

**PENGEMBANGAN APLIKASI *MOBILE* DURIPINTAR UNTUK
KLASIFIKASI JENIS DAN DETEKSI PENYAKIT TANAMAN DURIAN
BERBASIS *KOTLIN* DENGAN ARSITEKTUR *MVVM***

(Skripsi)

Oleh

MUCHAMMAD RAJA HAIKAL FIAUGUSTIAN

NPM 2155061016



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2026

**PENGEMBANGAN APLIKASI *MOBILE* DURIPINTAR UNTUK
KLASIFIKASI JENIS DAN DETEKSI PENYAKIT TANAMAN DURIAN
BERBASIS *KOTLIN* DENGAN ARSITEKTUR *MVVM***

Oleh

MUCHAMMAD RAJA HAIKAL FIAUGUSTIAN

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2026

ABSTRAK

PENGEMBANGAN APLIKASI *MOBILE* DURIPINTAR UNTUK KLASIFIKASI JENIS DAN DETEKSI PENYAKIT TANAMAN DURIAN BERBASIS *KOTLIN* DENGAN ARSITEKTUR *MVVM*

Oleh

MUCHAMMAD RAJA HAIKAL FIAUGUSTIAN

Durian (*Durio zibethinus*) merupakan salah satu komoditas buah tropis yang memiliki nilai ekonomi tinggi di Indonesia. Namun, proses identifikasi jenis durian serta deteksi penyakit pada tanaman durian masih sering dilakukan secara manual sehingga memerlukan waktu dan pengetahuan khusus. Penelitian ini bertujuan untuk mengembangkan aplikasi mobile DuriPintar berbasis Android yang dapat membantu pengguna dalam mengklasifikasikan jenis durian dan mendeteksi penyakit daun melalui pemindaian gambar. Pengembangan aplikasi dilakukan menggunakan bahasa pemrograman Kotlin dengan menerapkan arsitektur *Model–View–ViewModel (MVVM)*. Metode pengembangan sistem menggunakan *Software Development Life Cycle (SDLC)* dengan pendekatan Kanban yang meliputi tahapan studi literatur, pengumpulan kebutuhan pengguna, perancangan sistem menggunakan *UML*, implementasi, serta pengujian. Model kecerdasan buatan untuk klasifikasi jenis durian dan deteksi penyakit daun diintegrasikan ke dalam aplikasi melalui *Application Programming Interface (API)*. Hasil penelitian menunjukkan bahwa aplikasi DuriPintar berhasil dikembangkan dan mampu menampilkan hasil klasifikasi jenis durian serta deteksi penyakit daun secara langsung melalui perangkat mobile. Aplikasi ini diharapkan dapat membantu petani dan masyarakat dalam melakukan identifikasi tanaman durian secara lebih cepat dan praktis.

Kata kunci: Durian, *Android*, *Kotlin*, *MVVM*, Klasifikasi Durian, Deteksi Penyakit Tanaman, *Mobile*.

ABSTRACT

DEVELOPMENT OF A DURIPINTAR MOBILE APPLICATION FOR DURIAN VARIETY CLASSIFICATION AND DISEASE DETECTION USING KOTLIN WITH THE MVVM ARCHITECTURE

By

MUCHAMMAD RAJA HAIKAL FIAUGUSTIAN

Durian (Durio zibethinus) is one of the tropical fruit commodities with high economic value in Indonesia. However, the process of identifying durian varieties and detecting diseases in durian plants is still commonly performed manually, which requires considerable time and specific expertise. This study aims to develop a mobile application called DuriPintar based on the Android platform to assist users in classifying durian varieties and detecting leaf diseases through image scanning. The application was developed using the Kotlin programming language with the implementation of the Model–View–ViewModel (MVVM) architecture. The system development process follows the Software Development Life Cycle (SDLC) using the Kanban approach, which includes literature study, requirement gathering, system design using UML, implementation, and testing. Artificial intelligence models for durian variety classification and leaf disease detection are integrated into the application through an Application Programming Interface (API). The results of this study indicate that the DuriPintar application was successfully developed and is capable of displaying classification results of durian varieties and leaf disease detection directly through mobile devices. This application is expected to assist farmers and the general public in identifying durian plants more quickly and practically.

Keywords: Durian, Android, Kotlin, MVVM, Image Classification, Disease Detection, Mobile.

Judul Skripsi : PENGEMBANGAN APLIKASI
MOBILE DURIPINTAR UNTUK
KLASIFIKASI JENIS DAN DETEKSI
PENYAKIT TANAMAN DURIAN
BERBASIS KOTLIN DENGAN
ARSITEKTUR MVVM

Nama Mahasiswa : Muchammad Raja Haikal Fiaugustian

Nomor Pokok Mahasiswa : 2155061016

Program Studi : Teknik Informatika

Jurusan : Teknik Elektro

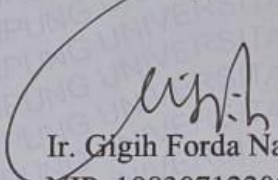
Fakultas : Teknik

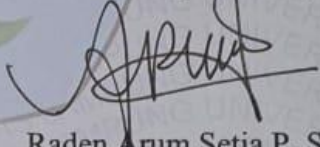
MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping


Ir. Gigih Forda Nama, S.T., M.T.I., IPM
NIP. 198307122008121003


Raden Arum Setia P, S.Si, M.T.I
NIP. 199009212019032025

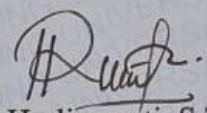
2. Mengetahui

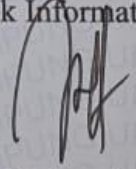
Ketua Jurusan

Ketua Program Studi

Teknik Elektro

Teknik Informatika

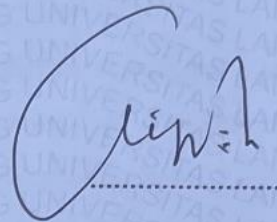

Herlinawati, S.T., M.T.
NIP. 197103141999032001


Yessi Mulyani, S.T., M.T.
NIP. 197312262000122001

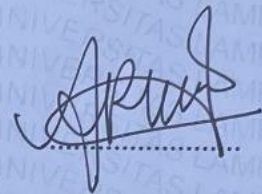
PENGESAHAN

1. Tim Penguji

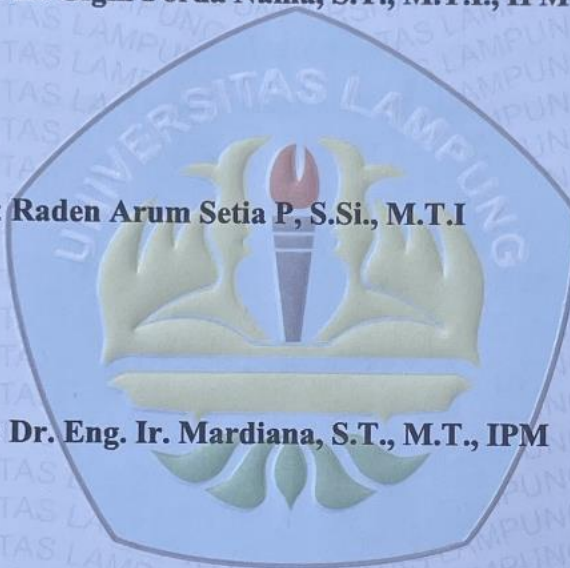
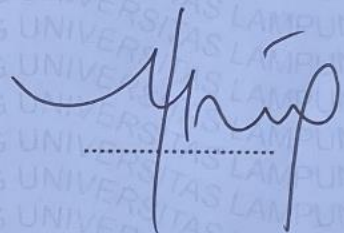
Ketua : **Ir. Gigih Forda Nama, S.T., M.T.I., IPM**



Sekretaris : **Raden Arum Setia P, S.Si., M.T.I**



Penguji : **Dr. Eng. Ir. Mardiana, S.T., M.T., IPM**



2. Dekan Fakultas Teknik


Dr. H. Ahmad Herison, S.T., M.T.
NIP. 196910302000031001

Tanggal Lulus Ujian Skripsi : 10 Februari 2026

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi berjudul “Pengembangan Aplikasi *Mobile* Duripintar Untuk Klasifikasi Jenis Dan Deteksi Penyakit Tanaman Durian Berbasis *Kotlin* Dengan Arsitektur *MVVM*” sepenuhnya merupakan hasil karya saya sendiri. Apabila di kemudian hari terbukti pernyataan ini tidak benar, saya siap menerima sanksi sesuai dengan ketentuan hukum yang berlaku.

Bandar Lampung, 10 Februari 2026

Penulis,



Muchammad Raja Haikal Fiaugustian

NPM. 2155061016

RIWAYAT HIDUP



Penulis bernama Muchammad Raja Haikal Fiaugustian dilahirkan di Bandarlampung pada tanggal 19 Agustus 2001, sebagai anak pertama dari pasangan Bapak Ir. Muchammad Puncak Stiawan, S.P., M.M dan Ibu Dr. Erfina, M.M.

Penulis menyelesaikan pendidikan Taman Kanak-Kanak(TK) pada tahun 2006-2007 di Yayasan Ath-Thoyyibah Menggala, Kemudian melanjutkan pendidikan Sekolah Dasar (SD) di SDN 01 Menggala Kota pada tahun 2007-2013, melanjutkan ke Sekolah Menengah Pertama SMP Al-Kautsar pada tahun 2013-2016, serta menamatkan pendidikan Sekolah Menengah Atas SMA pada tahun 2016-2019. Pada tahun 2021, penulis diterima sebagai mahasiswa Program Studi Teknik Informatika Fakultas Teknik Universitas Lampung melalui jalur seleksi SMMPTN barat.

Selama menjadi mahasiswa, penulis pernah berpartisipasi dalam berbagai kegiatan, antara lain:

1. Mengikuti Penelitian MBKM dengan Judul Penelitian Sistem Cerdas Rekomendasi Tanaman Perkebunan Berbasis Machine Learning Sebagai Adaptasi Perubahan Iklim di Taman Kupu-Kupu Gita Persada Lampung pada tahun 2024
2. Mengikuti kegiatan Studi Independen Bersertifikat dari Kementerian Pendidikan dan Budaya di mitra Bangkit Akademi pada tahun 2023.
3. Menjadi anggota Himpunan Mahasiswa Teknik Elektro pada tahun 2022-2023.

MOTTO

“Indeed, Allah loves a person who, when he performs a task, does it with excellence.”

(Hadith by At-Tabrani)

“Pain + Reflection = Progress.”

(Ray Dalio)

“Fortune Favors The Bold”

(Virgil)

PERSEMBAHAN

Bismillahirrahmanirrahim...

Segala puji bagi Allah Swt. Tuhan Semesta Alam yang telah memberikan segala nikmat dan anugerah. Sholawat serta salam selalu terlimpah kepada Nabi Muhammad SAW sebagai suri tauladan umat muslim. Sebagai rasa terima kasih kupersembahkan karya ini kepada:

Mama Tercinta (Dr. Erfina, M.M)

Yang telah memberikan cinta, kasih sayang, serta doa dan restu yang tiada henti. Mama selalu menjadi pendukung dan penyemangat dalam setiap langkah perjalanan saya. Terima kasih atas segala bimbingan, didikan, serta petuah dan saran yang membangun sehingga saya dapat sampai pada tahap ini.

Papa Tercinta (Ir. Muchammad Puncak Stiawan, S.P., M.M)

Yang telah memberikan nasihat, dukungan, serta teladan dalam kerja keras dan perjuangan. Setiap nasihat dan perjuangan Papa menjadi kekuatan bagi saya untuk terus berusaha dan tidak menyerah dalam menghadapi setiap tantangan. Semoga Papa selalu diberikan kesehatan, kebahagiaan, dan keberkahan oleh Allah SWT.

Almamater Tercinta Universitas Lampung

Sebagai tempat menimba ilmu, yang telah menjadikanku pribadi yang lebih baik dan mempertemukanku dengan orang-orang hebat.

SANWACANA

Puji syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Pengembangan Aplikasi *Mobile* DuriPintar Untuk Klasifikasi Jenis Dan Deteksi Penyakit Tanaman Durian Berbasis *Kotlin* Dengan Arsitektur *MVVM*” dengan baik.

Dalam kesempatan ini, penulis mengucapkan terima kasih kepada:

1. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., I.P.M., selaku Rektor Universitas Lampung;
2. Bapak Dr. Hi. Ahmad Herison, S.T., M.T. selaku Dekan Fakultas Teknik, Universitas Lampung.;
3. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung.;
4. Ibu Yessi Mulyani, S.T., M.T., selaku Ketua Program Studi Teknik Informatika Fakultas Teknik Universitas Lampung.;
5. Bapak Ir. Gigih Forda Nama, S.T., M.T.I., IPM, selaku Dosen Pembimbing Utama, yang dengan sabar, telaten, dan penuh perhatian dalam memberikan arahan, masukan, serta ilmu yang sangat berharga selama proses penelitian dan penulisan skripsi sampai selesai.;
6. Bapak Raden Arum Setia P, S.Si., M.T.I, selaku Dosen Pembimbing Pendamping, yang telah meluangkan waktu, tenaga, dan pikiran untuk memberikan bimbingan, saran, dan perbaikan skripsi sampai selesai.;
7. Ibu Dr. Eng. Ir. Mardiana, S.T., M.T., IPM, selaku Dosen Penguji, yang telah memberikan kritik, saran, dan masukan yang membangun demi penyempurnaan hasil skripsi ini hingga selesai.;

8. Bapak Mahendra Pratama, S.T., M.Eng., selaku Dosen Pembimbing Akademik, yang telah memberikan bimbingan, perhatian, serta motivasi selama penulis menjalani masa perkuliahan.;
9. Seluruh Dosen Program Studi Teknik Informatika Fakultas Teknik Universitas Lampung, yang telah memberikan ilmu, wawasan, dan pengalaman berharga selama masa perkuliahan.;
10. Rekan satu tim penelitian, Muchammad Raja Haikal Fiaugustian, Farhat Febrianto, Muhammad Rafi Rizanda, atas kerja sama yang solid, semangat kebersamaan, serta kontribusinya dalam pengembangan Aplikasi Duri Pintar ini.;
11. Aidita Kirana, S.Pd., *for her unwavering support, encouragement, and assistance in every aspect. Her patience, understanding, and constant motivation have been invaluable. I am truly grateful for your presence.*
12. Teman-teman Teknik Informatika 2021 yang tidak dapat disebutkan satu per satu, atas kebersamaan, dukungan, dan semangat yang diberikan selama masa perkuliahan hingga penyusunan skripsi ini.;
13. Keluargaku, yang selalu memberikan doa, dukungan, motivasi, bantuan, dan semangat dalam penyelesaian skripsi ini

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, namun besar harapan karya ini dapat memberikan manfaat serta menjadi referensi bagi pengembangan penelitian di bidang Rekayasa Perangkat Lunak dan bermanfaat bagi banyak orang.

Bandar Lampung, 10 Februari 2026

Penulis,

Muchammad Raja Haikal Fiaugustian
NPM. 2155061016

DAFTAR ISI

ABSTRAK	i
<i>ABSTRACT</i>	ii
PENGESAHAN	iv
SURAT PERNYATAAN	v
RIWAYAT HIDUP	vi
MOTTO	vii
PERSEMBAHAN.....	viii
SANWACANA	ix
DAFTAR ISI.....	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN	xvii
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah	3
1.6 Sistematika Penulisan Laporan.....	3
II. TINJAUAN PUSTAKA.....	5
2.1 Dasar Teori	5
2.1.1 Tanaman Durian (<i>Durio zibethinus</i>).....	5
2.1.2 <i>Unified Modeling Language</i> (UML)	6
2.1.3 Android Studio.....	7
2.1.4 <i>User Acceptance Test</i> (UAT)	8

2.1.5	Android 9	
2.1.6	Kotlin	10
2.1.7	Arsitektur MVVM (<i>Model-View-ViewModel</i>)	11
2.1.8	UI/UX (<i>User Interface</i> dan <i>User Experience</i>)	12
2.1.9	<i>Material Design</i>	13
2.1.10	API (<i>Application Programming Interface</i>)	14
2.1.11	<i>Blackbox Testing</i>	15
2.2	Penelitian Terkait	16
2.2.1	Penelitian Akurasi Diagnosa Penyakit Tanaman Padi menggunakan Kamera dengan Metode Klasifikasi Gambar pada Perangkat Bergerak Android	16
2.2.2	Aplikasi GreenAdvisor Sebagai Pendeteksi dan Rekomendasi Tanaman Berdasarkan Koordinat Berbasis Android Menggunakan Design Pattern	17
2.2.3	Analisis Klasifikasi Kesegaran Buah Apel Menggunakan Metode <i>Convolutional Neural Network</i> (CNN) Berbasis <i>Android</i>	19
2.2.4	Pengembangan Aplikasi Mobile Pendeteksi Penyakit pada Tanaman Cabai dengan menggunakan Teknologi Clarifai	20
2.2.5	Sistem Deteksi Penyakit Daun Apel Menggunakan Metode CNN Berbasis Mobile	21
2.2.6	Implementasi <i>Convolutional Neural Network</i> Menggunakan Arsitektur <i>Xception</i> Untuk Identifikasi Penyakit Daun Pada Tanaman Durian	23
2.2.7	Implementasi <i>Convolutional Neural Network</i> Berbasis Arsitektur <i>Efficientnet-B0</i> Untuk Klasifikasi Varietas Buah Durian	25
III.	METODE PENELITIAN	27
3.1	Waktu dan Tempat Penelitian	27
3.2	Alat dan Bahan Penelitian	27
3.3	Struktur Tim	28
3.4	Tahapan Penelitian	29

3.5	Studi literatur	30
3.6	SDLC menggunakan Kanban	30
3.6.1	<i>User's Stories</i>	31
3.6.2	<i>Requirements Gathering</i>	32
3.6.3	<i>Unified Modeling Language</i>	34
3.6.3.1	Use Case Diagram	34
3.6.3.2	Activity Diagram	34
3.6.4	<i>Product Backlog</i>	37
3.6.5	<i>Work in Progress</i>	37
3.6.6	<i>Software In Test</i>	37
3.6.7	<i>User Acceptance Test (UAT)</i>	38
3.7	Penulisan Laporan	39
IV.	HASIL DAN PEMBAHASAN.....	40
4.1	Studi Literatur	40
4.2	<i>Software Development Life Cycle</i> dengan Kanban.....	41
4.2.1	<i>User's Requests / Stories</i>	42
4.2.2	<i>Requirements Gathering</i>	43
4.2.2.1	Arsitektur MVVM	44
4.2.2.2	Desain UI/UX	47
4.2.3	<i>Backlog</i> 53	
4.2.4	<i>Work In Progress</i>	57
4.2.4.1	<i>Code Review</i>	71
4.2.5	<i>Software In Test</i>	72
4.2.6	<i>User Acceptance Test</i>	77
4.2.7	Pengujian <i>Usability</i> Menggunakan <i>System Usability Scale</i> (SUS) 80	
V.	KESIMPULAN DAN SARAN.....	83
5.1	Kesimpulan.....	83
5.2	Saran.....	83
	DAFTAR PUSTAKA	85
	LAMPIRAN.....	89

DAFTAR TABEL

Tabel 3.1 Waktu Penelitian	27
Tabel 3.2 Alat Penelitian	27
Tabel 3.3 <i>User Story</i>	32
Tabel 3.4 Kebutuhan Fungsional	33
Tabel 3.5 Kebutuhan Non Fungsional	33
Tabel 3.6 <i>Test Case Blackbox Testing</i>	38
Tabel 4.1 <i>User Story</i>	43
Tabel 4.2 Kebutuhan Fungsional	44
Tabel 4.3 Kebutuhan Non Fungsional	44
Tabel 4.4 <i>Unit Test</i>	73
Tabel 4.5 <i>Blackbox test final</i>	75
Tabel 4.6 Bobot Penilaian	77
Tabel 4.7 Kategori Interpretasi skor	78
Tabel 4.8 Hasil Kuisiomer User	78
Tabel 4.9 Hasil Kuesioner <i>System Usability Scale</i> (SUS)	80
Tabel 4.10 Kategori Interpretasi skor	81

DAFTAR GAMBAR

Gambar 2.1	Manfaat Durian.....	5
Gambar 2.2	<i>Android Studio</i>	7
Gambar 2.3	Tahapan UAT	8
Gambar 2.4	<i>Android</i>	9
Gambar 2.5	Kotlin	10
Gambar 2.6	Arsitektur MVVM	11
Gambar 2.7	<i>User Interface/User Experience</i>	12
Gambar 2.8	<i>Logo Material Design</i>	13
Gambar 2.9	<i>Application Programming Interface</i>	14
Gambar 2.10	<i>Blackbox Testing</i>	15
Gambar 2.11	Rancangan Halaman <i>Register</i> dan <i>Login</i>	18
Gambar 2.12	Use Case Diagram Aplikasi.....	19
Gambar 2.13	Tampilan Halaman Menu Utama Aplikasi	21
Gambar 2.14	Tampilan Halaman Deteksi Aplikasi	23
Gambar 2.15	Total Dataset	24
Gambar 2.16	<i>Confusion matrix Epoch 40</i>	26
Gambar 3.1	Struktur Tim Peneliti	28
Gambar 3.2	Tahapan Penelitian.....	29
Gambar 3.3	Kanban board.....	31
Gambar 3.4	<i>Use Case Diagram</i>	34
Gambar 3.5	<i>Activity Diagram</i> Daftar	35
Gambar 3.6	<i>Activity Diagram</i> Masuk.....	35
Gambar 3.7	Activity Diagram Deteksi Penyakit Durian.....	36
Gambar 3.8	Activity Diagram Deteksi Klasifikasi Durian.....	36
Gambar 4.1	Visualisasi Alur Kerja dengan <i>Kanban Board</i>	42

Gambar 4.2	Diagram Component Arsitektur <i>MVVM</i> Aplikasi DuriPintar.....	44
Gambar 4.3	Injeksi Dependensi menggunakan Koin	46
Gambar 4.4	Desain Halaman <i>Onboarding</i> DuriPintar.....	47
Gambar 4.5	Desain halaman Welcome, Login, dan Register	48
Gambar 4.6	Desain halaman <i>Profile</i>	49
Gambar 4.7	Desain halaman <i>Home</i>	50
Gambar 4.8	Desain halaman <i>Camera</i> dan <i>Scan Result</i>	51
Gambar 4.9	Desain halaman <i>History</i> dan <i>Note</i>	52
Gambar 4.10	Desain halaman <i>Disease & Variety</i>	53
Gambar 4.11	Tampilan UI Notion bagian <i>frontend mobile</i>	54
Gambar 4.12	Tampilan detail backlog <i>frontend mobile</i>	56
Gambar 4.13	Tampilan <i>notion</i> ketika pengambilan task masuk ke <i>Work In progress</i>	57
Gambar 4.14	Desain halaman Onboarding dan <i>Welcome</i>	58
Gambar 4.15	Persiapan Lingkungan Aplikasi	59
Gambar 4.16	Struktur Paket Proyek.....	60
Gambar 4.17	<i>XML</i> Halaman <i>Home</i>	61
Gambar 4.18	<i>XML</i> Halaman <i>Camera</i>	62
Gambar 4.19	<i>XML</i> Halaman <i>Camera</i>	63
Gambar 4.20	<i>XML</i> Halaman <i>Collection & Detail</i>	63
Gambar 4.21	<i>XML</i> Halaman <i>Note & History</i>	64
Gambar 4.22	<i>XML</i> Halaman <i>Auth Login</i>	65
Gambar 4.23	<i>XML</i> Halaman <i>Profile</i>	65
Gambar 4.24	<i>StateFlow login</i>	66
Gambar 4.25	Penggunaan <i>Parcelize</i>	66
Gambar 4.26	Kode <i>collect</i> untuk <i>error</i>	67
Gambar 4.27	Base URL Duripintar	67
Gambar 4.28	Kontrak Pemindaian.	68
Gambar 4.29	<i>Leaf</i> dan <i>Fruit Response</i>	69
Gambar 4.30	<i>Fallback</i> kartu cuaca	70
Gambar 4.31	<i>AuthRepository</i>	70

DAFTAR LAMPIRAN

Lampiran 1. Link Dokumentasi	89
Lampiran 2. Rancangan <i>Low Fidelity Wireframe</i>	89
Lampiran 3. Hasil <i>Quisioner User Acceptance Test</i>	90
Lampiran 4. Pengumpulan dataset dan survei ke petani	90

I. PENDAHULUAN

1.1 Latar Belakang

Durian atau disebut dalam bahasa Latin *Durio zabethinus Murr* merupakan salah satu jenis buah tropis yang tergolong dalam famili Bombacaceae yang memiliki nilai ekonomis tinggi. Buah ini memiliki cita rasa yang khas dan unik, sehingga diminati oleh berbagai kalangan masyarakat. Di Indonesia, durian menjadi salah satu komoditas buah dengan produksi nasional yang mengalami peningkatan setiap tahunnya[1]. Berdasarkan data yang dirilis oleh Badan Pusat Statistik, produksi durian di Indonesia mengalami peningkatan yang konsisten dari tahun ke tahun. Pada tahun 2020, total produksi durian tercatat sebesar 1.133.194 ton. Jumlah tersebut meningkat menjadi 1.352.957 ton pada tahun 2021, kemudian naik menjadi 1.582.171 ton pada tahun 2022. Tren kenaikan ini berlanjut pada tahun 2023 dengan total produksi sebesar 1.852.045 ton, dan pada tahun 2024 mencapai sekitar 1.961.486 ton. Data ini menunjukkan adanya pertumbuhan yang signifikan dalam sektor produksi durian selama lima tahun terakhir[2]. Peningkatan produksi durian menunjukkan dampak ekonomi yang signifikan bagi perekonomian di Indonesia. Harga jual durian yang relatif tinggi dibandingkan dengan komoditas buah lainnya menjadikannya sebagai salah satu peluang agribisnis yang potensial dalam meningkatkan pendapatan petani. Tingkat produksi durian yang terus meningkat setiap tahunnya membuka peluang untuk ekspansi ke pasar internasional[3]. Seiring dengan meningkatnya produksi durian, kebutuhan akan pemanfaatan teknologi dalam mendukung kegiatan budidaya, identifikasi varietas, hingga deteksi penyakit tanaman pun semakin relevan[4].

Penelitian sebelumnya telah mengembangkan sistem klasifikasi dan diagnosis penyakit durian berbasis kecerdasan buatan, seperti metode *Naive Bayes Classifier*,

Certainty Factor, dan *Dempster-Shafer* untuk mendiagnosis gejala berdasarkan data gejala visual atau masukan pengguna[4], [5]. Namun keterbatasan masih ditemukan pada sisi implementasi sistem dalam bentuk aplikasi web atau *desktop* yang belum mempertimbangkan aspek aksesibilitas dan kemudahan penggunaan di lapangan.[6] Sejalan dengan itu, studi terkini yang dilakukan oleh Kusuma et al., pendekatan berbasis *Convolutional Neural Network* (CNN) telah diintegrasikan ke dalam aplikasi mobile berbasis Android untuk mendeteksi penyakit tanaman padi melalui citra daun menggunakan arsitektur *MobileNetV3*[7]. Penerapan teknologi mobile seperti ini menjadi relevan karena memberikan kemudahan kepada pengguna akhir, dalam hal ini petani dan masyarakat yang baru memulai budidaya dalam melakukan diagnosis.[6] Meski demikian, sebagian besar penelitian masih menitikberatkan pada akurasi model atau algoritma yang digunakan dan belum memberikan perhatian pada desain antarmuka *frontend* sebagai aspek untuk menjembatani teknologi dengan pengguna lapangan. Padahal, desain antarmuka yang baik sangat dibutuhkan agar aplikasi benar-benar dapat diadopsi dan digunakan secara luas[8], [9]. Oleh karena itu, penelitian ini diharapkan mampu menjawab keterbatasan yang telah disampaikan sebelumnya dengan memfokuskan pengembangan aplikasi DuriPintar berbasis *Android* menggunakan arsitektur *MVVM* (Model–View–ViewModel) karena mampu memisahkan logika presentasi dari tampilan sehingga kode lebih terstruktur dan konsisten.[10]

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini adalah sebagai:

1. Bagaimana merancang dan membangun aplikasi *mobile* Duripintar berbasis *Android* menggunakan bahasa pemrograman Kotlin?
2. Bagaimana mengintegrasikan hasil dari model AI klasifikasi dan diagnosis penyakit daun tanaman durian ke dalam aplikasi?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Merancang dan membangun aplikasi *mobile* Duripintar berbasis *Android* menggunakan bahasa pemrograman Kotlin.
2. Mengintegrasikan hasil dari model AI klasifikasi dan diagnosis ke dalam aplikasi.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

1. Penelitian ini dapat memberikan kemudahan bagi petani dan masyarakat umum dalam mengidentifikasi jenis durian serta mendeteksi penyakit tanaman tersebut melalui aplikasi *mobile* Duripintar.
2. Penelitian ini diharapkan dapat menjadi referensi bagi peneliti lain dan pemanfaatan teknologi *mobile* terbaru seperti *kotlin* dan *MVVM*.

1.5 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah:

1. Penelitian ini berfokus pada pengembangan *frontend* aplikasi *android*.
2. Penelitian ini menggunakan *Kotlin* dengan arsitektur *MVVM* untuk pengembangan *frontend*.
3. Pengujian hanya dilakukan dari sisi pengguna (*end-user*) melalui evaluasi desain dan pengalaman penggunaan (*User Interface* dan *User Experience*), bukan pada performa atau akurasi model klasifikasi.

1.6 Sistematika Penulisan Laporan

Adapun sistematika yang digunakan dalam penulisan penelitian ini adalah sebagai berikut:

BAB I : PENDAHULUAN

Bab ini berisi latar belakang, perumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, serta sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Bab ini berisi tentang landasan teori yang mendukung dalam penelitian ini serta penelitian – penelitian sebelumnya yang berkaitan dengan penelitian ini.

BAB III : METODOLOGI PENELITIAN

Bab ini berisi mengenai waktu penelitian, tempat penelitian, alat penelitian serta metode dan tahap yang dilakukan dalam penelitian ini.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini berisi hasil dan pembahasan dari penelitian yang telah dilakukan.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan penelitian dan saran untuk penelitian yang berkaitan pada waktu setelahnya.

DAFTAR PUSTAKA :

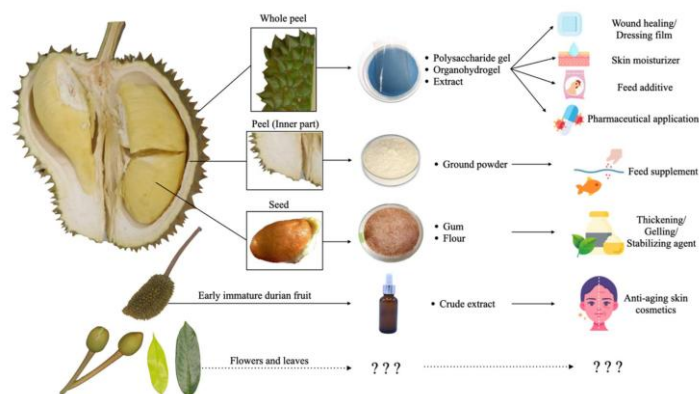
Bab ini memuat daftar sumber kutipan teori - teori yang dijadikan acuan penulis dalam menulis laporan.

II. TINJAUAN PUSTAKA

2.1 Dasar Teori

2.1.1 Tanaman Durian (*Durio zibethinus*)

Durian, dikenal sebagai "raja buah" di Asia Tenggara, merupakan buah tropis yang berasal dari genus *Durio*, dengan *Durio zibethinus* sebagai spesies yang paling umum dikonsumsi dan diperdagangkan secara internasional. Buah ini terkenal karena ukurannya yang besar, kulit berduri, dan aroma khas yang bagi sebagian orang memikat, namun bagi yang lain dianggap menyengat[11]. Secara nutrisi, durian menonjol dibandingkan buah tropis lainnya. Dalam 243 gram daging buah durian mengandung sekitar 357 kalori, 13 gram lemak, 66 gram karbohidrat, 9 gram serat, dan 4 gram protein. Durian juga kaya akan vitamin C (80% dari kebutuhan harian), vitamin B kompleks (termasuk B6, tiamin, riboflavin), kalium, magnesium, dan folat[12]. Selain itu, durian mengandung senyawa bioaktif seperti polifenol, flavonoid, dan karotenoid yang berperan sebagai antioksidan[13].



Gambar 2.1 Manfaat Durian

Beberapa manfaat kesehatan yang dikaitkan dengan konsumsi durian antara lain:

- a. Menurunkan risiko penyakit jantung: Kandungan serat dan lemak tak jenuh tunggal membantu mengatur kadar kolesterol dan tekanan darah[13].
- b. Mendukung kesehatan pencernaan: Kandungan serat tinggi membantu mencegah sembelit dan menjaga kesehatan usus[12].
- c. Meningkatkan sistem imun: Kandungan vitamin C dan antioksidan membantu melindungi tubuh dari infeksi dan kerusakan sel akibat radikal bebas[14].
- d. Potensi antikanker: Studi laboratorium menunjukkan bahwa ekstrak durian dapat menghambat pertumbuhan sel kanker payudara, meskipun penelitian lebih lanjut pada manusia diperlukan[11], [15].

Namun, konsumsi durian perlu diperhatikan secara moderat terutama pada individu dengan kondisi kesehatan tertentu seperti diabetes atau penyakit ginjal, mengingat tingginya kandungan kalori dan lemaknya, meskipun indeks glikemik durian relatif rendah dibandingkan buah tropis lainnya[13].

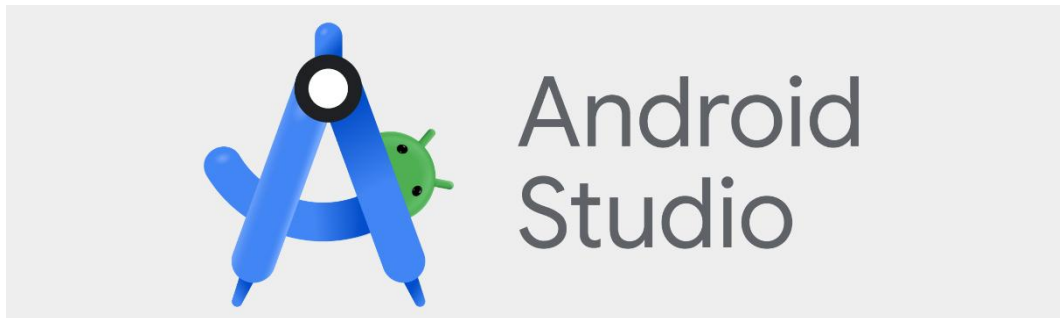
2.1.2 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan artefak perangkat lunak berbasis objek. UML dirancang untuk membantu para pengembang dan analis dalam memahami struktur serta perilaku sistem perangkat lunak yang kompleks. UML pertama kali dikembangkan oleh Grady Booch, Ivar Jacobson, dan James Rumbaugh—tiga tokoh penting dalam rekayasa perangkat lunak yang dikenal dengan sebutan "tiga amigo". Mereka menyatukan berbagai metode pemodelan sebelumnya menjadi satu standar terpadu. UML kemudian diadopsi oleh *Object Management Group (OMG)* pada tahun 1997 sebagai bahasa pemodelan standar industri[16].

UML mencakup beberapa jenis diagram, yang terbagi menjadi dua kelompok utama:

- a. *Use Case Diagram* merupakan diagram yang menggambarkan fungsionalitas yang diharapkan dari sistem serta interaksi antara aktor (manusia atau sistem lain) dan sistem itu sendiri[17].
- b. *Activity Diagram* merupakan diagram yang menunjukkan aliran aktivitas dalam sistem yang sedang berjalan[17].

2.1.3 Android Studio



Gambar 2.2 *Android Studio*

Android Studio merupakan lingkungan pengembangan terintegrasi (*Integrated Development Environment/IDE*) resmi dari Google untuk pengembangan aplikasi Android. IDE ini dirilis pertama kali pada tahun 2013 dan dibangun di atas *platform* IntelliJ IDEA dari JetBrains, yang menawarkan antarmuka cerdas dan alat bantu yang dirancang khusus untuk kebutuhan pengembang *Android*[18]. *Android Studio* menyediakan fitur lengkap, seperti sistem build berbasis *Gradle*, *Android Virtual Device* (AVD) untuk emulasi, editor visual untuk tata letak, pelengkapan kode secara otomatis (*code completion*), dan dukungan *debugging* yang terintegrasi[19]. Selain itu, *Android Studio* mendukung berbagai bahasa pemrograman seperti Java, Kotlin, dan C++. Sejak tahun 2017, Google secara resmi merekomendasikan Kotlin sebagai bahasa utama untuk pengembangan aplikasi android. karena sintaksisnya yang lebih ringkas dan fitur-fitur modern yang meningkatkan keamanan dan produktivitas pengembang[20].

2.1.4 *User Acceptance Test (UAT)*



Gambar 2.3 Tahapan UAT

User Acceptance Testing (UAT) adalah tahap akhir dalam proses pengujian perangkat lunak, di mana pengguna akhir atau perwakilan bisnis melakukan verifikasi terhadap sistem untuk memastikan bahwa perangkat lunak memenuhi kebutuhan dan persyaratan bisnis yang telah ditetapkan. UAT bertujuan untuk menilai apakah sistem siap untuk digunakan dalam lingkungan operasional sebenarnya. Berbeda dengan pengujian teknis lainnya seperti *unit testing* atau *system testing*, UAT fokus pada validasi dari perspektif pengguna. Pengujian ini biasanya dilakukan dalam lingkungan yang menyerupai kondisi produksi, dengan skenario yang mencerminkan penggunaan nyata oleh pengguna akhir. Tujuan utamanya adalah untuk memastikan bahwa sistem dapat mendukung proses bisnis sebagaimana mestinya dan memberikan nilai tambah bagi organisasi[21].

Dalam konteks pengembangan perangkat lunak berbasis *Agile*, UAT sering kali dilakukan secara iteratif dan kolaboratif, melibatkan pengguna sejak awal proses pengembangan. Pendekatan ini memungkinkan deteksi dini terhadap ketidaksesuaian antara sistem yang dikembangkan dengan ekspektasi pengguna, sehingga perbaikan dapat dilakukan lebih cepat dan efisien[22].

2.1.5 Android



Gambar 2.4 *Android*

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak dengan layar sentuh, seperti *smartphone* dan tablet. Sistem ini dikembangkan oleh *Android Inc.*, yang kemudian diakuisisi oleh Google pada tahun 2005. Versi komersial pertama Android dirilis pada tahun 2008 sebagai bagian dari Open Handset Alliance, sebuah konsorsium perusahaan teknologi yang bertujuan untuk memajukan standar terbuka dalam perangkat seluler[23]. Arsitektur Android terdiri atas beberapa lapisan, termasuk kernel Linux, pustaka sistem, *Android Runtime* (ART), kerangka kerja aplikasi, dan aplikasi pengguna. ART menggantikan *Dalvik Virtual Machine* sejak Android 5.0, menawarkan peningkatan kinerja melalui kompilasi *ahead of time* (AOT) dan *just in time* (JIT), serta pengelolaan memori yang lebih efisien[24]. Sebagai platform *open-source*, Android memungkinkan pengembang untuk mengakses dan memodifikasi kode sumbernya, yang berkontribusi pada ekosistem aplikasi yang luas dan beragam. Fleksibilitas ini juga memungkinkan produsen perangkat keras untuk menyesuaikan sistem operasi sesuai dengan kebutuhan perangkat mereka[25].

2.1.6 Kotlin

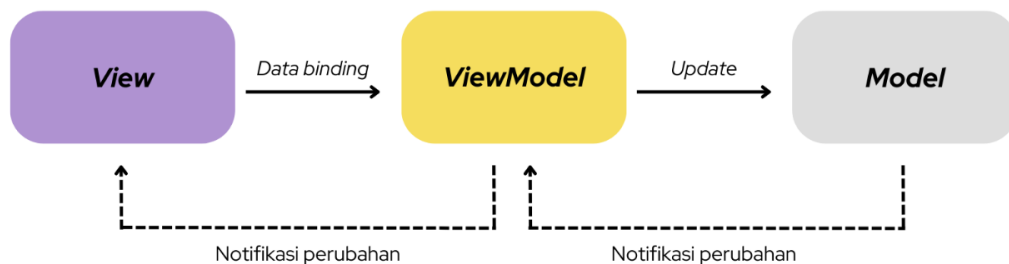


Gambar 2.5 Kotlin

Kotlin adalah bahasa pemrograman modern yang dikembangkan oleh JetBrains, dirilis pertama kali pada tahun 2011, dan berjalan pada *Java Virtual Machine* (JVM). Kotlin dirancang untuk meningkatkan produktivitas pengembang dengan sintaks yang ringkas, fitur *null safety*, serta dukungan untuk paradigma pemrograman berorientasi objek dan fungsional[26]. Google secara resmi mengadopsi Kotlin sebagai bahasa pemrograman utama untuk pengembangan Android pada tahun 2017 karena kemudahannya dalam integrasi dengan proyek Java yang sudah ada dan fitur-fitur modern yang ditawarkan[20]. Salah satu fitur unggulan Kotlin adalah *null safety*, yang dirancang untuk secara signifikan mengurangi risiko *null references*, juga dikenal sebagai "*The Billion-Dollar Mistake*"[27]. Kotlin juga menyediakan *coroutine*, yaitu fitur pemrograman asinkron yang lebih ringan dan mudah dikelola dibandingkan dengan *thread* tradisional. Penelitian menunjukkan bahwa pengembangan aplikasi Android menggunakan Kotlin dapat mempercepat waktu implementasi, meminimalisasi bug terkait *null pointer*, serta meningkatkan keterbacaan kode[28].

2.1.7 Arsitektur MVVM (*Model-View-ViewModel*)

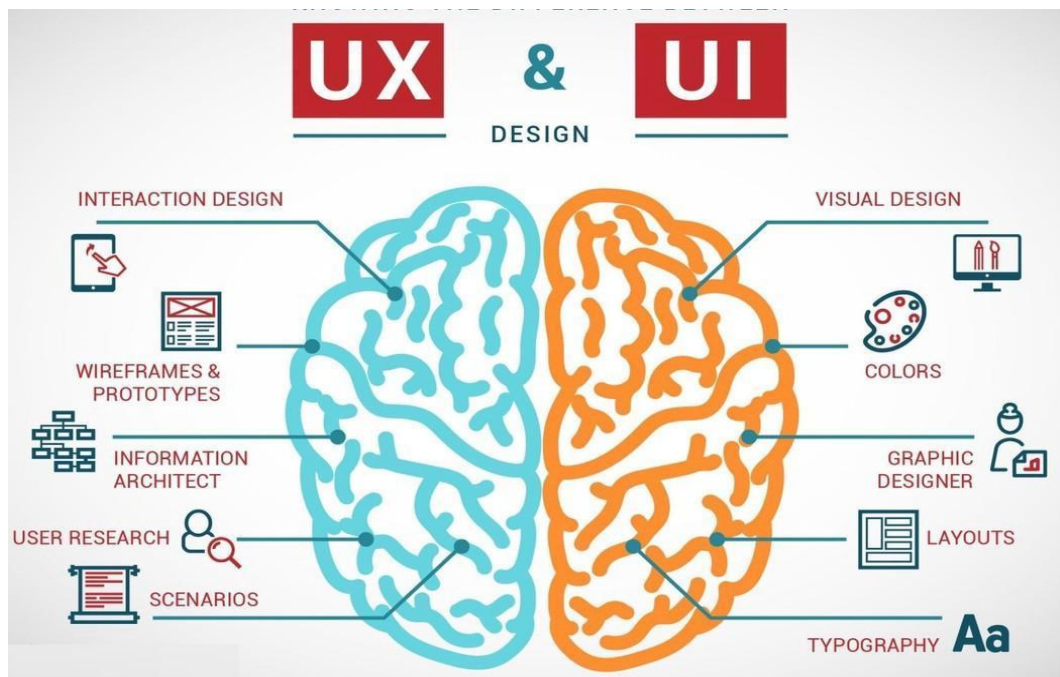
Model-View-ViewModel



Gambar 2.6 Arsitektur MVVM

Model-View-ViewModel (MVVM) merupakan pola arsitektur perangkat lunak yang memisahkan logika bisnis (*Model*), antarmuka pengguna (*View*), dan komponen penghubung (*ViewModel*). MVVM diperkenalkan oleh Microsoft dan banyak diadopsi dalam pengembangan aplikasi Android karena kemudahan pemeliharaan dan skalabilitasnya[29]. Dalam konteks Android, MVVM didukung oleh berbagai komponen seperti *LiveData*, *ViewModel*, dan *Data Binding*. Pola ini memungkinkan perubahan pada data ditampilkan secara otomatis di UI tanpa intervensi manual, yang mempercepat proses pengembangan dan mengurangi kemungkinan *bug*[30]. Studi kasus menunjukkan bahwa penerapan MVVM dalam proyek Android berdampak positif terhadap efisiensi pemrosesan data dan modularitas kode, terutama dalam aplikasi yang kompleks dan dinamis[31].

2.1.8 UI/UX (*User Interface* dan *User Experience*)



Gambar 2.7 *User Interface/User Experience*

User Interface (UI) dan *User Experience* (UX) adalah dua aspek penting dalam desain produk digital. UI merujuk pada elemen-elemen visual dan interaktif yang memungkinkan pengguna berinteraksi dengan perangkat lunak, seperti tombol, menu, dan ikon. Sementara itu, UX mencakup keseluruhan pengalaman pengguna saat berinteraksi dengan produk, termasuk kemudahan penggunaan, kepuasan, dan efisiensi[32]. Desain UI/UX yang baik dapat meningkatkan kepuasan pengguna, mengurangi tingkat kesalahan, dan meningkatkan retensi pengguna. Oleh karena itu, penting bagi pengembang untuk memahami kebutuhan dan preferensi pengguna melalui riset dan pengujian *usability* sebelum merancang antarmuka pengguna[33].

UI dan UX sering kali dianggap sebagai dua sisi dari koin yang sama. UI fokus pada aspek visual dan interaktif dari produk, sedangkan UX berfokus pada bagaimana perasaan dan pengalaman pengguna saat berinteraksi dengan produk tersebut. Keduanya harus bekerja sama untuk menciptakan produk yang tidak hanya menarik secara visual tetapi juga mudah digunakan dan memenuhi kebutuhan pengguna[34].

2.1.9 *Material Design*

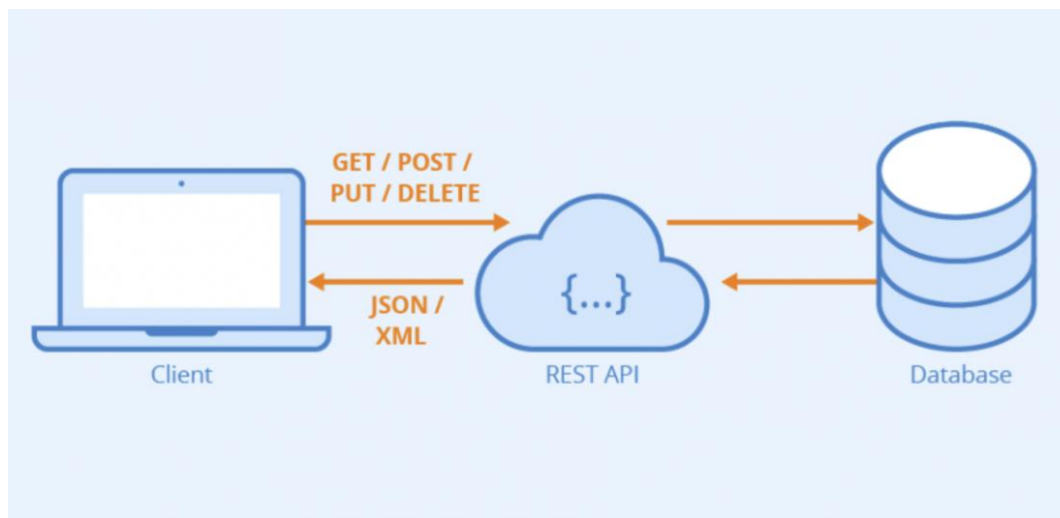
Material Design adalah sistem desain yang dikembangkan oleh Google untuk menciptakan antarmuka pengguna yang konsisten dan estetis di berbagai platform, seperti *Android*, *iOS*, dan web. Sistem desain ini diperkenalkan pertama kali pada konferensi Google I/O 2014 dan bertujuan untuk menyediakan panduan komprehensif bagi pengembang dan desainer dalam membangun aplikasi dengan pengalaman pengguna yang menyenangkan dan mudah digunakan[35]. Salah satu konsep utama dari *Material Design* adalah penggunaan efek kedalaman, seperti bayangan dan pencahayaan, yang memberikan kesan elemen-elemen antarmuka seperti tombol dan kartu bergerak dengan cara yang realistis. Selain itu, *Material Design* juga menekankan penggunaan grid yang fleksibel, animasi responsif, dan transisi yang halus. Dengan prinsip ini, aplikasi menjadi lebih intuitif dan lebih mudah dipahami oleh pengguna[36].



Gambar 2.8 Logo *Material Design*

Pada tahun 2021, *Material Design* diperbarui menjadi versi ketiga, yang dikenal sebagai *Material You*. Versi ini menambahkan tingkat personalisasi yang lebih tinggi, dengan memungkinkan pengguna untuk menyesuaikan tema dan warna antarmuka berdasarkan preferensi pribadi, misalnya dengan menyesuaikan warna tema aplikasi sesuai dengan gambar latar belakang[37].

2.1.10 API (*Application Programming Interface*)

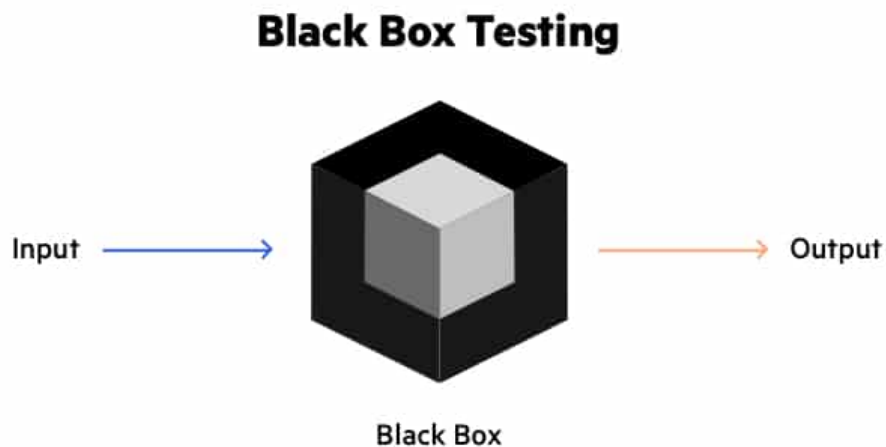


Gambar 2.9 *Application Programming Interface*

Application Programming Interface (API) adalah seperangkat aturan dan protokol yang memungkinkan perangkat lunak untuk berkomunikasi satu sama lain. API mendefinisikan metode dan struktur data yang digunakan untuk pertukaran informasi antar sistem, memungkinkan integrasi dan interoperabilitas aplikasi yang berbeda. Dalam pengembangan perangkat lunak modern, API memainkan peran krusial dalam membangun sistem modular dan *skalabel*. Misalnya, API memungkinkan aplikasi untuk mengakses layanan eksternal seperti pembayaran, peta, atau data cuaca tanpa perlu memahami implementasi internal layanan tersebut[38].

Pentingnya dokumentasi API juga tidak dapat diabaikan. Penelitian menunjukkan bahwa dokumentasi API yang jelas dan mudah dipahami dapat meningkatkan efisiensi pengembang dalam mengintegrasikan dan memanfaatkan API, serta mengurangi kemungkinan kesalahan dalam penggunaan[39].

2.1.11 Blackbox Testing



Gambar 2.10 Blackbox Testing

BlackBox Testing adalah metode pengujian perangkat lunak yang berfokus pada evaluasi fungsionalitas sistem berdasarkan spesifikasi tanpa mempertimbangkan struktur internal atau kode sumber dari perangkat lunak tersebut[40], [41]. Dalam pendekatan ini, penguji hanya mengetahui *input* yang diberikan dan *output* yang diharapkan, tanpa mengetahui bagaimana proses internal menghasilkan output tersebut. Tujuan utama dari *BlackBox Testing* adalah untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan [42].

Beberapa teknik umum yang digunakan dalam *BlackBox Testing* meliputi *Equivalence Partitioning*, *Boundary Value Analysis*, *Cause-Effect Graphing*, dan *State Transition Testing*. Teknik-teknik ini membantu dalam merancang kasus uji yang efektif untuk mengevaluasi berbagai aspek fungsional dari perangkat lunak tanpa memerlukan pengetahuan tentang implementasi internalnya. Penelitian terbaru menunjukkan bahwa *BlackBox Testing* efektif dalam mendeteksi cacat pada aplikasi, terutama ketika digunakan bersama dengan teknik seperti *Equivalence Partitioning*. Misalnya, dalam studi oleh Sasmito dan Mutasodirin, penerapan *BlackBox Testing* dengan teknik *Equivalence Partitioning* pada aplikasi Transcrop

berhasil mengidentifikasi area yang perlu perbaikan untuk meningkatkan kinerja aplikasi [41], [43].

Keunggulan utama dari *BlackBox Testing* adalah kemampuannya untuk dilakukan oleh penguji tanpa pengetahuan mendalam tentang struktur internal perangkat lunak, sehingga memungkinkan pengujian dari perspektif pengguna akhir. Namun, keterbatasannya termasuk kemungkinan tidak terdeteksinya kesalahan dalam logika internal yang tidak tercermin dalam *output* [42].

2.2 Penelitian Terkait

2.2.1 Penelitian Akurasi Diagnosa Penyakit Tanaman Padi menggunakan Kamera dengan Metode Klasifikasi Gambar pada Perangkat Bergerak Android

Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi berbasis Android yang dapat membantu petani dalam mendeteksi penyakit pada tanaman padi. Masalah utama yang dihadapi petani padi adalah gagal panen akibat penyakit yang menyebar tanpa disadari, sehingga mengurangi pasokan beras di Indonesia. Penelitian ini juga bertujuan untuk mengevaluasi pengaruh kualitas kamera Android terhadap deteksi gambar penyakit pada tanaman. Proses pengembangan aplikasi dilakukan dengan menggunakan metodologi SDLC *Waterfall* yang terdiri atas empat tahapan: pengumpulan data, pelatihan model, implementasi sistem, dan pengujian serta analisis. Data dikumpulkan dari website Kaggle yang menyediakan dataset penyakit tanaman padi, dan model *machine learning* dilatih menggunakan *platform Teachable Machine*. Selanjutnya, aplikasi dikembangkan di *platform* Android menggunakan TensorFlow Lite untuk mendeteksi penyakit. Setelah aplikasi dikembangkan, dilakukan pengujian untuk mengevaluasi kinerja aplikasi pada berbagai kelas smartphone, yaitu *low-end*, *mid-end*, dan *high-end*, serta mengukur pengaruh kualitas kamera terhadap akurasi deteksi penyakit.

Hasil dari penelitian ini menunjukkan bahwa aplikasi yang dikembangkan dapat membantu petani dalam mendeteksi penyakit padi lebih cepat dan lebih awal, yang diharapkan dapat mengurangi kerugian akibat gagal panen. Penelitian ini juga memberikan kontribusi dalam pemanfaatan teknologi Android dan *machine*

learning dalam sektor pertanian, khususnya dalam mendukung ketahanan pangan di Indonesia. Melalui penelitian ini, penulis berharap aplikasi ini dapat digunakan oleh petani untuk memantau kondisi tanaman mereka secara lebih efisien dan efektif, serta memberikan solusi praktis dalam mengatasi penyakit tanaman yang dapat mengancam hasil pertanian. Dengan demikian, penelitian ini tidak hanya menunjukkan penerapan teknologi di bidang pertanian, tetapi juga bagaimana teknologi *mobile* dapat dimanfaatkan untuk meningkatkan kualitas dan hasil pertanian, serta mendukung keberlanjutan sektor pertanian di Indonesia[44].

2.2.2 Aplikasi GreenAdvisor Sebagai Pendeteksi dan Rekomendasi Tanaman

Berdasarkan Koordinat Berbasis Android Menggunakan Design Pattern

Penelitian ini mengembangkan aplikasi berbasis Android bernama GreenAdvisor yang bertujuan untuk membantu petani dan pecinta tanaman dalam memilih tanaman yang sesuai dengan kondisi lingkungan, khususnya berdasarkan koordinat geografis. Aplikasi ini dikembangkan menggunakan konsep *Model-View-ViewModel* (MVVM) dan bahasa pemrograman Kotlin untuk memastikan kemudahan dalam pengembangan serta pemeliharaan aplikasi. Selain itu, aplikasi ini juga memanfaatkan teknologi *cloud computing* yang terintegrasi dengan fitur pengenalan gambar menggunakan *machine learning* untuk mendeteksi tanaman.

The image shows two side-by-side wireframes for a mobile application, enclosed in a green border. Both pages feature a globe icon in the top right corner and an illustration of a person using a smartphone on the left.

Left Page (Register):

- Greeting: **Selamat Datang,**
- Instruction: **Silahkan Buat Akun Untuk Masuk Ke Aplikasi**
- Form fields:
 - Nama:** Input field with placeholder "Green Advisor".
 - Email:** Input field with placeholder "Capstone2023@gmail.com".
 - Password:** Input field with masked characters "*****".
 - Confirm Password:** Input field with placeholder "Ketik Ulang Password".
- Link: [Sudah punya akun?](#)
- Button: **Daftar**

Right Page (Login):

- Confirmation: **Thank you!**
- Message: **Your account has been successfully registered.**
- Form fields:
 - Password:** Input field with masked characters "*****".
 - Confirm Password:** Input field with masked characters "*****".
- Link: [Sudah punya akun?](#)

Gambar 2.11 Rancangan Halaman *Register* dan *Login*

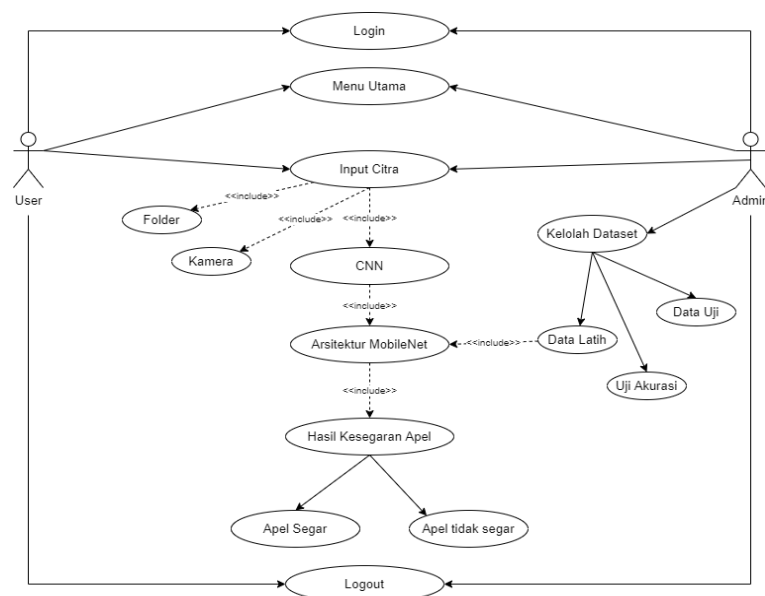
Pada tahap awal penelitian, dilakukan pengumpulan data mengenai tanaman dan kondisi lingkungan setempat yang digunakan dalam aplikasi. Penelitian ini berfokus pada pengembangan aplikasi yang tidak hanya memberikan rekomendasi tanaman berdasarkan kondisi tanah dan iklim, tetapi juga memungkinkan pengguna untuk mempelajari lebih lanjut mengenai tanaman tersebut melalui antarmuka yang sederhana dan mudah digunakan. Aplikasi ini bertujuan untuk memberikan solusi praktis bagi para petani yang kesulitan dalam menentukan jenis tanaman yang cocok untuk ditanam di daerah tertentu, mengingat perbedaan ketinggian dan kondisi iklim yang mempengaruhi hasil pertanian.

Metode yang digunakan dalam pengembangan aplikasi ini adalah metode *prototype*, di mana prototipe awal aplikasi dikembangkan untuk diuji coba dengan melibatkan pengguna. Pengujian dilakukan dengan menggunakan *functional testing* dan *performance evaluation*, serta diikuti dengan survei untuk mengevaluasi tingkat kepuasan pengguna. Hasil pengujian menunjukkan bahwa aplikasi ini sangat efektif dalam memberikan rekomendasi tanaman yang sesuai dengan kondisi

lingkungan pengguna, serta memberikan informasi yang berguna bagi petani dalam meningkatkan hasil pertanian mereka[9].

2.2.3 Analisis Klasifikasi Kesegaran Buah Apel Menggunakan Metode *Convolutional Neural Network (CNN)* Berbasis *Android*

Penelitian ini mengembangkan aplikasi *Android* untuk mengklasifikasikan tingkat kematangan buah apel menggunakan metode *Convolutional Neural Network (CNN)*. Buah apel merupakan salah satu komoditas pertanian penting, dan menentukan tingkat kematangan apel secara manual sering memakan waktu dan subjektif. Dengan menggunakan CNN, aplikasi ini dapat secara otomatis mengklasifikasikan tingkat kematangan apel dari gambar yang diambil dengan *smartphone* meningkatkan efisiensi dan akurasi dalam penentuan kualitas buah apel. Data citra apel pada berbagai tingkat kematangan dikumpulkan dan diproses untuk melatih model CNN yang kemudian dioptimalkan untuk aplikasi *Android*. Hasil dari model ini memungkinkan aplikasi mengenali tingkat kematangan apel dengan tingkat akurasi yang tinggi, memberikan manfaat signifikan bagi petani apel dalam mengelola panen mereka. Aplikasi ini juga menawarkan kemudahan bagi pengguna untuk mendapatkan informasi terkait kesegaran buah apel dengan cepat.



Gambar 2.12 Use Case Diagram Aplikasi

Aplikasi yang dikembangkan memberikan solusi praktis dalam manajemen panen, mengurangi ketergantungan pada penilaian subjektif dan meningkatkan kualitas produk apel. Hasil pengujian menunjukkan bahwa aplikasi ini efektif dalam memberikan informasi yang akurat dan cepat, sehingga dapat mendukung petani dalam proses panen apel yang lebih efisien. Dengan demikian, penggunaan CNN dalam aplikasi Android menawarkan alternatif yang lebih baik dibandingkan dengan metode manual, memperkenalkan cara baru dalam mengelola kualitas buah dengan teknologi canggih. Aplikasi ini dapat diimplementasikan secara luas untuk komoditas pertanian lainnya yang memerlukan penilaian kualitas berdasarkan citra[45].

2.2.4 Pengembangan Aplikasi Mobile Pendeteksi Penyakit pada Tanaman Cabai dengan menggunakan Teknologi Clarifai

Penelitian ini mengembangkan aplikasi mobile berbasis Android yang dirancang untuk mendeteksi penyakit pada tanaman cabai dengan memanfaatkan teknologi Clarifai, sebuah platform *Computer Vision* yang menggunakan pendekatan *Deep Convolutional Neural Network* (CNN). Tujuan aplikasi ini adalah untuk mengatasi kendala utama petani di Indonesia, yaitu kurangnya pengetahuan dalam mengenali jenis penyakit yang menyerang tanaman cabai. Teknologi Clarifai memungkinkan aplikasi mengenali dan mengklasifikasikan penyakit tanaman secara otomatis melalui gambar yang diambil dari kamera perangkat *smartphone* atau galeri pengguna. Metode pengembangan aplikasi yang digunakan dalam penelitian ini adalah metode *Waterfall*, yang dilakukan secara terstruktur mulai dari analisis kebutuhan, perancangan sistem menggunakan arsitektur *Model View Presenter* (MVP), hingga implementasi menggunakan bahasa pemrograman *Java*. Dalam proses implementasi, aplikasi terintegrasi dengan layanan *hosting* gambar *Imgur* untuk penyimpanan gambar serta menggunakan teknologi Clarifai untuk identifikasi penyakit cabai. Clarifai bekerja melalui tiga tahapan yaitu *Define* (pendefinisian label penyakit), *Train* (pelatihan model), dan *Recognize* (identifikasi penyakit dari gambar yang diunggah).



Gambar 2.13 Tampilan Halaman Menu Utama Aplikasi

Hasil pengujian menunjukkan bahwa aplikasi memiliki tingkat validasi fitur fungsional sebesar 100%, tingkat akurasi pendeteksian penyakit sebesar 79%, dan nilai *usability* sebesar 82,75% (kategori sangat baik). Pengujian kompatibilitas juga sukses, dengan aplikasi yang dapat berjalan baik pada perangkat Android minimal SDK level 23. Dengan demikian, aplikasi ini memberikan solusi praktis bagi petani dalam meningkatkan efisiensi pengelolaan penyakit tanaman cabai melalui teknologi *mobile* berbasis kecerdasan buatan[46].

2.2.5 Sistem Deteksi Penyakit Daun Apel Menggunakan Metode CNN Berbasis Mobile

Penelitian ini mengembangkan sistem deteksi penyakit daun apel berbasis mobile menggunakan algoritma *Convolutional Neural Network* (CNN) dengan arsitektur EfficientNet-B7 yang diimplementasikan dalam aplikasi menggunakan teknologi

Flutter. Sistem ini bertujuan mendeteksi secara dini berbagai penyakit daun apel, termasuk *Black Rot*, *Cedar Rust*, *Scab*, serta klasifikasi daun apel yang sehat. Data yang digunakan dalam penelitian ini sebanyak 8000 citra daun apel, yang diklasifikasikan ke dalam beberapa kategori penyakit untuk melatih model CNN secara efektif.

Metode pengembangan aplikasi meliputi tahap pengumpulan dan *preprocessing* data, pelatihan model CNN menggunakan arsitektur *EfficientNet-B7*, evaluasi model, dan implementasi dalam bentuk aplikasi mobile yang dirancang dengan antarmuka pengguna sederhana dan interaktif. Model CNN diuji menggunakan berbagai konfigurasi *optimizer* seperti SGD, Adam, dan RMSProp untuk menentukan hasil terbaik. Hasil pengujian menunjukkan model CNN mampu menghasilkan tingkat akurasi klasifikasi yang tinggi, dengan nilai *accuracy* tertinggi mencapai 99%. Dalam tahap implementasi, aplikasi memungkinkan pengguna mengunggah gambar daun apel langsung melalui perangkat mobile untuk kemudian diklasifikasikan secara otomatis dan cepat.



Gambar 2.14 Tampilan Halaman Deteksi Aplikasi

Hasil klasifikasi ditampilkan dalam antarmuka yang informatif sehingga pengguna dapat segera mengetahui kondisi daun apel tersebut. Dengan hasil pengujian akurasi yang tinggi serta kemudahan penggunaan, aplikasi ini memberikan solusi praktis bagi petani dalam melakukan diagnosa dini penyakit daun apel, membantu peningkatan produksi dan kualitas hasil panen melalui pendekatan teknologi berbasis kecerdasan buatan[47].

2.2.6 Implementasi *Convolutional Neural Network* Menggunakan Arsitektur *Xception* Untuk Identifikasi Penyakit Daun Pada Tanaman Durian

Penelitian ini mengembangkan model klasifikasi penyakit daun durian menggunakan algoritma *Convolutional Neural Network* (CNN), sebagai upaya untuk membantu petani dalam mengenali jenis penyakit sejak dini melalui

pemrosesan citra daun. Sistem ini dibangun menggunakan bahasa pemrograman *Python* dengan pemanfaatan platform *Google Colab* untuk pelatihan model, serta dilakukan serangkaian eksperimen untuk mendapatkan arsitektur dan konfigurasi model terbaik yang mampu mendeteksi penyakit secara optimal. Penelitian ini menjadi penting karena pengenalan penyakit secara manual memerlukan keahlian khusus dan memakan waktu, sementara pendekatan otomatis ini dinilai lebih efisien.

NO.	Kelas	Jumlah Data	Total Dataset	Resolusi	Ukuran File
1.	<i>Algal Leaf Spot</i>	733	4.437	224 x 224	12 – 15 kB
2.	<i>Phomopsis Leaf Spot</i>	878			
3.	<i>Leaf Blight</i>	937			
4.	<i>Healthy Leaf</i>	976			
5.	<i>Allocaridara Attack</i>	913			

Gambar 2.15 Total Dataset

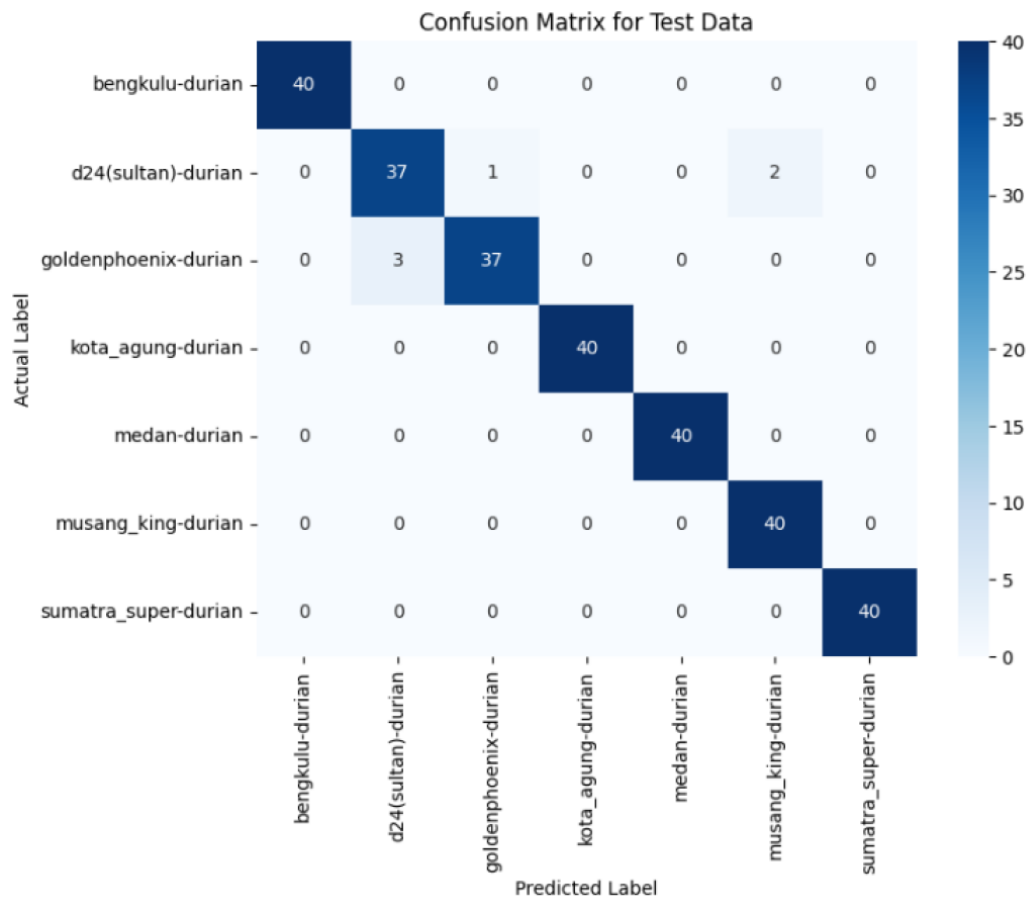
Total Dataset yang digunakan dalam penelitian ini berjumlah 4.437 gambar daun durian yang terbagi ke dalam lima kelas, yaitu *Algal Leaf Spot* (733 gambar), *Phomopsis Leaf Spot* (878 gambar), *Leaf Blight* (937 gambar), *Healthy Leaf* (976 gambar), dan *Allocaridara Attack* (913 gambar). Setiap gambar memiliki resolusi 224x224 piksel dengan ukuran file antara 12–15 kB. Dataset ini kemudian dibagi menjadi tiga bagian: 70% untuk data latih (*training*), 10% untuk data validasi (*Validate*) dan 20% untuk data uji (*testing*). Selain itu, teknik augmentasi data digunakan untuk memperluas variasi dataset dan mencegah *overfitting* saat proses pelatihan model.

Model CNN dirancang dengan beberapa lapisan *convolutional*, *pooling*, dan *fully connected layer* yang dikombinasikan dengan fungsi aktivasi *ReLU* serta optimasi menggunakan algoritma Adam. Evaluasi performa dilakukan menggunakan metrik

seperti akurasi, presisi, *recall*, dan *confusion matrix*. Hasil terbaik menunjukkan bahwa model CNN ini mampu mencapai akurasi yang sangat tinggi dalam mengenali kelima kelas daun durian, menunjukkan bahwa pendekatan ini sangat potensial untuk diterapkan dalam sistem deteksi penyakit berbasis citra. Model yang telah dilatih kemudian diekspor dalam format yang kompatibel untuk digunakan pada penelitian ini.

2.2.7 Implementasi *Convolutional Neural Network* Berbasis Arsitektur *Efficientnet-B0* Untuk Klasifikasi Varietas Buah Durian

Penelitian ini bertujuan untuk mengembangkan model klasifikasi jenis buah durian menggunakan pendekatan *Convolutional Neural Network* (CNN), yang dilatih dengan dataset citra dari tujuh jenis durian berbeda. Dataset awal terdiri atas total 1.050 gambar yang tidak seimbang antara kelas, seperti Musang King 200, Golden Phoenix 200, D24 Sultan 200, sedangkan kelas Medan hanya 156, Bengkulu 114, Kota Agung 123, dan Sumatra Super 57. Ketimpangan ini diatasi melalui proses augmentasi citra menggunakan teknik seperti rotasi, *flipping*, dan *zooming* untuk meningkatkan jumlah gambar secara proporsional hingga menjadi total 1.400 gambar (200 gambar per kelas).



Gambar 2.16 *Confusion matrix Epoch 40*

Model CNN yang digunakan dalam penelitian ini terdiri atas tiga lapisan konvolusi dan tiga lapisan *max-pooling*, dilanjutkan dengan dua lapisan *fully connected*. Pelatihan model dilakukan menggunakan dataset yang telah diaugmentasi, dengan proses validasi dan evaluasi model yang menunjukkan akurasi sebesar 95% pada data validasi. Evaluasi performa juga mencakup *confusion matrix* untuk mengukur keakuratan prediksi pada setiap kelas durian, di mana model menunjukkan performa yang cukup konsisten antar kelas. Model ini berhasil mempelajari karakteristik visual dari masing-masing jenis durian seperti bentuk duri, warna kulit, dan kontur buah, yang menjadi fitur utama dalam klasifikasi.

III. METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Pelaksanaan penelitian dan pengembangan alat ini dilakukan pada waktu dan tempat yang telah ditentukan sebagai berikut :

Waktu : Mei 2025 – September 2025

Tempat : Laboratorium Teknik Komputer

Tabel 3.1 Waktu Penelitian

No	Kegiatan	Bulan						
		Juni	Juli	Agustus	September	Oktober	November	Desember
1	Studi Literatur							
2	Pengumpulan Kebutuhan							
3	SDLC Kanban							
4	Penulisan Laporan							

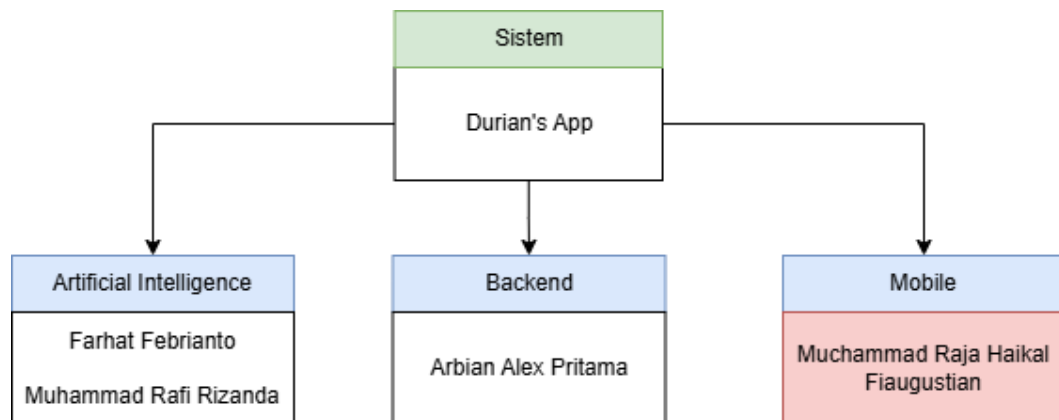
3.2 Alat dan Bahan Penelitian

Adapun alat dalam penelitian ini adalah sebagai berikut :

Tabel 3.2 Alat Penelitian

No.	Nama Alat	Spesifikasi	Deskripsi
1.	Laptop HP Spectre	Intel® Core™ i7-1250U, Memory 16GB LPDDR5, Storage 1 TB, Windows 11 64bit.	Perangkat keras yang digunakan untuk seluruh proses penelitian sampai dengan selesai
2.	Mouse	Logitech G102	Perangkat keras yang digunakan untuk menggerakkan kursor
3.	Android Studio	Koala	Perangkat lunak editor kode yang digunakan untuk pengembangan aplikasi.
4.	Kotlin	2.1.21	Bahasa pemrograman untuk pengembangan aplikasi.

3.3 Struktur Tim



Gambar 3.1 Struktur Tim Peneliti

Penelitian ini akan dilakukan oleh tim yang terdiri atas 4 orang yang secara individu mengerjakan bagian aplikasi terpisah namun saling terhubung. Bagian aplikasi yang dimaksud adalah *machine learning development*, *front end development*, dan *back end development*. Masing-masing anggota tim akan fokus mengerjakan bagiannya. Dua pengembang di bagian *machine learning development* yaitu Farhat Febrianto dan Muhammad Rafi Rizanda, satu pengembang di bagian ***front end development*** yaitu yaitu Muchammad Raja Haikal Fiaugustian, dan satu pengembang API *back end* yaitu Arbian Alex Pritama.

3.5 Studi literatur

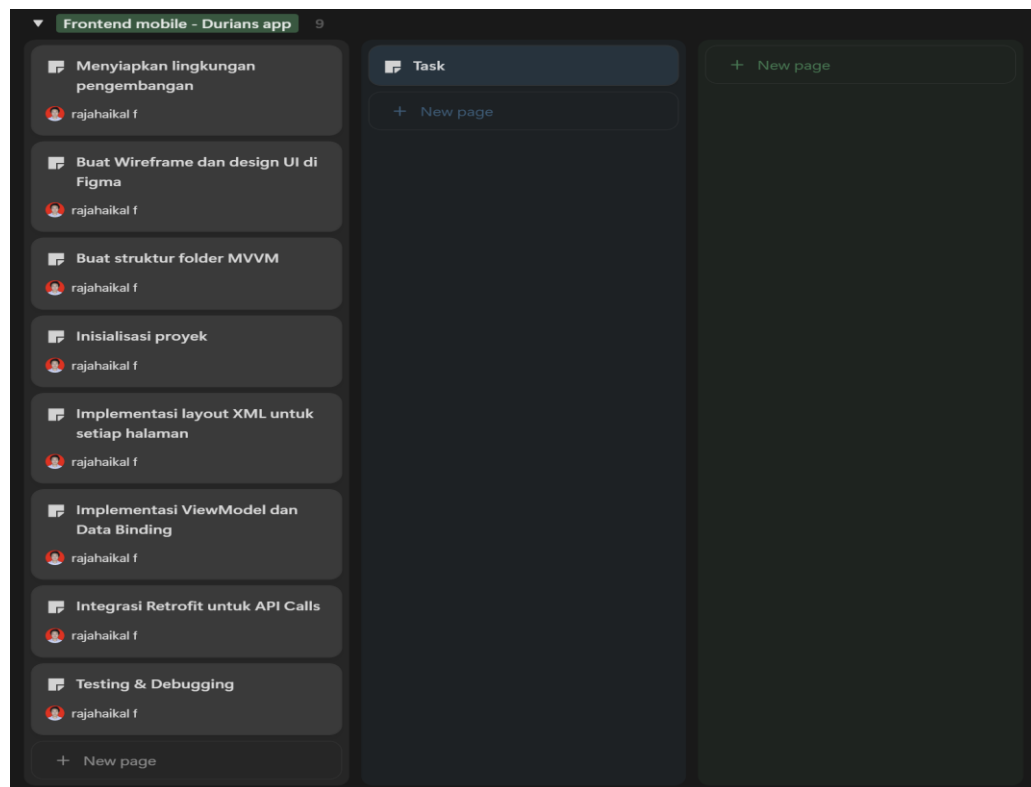
Tahapan studi literatur merupakan kegiatan awal dalam penelitian ini yang bertujuan mengumpulkan dan menganalisis berbagai sumber referensi untuk memperoleh landasan teoritis yang kuat terkait dengan topik penelitian. Dalam tahapan ini, peneliti melakukan kajian terhadap artikel ilmiah, jurnal, buku, maupun publikasi resmi lainnya yang relevan dengan penelitian yang dilakukan. Proses ini mencakup identifikasi dan pemilihan literatur yang sesuai, analisis secara mendalam terhadap teori-teori yang mendukung penelitian, serta pemetaan konsep-konsep penting terkait dengan topik pengembangan aplikasi mobile untuk klasifikasi jenis durian dan deteksi penyakit tanaman durian. Hasil dari tahapan ini adalah tersedianya dasar teori yang jelas, terstruktur, dan komprehensif sebagai acuan utama untuk mendukung tahapan selanjutnya dalam penelitian ini.

3.6 SDLC menggunakan Kanban

Kanban merupakan metode manajemen alur kerja yang berfokus pada visualisasi proses dan pembatasan jumlah pekerjaan yang sedang berlangsung (*Work in Progress/WIP*), sehingga dapat meningkatkan efisiensi dan mengurangi hambatan dalam proses pengembangan. SDLC dibagi ke dalam beberapa tahapan utama yang divisualisasikan dalam papan Kanban menggunakan aplikasi *Notion*, yaitu :

- a. **Backlog**: Berisi daftar seluruh kebutuhan atau fitur aplikasi yang telah dikumpulkan dari tahap *user stories* dan *requirements gathering*. Setiap item dalam *backlog* merupakan representasi dari kebutuhan pengguna yang telah dirumuskan dalam format yang jelas.
- b. **To Do**: Merupakan item yang telah diprioritaskan dari *backlog* dan siap untuk dikerjakan. Tahapan ini menjadi penanda bahwa fitur tersebut akan segera masuk ke proses pengembangan.
- c. **Work in Progress (WIP)**: Tahapan ini berisi fitur atau tugas yang sedang dalam proses pengembangan aktif oleh tim. Pada tahap ini, pengembang mulai menerjemahkan kebutuhan pengguna menjadi implementasi teknis dalam bentuk kode program dan antarmuka aplikasi.

- d. **Done:** Tugas yang telah selesai dan berhasil diuji akan masuk ke tahap selesai. Fitur ini dinyatakan siap digunakan oleh pengguna dan dapat dirilis sebagai bagian dari versi aplikasi.



Gambar 3.3 Kanban board

Dengan metode Kanban memudahkan tim untuk memantau progres pengembangan secara visual dan memungkinkan perubahan prioritas secara fleksibel selama proses berjalan.

3.6.1 User's Stories

Tahapan *User Stories* dilakukan setelah studi literatur, dengan tujuan mendefinisikan kebutuhan pengguna secara spesifik melalui perspektif pengguna itu sendiri. Pada tahap ini, peneliti mengidentifikasi kebutuhan serta harapan pengguna dalam penggunaan aplikasi yang dirancang. *User story* dirumuskan berdasarkan hasil diskusi dengan petani durian di Bumi Kedaton dan tim pengembang. Tahapan ini dilakukan dengan pendekatan berbasis narasi singkat dari

sisi pengguna yang mencerminkan fungsionalitas serta interaksi pengguna dengan aplikasi. Setiap *user story* akan berisi deskripsi singkat yang menunjukkan siapa pengguna (*user*), apa yang diinginkan pengguna (fitur atau kemampuan aplikasi), serta alasan atau tujuan di balik kebutuhan tersebut. Setiap *user story* yang dikembangkan pada tahapan ini akan dilanjutkan ke tahap *requirements gathering*, lalu masuk ke dalam *product backlog* agar bisa dikembangkan secara bertahap pada fase berikutnya dari penelitian ini.

Tabel 3.3 *User Story*

No.	ID	User Story
1.	US-01	Sebagai pengguna, saya ingin dapat <i>login</i> agar saya bisa mengakses fitur aplikasi secara personal.
2.	US-02	Sebagai pengguna, saya ingin dapat memindai gambar daun durian agar saya bisa mengetahui apakah tanaman saya terkena penyakit.
3.	US-03	Sebagai pengguna, saya ingin dapat memindai gambar buah durian agar saya bisa mengetahui jenis durian.
4.	US-04	Sebagai pengguna, saya ingin melihat hasil klasifikasi secara langsung setelah memindai gambar.
5.	US-05	Sebagai pengguna, saya ingin menyimpan dan melihat riwayat scan sebelumnya.
6.	US-06	Sebagai pengguna, saya ingin melihat informasi cuaca terkini yang berkaitan dengan lokasi saya.

3.6.2 *Requirements Gathering*

Tahapan *requirements gathering* merupakan kelanjutan dari tahapan *user stories*, di mana setiap cerita pengguna yang telah dibuat akan dianalisis lebih mendalam untuk mendapatkan spesifikasi kebutuhan secara detail. Dalam tahapan ini, kebutuhan pengguna yang sebelumnya dirumuskan melalui *user stories* dikaji kembali oleh tim pengembang untuk menentukan detail teknis, cakupan fitur, serta kelayakan implementasi fitur tersebut dalam aplikasi. Kemudian dari hasil diskusi, diperoleh daftar kebutuhan pengguna yang akan diterapkan dalam perancangan sistem ini. Selanjutnya, kebutuhan pengguna tersebut diuraikan lebih lanjut menjadi kebutuhan fungsional dan nonfungsional. Adapun daftar kebutuhan pengguna hasil dari diskusi tim perancangan dengan petani adalah sebagai berikut:

1. Dapat mengklasifikasikan jenis durian berdasarkan gambar buah yang dipindai.
2. Dapat mendeteksi penyakit daun durian melalui pemindaian daun.

3. Dapat melihat informasi cuaca terkini untuk membantu dalam perawatan pohon durian.
4. Dapat melihat harga durian di pasaran sebagai acuan penjualan.
5. Dapat menyimpan dan melihat riwayat hasil scan untuk memantau kondisi tanaman.
6. Dapat membuat catatan (*notes*) untuk kegiatan perawatan pohon durian.

Adapun daftar kebutuhan fungsional hasil dari diskusi bersama dengan tim perancangan adalah sebagai berikut:

Tabel 3.4 Kebutuhan Fungsional

ID	Penjelasan
KF-01	Sistem dapat melakukan proses login dan logout untuk pengguna terdaftar.
KF-02	Sistem dapat melakukan pemindaian gambar buah durian
KF-03	Sistem dapat melakukan pemindaian gambar daun durian
KF-04	Sistem dapat menampilkan hasil klasifikasi jenis durian kepada pengguna.
KF-05	Sistem dapat menampilkan jenis penyakit daun yang terdeteksi kepada pengguna.
KF-06	Sistem dapat memungkinkan pengguna untuk membuat catatan
KF-07	Sistem dapat mengambil data cuaca

Adapun daftar kebutuhan non-fungsional hasil dari diskusi bersama dengan tim perancangan adalah sebagai berikut:

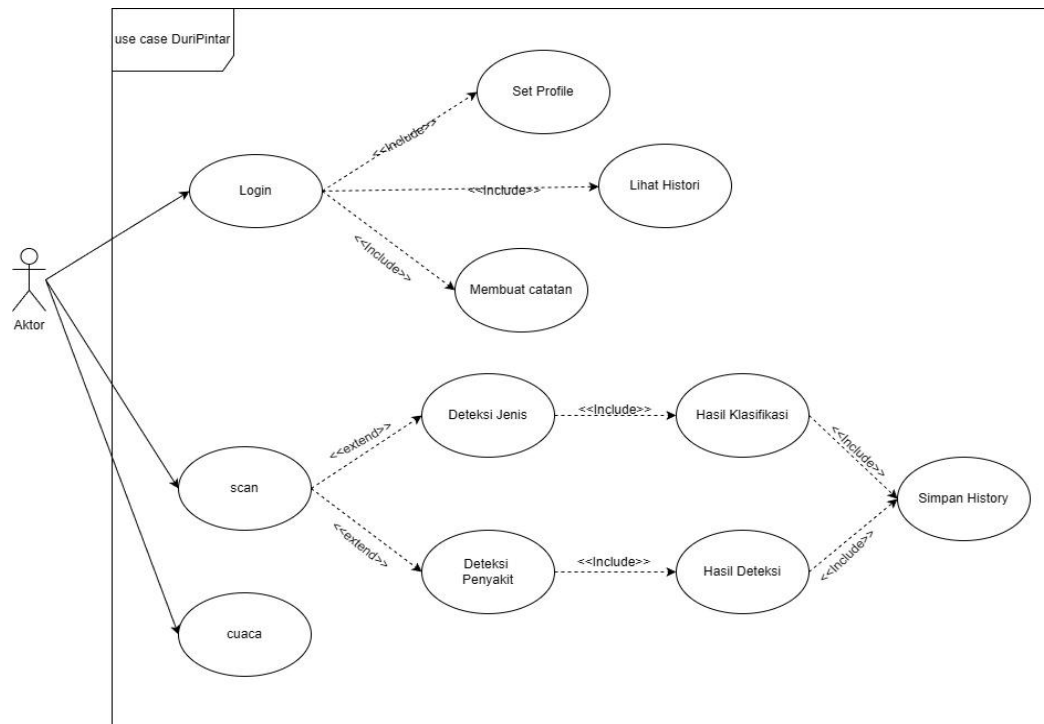
Tabel 3.5 Kebutuhan Non Fungsional

ID	Parameter	Penjelasan
KnF-01	<i>Availability</i>	1. Sistem dapat tersedia dan dapat diakses oleh pengguna selama 24 jam dalam sehari
KnF-02	<i>Reliability</i>	1. Sistem dapat memberikan hasil klasifikasi durian dan deteksi penyakit daun. 2. Sistem dapat menyimpan dan menampilkan riwayat hasil scan pengguna.
KnF-03	<i>Ergonomy</i>	1. Sistem dapat memiliki antarmuka pengguna (UI) yang mudah dipahami. 2. Sistem dapat menggunakan tata letak dan elemen UI yang konsisten untuk memudahkan navigasi pengguna.
KnF-04	<i>Portability</i>	1. Sistem dapat diinstal dan dijalankan pada perangkat Android dengan versi minimal Android 6.0 (Marshmallow).
KnF-05	<i>Safety</i>	1. Sistem harus melakukan validasi pada setiap data yang dimasukkan pengguna untuk mencegah kesalahan input atau data yang tidak valid
KnF-06	<i>Usability</i>	1. Sistem dapat menampilkan panduan penggunaan aplikasi bagi pengguna baru (<i>onboarding</i>).

3.6.3 Unified Modeling Language

3.6.3.1 Use Case Diagram

Adapun *use case* dari aplikasi yang dirancang adalah sebagai berikut:



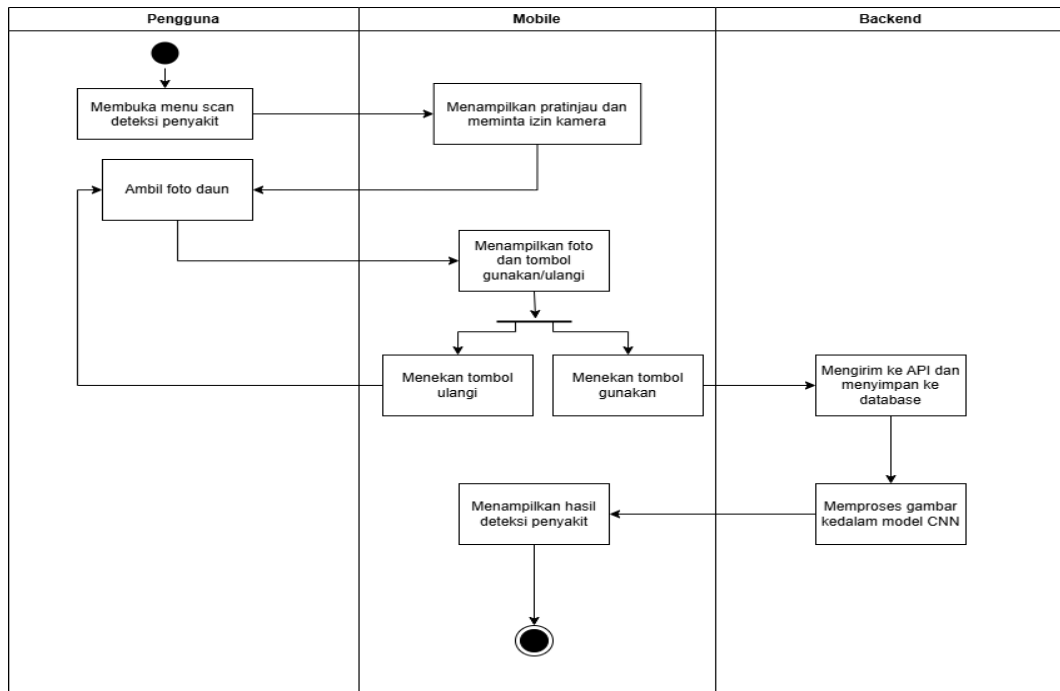
Gambar 3.4 Use Case Diagram

3.6.3.2 Activity Diagram

Activity Diagram adalah diagram yang menggambarkan alur aktivitas atau proses dalam sistem, mulai dari awal hingga akhir, termasuk keputusan atau percabangan yang terjadi. Diagram ini membantu memahami bagaimana aktivitas saling terkait dan berurutan dalam sistem.

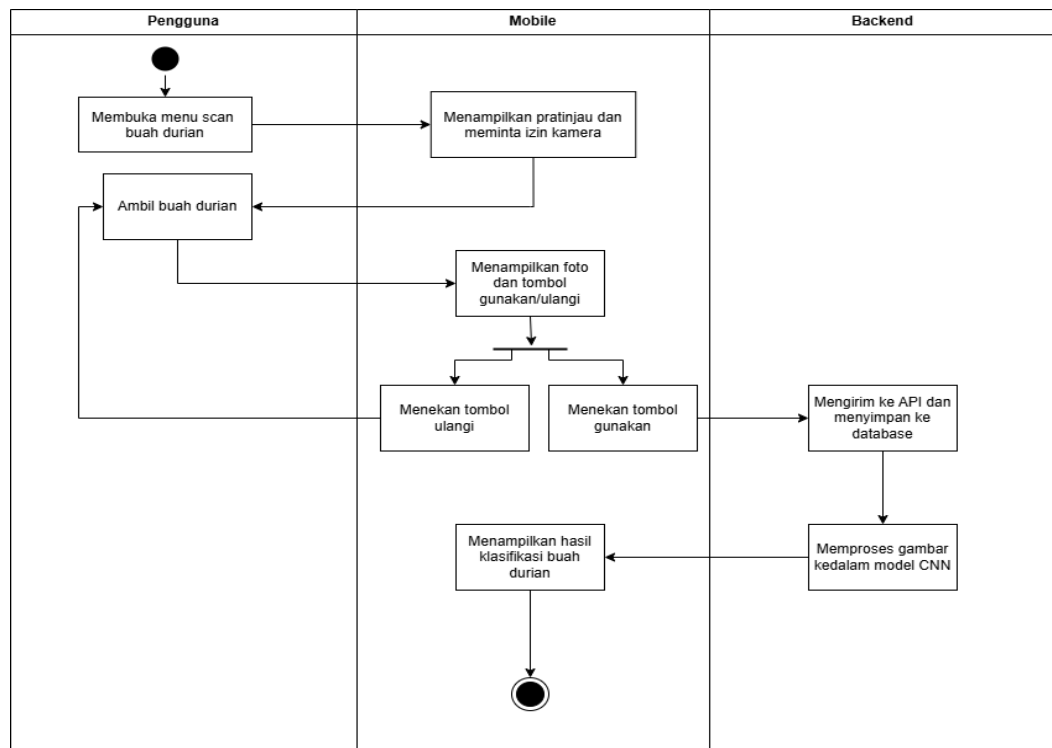
a. *Activity Diagram* Daftar

c. *Activity Diagram* Deteksi Penyakit Durian



Gambar 3.7 Activity Diagram Deteksi Penyakit Durian

d. *Activity Diagram* Deteksi Klasifikasi Durian



Gambar 3.8 Activity Diagram Deteksi Klasifikasi Durian

3.6.4 Product Backlog

Pada tahap ini, dilakukan diskusi untuk menyusun *product backlog* dengan merujuk pada daftar kebutuhan pengguna yang telah diidentifikasi sebelumnya. Diskusi diawali dengan menetapkan *product goal* sebagai tujuan utama yang ingin dicapai oleh produk. *Backlog* bersifat dinamis, artinya isi *backlog* dapat ditambahkan seiring perjalanan proyek semisal ada kebutuhan baru, ataupun terdapat umpan balik dari pihak terkait. Selanjutnya, item-item *backlog* yang mencakup fitur-fitur utama yang perlu dikembangkan, tugas-tugas teknis dan non-teknis, disusun guna mendukung proses pengembangan aplikasi.

3.6.5 Work in Progress

Work in Progress (WIP) adalah tahap dalam metode Kanban yang menggambarkan pekerjaan yang sedang dikerjakan oleh tim pengembangan aplikasi *mobile*. Pada tahap ini, setiap task yang telah direncanakan dalam *Product Backlog* dipindahkan ke WIP ketika pengerjaannya dimulai. Pengerjaan/pembangunan aplikasi *mobile* ini menggunakan aplikasi *Android Studio*, karena proses SDLC pada aplikasi ini menggunakan metode Kanban jumlah tugas yang bisa dikerjakan dalam satu waktu harus dibatasi. Dalam kasus ini, hal yang membatasi jumlah pekerjaan dalam proses adalah jumlah anggota tim *mobile development*. Oleh karena hanya terdiri atas satu orang, jumlah yang bisa dikerjakan dalam satu waktu adalah satu tugas.

3.6.6 Software In Test

Tahapan *software in test* dilakukan setelah seluruh fitur dalam aplikasi berhasil dikembangkan pada tahap work in progress. Dalam tahapan ini, pengujian aplikasi dilaksanakan dengan metode pengujian *blackbox testing*, yaitu metode yang menguji aplikasi berdasarkan fungsionalitas eksternal tanpa melihat struktur kode atau implementasi internalnya. Pengujian dilakukan dengan memberikan *input* tertentu ke dalam sistem, kemudian diamati hasil keluaran atau respons sistem untuk memastikan bahwa aplikasi berjalan sesuai dengan spesifikasi kebutuhan yang telah ditetapkan sebelumnya.

Tabel 3.6 *Test Case Blackbox Testing*

Id Test	Skenario	Input	Hasil yang diharapkan
TC-01	Pengguna mendeteksi jenis durian dengan foto yang jelas	Foto buah durian yang jelas	Sistem menampilkan jenis durian dengan benar
TC-02	Pengguna mendeteksi penyakit pada tanaman buah durian	Foto daun durian	Sistem menampilkan hasil penyakit durian dengan benar
TC-03	Pengguna dapat melihat riwayat deteksi setelah login	Login dengan akun terdaftar	Riwayat deteksi ditampilkan dengan benar dan jelas
TC-04	Pengguna dapat membuat catatan pengingat	Input tanggal, waktu, dan catatan	Reminder berhasil dibuat dan muncul notifikasi
TC-05	Pengguna dapat melihat informasi cuaca di wilayah pengguna	Mengizinkan lokasi dan Membuka halaman utama	Informasi cuaca terkini muncul sesuai lokasi pengguna

Setiap skenario pengujian dalam metode *blackbox testing* disusun berdasarkan kebutuhan dan *user stories* yang sudah dibuat pada tahapan sebelumnya. Fokus utama pengujian ini adalah validasi terhadap fungsionalitas dan interaksi pengguna dengan aplikasi secara keseluruhan. Jika ditemukan kesalahan atau ketidaksesuaian dalam hasil pengujian, maka hasil tersebut dicatat untuk diperbaiki pada tahapan selanjutnya. Keluaran dari tahapan *software in test* ini berupa laporan hasil pengujian yang detail, termasuk status fitur-fitur aplikasi yang telah diuji. Tahapan ini sangat penting untuk memastikan bahwa aplikasi benar-benar memenuhi kebutuhan dan ekspektasi pengguna sebelum lanjut ke tahap evaluasi pengguna akhir atau *User Acceptance Testing* (UAT).

3.6.7 *User Acceptance Test* (UAT)

Pada tahap ini sistem yang telah selesai dirancang diuji secara menyeluruh kepada pengguna akhir (*end user*) guna memastikan kualitas dan kesesuaian terhadap spesifikasi yang telah ditetapkan. Untuk mengumpulkan kebutuhan tersebut dilakukan survei di lokasi budidaya durian yang terkenal di Bandarlampung yaitu Kelurahan Bumi Kedaton Kecamatan Teluk Betung Utara. Sampel pada penelitian ini menggunakan *purposif sampling* yaitu memilih responden secara sengaja berdasarkan kriteria tertentu. Responden berjumlah 10 orang *end user* yang memiliki pengalaman atau keterlibatan langsung dalam penggunaan aplikasi untuk klasifikasi dan deteksi tanaman durian. Populasi sasaran yang dipilih adalah seluruh

petani durian yang ada di lokasi Kelurahan Bumi Kedaton Kecamatan Teluk Betung Utara. Pemilihan sampel Pengujian diawali dengan pembuatan skenario UAT berdasarkan proses bisnis utama yang dilakukan oleh pengguna. Kemudian setiap skenario akan dijabarkan menjadi *test case* yang mencakup langkah-langkah pengujian. Selanjutnya skenario pengujian akan diuji oleh pengguna akhir menggunakan aplikasi yang siap diluncurkan. Kemudian pengguna akhir mengisi kuissoner untuk menyampaikan pengalamannya setelah melakukan pengujian. Kuesioner ini mencakup pertanyaan mengenai reaksi pengguna terhadap sistem, antarmuka pengguna, kemudahan dalam mengoperasikan sistem, dan fungsionalitas yang tersedia. Hasil kuesioner kemudian dianalisis menggunakan skala likert untuk mengukur pendapat atau persepsi individu terhadap sistem.

3.7 Penulisan Laporan

Tahap akhir dari penelitian ini meliputi penyusunan laporan yang berisi proses, hasil, dan temuan terkait pengembangan aplikasi klasifikasi dan deteksi penyakit buah durian. Termasuk penarikan kesimpulan serta penyusunan saran untuk pengembangan lebih lanjut, yang bertujuan untuk mendokumentasikan hasil penelitian sehingga dapat digunakan sebagai referensi oleh pembaca maupun menjadi dasar bagi penelitian lanjutan.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan yang diperoleh dari penelitian ini, terdapat beberapa kesimpulan yang diambil sebagai berikut:

1. Aplikasi mobile DuriPintar dirancang dan dibangun sebagai aplikasi Android Native menggunakan bahasa pemrograman Kotlin dengan arsitektur MVVM dan pendekatan Kanban. Seluruh fitur utama, mulai dari autentikasi, pemindaian daun dan buah, penayangan hasil klasifikasi varietas dan deteksi penyakit, informasi cuaca, riwayat pemindaian, hingga fitur pencatatan, telah berhasil diimplementasikan.
2. Integrasi antara aplikasi *mobile* dan *backend* berhasil diwujudkan melalui pemanggilan API untuk layanan klasifikasi varietas, deteksi penyakit daun durian, serta informasi cuaca. Proses pengambilan gambar, pengiriman ke server, pemrosesan oleh model, hingga penayangan hasil dan penyimpanan ke histori.

5.2 Saran

Berdasarkan penelitian yang dilakukan, terdapat beberapa saran untuk pengembangan selanjutnya yakni sebagai berikut:

1. Disarankan untuk menambahkan mekanisme otomasi pada proses pengujian berbasis *blackbox testing* sehingga pelaksanaan pengujian menjadi lebih cepat, konsisten, dan dapat membantu memastikan tidak muncul kerusakan pada fitur yang sudah ada ketika dilakukan penambahan atau pembaruan fitur baru.

2. Agar aplikasi dapat menjangkau lebih banyak pengguna, pengembangan tidak perlu dibatasi pada Android Native saja, tetapi dapat diperluas ke platform lain seperti iOS.
3. Penelitian selanjutnya dapat memperluas ruang lingkup dengan mengevaluasi integrasi data agronomi lain (misalnya harga pasar atau rekomendasi tindakan perawatan).

DAFTAR PUSTAKA

- [1] N. Najira, E. Selviyanti, Y. Br. Tobing, K. Kasmawati, R. Sianturi, and A. B. Suwardi, "Diversitas Kultivar tanaman Durian (*Durio zabethinus* Murr.) Ditinjau dari Karakter Morfologi," *JBT*, vol. 20, no. 2, pp. 185–193, May 2020, doi: 10.29303/jbt.v20i2.1871.
- [2] Singh, M., Swetha, B., Lakshmi, M., Manasa, M., & Hamsa, M., *Plant disease detection using deep learning and Android-based mobile application*, *International Journal of Computer Application*, 15(6), 80–88 (2025).
- [3] Khaira, U., Weni, I., & Wilia, W., *Rancang bangun aplikasi deteksi penyakit tanaman jagung berbasis Android menggunakan CNN MobileNetV2*, *Jurnal Pepadun*, 5(1), 1–11, 2024.
- [4] P. L. Turnip, M. Ramadhan, and R. Kustini, "Sistem Pakar Mendiagnosa Penyakit pada Tanaman Durio Zibethinus (Durian) Menggunakan Metode Dempster Shafer," *Jurnal CyberTech*, vol. 4, no. 2, pp. 1–10, 2021.
- [5] M. R. A. Jauhari, G. F. Laxmi, and P. J. Santoso, "Sistem Pakar Untuk Mendiagnosis Penyakit Tanaman Durian Menggunakan Certainty Factor Dan Naive Bayes Classifier," *JURNAL RESPONSIF*, vol. 6, no. 2, pp. 175–184, 2024.
- [6] P. Shehrawat, Aditya, S. Singh, and B. Arulmanikandan, "Farmers' awareness level and adoption regarding usage of ICT for crop production," *International Journal of Agriculture Extension and Social Development*, vol. 7, pp. 167–173, Nov. 2024, doi: 10.33545/26180723.2024.v7.i11Sc.1366.
- [7] B. M. Kusuma, T. I. Hermanto, and D. Lestari, "Optimasi dan Integrasi Convolutional Neural Network pada Aplikasi Android untuk Deteksi Penyakit Daun Padi," *Jurnal Teknologi dan Sistem Informasi*, vol. 9, no. 2, pp. 251–258, 2024.
- [8] N. R. Wiwesa, "User Interface Dan User Experience Untuk Mengelola Kepuasan Pelanggan," vol. 3, no. 2, 2021.
- [9] R. Apriyanto, "Aplikasi Greenadvisor Sebagai Pendeteksi Dan Rekomendasi Tanaman Berdasarkan Koordinat Berbasis Android Menggunakan Design Pattern," Universitas Semarang, 2024.
- [10] Vijaywargi, A. & Boddapati, U. K., *Architectural Patterns in Android Development: Comparing MVP, MVVM, and MVI*, *IJRASET Journal For Research in Applied Science and Engineering Technology*, DOI:10.22214/ijraset.2024.60762, membahas dan membandingkan beberapa pola arsitektur aplikasi Android yang umum diterapkan termasuk MVVM dan MVP untuk pengembangan aplikasi mobile, 2024.

- [10] G. Khaksar, S. Kasemcholathan, and S. Sirikantaramas, "Durian (*Durio zibethinus* L.): Nutritional Composition, Pharmacological Implications, Value-Added Products, and Omics-Based Investigations," Mar. 2024.
- [11] P. Thiyajai, S. Charoenkiatkul, K. Kulpradit, P. Sridonpai, and K. Judprasong, "Nutritional composition of indigenous durian varieties," *Malaysian Journal of Nutrition*, vol. 26, no. 1, p. 7, 2020.
- [12] N. A. A. Aziz and A. M. M. Jalil, "Bioactive Compounds, Nutritional Value, and Potential Health Benefits of Indigenous Durian (*Durio Zibethinus* Murr.): A Review," 2019.
- [13] G. Khaksar, "Durian fruit pulp extract enhances intracellular glutathione levels, mitigating oxidative stress and inflammation for neuroprotection," 2024.
- [14] M. M. Ali, "Exploring the chemical composition, emerging applications, potential uses, and health benefits of durian: A review," 2020.
- [15] Object Management Group, "Unified Modeling Language (UML) Version 2.5.1."
- [16] Z. H. Muhamad, D. A. Abdulmonim, and B. Alathari, "An integration of uml use case diagram and activity diagram with Z language for formalization of library management system," *IJECE*, vol. 9, no. 4, p. 3069, Aug. 2019, doi: 10.11591/ijece.v9i4.pp3069-3076.
- [17] Diansyah, H. & Syafrinal, S., *Design and Development of a Mobile Application Using Android Studio and Flutter*, *Journal Mobile Technologies (JMS)*, 3(2), 69–76, 2025.
- [18] Ikram, M. F. D., S. Fachrurrazi, & R. P. Fhonna, *Media Pembelajaran Interaktif Membangun Android Application for Beginners Berbasis Multimedia*, *Sisfo: Jurnal Ilmiah Sistem Informasi*, 2025.
- [19] Mahmud, T., *Why Android App Testing Falls Short: Empirical Insights from Open-Source Projects*, *Empirical Software Engineering* — memberikan wawasan empiris tentang tantangan pengujian (testing) aplikasi Android modern, yang berkaitan erat dengan proses debugging untuk meningkatkan kualitas aplikasi, 2025.
- [20] Afrianto, A. Heryandi, A. Finadhita, and S. Atin, "User Acceptance Test For Digital Signature Application In Academic Domain To Support The Covid-19 Work From Home Program," *International Journal of Information System and Technology*, vol. 5, no. 2, 2021.
- [21] N. Sowri Raja Pillai and R. Rani Hemamalini, "Hybrid User Acceptance Test Procedure to Improve the Software Quality," *IAJIT*, vol. 19, no. 6, 2022, doi: 10.34028/iajit/19/6/14.
- [22] Putra, M. A., Yasirandi, R., & Oktaria, D., *Implementation and Analysis of Service-Based Integrated Platform for Research Center Using Thomas Erl Method*, *Jurnal Media Informatika Budidarma*, 2025.
- [23] Bappah, B., Bristol, L. G., Nouredine, L., Bello, S., Farooq, U., & Ali-Gombe, A., *Exploring Runtime Evolution in Android: A Cross-Version Analysis and Its Implications for Memory Forensics*, *arXiv preprint*, 2025.
- [24] Supriadi, D., Suryadi, E., Muslim, R., & Samsumar, L. D., *Implementasi Vulnerability Assessment OWASP (Open Web Application Security Project) pada Website Universitas Teknologi Mataram*, *Journal of Data Analytics, Information, and Computer Science*, 1(4), 232–240
- [25] *Information, and Computer Science*, 1(4), 232–240

- [26] Firnanda, M. Y., et al., *Comparative Analysis of Penetration Testing Frameworks: OWASP, PTES, and NIST SP 800-115 for Detecting Web Application Vulnerabilities*, *Jurnal Advanced in Information and Industrial Technology*
- [27] Sugara, V. I., & Sriyasa, I. W., *Analisis Keamanan Web Menggunakan Open Web Application Security Web (OWASP)*, *The Indonesian Journal of Computer Science*, 13(2)
- [28] J. J. Sanjaya and J. Susilo, “Perbandingan Performa Kotlin vs Java dalam Pengembangan Android dengan Metode Iterasi While,” *bit-Tech*, vol. 7, no. 2, pp. 545–553, Dec. 2024, doi: 10.32877/bt.v7i2.1898.
- [29] H. Zakaria and I. K. D. Nuryana, “MVVM Perangkat Lunak Android dan Analisis Arsitektur MVP dengan Studi Kasus Aplikasi iTourism,” *JINACS*, vol. 4, no. 4, 2023.
- [30] J. Kouraklis, “MVVM as Design Pattern,” in *MVVM in Delphi*, Berkeley, CA: Apress, 2016, pp. 1–12. doi: 10.1007/978-1-4842-2214-0_1.
- [31] B. Arfianto and A. Prapanca, “Analisis Perbandingan Performa Pola Arsitektur Model- View-ViewModel (MVVM) dan Model-View-Presenter (MVP) pada Pengembangan Aplikasi Desa Wisata Berbasis Android,” *JINACS*, vol. 06, no. 02, 2024.
- [32] H. Joo, “A Study on Understanding of UI and UX, and Understanding of Design According to User Interface Change,” vol. 12, no. 20, 2017.
- [33] S. Hellweger and X. Wang, “What is User Experience Really: towards a UX Conceptual Framework,” 2015.
- [34] B. Paneru, B. Paneru, R. Poudyal, and K. Bikram Shah, “Exploring the Nexus of User Interface (UI) and User Experience (UX) in the Context of Emerging Trends and Customer Experience, Human Computer Interaction, Applications of Artificial Intelligence,” *INJIISCOM*, vol. 5, no. 1, pp. 102–113, Mar. 2024, doi: 10.34010/injiiscom.v5i1.12488.
- [35] Google, “Introduction to Material Design,” Material Design. [Online]. Available: <https://m2.material.io/design/introduction#theming>
- [36] Samvid Mistry and Prakash Pate, “Material Design: A Modern Software Design Language,” *Open Source For You*, no. 2016, pp. 64–66.
- [37] Google, “Material Design 3 – Google’s Latest Open Source Design System,” Material Design. [Online]. Available: <https://m3.material.io/get-started>
- [38] Office of the Executive Director for Systems Engineering and Architecture and Office of the Under Secretary of Defense for Research and Engineering, “Application Programming Interface (API) Technical Guidance,” Department of Defense (DoD), Technical Guidance, 2023.
- [39] J. Y. Khan, Md. Tawkat Islam Khondaker, G. Uddin, and A. Iqbal, “Automatic Detection of Five API Documentation Smells: Practitioners’ Perspectives,” *IEEE*, 2021, pp. 318–329. doi: 10.1109/SANER50967.2021.00037.
- [40] M. Ehmer and F. Khan, “A Comparative Study of White Box, Black Box and Grey Box Testing Techniques,” *IJACSA*, vol. 3, no. 6, 2012, doi: 10.14569/IJACSA.2012.030603.
- [41] H. Wang, Y. Zhang, and X. Fu, “Innovative applications of black box testing: a new strategy for teaching assessment in higher education,” *IEEE Access*, pp. 1–1, 2025, doi: 10.1109/ACCESS.2025.3568825.

- [42] T. Y. Chen and P.-L. Poon, "Experience With Teaching Black-Box Testing in a Computer Science/Software Engineering Curriculum," *IEEE Trans. Educ.*, vol. 47, no. 1, pp. 42–50, Feb. 2004, doi: 10.1109/TE.2003.817617.
- [43] G. W. Sasmito and M. A. Mutasodirin, "Black Box Testing with Equivalence Partitions Techniques in Transcrop Applications," Lombok, Indonesia: IEEE, Sep. 2023, pp. 53–58. doi: 10.1109/IC2IE60547.2023.10331562.
- [44] B. A. Putra, A. P. Kharisma, and F. A. Huda, "Penelitian Akurasi Diagnosa Penyakit Tanaman Padi menggunakan Kamera dengan Metode Klasifikasi Gambar pada Perangkat Bergerak Android," vol. 6, no. 10, 2022.
- [45] R. Namruddin, "Klasifikasi Kesegaran Buah Apel Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Android," 2023.
- [46] N. B. A. Pratama, I. Arwani, and Handoko, "Pengembangan Aplikasi Mobile Pendeteksi Penyakit pada Tanaman Cabai dengan menggunakan Teknologi Clarifai," 2020.
- [47] M. K. Anwar, "Sistem Deteksi Penyakit Daun Apel Menggunakan Metode Convolutional Neural Network Berbasis Mobile," s1, 2024.