendala

1. Pengguna kurang memahami cara memakai aplikasi
2. Sistem tidak dapat mendeteksi semua jenis penyakit cabai
3. Kualitas gambar yang diunggah rendah
4. Koneksi internet tidak stabil

### C. Ketergantungan

1. Ketersediaan data tentang penyakit-penyakit pada tanaman cabai
2. Ketersediaan *software* dan *hardware* yang memadai.

## 6. Struktur Organisasi Proyek

Identifikasi Pemangku kepentingan utama dan anggota tim berdasarkan fungsi, nama dan peran

Nama	Job desk	Deskripsi Tugas
Reza Sefiyanti	<i>Project Manager,</i> Sekretaris	Perencanaan keseluruhan proyek, melakukan pengawasan dan koordinasi tim, memastikan bahwa proyek berjalan sesuai rencana, membuat laporan pengerjaan proyek, membuat

Nama	<i>Job desk</i>	Deskripsi Tugas
		proposal, membuat <i>Logbook</i> kelompok, dan membuat MOU.
Vicha Farazahla	Data <i>Analyst</i> , Bendahara	Menganalisis data terkait dengan pendekripsi penyakit pada tanaman cabai, mengumpulkan dan menyajikan data relevan untuk membantu dalam pengambilan keputusan proyek, menyusun anggaran terhadap penggunaan dana, berkoordinasi dengan tim terkait untuk memastikan anggaran yang telah disetujui.
Dea Amelia Setyorini	UI/UX <i>Designer</i>	Merancang antarmuka pengguna (UI) dan pengalaman pengguna (UX), memastikan aplikasi memiliki tampilan dan interaksi yang menarik serta mudah digunakan.
Salih Arya Gumlilang	<i>Mobile Developer</i>	Pengembangan perangkat lunak Aplikasi cek cabai, menerjemahkan konsep dan desain ke dalam kode komputer.

Nama	Job desk	Deskripsi Tugas
Yudit Aditya Nugraha	<i>Quality Assurance</i>	Bertanggung jawab untuk menguji Aplikasi CekCabai guna memastikan bahwa tidak ada <i>bug</i> atau masalah yang signifikan, melakukan pengujian fungsionalitas, mengidentifikasi masalah yang perlu diperbaiki sebelum peluncuran.

## 7. Analisis SWOT

### A. *Strength* (Kekuatan)

- a) Banyaknya data pertanian yang tersedia, mulai dari data pertanian tradisional hingga data modern.
- b) Cabai merupakan salah satu komoditas tanaman pangan utama di Indonesia, sehingga ada dorongan untuk menjaga dan melindungi tanaman dari penyakit.
- c) Kesadaran petani dalam mengurangi kerugian dan meningkatkan produktivitas.

### B. *Weakness* (Kelemahan)

- a) Data yang tersedia bisa saja tidak lengkap dan tidak terbaru sehingga menghambat pengembangan dan akurasi model deteksi penyakit.
- b) Banyaknya petani di Indonesia yang mungkin tidak memiliki akses penuh terhadap teknologi.
- c) Penyakit pada tanaman cabai bisa sangat beragam yang dapat menambah kompleksitas dalam mendiagnosis penyakit secara akurat dan tepat waktu.

### C. *Opportunities* (Peluang)

- a) Mengembangkan alat deteksi yang lebih canggih dan akurat.

- b) Kesempatan bekerja sama dengan ahli pertanian untuk mengembangkan solusi yang tepat.
- c) Potensi pasar yang besar.

#### D. Threat (Ancaman)

- a) Meningkatnya persaingan teknologi baru yang terus berkembang.
- b) Keterbatasan infrastruktur pada beberapa daerah.
- c) Perubahan iklim yang dapat mempengaruhi penyebaran penyakit pada tanaman.

### 8. Work Breakdown Structure (WBS)

1. Analisis dan Perencanaan
  - 1.1. Identifikasi kebutuhan pengguna
  - 1.2. Identifikasi kebutuhan Sistem
  - 1.3. Identifikasi kebutuhan Data
2. Desain
  - 2.1. Desain UX/UI
    - 2.1.1. Kerangka Kerja (*Wireframes*)
  - 2.2. Desain Basis Data
    - 2.2.1. Diagram Aliran Data
  - 2.3. Desain Arsitektur Sistem
    - 2.3.1. Tentukan Komponen Sistem
    - 2.3.2. Pilih Teknologi
  - 2.4. Pembuatan UI/UX
3. Membangun Konstruksi Aplikasi
  - 3.1. Implementasi Desain UI/UX
  - 3.2. Pengembangan Model AI
  - 3.3. Implementasi AI
  - 3.4. Pembangunan Database Firebase
4. Pengujian
  - 4.1. SUS

- 4.2. BlackBox
- 5. Pengembangan Sistem
  - 5.1. Evaluasi
  - 5.2. Pemeliharaan sistem

## 9. Jadwal Pelaksanaan Proyek

No	Kegiatan	Bulan Pelaksanaan				
		Maret	April	Mei	Juni	Juli
<b>1. Analisis dan Perencanaan</b>						
1.1	Identifikasi kebutuhan pengguna	Red				
1.2	Identifikasi kebutuhan Sistem	Red				
1.3	Identifikasi kebutuhan Data		Red			
<b>2. Desain</b>						
2.1	Desain UI/UX			Red		
2.1.1	Kerangka Kerja ( <i>wireframes</i> )			Red		
2.2	Desain Basis Data			Red		
2.2.1	Diagram Aliran Data			Red		
2.3	Desain Arsitektur Sistem			Red		
2.3.1	Tentukan komponen sistem			Red		
2.3.2	Pilih teknologi			Red		
2.4	Pembuatan UI/UX			White	Red	
<b>3. Membangun Konstruksi Aplikasi</b>						
3.1	Implementasi Desain UI/UX			Red		
3.2	Implementasi AI			Red		
3.3	Pembangunan <i>Database</i>			White	Red	
<b>4. Pengujian</b>						
4.2	<i>Unit testing</i>			Red		
4.2	<i>Integration Testing</i>			Red		
4.3	<i>System Testing</i>			White	Red	
<b>5. Pengembangan Sistem</b>						
5.1	Evaluasi			White	Red	

5.2	Pemeliharaan sistem									
-----	---------------------	--	--	--	--	--	--	--	--	--

#### 10. Tabel Rencana Pembiayaan

Kegiatan	Bulan						Total
	Maret	April	Mei	Juni	Juli		
<b>1. Analisis dan Perencanaan</b>							
1.1 Identifikasi Kebutuhan Pengguna	35.000						35.000
1.2 Identifikasi Kebutuhan Sistem	35.000						35.000
1.3 Identifikasi Kebutuhan Data	35.000						35.000
<b>2. Desain</b>							
2.1 Desain UI/UX		100.000					100.000
2.1.1 Kerangka Kerja		40.000					40.000
2.2 Desain Basis Data		30.000					30.000
2.2.1 Diagram Aliran Data		25.000					25.000
2.3 Desain Arsitektur Sistem			45.000				45.000
2.3.1 Tentukan Komponen Sistem			20.000				20.000
2.3.2 Pilih Teknologi			20.000				20.000
2.4 Pembuatan UI/UX				100.000			100.000

<b>3. Membangun Aplikasi <i>Android</i></b>						
3.1 Implementasi UI/UX				100.000		100.000
3.2 Implementasi AI				100.000		100.000
3.3 Pembangunan <i>Database</i>				70.000		70.000
<b>4. Pengujian</b>						
4.1 <i>Unit testing</i>					35.000	35.000
4.2 <i>Integration Testing</i>					35.000	35.000
4.3 <i>System Testing</i>					45.000	45.000
<b>5. Pengembangan Sistem</b>						
5.1 Evaluasi					40.000	40.000
5.2 Pemeliharaan Sistem					40.000	40.000
<b>Total</b>	105.000	195.000	85.000	300.000	195.000	850.000

**Lampiran 2 Surat Perintah Kerja / Kontrak Kerja dengan Mitra / Client**

**SURAT PERJANJIAN KERJASAMA**

**PROYEK APLIKASI PENDETEKSI PENYAKIT PADA TANAMAN CABAI**

Pada hari ini, 08 Juli 2024 bertempat di Klaten, telah ditandatangani surat perjanjian kerjasama antara :

**Pihak Pertama**

Nama : Reza Sefiyanti

Jabatan : Proyek Manager

Alamat : Pandeyan, Umbulharjo, Kota Yogyakarta. DI Yogyakarta

**Pihak Kedua**

Nama : Endri Biniwarni

Jabatan : Pemilik dan Pengelola Utama

Alamat : Desa Gantiwarno, Kecamatan Gantiwarno, Kabupaten Klaten.

Selanjutnya disebut dengan **PIHAK PERTAMA** dan **PIHAK KEDUA**, dengan ini telah sepakat untuk mengadakan perjanjian kerjasama dalam hal pengembangan aplikasi Pendekripsi Penyakit Pada Tanaman Cabai, dengan ketentuan ketentuan sebagai berikut:

## **PASAL 1**

### **Ruang Lingkup Kerja sama**

1. **Pihak Pertama** dan **Pihak Kedua** sepakat untuk bersama-sama mengembangkan Aplikasi Pendekripsi Penyakit pada Tanaman Cabai.
2. Aplikasi ini akan menggunakan teknologi *Tensorflow Lite* untuk mendekripsi penyakit pada tanaman cabai.
3. **Pihak Pertama** bertanggung jawab untuk:
  - a. Mengembangkan aplikasi pendekripsi penyakit pada tanaman cabai.
  - b. Melakukan pelatihan kepada **Pihak Kedua** tentang penggunaan aplikasi.
4. **Pihak Kedua** bertanggung jawab untuk:
  - a. Menyediakan data dan informasi tentang penyakit pada tanaman cabai.
  - b. Melakukan sosialisasi dan implementasi aplikasi kepada petani cabai.

## **PASAL 2**

### **Tujuan**

Tujuan dari perjanjian kerja sama ini adalah:

1. Meningkatkan kualitas dan produktivitas tanaman cabai di Indonesia.
2. Meningkatkan efisiensi dan efektivitas pengelolaan tanaman cabai.
3. Meningkatkan hasil panen cabai.
4. Mengurangi kerugian akibat penyakit pada tanaman cabai.

## **PASAL 3**

### **Pendanaan**

1. Biaya pengembangan Aplikasi Pendekripsi Penyakit Pada Tanaman Cabai akan ditanggung oleh **Pihak Kedua**.
2. Biaya sosialisasi dan implementasi aplikasi kepada petani cabai akan ditanggung oleh **Pihak Kedua**.

## **PASAL 4**

### **Hak dan Kewajiban**

#### **Hak dan kewajiban Pihak Pertama:**

##### **1. Hak:**

- a. Menerima pembayaran atas jasanya dalam mengembangkan dan mengimplementasikan aplikasi.
- b. Mendapatkan pengakuan atas hak kekayaan intelektualnya atas aplikasi.

##### **2. Kewajiban:**

- a. Menyelesaikan pengembangan aplikasi sesuai dengan waktu yang telah disepakati.
- b. Memberikan pelatihan kepada **Pihak Kedua** tentang penggunaan aplikasi.

#### **Hak dan Kewajiban Pihak Kedua**

##### **1. Hak:**

- a. Menggunakan aplikasi pendekripsi penyakit pada tanaman cabai untuk meningkatkan produktivitas tanaman cabai.
- b. Mendapatkan pengetahuan dan keterampilan baru dalam mengidentifikasi dan mengendalikan penyakit tanaman cabai.

##### **2. Kewajiban:**

- a. Menyediakan pendanaan yang dibutuhkan untuk pengembangan aplikasi.
- b. Menyediakan data dan informasi tentang penyakit tanaman cabai di wilayahnya.
- c. Membantu dalam pengujian dan pengembangan aplikasi
- d. Menbantu dalam sosialisasi dan implementasi aplikasi kepada petani cabai di wilayahnya.

## **PASAL 5**

### **Jangka Waktu**

1. Perjanjian kerja sama ini berlaku selama 4 Bulan, terhitung sejak tanggal ditandatanganinya perjanjian ini.
2. Perjanjian kerja sama ini dapat diperpanjang atas kesepakatan kedua belah pihak.

## **PASAL 6**

### **Pengalihan Perjanjian**

Pihak Pertama dan Pihak Kedua tidak boleh mengalihkan perjanjian kerja sama ini kepada pihak lain tanpa persetujuan tertulis dari Pihak Pertama dan Pihak Kedua.

## **PASAL 7**

### **Penyelesaian Perselisihan**

1. Apabila terjadi sengketa dalam pelaksanaan perjanjian kerja sama ini, kedua belah pihak akan berusaha menyelesaikan dengan cara musyawarah mufakat.
2. Apabila musyawarah mufakat tidak tercapai, maka sengketa akan diselesaikan melalui Arbitrase (Pihak ketiga yang netral).

## **PASAL 8**

### **Penutup**

1. Perjanjian kerja sama ini dibuat dalam rangkap 2 (Dua) eksemplar, masing masing satu eksemplar untuk **Pihak Pertama, dan Pihak Kedua**.
2. Perjanjian kerja sama ini ditandatangani pada hari, tanggal, tempat yang tersebut di atas.

Klaten, 8 Juli 2024

#### **PIHAK PERTAMA**



Reza Sefiyanti

#### **PIHAK KEDUA**



Endri Biniwarni

**Saksi I**



---

**Yudit Aditya Nugraha**

**Saksi II**



---

**Vicha Farazahla**

### Lampiran 3 Log Book Kelompok sudah terisi minimal 7

LOG BOOK MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2021/2022 (WAJIB DIISI DAN MASUK DALAM PENILAIAN)																											
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="3" style="padding: 2px;">Nama Kelompok MPTI :</td> </tr> <tr> <td style="width: 10%;">No</td> <td style="width: 20%;">NIM</td> <td style="width: 70%;">NAMA</td> </tr> <tr> <td>1</td> <td>2100018400</td> <td>Vicha Farazahla</td> </tr> <tr> <td>2</td> <td>2100018429</td> <td>Reza Sefiyanti</td> </tr> <tr> <td>3</td> <td>2100018438</td> <td>Salih Arya Gumilang</td> </tr> <tr> <td>4</td> <td>2100018445</td> <td>Dea Amelia Setyorini</td> </tr> <tr> <td>5</td> <td>2100018464</td> <td>Yudit Aditya Nugraha</td> </tr> </table>							Nama Kelompok MPTI :			No	NIM	NAMA	1	2100018400	Vicha Farazahla	2	2100018429	Reza Sefiyanti	3	2100018438	Salih Arya Gumilang	4	2100018445	Dea Amelia Setyorini	5	2100018464	Yudit Aditya Nugraha
Nama Kelompok MPTI :																											
No	NIM	NAMA																									
1	2100018400	Vicha Farazahla																									
2	2100018429	Reza Sefiyanti																									
3	2100018438	Salih Arya Gumilang																									
4	2100018445	Dea Amelia Setyorini																									
5	2100018464	Yudit Aditya Nugraha																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="3" style="padding: 2px;">Judul Manajemen Tugas Proyek : Aplikasi Pendekripsi Penyakit pada Tanaman Cabai</td> </tr> <tr> <td colspan="3" style="padding: 2px;">Dosen Pembimbing : Taufiq Ismail, S.T., M.Cs</td> </tr> </table>							Judul Manajemen Tugas Proyek : Aplikasi Pendekripsi Penyakit pada Tanaman Cabai			Dosen Pembimbing : Taufiq Ismail, S.T., M.Cs																	
Judul Manajemen Tugas Proyek : Aplikasi Pendekripsi Penyakit pada Tanaman Cabai																											
Dosen Pembimbing : Taufiq Ismail, S.T., M.Cs																											
<b>Logbook Minggu 4 sd 7 (sebelum UTS)</b>																											
No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)	Paraf MITRA/CLIENT	Paraf Dosen Pembimbing MPTI																				
		Hari/TGL	Durasi (Jam)																								
1.	Menentukan calon mitra dan dosen pembimbing	22 Februari 2024	1	Mendapatkan rencana calon mitra																							
2.	Menghubungi pak Jefree sebagai dosen pembimbing	06 Maret 2024	1	Konfirmasi Pak Jefree sebagai dosen pembimbing	Tidak terjadi kesepakatan																						
3.	Melakukan bimbingan bersama pak Jefree membahas mitra	13 Maret 2024	2	Pak Jefree menghubungi lazismu																							
4.	Memutuskan untuk	15 Maret	3	Mendapatkan calon mitra																							

5.	mengganti mitra	2024		Menghubungi mitra dan dosen pembimbing			
6.	Menghubungi mitra dan dosen pembimbing baru	26 Maret 2024	2	Menghubungi mitra dan dosen pembimbing			
7.	Bimbingan dengan dosen pembimbing	2 April 2024	2	Mendapatkan arahan penggerakan proyek			
8.	Mengubah desain aplikasi	23 April 2024	4	Desain aplikasi menjadi lebih sederhana			
9.	Menghubungi mitra baru dan membuat starter projek aplikasi serta mengubah proposal	26 April 2024	4	Proposal ditandatangani mitra			
10.	Melanjutkan programming	27 April 2024	12	Progress aplikasi			
11.	Mengadakan pertemuan dengan mitra untuk membahas proyek	28 April 2024	4	Mencatat kebutuhan mitra aplikasi			
12.	Melakukan bimbingan dengan pak Taufiq dan membahas pertambahan dosipem sementara	14 Mei 2024	2				
	Mengerjakan laporan MPTI	16 Mei 2024	4	Progress laporan MPTI			

Catatan Pembimbing Lapangan/Dosen Pembimbing MPTI/Dosen Pengampu Kelas MPTI :

---



---



---



---

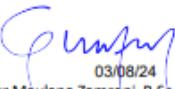
Logbook Minggu 9 sd 11 (setelah UTS)

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)	Paraf MITRA/CLIENT	Paraf Dosen Pembimbing MPTI
		Hari/TGL	Jam				
1	Mengerjakan Laporan	18 Mei – 1 Agustus 2024	48	Laporan telah selesai			
2	Melanjutkan pengembangan Aplikasi	18 Mei – 1 Agustus 2024	72	Model AI berhasil dibuat dan diintegrasikan ke dalam aplikasi android	Beberapa arsitektur CNN tidak cocok apabila digunakan pada android. Ganti model		

Yogyakarta, 3 Agustus 2024

Dosen Pengampu Kelas MPTI

Ketua MPTI



03/08/24  
Guntur Maulana Zamroni, B.Sc., M.Kom.  
NIY. 60181172



Reza Sefiyanti  
2100018429

## Lampiran 4 Log Book Individu sudah terisi minimal 7x

### Logbook Reza Sefiyanti

LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2023/2024 (WAJIB DIISI DAN TERLAMPIR DALAM LAPORAN)					
Nama Kelompok MPTI : FundEASE			Petunjuk Pengisian Log Book		
Nim : 2100018429			1. Log book dibuat per kegiatan		
Nama : Reza Sefiyanti			2. Log book ditulis tangan		
Jobdesk : PROYER Manager			3. Log book per kegiatan individu dan di paraf oleh dosen pengampu kelas MPTI saat UTS		
Judul MPTI: Apakah Pendekripsi penyakit pada Tanaman cabai					
Dosen Pembimbing MPTI: Taufiq Ismail, S.T., M.CS					
No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)
		Hari/TGL	Durasi (Jam)		
1.	Pertem dengan tim untuk membahas proyek dan mendapatkan laporan kerjaan membuat grup di what'sapp dan memintaikan semua anggota.	Senin/ 5 Maret 2024	2 jam	- Mendapatkan proyek yang akan dilakukan. - semua Anggota tim terangabung kedalam grup what'sapp.	-
2.	mengarahkan sibdes selap Anggara	Rabu/ 6 Maret 2024	1 jam	-rap Anggota mendikte sibdes masina masing	-
3.	Membuat dan memperbarui protosai proyek .	Rabu/ 13 Maret 2024	2 Jam	- proposal yang telah diperbarui	- sibdes kurang lengkap
4.	mengadakan pertemuan dengan tim	Senin/ 2 April 2024	2 jam	- Evaluasi kinerja tim	-
5.	Menemu K Memvalidas° kerjaan yang telah diselesaikan oleh tim.	Senin/ 26 April 2024	1 jam	- Proses kerjaan terorganisir	-
6.	Menghubungi calon mitra dan membuat jadwal temu	Senin/ 26 April 2024	1 jam	- Calon mitra mengelugor untuk sangg° temu.	-
7.	Mengadakan pertemuan dengan klien atau calon mitra untuk membahas tentang proyek.	Senin/ 26 April 2024	2 jam	- klien atau calon mitra mengelugor kerja sama proyek.	- tidak dapat menemu klien secara langsung. sehingga dimediasi oleh anggota tim.
8.	Membuat dan memperbarui documen proyek	Senin/ 3 Mei 2024	2 jam	- Dokumen proyek terbaru yang telah diperbarui.	-

Catatan Ketua Tim MPTI :

Nb; Untuk ketua tim di ttd verifikasi oleh anggota lain dalam satu kelompok

Yogyakarta, 16 Mei 2024  
Ketua MPTI  
  
(Safit Arya Gunilang.)

## Logbook Dea Amelia Setyorini

**LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA  
PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2023/2024  
(WAJIB DIISI DAN TERLAMPIR DALAM LAPORAN)**

Nama Kelompok MPTI : Fundease
Nim : 2100018445
Nama : Dea Amelia Setyorini
Jobdesk : UI/UX Designer
Judul MPTI : Aplikasi Pendekteksi Penyakit pada Tanaman Cabai
Dosen Pembimbing MPTI : Taufiq Ismail, S.T., M.Cs

**Petunjuk Pengisian Log Book**

1. Log book dibuat per kegiatan
2. Log book ditulis tangan
3. Log book per kegiatan individu dan di paraf oleh dosen pengampu kelas MPTI saat UTS

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)
		Hari/TGL	Durasi (Jam)		
1	Membuat Wireframe awal untuk halaman utama aplikasi	Sabtu, 09 Maret 2024	2,5 Jam	Sketsa awal Aplikasi	
2	Menentukan logo aplikasi	Senin, 11 Maret 2024	2 Jam	Menggunakan Logo yang cocok dengan aplikasi dan disetujui Tim	
3	Membuat Design register dan login	Selasa, 12 Maret 2024	1,5 Jam	Tampilan visual halaman register dan login yang mencakup logo aplikasi, form, tombol register dan login	
4	Membuat Design Menu dan akun Profil	Rabu, 13 Maret 2024	2 Jam	Tampilan visual dari menu utama dan akun Profil	

5	Membuat Design Analisis tanaman dan hasil penyakit	Rabu, 13 Maret 2024	2 Jam	Tampilan Visual halaman analisis tanaman dan hasil penyakit	
6	Membuat Prototype Interaktif	Kamis, 14 Maret 2024	1 Jam	prototype Interaktif yang diuji oleh tim proyek	
7	Mengubah Design aplikasi dan Prototype	Selasa, 23 April 2024	3,5 Jam	Design diubah sesuai proyek	
8	Membuat Design Poster	Kamis, 18 Juli 2024	1 Jam	Poster aplikasi	

Catatan Ketua Tim MPTI :

---



---



---



---

Nb; Untuk ketua tim di ttd verifikasi oleh anggota lain dalam satu kelompok

Yogyakarta, 3 Agustus 2024

Ketua MPTI



( Reza Sefiyanti)

## Logbook Vicha Farazahla

**LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA  
PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2023/2024  
(WAJIB DIISI DAN TERLAMPIR DALAM LAPORAN)**

Nama Kelompok MPTI : FundEase
Nim : 2100018400
Nama : Vicha Farazahla
Jobdesk : Data Analyst, Bendahara
Judul MPTI : Aplikasi Pendekripsi Penyakit pada Tanaman Cabai
Dosen Pembimbing MPTI : Taufiq Ismail, S.T., M.Cs.

Petunjuk Pengisian Log Book  
 1. Log book dibuat per kegiatan  
 2. Log book ditulis tangan  
 3. Log book per kegiatan individu dan di paraf oleh dosen pengampu kelas MPTI saat UTS

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)
		Hari/TGL	Durasi (jam)		
1	Analisis Kebutuhan Proses Bisnis	Kamis, 7 Maret 2024	1 jam	Mengumpulkan kebutuhan mitra untuk proses bisnis baru	-
2	Analisis Kebutuhan Aplikasi	Selasa, 12 Maret 2024	1.5 jam	Luaran yang diharapkan dari proyek	-
3	Analisis Kebutuhan Pengguna, Sistem, dan Data	Rabu, 13 Maret 2024	1.5 jam	Mengumpulkan kebutuhan pengguna, sistem, dan data	-
4	Analisis SWOT		1.5 jam		-

5	Pembuatan dan Pembaruan Dokumen Proyek	Kamis, 14 Maret 2024		Mengidentifikasi kekuatan, kelemahan, peluang, dan ancaman yang mempengaruhi proyek.	
6	Membuat Rancangan Anggaran Proyek	Jumat, 3 Mei 2024	2.5 jam	Dokumen dengan analisis kebutuhan proyek yang terbaru	-
7	Membuat Jadwal Pelaksanaan Proyek	24 Mei 2024	2 jam	Rancangan Anggaran Proyek	-
		13 Juni 2024	1 jam	Rancangan Jadwal Pelaksanaan Proyek	-

Catatan Ketua Tim MPTI :

---



---



---



---

Nb; Untuk ketua tim di ttd verifikasi oleh anggota lain dalam satu kelompok

Yogyakarta, 3 Agustus 2024

Ketua MPTI

(Reza Sefiyanti)

## Logbook Salih Arya GumiLang

**LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA  
PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2023/2024  
(WAJIB DIISI DAN TERLAMPIR DALAM LAPORAN)**

Nama Kelompok MPTI : FundEase			<b>Petunjuk Pengisian Log Book</b> 1. Log book dibuat per kegiatan 2. Log book ditulis tangan 3. Log book per kegiatan individu dan di paraf oleh dosen pengampu kelas MPTI saat UTS	
Nim : 2100018438				
Nama : Salih Arya GumiLang				
Jobdesk : Programmer				
Judul MPTI : APLIKASI PENDETEKSI PENYAKIT PADA TANAMAN CABAI				
Dosen Pembimbing MPTI : Taufiq Ismail S.T., M.Cs.				

No	Kegiatan	waktu pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari/Tgl	Durasi		
1	Membuat Project Android dan Publish ke github	Jumat, 29 Maret 2024	0,5 Jam	Repository Github	-
2	Membuat UI layout greeting pages	Sabtu, 30 Maret 2024	0,5 Jam	Layout greeting pages (Jetpack Compose)	-
3	Implementasi Viewpager greeting pages	Minggu, 31 Maret 2024	2 Jam	Viewpager greeting sudah bisa slide sesuai indeks	-
4	Membuat login activity dan setup Firebase	Kamis, 25 April 2024	3 Jam	fitur Login sudah bisa digunakan	-
5	Membuat Register activity dan setup Firebase	Semin, 15 April 2024	3 Jam	fitur register sudah bisa digunakan	-
6	Membangun model AI tensorflow lite pengenalan citra	Jumat, 5 Juli 2024	6 Jam	Model dapat mendeteksi beberapa penyakit tanaman cabai	Keterbatasan limit resource google collab
7	Mengintegrasikan model AI ke dalam aplikasi android	5 Juli - 20 Juli	48 Jam	Aplikasi android dapat memindai penyakit tanaman cabai	Mencoba beberapa arsitektur model AI, terdapat beberapa arsitektur yang tidak cocok dengan android
8	Membuat tampilan beberapa halaman aplikasi android	21 Juli	10 Jam	Dashboard, Camera, Analysis Screen, Recommendation Screen, Profile, History	-
9	Mengatur alur data pemindaiannya yang didapatkan	22 Juli	10 Jam	Hasil deteksi dapat digunakan pada analysis screen dan recommendation screen	kesulitan mengintegrasikan hasil deteksi dari camera activity dengan composable screen lainnya

## Logbook Yudit Aditya Nugraha

**LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA  
PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2023/2024  
(WAJIB DIISI DAN TERLAMPIR DALAM LAPORAN)**

Nama Kelompok MPTI : FUNDEASE			<b>Petunjuk Pengisian Log Book</b> 1. Log book dibuat per kegiatan 2. Log book ditulis tangan 3. Log book per kegiatan individu dan di paraf oleh dosen pengampu kelas MPTI saat UTS	
Nim : 2100018464				
Nama : Yudit Aditya Nugraha				
Jobdesk : Quality Assurance				
Judul MPTI : Aplikasi Pendekripsi Penyakit pada Tanaman Cabai				
Dosen Pembimbing MPTI : Taufiq Ismail, S.T., M.Cs				

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)
		Hari/TGL	Durasi (Jam)		
1.	Memastikan Greetings page pada aplikasi berjalan sesuai Rencana dapat di slide	Minggu / 31 Maret 2024	½ Jam	Greetings Page dapat di slide serta terdapat indikator Jumlah Pages yang ditampilkan	
2.	Membuat Test terhadap fungsi register pada halaman awal setelah greetings page	Sabtu / 20 April 2024	1 Jam	Fungsi Register berhasil berjalan dengan semestinya atau sesuai skenario	
3.	Menyiapkan kebutuhan untuk pembuatan form SUS	Rabu / 1 Mei 2024	1 Jam	Mendapatkan pertanyaan pertanyaan untuk pengujian	
4.	Membuat form SUS dan memberikan kepada responden	Senin / 1 Juli 2024	1,5 Jam	Mendapatkan jawaban dari responden	

5.	Mempersiapkan kebutuhan untuk pengujian blackbox	Kamis / 15 Juli 2024	1 Jam	Mendapat Rencana pengujian dan data uji	
6.	Melakukan pengujian blackbox	Senin / 22 Juli 2024	2,5 Jam	Mendapatkan skenario dan hasil uji	
7.	Melampirkan laporan	Kamis / 1 Agustus 2024	1 Jam	Hasil uji terlampir dalam laporan	

Catatan Ketua Tim MPTI :

---



---



---



---

Nb; Untuk ketua tim di ttd verifikasi oleh anggota lain dalam satu kelompok

Yogyakarta, 3 Agustus 2024

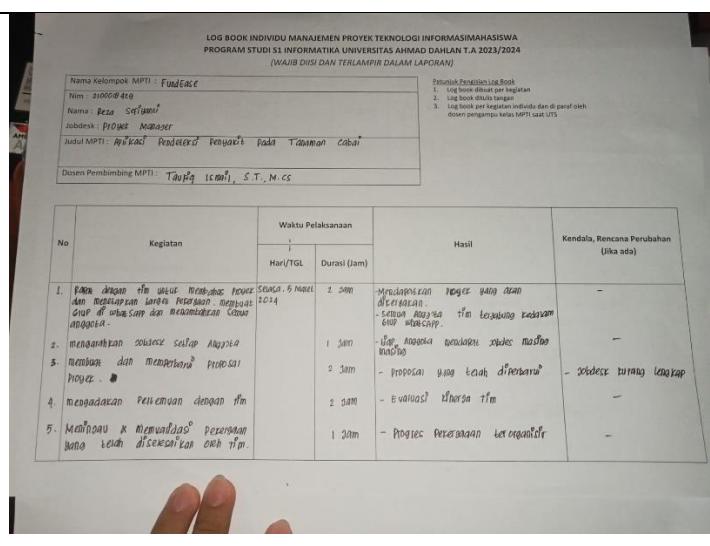
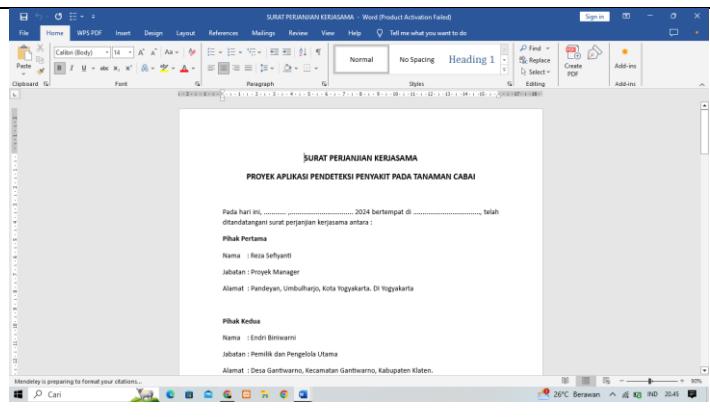
Ketua MPTI

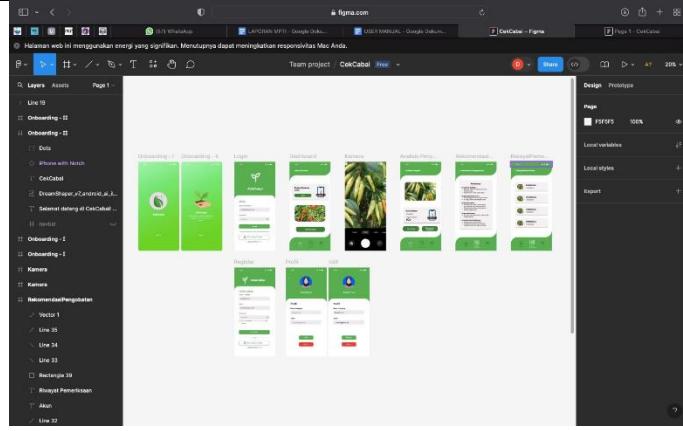
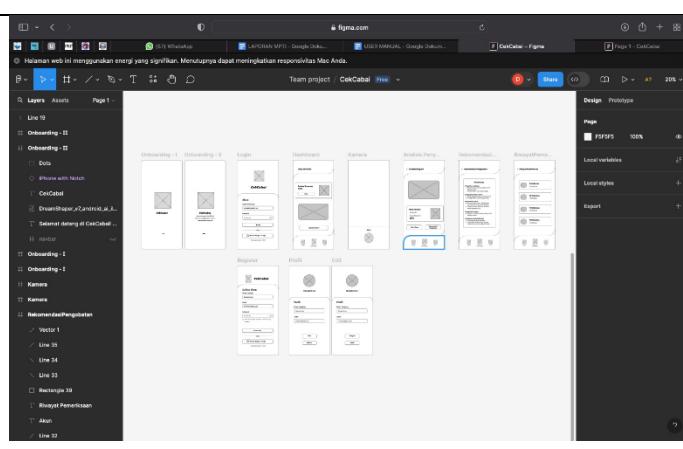
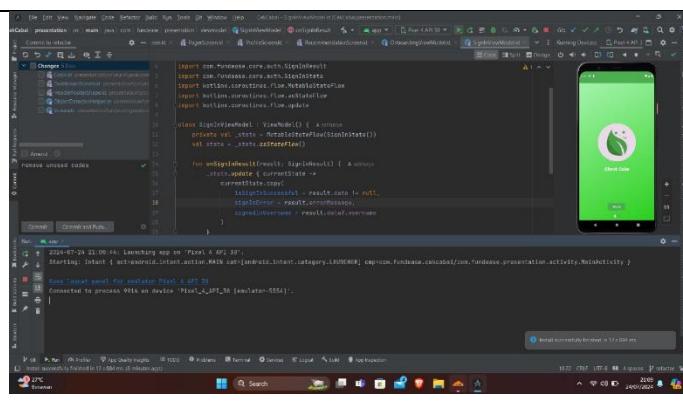


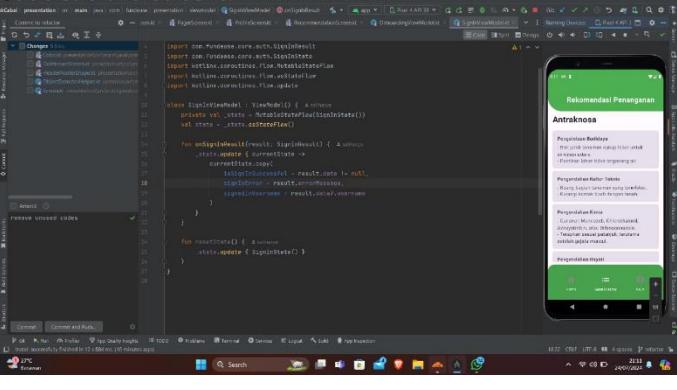
(Reza Sefiyanti)

## Lampiran 5 Foto Dokumentasi Kegiatan Proyek

No	Foto Dokumentasi	Keterangan
1		Bimbingan MPTI kepada Dosen Pembimbing.
2		Pembuatan desain UI.
3		<p>Penulisan proposal yang akan diajukan ke klien / mitra.</p> <p><b>Reza Sefiyanti</b> 2100018429</p>

4		<p>Bimbingan MPTI, membahas terkait anggaran dana dan proposal proyek.</p>																																						
5	 <p><b>LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2023/2024 (WAJIB DIISI DAN TERLAMPIR DALAM LAPORAN)</b></p> <p><b>Nama Kelompok MPTI : Fundcase</b> Nim : 31008468 Nama : Reza Syafiqi Jabatan : Projek Manager Jadwal MPTI : Apa yang Pendekar Penyakti pada Tanaman Cabai</p> <p>Dosen Pembimbing MPTI : Taufiq Ismail, S.T., M.CS</p> <table border="1"> <thead> <tr> <th rowspan="2">No</th> <th rowspan="2">Kegiatan</th> <th colspan="2">Waktu Pelaksanaan</th> <th rowspan="2">Hasil</th> <th rowspan="2">Kendala, Rencana Perubahan (Jika ada)</th> </tr> <tr> <th>Har/TGL</th> <th>Durasi (jam)</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>Pada akhir tlm wktu pelaksanaan kegiatan 5 bulan</td> <td>2024</td> <td>1 jam</td> <td> <ul style="list-style-type: none"> <li>Mendapatkan reaksi yang baik</li> <li>- Siswa yang tidak mengikuti kegiatan</li> </ul> </td> <td>-</td> </tr> <tr> <td>2.</td> <td>mengantarkan solusi cepat Algoritma</td> <td></td> <td>1 jam</td> <td> <ul style="list-style-type: none"> <li>- Solusi yang diberikan tidak masuk ke dalam</li> </ul> </td> <td>- Solusi kurang lengkap</td> </tr> <tr> <td>3.</td> <td>membaca dan memperbaiki Proposal</td> <td></td> <td>2 jam</td> <td> <ul style="list-style-type: none"> <li>- Proposial yang telah diperbaiki</li> </ul> </td> <td>-</td> </tr> <tr> <td>4.</td> <td>megadakan pertemuan dengan tim</td> <td></td> <td>2 jam</td> <td> <ul style="list-style-type: none"> <li>- Evaluasi tlm</li> </ul> </td> <td>-</td> </tr> <tr> <td>5.</td> <td>Membuat k memorandum persetujuan yang telah disesuaikan oleh tlm</td> <td></td> <td>1 jam</td> <td> <ul style="list-style-type: none"> <li>- Proses persetujuan berjalan</li> </ul> </td> <td>-</td> </tr> </tbody> </table>	No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)	Har/TGL	Durasi (jam)	1.	Pada akhir tlm wktu pelaksanaan kegiatan 5 bulan	2024	1 jam	<ul style="list-style-type: none"> <li>Mendapatkan reaksi yang baik</li> <li>- Siswa yang tidak mengikuti kegiatan</li> </ul>	-	2.	mengantarkan solusi cepat Algoritma		1 jam	<ul style="list-style-type: none"> <li>- Solusi yang diberikan tidak masuk ke dalam</li> </ul>	- Solusi kurang lengkap	3.	membaca dan memperbaiki Proposal		2 jam	<ul style="list-style-type: none"> <li>- Proposial yang telah diperbaiki</li> </ul>	-	4.	megadakan pertemuan dengan tim		2 jam	<ul style="list-style-type: none"> <li>- Evaluasi tlm</li> </ul>	-	5.	Membuat k memorandum persetujuan yang telah disesuaikan oleh tlm		1 jam	<ul style="list-style-type: none"> <li>- Proses persetujuan berjalan</li> </ul>	-	<p>Pengisian <i>Logbook</i> Individu.</p>
No	Kegiatan			Waktu Pelaksanaan				Hasil	Kendala, Rencana Perubahan (Jika ada)																															
		Har/TGL	Durasi (jam)																																					
1.	Pada akhir tlm wktu pelaksanaan kegiatan 5 bulan	2024	1 jam	<ul style="list-style-type: none"> <li>Mendapatkan reaksi yang baik</li> <li>- Siswa yang tidak mengikuti kegiatan</li> </ul>	-																																			
2.	mengantarkan solusi cepat Algoritma		1 jam	<ul style="list-style-type: none"> <li>- Solusi yang diberikan tidak masuk ke dalam</li> </ul>	- Solusi kurang lengkap																																			
3.	membaca dan memperbaiki Proposal		2 jam	<ul style="list-style-type: none"> <li>- Proposial yang telah diperbaiki</li> </ul>	-																																			
4.	megadakan pertemuan dengan tim		2 jam	<ul style="list-style-type: none"> <li>- Evaluasi tlm</li> </ul>	-																																			
5.	Membuat k memorandum persetujuan yang telah disesuaikan oleh tlm		1 jam	<ul style="list-style-type: none"> <li>- Proses persetujuan berjalan</li> </ul>	-																																			
6	 <p><b>SURAT PERJANJIAN KERJASAMA</b> PROYEK APLIKASI PENDETEKSI PENYAKIT PADA TANAMAN CABAI</p> <p>Pada hari ini, ..... 2024 bertempat di ....., telah ditandatangani surat perjanjian kerjasama antara :</p> <p><b>Pihak Pertama</b></p> <p>Nama : Reza Syafiqi Jabatan : Projek Manager Alamat : Pandeyan, Umbulharjo, Kota Yogyakarta, DI Yogyakarta</p> <p><b>Pihak Kedua</b></p> <p>Nama : Endri Biniwari Jabatan : Pemilik dan Pengelola Utama Alamat : Desa Gantungwaro, Kecamatan Gantungwaro, Kabupaten Klaten.</p>	<p>Penulisan MOU.</p>																																						

7		<p>Pembuatan desain UI lanjutan.</p>
8		<p>Pembuatan wireframe.</p>
9		<p>Pertemuan tim, membuat proses bisnis.</p>
10		<p>Implementasi desain UI/UX ke dalam codingan.</p>

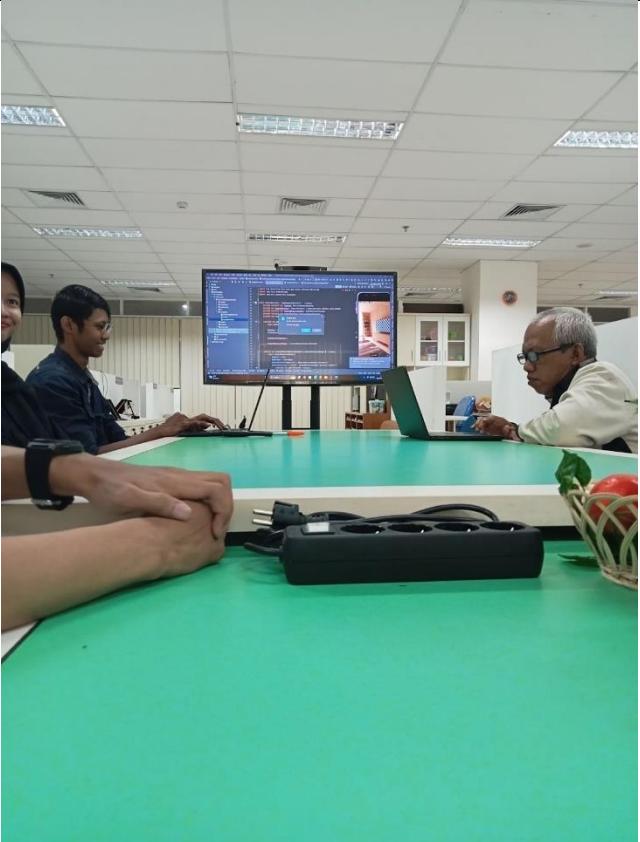
11		Pertemuan tim, membuat Laporan dan peng�aan proyek lanjutan.
12		Pertemuan tim, melanjutkan peng�aan proyek.
13		Penyelesaian pembuatan aplikasi sebelum penyerahan ke klien / mitra.

14	 A group of four people are gathered around a row of chili plants in a garden. They are examining the plants and some are holding small containers. The plants are lush and green.	Pertemuan dengan klien/mitra dan mengecek kebun cabai.
15	 Three women are sitting on a tiled floor in a living room, signing a document. One woman in a pink hijab is on the left, one in a brown hijab is in the center, and one in a black hijab is on the right. They are all smiling and looking at the document.	Penandatanganan MOU oleh klien/mitra sebagai Pihak Kedua.

16		<p>Penandatanganan MOU oleh <i>Project Manager</i>.</p>
17		<p>Penandatanganan MOU oleh para Saksi.</p>

18		Sosialisasi penggunaan Aplikasi Pendekripsi Penyakit pada Tanaman Cabai kepada klien/mitra.
19		Pertemuan tim, menyelesaikan laporan, membuat <i>User Manual</i> , & membuat Poster Aplikasi.
20		Pertemuan tim, menyelesaikan laporan dan membuat realisasi jadwal pelaksanaan proyek.

21		Pertemuan tim, menyelesaikan laporan akhir.
22		Bimbingan MPTI ke dosen pembimbing.



23

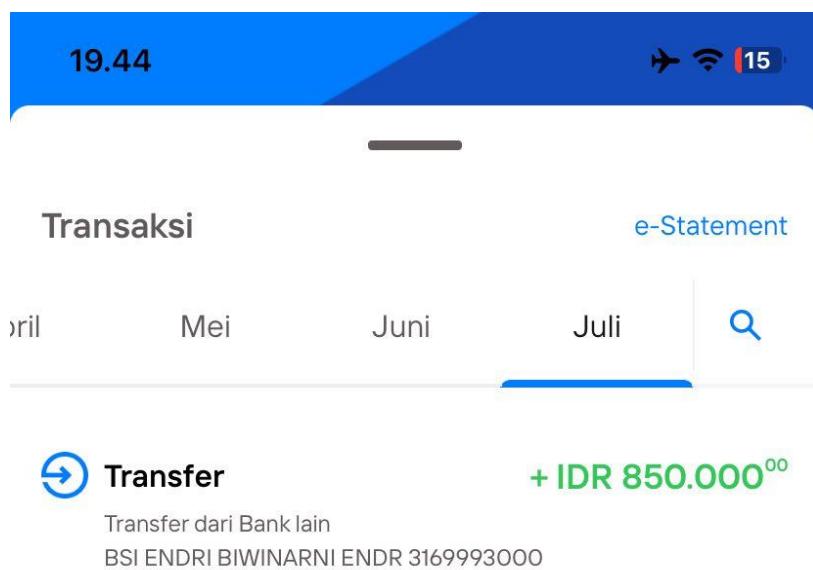


Pengerjaan Revisi Laporan Akhir.

**Lampiran 6 Bukti Serah Terima Proyek**



## Lampiran 7 Bukti Pembayaran Proyek



## Lampiran 8 Tools: source code & user manual

### A. Source Code

App.kt

```
class App : Application() {  
    override fun onCreate() {  
        super.onCreate()  
  
        startKoin {  
            AndroidLogger()  
            AndroidContext(this@App)  
            modules(  
                commonModule,  
                coreModule,  
                presentationModule  
            )  
        }  
    }  
}
```

```
    FirebaseApp.initializeApp(this)
}
}
```

### ObjectDetectorHelper.kt

```
class ObjectDetectorHelper(
    private val context: Context,
    private val modelName: String = "ssd_mobilenet_fpnLite_metadata.tflite",
    private val objectDetectorRepository: ObjectDetectorRepository,
    private val detectorListener: DetectorListener?
) {

    private var isTFLiteVisionInitialized = false

    init {
        initializeTFLite()
    }

    private fun initializeTFLite() {
        TfLiteGpu.isGpuDelegateAvailable(context).onSuccessTask { gpuAvailable ->
            val optionsBuilder = TfLiteInitializationOptions.builder()
            if (gpuAvailable) {
                optionsBuilder.setEnableGpuDelegateSupport(true)
            }
            TfLiteVision.initialize(context, optionsBuilder.build())
        }.addOnSuccessListener {
            isTFLiteVisionInitialized = true
            objectDetectorRepository.initializeObjectDetector(context, modelName)
        }.addOnFailureListener { e ->
            val errorMessage = "TFLite initialization failed: ${e.message}"
        }
    }
}
```

```

        Log.e(TAG, errorMessage, e)
        detectorListener?.onError(errorMessage)
    }

}

fun isInitialized(): Boolean {
    return isTFLiteVisionInitialized
}

fun detectObject(image: ImageProxy) {
    if (!isTFLiteVisionInitialized) {
        val errorMessage = "TFLite is not initialized yet."
        Log.e(TAG, errorMessage)
        detectorListener?.onError(errorMessage)
        return
    }

    val objectDetector = objectDetectorRepository.getObjectDetector()
    if (objectDetector == null) {
        val errorMessage = "ObjectDetector is not initialized."
        Log.e(TAG, errorMessage)
        detectorListener?.onError(errorMessage)
        return
    }

    val imageProcessor = ImageProcessor.Builder()
        .add(Rot90Op(-image.imageInfo.rotationDegrees / 90))
        .add(NormalizeOp(-1.0f, 2.0f))
        .build()
}

```

```

    val tensorImage = imageProcessor.process(TensorImage.fromBitmap(toBitmap(image)))
}

var inferenceTime = SystemClock.uptimeMillis()
val results = objectDetector.detect(tensorImage)
inferenceTime = SystemClock.uptimeMillis() - inferenceTime

detectorListener?.onResults(
    results,
    inferenceTime,
    tensorImage.height,
    tensorImage.width
)
}

private fun toBitmap(image: ImageProxy): Bitmap {
    val bitmapBuffer = Bitmap.createBitmap(
        image.width,
        image.height,
        Bitmap.Config.ARGB_8888
    )
    image.use { bitmapBuffer.copyPixelsFromBuffer(image.planes[0].buffer) }
    image.close()
    return bitmapBuffer
}

interface DetectorListener {
    fun onError(error: String)
    fun onResults(
        results: MutableList<Detection>?,
        inferenceTime: Long,
    )
}

```

```
    imageHeight: Int,  
    imageWidth: Int  
)  
  
}  
  
companion object {  
    private const val TAG = "ObjectDetectorHelper"  
}  
}
```

#### ObjectDetectorRepository.kt

```
interface ObjectDetectorRepository {  
    fun initializeObjectDetector(context: Context, modelName: String)  
    fun getObjectDetector(): ObjectDetector?  
}
```

#### ObjectDetectorRepositoryImpl.kt

```
class ObjectDetectorRepositoryImpl : ObjectDetectorRepository {  
  
    private var objectDetector: ObjectDetector? = null  
  
    override fun initializeObjectDetector(context: Context, modelName: String) {  
        val optionBuilder = ObjectDetector.ObjectDetectorOptions.builder()  
            .setScoreThreshold(0.7f)  
            .setMaxResults(2)  
        val baseOptionBuilder = BaseOptions.builder()  
        when {  
            CompatibilityList().isDelegateSupportedOnThisDevice ->  
                baseOptionBuilder.useGpu()  
        }  
        objectDetector = ObjectDetector.create(context, modelName, optionBuilder, baseOptionBuilder)  
    }  
    ...  
}
```

```

Build.VERSION.SDK_INT      >=      Build.VERSION_CODES.O_MR1    ->
baseOptionBuilder.useNnapi()
else -> baseOptionBuilder.setNumThreads(4)
}
optionBuilder.setBaseOptions(baseOptionBuilder.build())

try {
    objectDetector = ObjectDetector.createFromFileAndOptions(
        context,
        modelName,
        optionBuilder.build()
    )
} catch (e: IllegalStateException) {
    val errorMessage = "ObjectDetector initialization failed: ${e.message}"
    Log.e("ObjectDetectorRepository", errorMessage, e)
}
}

override fun getObjectDetector(): ObjectDetector? {
    return objectDetector
}
}

```

### GoogleAuthClient.kt

```

class GoogleAuthClient(
    private val context: Context,
    val oneTapClient: SignInClient
) {
    private val auth: FirebaseAuth = FirebaseAuth.getInstance()
}

```

```

suspend fun signIn(): IntentSender? {
    val result = try {
        oneTapClient.beginSignIn(buildSignInRequest()).await()
    } catch (e: Exception) {
        Log.e("GoogleAuthClient", "Begin sign-in failed: ${e.localizedMessage}")
        if (e is CancellationException) throw e
        null
    }
    return result?.pendingIntent?.intentSender
}

suspend fun signInWithIntent(intent: Intent): SignInResult {
    val credential = try {
        oneTapClient.getSignInCredentialFromIntent(intent)
    } catch (e: ApiException) {
        Log.e("GoogleAuthClient", "Error getting sign-in credential: ${e.localizedMessage}")
        return SignInResult(null, e.localizedMessage)
    }

    val googleIdToken = credential.googleIdToken
    val googleCredentials = GoogleAuthProvider.getCredential(googleIdToken, null)
    return try {
        val user = auth.signInWithCredential(googleCredentials).await().user
        SignInResult(
            data = user?.run {
                UserData(
                    userId = uid,
                    username = displayName,
                    profilePictureUrl = photoUrl?.toString()
                )
            }
        )
    } catch (e: ApiException) {
        Log.e("GoogleAuthClient", "Error signing in with credential: ${e.localizedMessage}")
        return SignInResult(null, e.localizedMessage)
    }
}

```

```

    },
    errorMessage = null
)
} catch (e: Exception) {
    Log.e("GoogleAuthClient", "Sign-in with credential failed: ${e.localizedMessage}")
    if (e is CancellationException) throw e
    SignInResult(
        data = null,
        errorMessage = e.message
    )
}
}

suspend fun signOut() {
    try {
        oneTapClient.signOut().await()
        auth.signOut()
    } catch (e: Exception) {
        Log.e("GoogleAuthClient", "Sign-out failed: ${e.localizedMessage}")
        if (e is CancellationException) throw e
    }
}

fun getSignedInUser(): UserData? = auth.currentUser?.run {
    UserData(
        userId = uid,
        username = displayName,
        profilePictureUrl = photoUrl?.toString()
    )
}

```

```
private fun buildSignInRequest(): BeginSignInRequest {
    return BeginSignInRequest.builder()
        .setGoogleIdTokenRequestOptions(
            BeginSignInRequest.GoogleIdTokenRequestOptions.builder()
                .setSupported(true)
                .setServerClientId(context.getString(R.string.web_client_id))
                .setFilterByAuthorizedAccounts(false)
                .build()
        )
        .build()
}
```

### SignInResult.kt

```
data class SignInResult(
    val data: UserData?,
    val errorMessage: String?
)
```

### SignInState.kt

```
data class SignInState(
    val signInSuccessful: Boolean = false,
    val signInError: String? = null,
    val signedInUsername: String? = null
)
```

### UserData.kt

```
data class UserData(
```

```
    val userId: String?,  
    val username: String?,  
    val profilePictureUrl: String?  
)  
}
```

### CameraActivity.kt

```
class CameraActivity : AppCompatActivity() {  
    private lateinit var binding: ActivityCameraBinding  
    private var cameraSelector: CameraSelector =  
        CameraSelector.DEFAULT_BACK_CAMERA  
    private lateinit var objectDetectorHelper: ObjectDetectorHelper  
    private lateinit var cameraProvider: ProcessCameraProvider  
    private val modelName = "ssd_mobilenet_fpnLite_metadata.tflite"  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
  
        binding = ActivityCameraBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
  
        objectDetectorHelper = createObjectDetectorHelper()  
    }  
  
    private fun createObjectDetectorHelper(): ObjectDetectorHelper {  
        return ObjectDetectorHelper(  
            context = this,  
            modelName = modelName,  
            objectDetectorRepository = ObjectDetectorRepositoryImpl(),  
            detectorListener = object : ObjectDetectorHelper.DetectorListener {  
                override fun onError(error: String) {  
                    // Handle error  
                }  
            }  
        )  
    }  
}
```

```
    runOnUiThread {
        Toast.makeText(
            this@CameraActivity,
            error,
            Toast.LENGTH_SHORT
        ).show()
    }
}

override fun onResults(
    results: MutableList<Detection>?,
    inferenceTime: Long,
    imageHeight: Int,
    imageWidth: Int
) {
    runOnUiThread {
        handleDetectionResults(results, inferenceTime, imageHeight, imageWidth)
    }
}
}

private fun handleDetectionResults(
    results: MutableList<Detection>?,
    inferenceTime: Long,
    imageHeight: Int,
    imageWidth: Int
) {
    results?.let {
        if (it.isNotEmpty() && it[0].categories.isNotEmpty()) {
```

```
binding.overlay.setResults(results, imageHeight, imageWidth)

val builder = StringBuilder()
for (result in results) {
    val displayResult =
        "${result.categories[0].label} " + NumberFormat.getPercentInstance()
            .format(result.categories[0].score).trim()
    builder.append("$displayResult \n")
}

binding.tvResult.text = builder.toString()
binding.tvInferenceTime.text = "$inferenceTime ms"
} else {
    clearDetectionUI()
}
?: clearDetectionUI()

binding.overlay.invalidate()
}

private fun clearDetectionUI() {
    binding.overlay.clear()
    binding.tvResult.text = ""
    binding.tvInferenceTime.text = ""
}

public override fun onResume() {
    super.onResume()
    hideSystemUI()
    val cameraProviderFuture = ProcessCameraProvider.getInstance(this)
    cameraProviderFuture.addListener({
```

```

    cameraProvider = cameraProviderFuture.get()
    startCamera()
}, ContextCompat.getMainExecutor(this))

}

private fun startCamera() {
    binding.viewFinder.implementationMode =
PreviewView.ImplementationMode.COMPATIBLE
    val resolutionSelector = ResolutionSelector.Builder()

.setAspectRatioStrategy(AspectRatioStrategy.RATIO_16_9_FALLBACK_AUTO_STRATEGY)
.build()

    val imageAnalyzer = ImageAnalysis.Builder()
        .setResolutionSelector(resolutionSelector)
        .setTargetRotation(binding.viewFinder.display.rotation)
        .setBackpressureStrategy(ImageAnalysis.STRATEGY_KEEP_ONLY_LATEST)
        .setOutputImageFormat(ImageAnalysis.OUTPUT_IMAGE_FORMAT_RGBA_8888)
        .build()

    val preview = Preview.Builder().build().also {
        it.setSurfaceProvider(binding.viewFinder.surfaceProvider)
    }

    imageAnalyzer.setAnalyzer(Executors.newSingleThreadExecutor()) { image ->
        if (objectDetectorHelper.isInitialized()) {
            objectDetectorHelper.detectObject(image)
        } else {
            Log.e(TAG, "TFLite is not initialized yet.")
        }
    }
}

```

```
try {
    cameraProvider.unbindAll()
    cameraProvider.bindToLifecycle(
        this, cameraSelector, preview, imageAnalyzer
    )
} catch (exc: Exception) {
    Toast.makeText(
        this@CameraActivity, "Failed to start camera.", Toast.LENGTH_SHORT
    ).show()
    Log.e(TAG, "startCamera: ${exc.message}")
}

private fun hideSystemUI() {
    @Suppress("DEPRECATION") if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R)
    {
        window.insetsController?.hide(WindowInsets.Type.statusBars())
    } else {
        window.setFlags(
            WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN
        )
    }
    supportActionBar?.hide()
}

companion object {
    private const val TAG = "CameraActivity"
}
```

### MainActivity.kt

```
class MainActivity : ComponentActivity() {

    private val onboardingViewModel: OnboardingViewModel by viewModels()
    private val signInViewModel: SignInViewModel by viewModels()

    private val REQ_ONE_TAP = 2

    private val googleAuthClient by lazy {
        GoogleAuthClient(
            this, oneTapClient = Identity.getSignInClient(this)
        )
    }

    private val signInLauncher = registerForActivityResult(
        ActivityResultContracts.StartIntentSenderForResult()
    ) { result ->
        if (result.resultCode == RESULT_OK) {
            lifecycleScope.launch {
                val signInResult = googleAuthClient.signInWithIntent(
                    intent = result.data ?: return@launch
                )
                signInViewModel.onSignInResult(signInResult)
            }
        }
    }

    @OptIn(KoinExperimentalAPI::class)
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
```

```

    val startDestination = googleAuthClient.getSignedInUser()?.let {
        Screen.Dashboard.route
    } ?: Screen.Onboarding.route

    setContent {
        KoinAndroidContext {
            CekCabaiTheme {
                SetupNavHost(
                    onboardingViewModel = onboardingViewModel,
                    signInViewModel = signInViewModel,
                    googleAuthClient = googleAuthClient,
                    signInRequestLauncher = signInLauncher,
                    startDestination = startDestination
                )
            }
        }
    }
}

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)

    if (requestCode == REQ_ONE_TAP) {
        try {
            val credential = googleAuthClient.oneTapClient.getSignInCredentialFromIntent(data)
            val idToken = credential.googleIdToken
            when {
                idToken != null -> {
                    Log.d("MainActivity", "Got ID token.")
                }
            }
        }
    }
}

```

```
        else -> {
            Log.d("MainActivity", "No ID token!")
        }
    }
} catch (e: ApiException) {
    Log.e("MainActivity", "Error getting sign-in credential: ${e.localizedMessage}")
}
}
}
}
```

#### AuthButton.kt

```
@Composable
fun AuthButton(isLoginScreen: Boolean) {
    Button(
        onClick = {},
        modifier = Modifier
            .fillMaxWidth()
            .padding(vertical = 2.dp),
        colors = ButtonDefaults.buttonColors(containerColor = Color(0xFF4CAF50))
    ) {
        Text(text = if (isLoginScreen) "Masuk" else "Daftar", color = Color.White)
    }
}
```

#### AuthCard.kt

```
@Composable
fun AuthCard(
```

```

isLoginScreen: Boolean,
navController: NavController,
onGoogleLoginClick: () -> Unit
) {
    val scrollState = rememberScrollState()
    val cardHeight = 400.dp

    ElevatedCard(
        modifier = Modifier
            .fillMaxWidth()
            .padding(top = 12.dp)
            .height(cardHeight)
            .then(if (!isLoginScreen) Modifier.verticalScroll(scrollState) else Modifier),
        shape = RoundedCornerShape(topStart = 32.dp, topEnd = 32.dp)
    ) {
        Box(
            modifier = Modifier
                .fillMaxWidth()
                .padding(top = 32.dp, start = 32.dp, end = 32.dp)
        ) {
            Column {
                AuthHeaderText(isLoginScreen)
                AuthField(isLoginScreen)
                if (!isLoginScreen) {
                    AuthTermsAndConditions()
                }
                AuthButton(isLoginScreen)
                OrDivider()
                GoogleLoginButton(onGoogleLoginClick)
                SwitchAuthPrompt(isLoginScreen, navController)
            }
        }
    }
}

```

```
    }
}
}
```

### AuthField.kt

```
@Composable
```

```
fun AuthField(isLoginScreen: Boolean) {
    if (!isLoginScreen) {
        AuthLabeledTextField(
            label = "Nama Lengkap",
            value = "Fulan",
            onValueChange = {}
        )
        AuthLabeledTextField(
            label = "Email/Username",
            value = "fulan@gmail.com",
            onValueChange = {}
        )
        AuthLabeledTextField(
            label = "Password",
            value = "*****",
            onValueChange = {},
            trailingIcon = {
                Icon(
                    imageVector = Icons.Default.Lock,
                    contentDescription = null
                )
            }
        )
    }
}
```

### AuthHeaderText.kt

```
@Composable
fun AuthHeaderText(isLoginScreen: Boolean) {
    Text(
        text = if (isLoginScreen) "Login" else "Daftar",
        fontWeight = FontWeight.Bold,
        fontSize = 20.sp,
        color = Color(0xFF4CAF50)
    )
}
```

### AuthLabeledTextField.kt

```
@Composable
fun AuthLabeledTextField(
    label: String,
    value: String,
    onValueChange: (String) -> Unit,
    trailingIcon: @Composable () -> Unit)? = null
) {
    Column(
        modifier = Modifier
            .fillMaxWidth()
            .padding(top = 8.dp)
    ) {
        Text(
            text = label,
            color = Color.Gray,
        )
        TextField(

```

```

    value = value,
    onValueChange = onValueChange,
    modifier = Modifier
        .fillMaxWidth(),
    trailingIcon = trailingIcon,
    textStyle = LocalTextStyle.current.copy(fontSize = 14.sp)
)
}
}

```

### AuthLogoSection.kt

```

@Composable
fun AuthLogoSection() {
    Column(
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        Image(
            painter = painterResource(id = R.drawable.plant),
            contentDescription = null,
            modifier = Modifier.size(120.dp)
        )
        Text(
            text = "CheckCabe",
            modifier = Modifier.padding(top = 24.dp),
            fontFamily = FontFamily.SansSerif,
            fontSize = 24.sp,
            fontStyle = FontStyle.Italic,
            fontWeight = FontWeight.ExtraBold,
            lineHeight = 44.sp,

```

```
    textAlign = TextAlign.Center,  
    color = Color.White  
)  
}  
}
```

### AuthOrDivider.kt

```
@Composable  
fun OrDivider() {  
    Row(  
        modifier = Modifier  
            .fillMaxWidth()  
            .padding(vertical = 2.dp),  
        verticalAlignment = Alignment.CenterVertically  
    ) {  
        Divider(modifier = Modifier.weight(1f))  
        Text(  
            text = "atau",  
            modifier = Modifier.padding(horizontal = 8.dp),  
            color = Color.Gray  
        )  
        Divider(modifier = Modifier.weight(1f))  
    }  
}
```

### AuthTermsAndConditions.kt

```
@Composable  
fun AuthTermsAndConditions() {  
    var checked by remember { mutableStateOf(false) }
```

```

Row(
    modifier = Modifier
        .fillMaxWidth()
        .padding(vertical = 2.dp),
    verticalAlignment = Alignment.CenterVertically
) {
    Checkbox(
        checked = checked,
        onCheckedChange = { checked = it }
    )
    Text(
        text = "Saya setuju dengan syarat dan ketentuan yang berlaku",
        modifier = Modifier.padding(start = 2.dp),
        color = Color.Gray,
        fontSize = 12.sp
    )
}
}

```

### BottomAppBarComponent.kt

```

@Composable
fun BottomAppBarComponent(navController: NavController) {
    val navBackStackEntry by navController.currentBackStackEntryAsState()
    val currentRoute = navBackStackEntry?.destination?.route

    Surface(
        modifier = Modifier
            .fillMaxWidth()
            .height(120.dp),
        color = GreenTheme,
    )
}

```

```
    shape = RoundedCornerShape(topStart = 24.dp, topEnd = 24.dp)

) {

Row(
    modifier = Modifier
        .fillMaxWidth()
        .background(Color(0xFF48A14D)),
    horizontalArrangement = Arrangement.SpaceEvenly,
    verticalAlignment = Alignment.CenterVertically
) {

BottomAppBarItem(
    icon = Icons.Default.Home,
    label = "Home",
    onClick = { navController.navigate(Screen.Dashboard.route) },
    isActive = currentRoute == Screen.Dashboard.route
)

BottomAppBarItem(
    icon = Icons.Filled.List,
    label = "Scan History",
    onClick = { navController.navigate(Screen.History.route) },
    isActive = currentRoute == Screen.History.route ||
        currentRoute == Screen.Analysis.route ||
        currentRoute == Screen.Recommendation.route
)

BottomAppBarItem(
    icon = Icons.Default.AccountCircle,
    label = "Akun",
    onClick = { navController.navigate(Screen.Profile.route) },
    isActive = currentRoute == Screen.Profile.route
)
}
}
```

```
}
```

### DotIndicators.kt

```
@OptIn(ExperimentalFoundationApi::class)
@Composable
fun DotIndicators(pagerState: PagerState) {
    Row(
        modifier = Modifier
            .padding(16.dp)
            .fillMaxWidth(),
        horizontalArrangement = Arrangement.Center
    ) {
        repeat(pagerState.pageCount) { index ->
            Box(
                modifier = Modifier
                    .padding(horizontal = 4.dp)
                    .size(12.dp)
                    .clip(CircleShape)
                    .background(if (pagerState.currentPage == index) Color.White else Color.Gray)
            )
        }
    }
}
```

### GoogleButton.kt

```
@Composable
fun GoogleLoginButton(
    onGoogleLoginClick: () -> Unit
){
```

```
OutlinedButton(  
    onClick = onGoogleLoginClick,  
    modifier = Modifier  
        .fillMaxWidth()  
        .height(40.dp),  
    shape = RoundedCornerShape(8.dp)  
) {  
    Icon(  
        painter = painterResource(id = R.drawable.ic_google),  
        contentDescription = null,  
        tint = Color.Unspecified,  
        modifier = Modifier.size(24.dp)  
    )  
    Text(  
        text = "Masuk dengan Google",  
        modifier = Modifier.padding(start = 8.dp),  
        color = Color.Gray  
    )  
}  
}
```

### NavigateToLoginButton.kt

```
@Composable  
fun NavigateToLoginButton(modifier: Modifier, onClick: () -> Unit) {  
    Button(  
        onClick = onClick,  
        modifier = modifier,  
        shape = RoundedCornerShape(8.dp)  
) {  
    Text(text = "Masuk")  
}
```

```
    }  
}
```

### PageContent.kt

```
data class PageContentData(  
    val imageResId: Int,  
    val text: String,  
    val description: String? = null  
)  
  
@Composable  
fun PageContent(data: PageContentData) {  
    Column(  
        modifier = Modifier  
            .fillMaxSize(),  
        verticalArrangement = Arrangement.Center,  
        horizontalAlignment = Alignment.CenterHorizontally  
    ) {  
        Image(  
            painter = painterResource(id = data.imageResId),  
            contentDescription = "Greeting Image",  
            modifier = Modifier.size(200.dp)  
        )  
        Text(  
            text = data.text,  
            modifier = Modifier.padding(top = 24.dp),  
            fontFamily = FontFamily.SansSerif,  
            fontSize = 24.sp,  
            fontStyle = FontStyle.Italic,  
            fontWeight = FontWeight.ExtraBold,  
        )  
    }  
}
```

```
        lineHeight = 44.sp,  
        textAlign = TextAlign.Center,  
        color = Color.White  
    )  
  
    data.description?.let { description ->  
  
        Text(  
            text = description,  
            modifier = Modifier.padding(horizontal = 64.dp, vertical = 8.dp),  
            fontSize = 12.sp,  
            fontWeight = FontWeight.Bold,  
            lineHeight = 24.sp,  
            textAlign = TextAlign.Center,  
            color = Color.White  
        )  
    }  
}  
}
```

### SwitchAuthPrompt.kt

```
@Composable  
fun SwitchAuthPrompt(isLoginScreen: Boolean, navController: NavController) {  
  
    val text = if (isLoginScreen) "Belum punya akun? Daftar" else "Sudah punya akun?  
Masuk"  
    val route = if (isLoginScreen) Screen.Register.route else Screen.Login.route  
  
    Box(modifier = Modifier.fillMaxWidth()) {  
        Text(  
            text = text,  
            modifier = Modifier
```

```
.align(Alignment.Center)
.padding(vertical = 16.dp)
.clickable { navController.navigate(route) },
color = Color(0xFF4CAF50),
fontWeight = FontWeight.Bold
)
}
}
```

### TopAppBarComponent.kt

```
@Composable
fun TopAppBarComponent(userName: String) {
    Box(
        modifier = Modifier
            .fillMaxWidth()
            .height(80.dp)
            .clip(RoundedCornerShape(bottomStart = 48.dp))
            .background(GreenTheme)
            .padding(horizontal = 16.dp, vertical = 24.dp)
    ) {
        Column(
            modifier = Modifier.align(Alignment.CenterStart)
        ) {
            Text(
                text = "Hello, $userName!",
                color = Color.White,
                fontSize = 24.sp,
                fontWeight = FontWeight.Bold,
                modifier = Modifier.padding(start = 48.dp)
            )
        }
    }
}
```

```
    }  
}  
}
```

### PresentationModule.kt

```
val presentationModule = module{  
    viewModel { SignInViewModel() }  
    viewModel { OnboardingViewModel() }  
}
```

### BottomAppBarItem.kt

```
@Composable  
fun BottomAppBarItem(  
    icon: Androidx.compose.ui.graphics.vector.ImageVector,  
    label: String,  
    onClick: () -> Unit,  
    isActive: Boolean = false  
) {  
    Column(  
        modifier = Modifier.wrapContentSize(),  
        horizontalAlignment = Alignment.CenterHorizontally  
) {  
        IconButton(onClick = onClick) {  
            Icon(  
                imageVector = icon,  
                contentDescription = label,  
                tint = if (isActive) Color.White else Color.Gray,  
                modifier = Modifier.size(30.dp)  
        )  
    }
```

```
    }

    Text(
        text = label,
        fontSize = 12.sp,
        color = if (isActive) Color.White else Color.White,
        fontWeight = if (isActive) FontWeight.Bold else FontWeight.Normal
    )
}

}
```

### Screen.kt

```
sealed class Screen(val route: String) {

    data object Login : Screen("login")
    data object Register : Screen("register")
    data object Camera : Screen("camera")
    data object Dashboard : Screen("dashboard")
    data object History : Screen("history")
    data object Profile : Screen("profile")
    data object Analysis : Screen("analysis")
    data object Onboarding : Screen("onboarding")
    data object Recommendation : Screen("recommendation")
}
```

### SetupNavHost.kt

```
@Composable
fun SetupNavHost(
    onboardingViewModel: OnboardingViewModel,
    signInViewModel: SignInViewModel,
    googleAuthClient: GoogleAuthClient,
```

```

signInRequestLauncher: ActivityResultLauncher<IntentSenderRequest>,
startDestination: String
) {
    val navController = rememberNavController()

    NavHost(navController = navController, startDestination = startDestination) {
        composable(Screen.Onboarding.route) {
            val signInState by signInViewModel.state.collectAsState()
            val context = LocalContext.current

            SideEffect {
                if (signInState.isSignInSuccessful) {
                    Toast.makeText(context, "Login is Successful", Toast.LENGTH_LONG).show()
                    navController.navigate(Screen.Dashboard.route) {
                        popUpTo(Screen.Onboarding.route) { inclusive = true }
                    }
                } else if (signInState.signInError != null) {
                    Toast.makeText(
                        context,
                        "Login failed: ${signInState.signInError}",
                        Toast.LENGTH_LONG
                    ).show()
                }
            }
        }
        MainScreen(onboardingViewModel, navController)
    }
    composable(Screen.Login.route) {
        AuthScreen(
            isLoginScreen = true,
            navController = navController,

```

```
        googleAuthClient = googleAuthClient,
        signInLauncher = signInRequestLauncher
    )
}

composable(Screen.Register.route) {
    AuthScreen(
        isLoginScreen = false,
        navController = navController,
        googleAuthClient = googleAuthClient,
        signInLauncher = signInRequestLauncher
    )
}

composable(Screen.Dashboard.route) {
    DashboardScreen(navController)
}

composable(Screen.Analysis.route) {
    AnalysisScreen(navController)
}

composable(Screen.History.route) {
    HistoryScreen(
        navController = navController,
        historyItems = listOf()
    )
}

composable(Screen.Profile.route) {
    ProfileScreen(navController)
}

composable(Screen.Recommendation.route) {
    RecommendationScreen(navController)
}
}
```

```
}
```

### AnalysisScreen.kt

```
@Composable
fun AnalysisScreen(navController: NavController) {
    Scaffold(
        topBar = {
            Box(
                modifier = Modifier
                    .fillMaxWidth()
                    .height(80.dp)
                    .clip(RoundedCornerShape(bottomStart = 48.dp))
                    .background(GreenTheme)
                    .padding(horizontal = 16.dp, vertical = 24.dp)
            )
        },
        Column(
            modifier = Modifier.align(Alignment.CenterStart)
        ) {
            Text(
                text = "Analisis Penyakit",
                color = Color.White,
                fontSize = 24.sp,
                fontWeight = FontWeight.Bold,
                modifier = Modifier.padding(start = 48.dp)
            )
        }
    ),
    bottomBar = { BottomAppBarComponent(navController) }
) { padding ->
```

```

Column(
    modifier = Modifier
        .fillMaxSize()
        .padding(horizontal = 16.dp),
    horizontalAlignment = Alignment.CenterHorizontally,
    verticalArrangement = Arrangement.Center
) {
    Image(
        painter = painterResource(id = R.drawable.ic_chilli),
        contentDescription = "Image of the chili plant",
        modifier = Modifier.fillMaxWidth()
    )
    Spacer(modifier = Modifier.height(16.dp))
    Surface(
        modifier = Modifier
            .padding(16.dp)
            .fillMaxWidth()
            .height(150.dp),
        shape = RoundedCornerShape(8.dp),
        color = Color(0xFFFF5F5F5)
    ) {
        Row(
            modifier = Modifier.padding(horizontal = 16.dp),
            verticalAlignment = Alignment.CenterVertically,
            horizontalArrangement = Arrangement.SpaceBetween
        ) {
            Column {
                Text(
                    text = "Hasil Analisis",
                    style = MaterialTheme.typography.headlineSmall.copy(fontWeight =
FontWeight.Bold),

```

```

        color = Color.Black
    )

Spacer(modifier = Modifier.height(8.dp))

Text(
    text = "Antraknosa",
    style = MaterialTheme.typography.headlineSmall,
    color = Color.Black
)

Spacer(modifier = Modifier.height(8.dp))

Text(
    text = "Tingkat Keyakinan :",
    style = MaterialTheme.typography.bodyLarge,
    color = Color.Black
)

Text(
    text = "90%",
    style = MaterialTheme.typography.headlineSmall.copy(fontWeight =
FontWeight.Bold),
    color = Color.Black
)

}

Box(
    modifier = Modifier
        .size(80.dp)
        .background(Color.Transparent)
) {
    Image(
        painter = rememberAsyncImagePainter(R.drawable.ic_plant_scan),
        contentDescription = null,
        modifier = Modifier.fillMaxSize(),
        contentScale = ContentScale.Fit
    )
}

```

```
        )
    }
}

Spacer(modifier = Modifier.height(16.dp))

Row(
    modifier = Modifier
        .fillMaxWidth(),
    horizontalArrangement = Arrangement.SpaceAround
) {
    Button(
        onClick = { navController.navigate(Screen.Camera.route) },
        modifier = Modifier.weight(1f),
        shape = RoundedCornerShape(8.dp)
    ) {
        Text(text = "Pindai Ulang")
    }

    Spacer(modifier = Modifier.width(16.dp))

    Button(
        onClick = { navController.navigate(Screen.Recommendation.route) },
        modifier = Modifier.weight(1f),
        shape = RoundedCornerShape(8.dp)
    ) {
        Text(
            text = "Rekomendasi Penanganan",
            textAlign = TextAlign.Center,
            fontSize = 10.sp
        )
    }
}
```

```
        }
    }
}

@Preview(showBackground = true)
@Composable
fun AnalysisScreenPreview() {
    val navController = rememberNavController()
    AnalysisScreen(navController)
}
```

### AuthScreen.kt

```
@Composable
fun AuthScreen(
    isLoginScreen: Boolean,
    navController: NavController,
    googleAuthClient: GoogleAuthClient,
    signInLauncher: ActivityResultLauncher<IntentSenderRequest>,
    modifier: Modifier = Modifier
) {
    val coroutineScope = rememberCoroutineScope()

    val onGoogleLoginClick: () -> Unit = {
        coroutineScope.launch {
            val intentSender = googleAuthClient.signIn()
            if (intentSender != null) {
                val intentSenderRequest = IntentSenderRequest.Builder(intentSender).build()
                signInLauncher.launch(intentSenderRequest)
            }
        }
    }
}
```

```

        }

    }

Box(
    modifier = modifier
        .fillMaxSize()
        .background(GreenTheme)
) {
    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Bottom,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        AuthLogoSection()
        AuthCard(
            isLoginScreen,
            navController,
            onGoogleLoginClick
        )
    }
}
}

```

### ***DashboardScreen.kt***

```

@Composable
fun DashboardScreen(navController: NavController) {
    val signInViewModel: SignInViewModel = viewModel()
    val context = LocalContext.current

    var selectedImageUri by remember { mutableStateOf<Uri?>(null) }

```

```

val launchCameraLauncher = rememberLauncherForActivityResult(
    contract = ActivityResultContracts.StartActivityForResult()
) { result ->
    if (result.resultCode == Activity.RESULT_OK) {
        val uriString = result.data?.getStringExtra("captured_image_uri")
        if (uriString != null) {
            selectedImageUri = Uri.parse(uriString)
            Log.d("Camera Result", "Captured URI: $selectedImageUri")
        }
    }
}

val requestPermissionLauncher = rememberLauncherForActivityResult(
    contract = ActivityResultContracts.RequestPermission()
) { isGranted ->
    if (isGranted) {
        launchCameraLauncher.launch(Intent(context, CameraActivity::class.java))
    }
}

val galleryLauncher = rememberLauncherForActivityResult(
    contract = ActivityResultContracts.PickVisualMedia()
) { uri ->
    if (uri != null) {
        selectedImageUri = uri
        Log.d("Photo Picker", "Selected URI: $uri")
    } else {
        Log.d("Photo Picker", "No media selected")
    }
}

```

```
val state by signInViewModel.state.collectAsState()

LaunchedEffect(Unit) {
    val permissionGranted = ContextCompat.checkSelfPermission(
        context,
        Manifest.permission.CAMERA
    ) == PackageManager.PERMISSION_GRANTED

    if (!permissionGranted) {
        requestPermissionLauncher.launch(Manifest.permission.CAMERA)
    }
}

Scaffold(
    topBar = { TopAppBarComponent(userName = state.signedInUsername ?: "Guest") },
    bottomBar = { BottomAppBarComponent(navController) }
) { padding ->
    DashboardContent(
        padding = padding,
        selectedImageUri = selectedImageUri,
        onCameraClick = {
            val permissionGranted = ContextCompat.checkSelfPermission(
                context,
                Manifest.permission.CAMERA
            ) == PackageManager.PERMISSION_GRANTED

            if (permissionGranted) {
                launchCameraLauncher.launch(Intent(context, CameraActivity::class.java))
            } else {
                requestPermissionLauncher.launch(Manifest.permission.CAMERA)
            }
        }
    )
}
```

```
        },
        onGalleryClick = {
            galleryLauncher.launch(PickVisualMediaRequest(ActivityResultContracts.PickVisualMedia.a.ImageOnly))
        },
        onAnalysisClick = { navController.navigate(Screen.Analysis.route) }
    )
}
}

@Composable
fun DashboardContent(
    padding: PaddingValues,
    selectedImageUri: Uri?,
    onCameraClick: () -> Unit,
    onGalleryClick: () -> Unit,
    onAnalysisClick: () -> Unit
) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(padding)
            .background(Color.White)
    ) {
        PlantCheckCard(onCameraClick)
        PlantImageCard(selectedImageUri)
        Spacer(modifier = Modifier.height(16.dp))
        UploadButton(onGalleryClick)
        Spacer(modifier = Modifier.weight(1f))
    }
}
```

```
    AnalysisButton(onAnalysisClick)
    Spacer(modifier = Modifier.weight(1f))
}

}

@Composable
fun PlantCheckCard(onCameraClick: () -> Unit) {
    Card(
        modifier = Modifier
            .fillMaxWidth()
            .padding(horizontal = 32.dp, vertical = 16.dp),
        elevation = CardDefaults.cardElevation(defaultElevation = 4.dp),
        shape = RoundedCornerShape(16.dp)
    ) {
        Row(
            modifier = Modifier
                .padding(16.dp)
                .fillMaxWidth(),
            verticalAlignment = Alignment.CenterVertically,
            horizontalArrangement = Arrangement.SpaceBetween
        ) {
            Column(modifier = Modifier.weight(1f)) {
                Text(
                    text = "Periksa Tanaman Anda",
                    fontSize = 20.sp,
                    fontWeight = FontWeight.Bold
                )
                Spacer(modifier = Modifier.height(8.dp))
                Button(
                    onClick = onCameraClick,
                    modifier = Modifier.fillMaxWidth(),

```

```

        shape = RoundedCornerShape(16.dp),
        colors = ButtonDefaults.buttonColors(containerColor = GreenTheme)

    ) {
        Text(
            text = "Mulai",
            color = Color.White,
            textAlign = TextAlign.Center
        )
    }
}

Image(
    painter = painterResource(id = R.drawable.ic_plant_scan),
    contentDescription = "Scan Plant",
    modifier = Modifier
        .size(100.dp)
        .padding(start = 8.dp)
)
}
}

}

@Composable
fun PlantImageCard(selectedImageUri: Uri?) {
    Card(
        modifier = Modifier
            .fillMaxWidth()
            .padding(horizontal = 32.dp),
        elevation = CardDefaults.cardElevation(defaultElevation = 4.dp),
        shape = RoundedCornerShape(16.dp)
    ) {
        Image(

```

```

painter = if (selectedImageUri != null) {
    rememberAsyncImagePainter(selectedImageUri)
} else {
    painterResource(
        id = R.drawable.ic_chilli
    )
},
contentDescription = "Chilli Plant",
modifier = Modifier
    .fillMaxWidth()
    .height(200.dp)
)
}
}

@Composable
fun UploadButton(onGalleryClick: () -> Unit) {
    Button(
        onClick = onGalleryClick,
        modifier = Modifier
            .fillMaxWidth()
            .padding(horizontal = 16.dp),
        shape = RoundedCornerShape(16.dp),
        colors = ButtonDefaults.buttonColors(containerColor = GreenTheme)
    ) {
        Text(
            text = "Upload via Gallery",
            color = Color.White,
            textAlign = TextAlign.Center
        )
    }
}

```

```
}
```

```
@Composable
```

```
fun AnalysisButton(onAnalysisClick: () -> Unit) {  
    Button(  
        onClick = onAnalysisClick,  
        modifier = Modifier  
            .fillMaxWidth()  
            .padding(horizontal = 16.dp),  
        shape = RoundedCornerShape(16.dp),  
        colors = ButtonDefaults.buttonColors(containerColor = GreenTheme)  
    ) {  
        Text(  
            text = "Analisis",  
            color = Color.White,  
            textAlign = TextAlign.Center  
        )  
    }  
}
```

```
@Preview(showBackground = true)
```

```
@Composable
```

```
fun DashboardScreenPreview() {  
    val navController = rememberNavController()  
    DashboardScreen(navController)  
}
```

```
HistoryItem.kt
```

```
data class HistoryItem(  
    val diseaseName: String? = null,
```

```
    val date: String? = null,  
    val imageUri: Uri? = null  
)
```

### HistoryItemCard.kt

```
@Composable  
fun HistoryItemCard(historyItem: HistoryItem) {  
    Card(  
        modifier = Modifier  
            .fillMaxWidth()  
            .padding(8.dp),  
        shape = RoundedCornerShape(12.dp),  
        elevation = CardDefaults.cardElevation(defaultElevation = 4.dp)  
    ) {  
        Row(  
            modifier = Modifier  
                .padding(16.dp)  
                .fillMaxWidth(),  
            verticalAlignment = Alignment.CenterVertically,  
            horizontalArrangement = Arrangement.Start  
        ) {  
            Image(  
                painter = if (historyItem.imageUri != null) {  
                    rememberAsyncImagePainter(model = historyItem.imageUri)  
                } else {  
                    painterResource(id = R.drawable.ic_chilli)  
                },  
                contentDescription = null,  
                modifier = Modifier  
                    .size(50.dp)
```

```
.clip(RoundedCornerShape(8.dp))  
)  
  
Spacer(modifier = Modifier.width(16.dp))  
  
Column {  
    Text(  
        text = historyItem.diseaseName ?: "Antrknosa",  
        fontWeight = FontWeight.Bold,  
        fontSize = 16.sp  
    )  
    Text(  
        text = historyItem.date ?: "29 Januari 2001",  
        fontSize = 14.sp,  
        color = Color.Gray  
    )  
}  
}  
}
```

### HistoryList.kt

```
@Composable  
fun HistoryList(historyItems: List<HistoryItem>) {  
    LazyColumn {  
        items(historyItems) { historyItem ->  
            HistoryItemCard(historyItem = historyItem)  
        }  
    }  
}
```

### HistoryScreen.kt

```
@Composable
fun HistoryScreen(navController: NavController, historyItems: List<HistoryItem>) {
    Scaffold(
        topBar = {
            Box(
                modifier = Modifier
                    .fillMaxWidth()
                    .height(80.dp)
                    .clip(RoundedCornerShape(bottomStart = 48.dp))
                    .background(GreenTheme)
                    .padding(horizontal = 16.dp, vertical = 24.dp)
            )
        },
        Column(
            modifier = Modifier.align(Alignment.CenterStart)
        ) {
            Text(
                text = "History Pemindaian",
                color = Color.White,
                fontSize = 24.sp,
                fontWeight = FontWeight.Bold,
                modifier = Modifier.padding(start = 48.dp)
            )
        }
    ),
    bottomBar = { BottomAppBarComponent(navController) }
) { padding ->
    Box(modifier = Modifier.padding(padding)) {
        HistoryList(historyItems = historyItems)
    }
}
```

```
    }  
}  
}
```

### OnboardingScreen.kt

```
@OptIn(ExperimentalFoundationApi::class)  
  
@Composable  
fun OnboardingScreen(viewModel: OnboardingViewModel, navController:  
NavController) {  
    val scope = rememberCoroutineScope()  
    val currentPage by viewModel.currentPage.collectAsState()  
    val pageCount by viewModel.pageCount.collectAsState()  
  
    val pagerState = rememberPagerState(  
        initialPage = currentPage,  
        pageCount = { pageCount }  
    )  
  
    Box(  
        modifier = Modifier  
            .fillMaxSize()  
            .background(  
                brush = Brush.linearGradient(  
                    colors = listOf(GreenStart, GreenEnd),  
                    start = Offset(0f, 0f),  
                    end = Offset(Float.POSITIVE_INFINITY, Float.POSITIVE_INFINITY)  
                )  
,  
        contentAlignment = Alignment.Center  
    ){
```

```

    PagerScreen(
        pagerState = pagerState,
        onPageChanged = { page ->
            scope.launch {
                viewModel.setCurrentPage(page)
            }
        }
    )
    NavigateToLoginButton(
        modifier = Modifier
            .padding(48.dp)
            .align(Alignment.BottomCenter),
        onClick = { navController.navigate(Screen.Login.route) }
    )
}
}
}

```

### PagerScreen.kt

```

@OptIn(ExperimentalFoundationApi::class)
@Composable
fun PagerScreen(pagerState: PagerState, onPageChanged: (Int) -> Unit) {
    Column(modifier = Modifier.fillMaxSize()) {
        HorizontalPager(
            state = pagerState,
            modifier = Modifier.weight(1f)
        ) { page ->
            onPageChanged(page)
            val pageContentData = when (page) {
                0 -> PageContentData(
                    imageResId = R.drawable.greet1,

```

```

        text = "Check Cabe"
    )

    1 -> PageContentData(
        imageResId = R.drawable.greet2,
        text = "Check Cabe",
        description = "Selamat datang di PeduliCabe! Bersama, kita jaga kebun
cabemu tetap sehat dan sejahtera"
    )

    else -> throw IllegalStateException("Unexpected page index: $page")
}

PageContent(pageContentData)
}

DotIndicators(pagerState = pagerState)
}

}

```

### **ProfileScreen.kt**

```

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ProfileScreen(navHostController: NavHostController) {
    Scaffold(
        topBar = {
            TopAppBar(
                navigationIcon = {
                    IconButton(onClick = { navHostController.popBackStack() }) {
                        Icon(
                            painter = painterResource(id = R.drawable.ic_google),
                            contentDescription = null,

```

```
        tint = Color.Unspecified,  
        modifier = Modifier.size(24.dp)  
    )  
}  
,  
title = {  
    Spacer(modifier = Modifier.width(0.dp))  
},  
)  
,  
bottomBar = { BottomAppBarComponent(navHostController) },  
content = { paddingValues ->  
    Column(  
        modifier = Modifier  
            .fillMaxSize()  
            .padding(paddingValues)  
            .padding(horizontal = 16.dp),  
        horizontalAlignment = Alignment.CenterHorizontally,  
        verticalArrangement = Arrangement.Top  
) {  
    Image(  
        painter = painterResource(id = R.drawable.greet1),  
        contentDescription = "Profile picture",  
        modifier = Modifier  
            .size(100.dp)  
            .clip(CircleShape)  
    )  
    Spacer(modifier = Modifier.height(8.dp))  
    Text(  
        text = "Michelle Putri",  
        fontWeight = FontWeight.Bold,  
    )  
}
```

```
fontSize = MaterialTheme.typography.headlineMedium.fontSize  
)  
  
Spacer(modifier = Modifier.height(16.dp))  
  
Card(  
    modifier = Modifier  
        .fillMaxWidth(),  
    shape = RoundedCornerShape(8.dp)  
) {  
    Column(  
        modifier = Modifier  
            .fillMaxWidth()  
            .padding(16.dp),  
        horizontalAlignment = Alignment.CenterHorizontally  
) {  
    Text(  
        text = "Profil",  
        fontWeight = FontWeight.Bold,  
        fontSize = MaterialTheme.typography.headlineSmall.fontSize  
    )  
    Spacer(modifier = Modifier.height(8.dp))  
    OutlinedTextField(  
        value = "Michelle Putri",  
        onValueChange = { /TODO/ },  
        label = { Text("Nama Lengkap") },  
        modifier = Modifier.fillMaxWidth()  
    )  
    Spacer(modifier = Modifier.height(8.dp))  
    OutlinedTextField(  
        value = "michelle@gmail.com",
```

```
    onValueChange = { /TODO/ },
    label = { Text("Email") },
    modifier = Modifier.fillMaxWidth()
)
Spacer(modifier = Modifier.height(16.dp))
Row(
    modifier = Modifier
        .fillMaxWidth(),
    horizontalArrangement = Arrangement.SpaceAround
) {
    Button(
        onClick = { /TODO/ },
        modifier = Modifier.weight(1f),
        shape = RoundedCornerShape(8.dp)
    ) {
        Text(text = "Edit")
    }
}

Spacer(modifier = Modifier.width(16.dp))

Button(
    onClick = { /TODO/ },
    modifier = Modifier.weight(1f),
    shape = RoundedCornerShape(8.dp),
    colors = ButtonDefaults.buttonColors(
        containerColor = Color.Red
    )
) {
    Text(text = "Keluar")
}
}
```

```
        }
    }
}
}

@Preview
@Composable
fun ProfileScreenPreview() {
    val navController = rememberNavController()
    ProfileScreen(navController)
}
```

### OnboardingViewModel.kt

```
class OnboardingViewModel : ViewModel() {
    private val _currentPage = MutableStateFlow(0)
    val currentPage: StateFlow<Int> = _currentPage

    private val _pageCount = MutableStateFlow(2)
    val pageCount: StateFlow<Int> = _pageCount

    private val _navigateToRegisterScreen = MutableStateFlow(false)
    val navigateToRegisterScreen: StateFlow<Boolean> = _navigateToRegisterScreen

    fun setCurrentPage(page: Int) {
        _currentPage.value = page
    }

    fun setPageCount(count: Int) {
```

```
_pageCount.value = count  
}  
  
fun navigateToRegisterScreen() {  
    _navigateToRegisterScreen.value = true  
}  
  
fun navigateToRegisterScreenComplete() {  
    _navigateToRegisterScreen.value = false  
}  
}
```

### SignInViewModel.kt

```
class SignInViewModel : ViewModel() {  
    private val _state = MutableStateFlow(SignInState())  
    val state = _state.asStateFlow()  
  
    fun onSignInResult(result: SignInResult) {  
        _state.update { currentState ->  
            currentState.copy(  
                isSignInSuccessful = result.data != null,  
                signInError = result.errorMessage,  
                signedInUsername = result.data?.username  
            )  
        }  
    }  
  
    fun resetState() {  
        _state.update { SignInState() }  
    }  
}
```

```
}
```

## OverlayView.kt

```
class OverlayView @JvmOverloads constructor(  
    context: Context,  
    attrs: AttributeSet? = null  
) : View(context, attrs) {  
  
    private var boxPaint = Paint()  
    private var textBackgroundPaint = Paint()  
    private var textPaint = Paint()  
  
    private var results: List<Detection> = LinkedList<Detection>()  
    private var scaleFactor: Float = 1f  
  
    private var bounds = Rect()  
  
    init {  
        initPaints()  
    }  
  
    private fun initPaints() {  
        boxPaint.color = ContextCompat.getColor(context, R.color.purple_700)  
        boxPaint.style = Paint.Style.STROKE  
        boxPaint.strokeWidth = 8f  
  
        textBackgroundPaint.color = Color.BLACK  
        textBackgroundPaint.style = Paint.Style.FILL  
        textBackgroundPaint.textSize = 50f  
    }  
}
```

```
textPaint.color = Color.WHITE
textPaint.style = Paint.Style.FILL
textPaint.setTextSize = 50f
}

fun setResults(
    detectionResults: MutableList<Detection>,
    imageHeight: Int,
    imageWidth: Int,
) {
    results = detectionResults

    scaleFactor = max(width * 1f / imageWidth, height * 1f / imageHeight)
}

override fun draw(canvas: Canvas) {
    super.draw(canvas)

    for (result in results) {
        val boundingBox = result.boundingBox

        val left = boundingBox.left * scaleFactor
        val top = boundingBox.top * scaleFactor
        val right = boundingBox.right * scaleFactor
        val bottom = boundingBox.bottom * scaleFactor

        val drawableRect = RectF(left, top, right, bottom)
        canvas.drawRect(drawableRect, boxPaint)

        val drawableText = "${result.categories[0].label} " +
            NumberFormat.getPercentInstance().format(result.categories[0].score)
```

```
    textBackgroundPaint.getTextBounds(drawableText, 0, drawableText.length,
bounds)

    val textWidth = bounds.width()
    val textHeight = bounds.height()
    canvas.drawRect(
        left,
        top,
        left + textWidth + BOUNDING_RECT_TEXT_PADDING,
        top + textHeight + BOUNDING_RECT_TEXT_PADDING,
        textBackgroundPaint
    )

    canvas.drawText(drawableText, left, top + bounds.height(), textPaint)
}

}

fun clear() {
    boxPaint.reset()
    textBackgroundPaint.reset()
    textPaint.reset()
    invalidate()
    initPaints()
}

companion object {
    private const val BOUNDING_RECT_TEXT_PADDING = 8
}
}
```

## B. User Manual



# USER MANUAL

*PRESENTED BY*  
**FUNDEASE**



CEKCABAI

**DAFTAR ISI**

## DAFTAR ISI

### Petunjuk penggunaan CekCabai

A. Menginstal Aplikasi CekCabai	3
B. Melakukan <i>Login</i> pada Aplikasi	3
C. Tampilan Awal CekCabai	4
Setelah <i>login</i> , aplikasi akan meminta persetujuan untuk isin ambil gambar dan merekam video. Klik “saat aplikasi digunakan”	
	5
D. Pindai penyakit	6
E. <i>Upload</i> via Galeri	6
F. Analisis penyakit	7
G. Rekomendasi penanganan	8
H. Riwayat pemeriksaan	9
I. Profil pengguna	9

## Petunjuk Penggunaan CekCabai

CekCabai adalah aplikasi yang dikembangkan oleh Tim FundEase untuk mendeteksi penyakit pada tanaman cabai. Aplikasi ini dapat digunakan tanpa koneksi internet dan memudahkan pengguna dalam mendeteksi penyakit kapan saja.

### A. Menginstall Aplikasi CekCabai

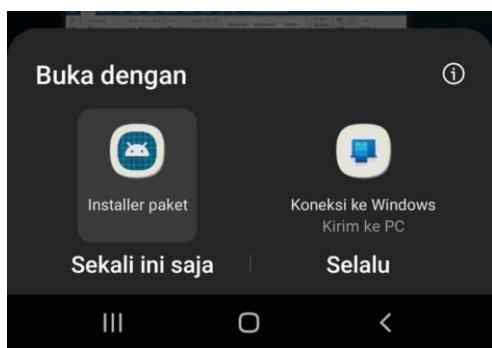
- Dapatkan aplikasi dari link Google Drive yang diberikan oleh tim kami.

([https://drive.google.com/drive/folders/1D7vZi4gmWVsViarjpgIgMU4mZ1AqgOrh?usp=drive\\_link](https://drive.google.com/drive/folders/1D7vZi4gmWVsViarjpgIgMU4mZ1AqgOrh?usp=drive_link))

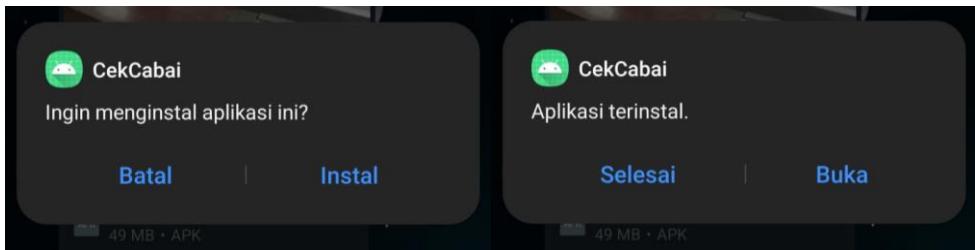
- Buka *file* unduhan dan ikuti petunjuk instalasi di layar.
- Pastikan perangkat Anda mengizinkan instalasi dari sumber tidak dikenal (bila diperlukan).



- Pilih *installer* paket



- Klik instal dan buka aplikasi yang sudah terinstal



## B. Melakukan *Login* pada Aplikasi

- Usap layar ke kiri.



- Klik "masuk" untuk melanjutkan.



- Jika belum memiliki akun, pilih opsi untuk mendaftar.

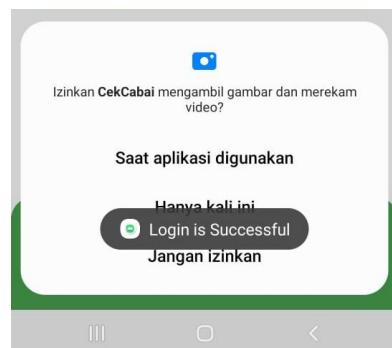


- Alternatif lain, klik "Masuk dengan Google" dan pilih akun yang akan anda gunakan untuk memakai aplikasi.



### C. Tampilan Awal CekCabai

Setelah *login*, aplikasi akan meminta persetujuan Anda untuk izin mengambil gambar dan merekam video. Klik “Saat aplikasi digunakan”.



Lalu Anda akan disambut dengan layar utama aplikasi CekCabai yang menampilkan beberapa menu penting.

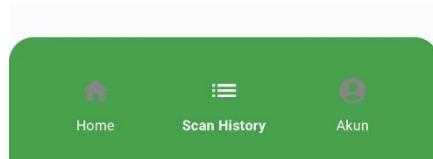
Menu yang tersedia:

- Pindai Penyakit dengan Kamera

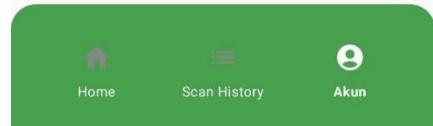


- Pindai Penyakit dengan *Upload Foto*

- Riwayat Pemeriksaan



- Profil Pengguna



#### D. Pindai Penyakit

Pilih menu "Mulai", untuk menu ini digunakan untuk memindai penyakit pada tanaman cabai.



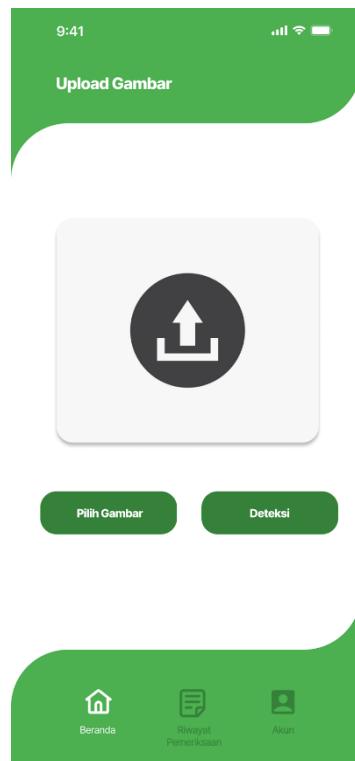
Pastikan kamera *smartphone* berfungsi dengan baik agar proses pemindaian lancar. Arahkan kamera ke cabai yang berpotensi terkena penyakit.



Klik “Ambil Gambar”, aplikasi akan menganalisa penyakit dari tanaman cabai dan menampilkan hasil deteksi.

#### E. *Upload via Galeri*

Selain melalui kamera pada aplikasi, Anda juga dapat mengunggah foto dari galeri untuk mencari tahu penyakit yang menyerang tanaman cabai. Klik “Pilih Gambar”, lalu pilih gambar tanaman cabai.



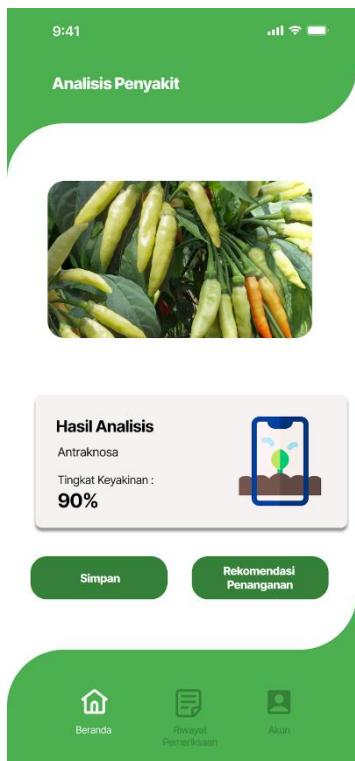
Jika pada kotak abu-abu sudah berubah sesuai dengan gambar yang telah diunggah.



Klik "Deteksi", aplikasi akan menganalisis penyakit apa yang telah menyerang tanaman cabai Anda.

#### F. Analisis Penyakit

Setelah diketahui penyakit pada cabai, *user* dapat mengetahui hasil analisis penyakit.



Dapat dilihat bahwa tanaman cabai yang sudah di *scan* terkena penyakit Antraknosa dengan tingkat keyakinan sebesar 90%.

Hasil pemeriksaan dapat disimpan dengan menekan tombol “Simpan”, sehingga Anda dapat memantau tanaman cabai.

Simpan

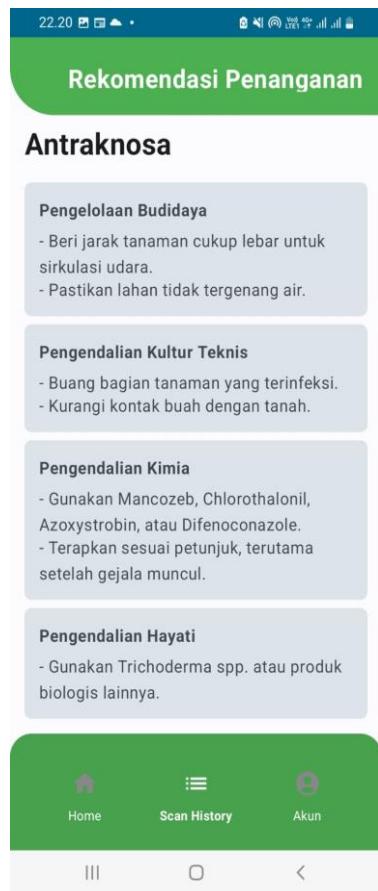
#### G. Rekomendasi Penanganan

Anda dapat mengetahui rekomendasi penanganan tentang penyakit yang terjadi pada cabai dengan menekan tombol “Rekomendasi Penanganan”.

Pindai Ulang

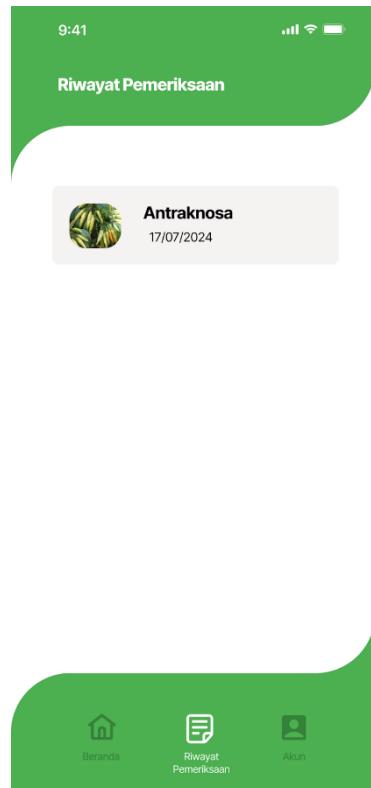
Rekomendasi  
Penanganan

Aplikasi akan menampilkan rekomendasi penanganan terkait dengan penyakit yang menyerang tanaman cabai.



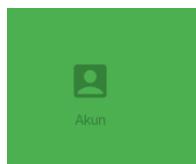
## H. Riwayat Pemeriksaan

Anda dapat melihat riwayat pemeriksaan yang telah dilakukan sebelumnya, terdapat tanggal dan nama penyakit.



## I. Profil Pengguna

Masuk ke menu "Profil Pengguna" untuk memperbarui informasi pribadi Anda.



Pastikan data yang Anda masukkan selalu yang terbaru.

Profil

Nama Lengkap

Michelle Putri

Email

michelle@gmail.com

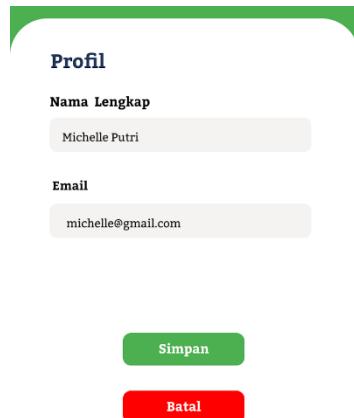
Edit

Keluar

Jika ada perubahan nama atau *email*, Anda dapat menekan tombol "Edit".

Edit

Profil dapat diubah menjadi data yang terbaru.



Setelah dipastikan benar Anda dapat menekan tombol “Simpan”.

Simpan

Namun, jika tidak ada data yang perlu diperbarui. Anda dapat menekan tombol “Batal”.

Batal

Apabila Anda ingin mengeluarkan akun dari Aplikasi, dapat menekan tombol “Keluar” yang berwarna “Merah”.

Keluar

## Lampiran 9 Link Video Profil Produk Luaran Proyek

[https://drive.google.com/drive/folders/1sb82EB3WvALgLIp\\_fDARi7bjNkAoepFv?usp=sharing](https://drive.google.com/drive/folders/1sb82EB3WvALgLIp_fDARi7bjNkAoepFv?usp=sharing)

## Lampiran 10 Poster Produk Luaran Proyek

The poster features a woman with long brown hair, wearing a white shirt, smiling as she looks at her smartphone. The phone screen displays the CEKCABAI app interface, which includes a green header with 'Hello, Michelle!', a main menu with 'Periksa Tanaman Anda' and 'Mulai', a photo of red and yellow chili peppers, and a 'Upload via Gallery' button. Below the phone, four numbered steps explain the app's functionality: 1. Ambil foto daun dan buah cabai untuk mendeteksi penyakit. 2. Otomatis mendeteksi penyakit dengan cepat dan akurat. 3. Dapatkan saran perawatan sesuai jenis penyakit. 4. Lihat riwayat pemeriksaan untuk memantau kesehatan tanaman. To the right of these steps is a detailed description of the app's purpose: 'Aplikasi ini dirancang untuk membantu petani dalam mendeteksi penyakit pada tanaman cabai secara cepat dan akurat. Dengan teknologi analisis gambar berbasis AI, pengguna hanya perlu mengambil foto daun tanaman yang terinfeksi, dan aplikasi akan menganalisis serta memberikan diagnosis dalam hitungan detik.' At the bottom left, there is a QR code with the text 'SCAN ME!' and 'DOWNLOAD NOW'.

CEKCABAI

Identifikasi penyakit pada tanaman cabai dengan cepat dan akurat.

- 1 Ambil foto daun dan buah cabai untuk mendeteksi penyakit.
- 2 Otomatis mendeteksi penyakit dengan cepat dan akurat.
- 3 Dapatkan saran perawatan sesuai jenis penyakit.
- 4 Lihat riwayat pemeriksaan untuk memantau kesehatan tanaman.

Aplikasi ini dirancang untuk membantu petani dalam mendeteksi penyakit pada tanaman cabai secara cepat dan akurat. Dengan teknologi analisis gambar berbasis AI, pengguna hanya perlu mengambil foto daun tanaman yang terinfeksi, dan aplikasi akan menganalisis serta memberikan diagnosis dalam hitungan detik.

SCAN ME!

DOWNLOAD NOW



**Lampiran 11 Slide presentasi proyek**

[https://www.canva.com/design/DAGL3z9G1BU/XToUePiYC3Ph5pBIhfAXxQ/edit?utm\\_content=DAGL3z9G1BU&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAGL3z9G1BU/XToUePiYC3Ph5pBIhfAXxQ/edit?utm_content=DAGL3z9G1BU&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)