

## **BAB III**

### **METODE PENELITIAN**

#### **3.1. Metode *Machine Learning Life Cycle***

Metode *machine learning life cycle* adalah metode pengembangan perangkat lunak yang digunakan pada penelitian ini.

##### **3.1.1. *Model Requirements***

Pada tahap model *requirement* dilakukan Analisa mulai dari pengumpulan data yang akan di lakukan seperti:

- Observasi di lakukan di dua tempat yaitu *Green House* IIB Darmajaya dan kebun Balai Pelatihan Pertanian Lampung. Pada observasi tersebut terdapat tanaman dengan kondisi daun yang tidak sehat. Dalam hasil observasi yang di lakukan cara mendeteksi penyakit yang dilakukan oleh petani masih secara manual. Hasil observasi dari *Green House* IIB Darmajaya dan BPP Lampung adalah:

##### 1. *Green House* IIB Darmajaya Lampung

- Observasi dilakukan pada tanggal 28 Februari sampai 3 Maret 2022.
- Terdapat 200 Tanaman melon jenis *makwari* yang ditanam pada dua kebun.
- Terdapat 160 tanaman yang mengalami penyakit daun antara lain layu *fusarium*, kutu daun dan ulat daun.

##### 2. BPP Lampung

- Observasi dilakukan pada tanggal 22 hingga 24 Maret 2022.
- Terdapat 45 tanaman melon jenis *sky rocket rock*, *golden melon* yang ditanam pada satu kebun terbuka (tidak berbasis *green house* / IoT).
- Data yang di temukan pada BPP adalah melon dengan kondisi sehat.

- Wawancara pada Pengelola *Green house* IIB Darmajaya dan staf ahli BPP Lampung. Hasil Pertanyaan dapat di lihat pada tabel 3.1 dan 3.2.

**Tabel 3.1** Hasil Wawancara dengan Pengelola *Green House* IIB Darmajaya

Narasumber		Bpk. Dodi Yudo Setiawan
No.	Pertanyaan	Jawaban
1.	Konsep pertanian melon yang dilaksanakan di <i>green house</i> IIB Darmajaya?	Pertanian yang dilaksanakan menggunakan konsep <i>smart farming</i> menggunakan <i>Internet of Things</i> . Pertanian dilaksanakan pada dua kebun yang berbeda. Tiap kebun memiliki dua sensor yaitu sensor suhu dan kelembapan tanah. Untuk proses penyiraman dan pendinginan dilakukan secara otomatis yang dikontrol melalui aplikasi Tani cerdas. Pertanian dikelola oleh Dosen dan mahasiswa yang bekerja sama dengan BPP Lampung dan Politeknik Negeri Lampung.
2.	Berapa jumlah tanaman melon dan jenisnya.	Jumlah tanaman melon di IIB <i>Green house</i> adalah 200 tanaman melon berjenis melon <i>makwari</i> .
3.	Apakah ada kesulitan dalam pengelolaannya?	Belum adanya pendeteksi tanaman pada aplikasi Tani Cerdas. Berbasis kecerdasan buatan sehingga pendeteksian masih dilakukan secara manual dengan mengobservasi langsung pada tanaman melon dan berkonsultasi kepada peneliti di Polinela. Sehingga sering kali deteksi penyakit berjalan dengan lambat yang mengakibatkan penyakit tidak cepat di tangani sehingga penyakit menular dengan cepat.

**Tabel 3.2** Hasil Wawancara Oleh Pihak BPP Lampung

Narasumber		<ul style="list-style-type: none"> <li>• Bpk. Dr Adi Destriadi Sutisna, SP, MP</li> <li>• Bpk. Farizal</li> <li>• Sulis Nur Wahida</li> </ul>
No.	Pertanyaan	Jawaban
1.	Berapa jumlah tanaman melon ?	Terdapat 45 tanaman melon dengan 3 jenis varian.
2.	Apa saja jenis varian melon ?	Jenis varian melon yang ada di BPP lampung yaitu <i>sky rocket rock</i> , <i>golden melon</i> .

**Tabel 3.2** Hasil Wawancara Oleh Pihak BPP Lampung (Lanjutan)

No.	Pertanyaan	Jawaban
3.	Apakah tanaman melon di sini sering terkena penyakit atau hama ?	Penyakit dan hama sering terjadi pada tanaman melon disebabkan lahannya terbuka.
4.	Jenis penyakit dan hama apa saja yang sering di alami oleh tanaman melon ?	Penyakit yang sering di alami bercak kuning, busuk buah, ulat daun, lalat buah, <i>oteng-oteng</i> namun tidak banyak.
5.	Apakah penyakit tersebut bisa di temukan di BPP secara banyak untuk pengambilan data ?	Penyakit tersebut tidak bisa ditemukan secara banyak hanya satu atau dua tanaman saja yang terkena, namun jika tanaman tersebut sudah terkena hama atau virus maka segera dilakukan penindakan untuk mencegah penularan atau kematian pada tanaman melon tersebut.
6.	Bagaimana cara mengantisipasi tanaman melon agar terhindar dari penyakit ?	Cara mengantisipasi tanaman melon agar terhindar dari penyakit adalah dengan melakukan penyemprotan secara rutin.
7.	Apakah teknologi informasi telah diterapkan dalam mendeteksi penyakit ?	Untuk saat ini masih belum ada teknologi informasi yang di terapkan untuk mendeteksi penyakit tanaman di BPP Lampung.
8.	Apakah ada kesulitan dalam pengelolaannya ?	Tidak ada kesulitan karena setiap hari tanaman melon dilakukan penyiraman dan pengamatan oleh petugas piket dan mahasiswa yang sedang melakukan penelitian dan diawasi oleh tenaga ahli.
9.	Apakah pada <i>green house</i> melon ini sudah diterapkan <i>teknologi IoT</i>	Belum, manajemen dan <i>controlling</i> masih menggunakan teknik konvensional. IoT baru di terapkan secara parsial pada tanaman sayur dan buah sayur.

Selain melakukan observasi dan wawancara tahapan ini juga menentukan jumlah data gambar yang akan di ambil. Kemudian rancangan arsitektur CNN yang akan di buat, evaluasi dan akurasi, sampai tahap implementasi ke dalam aplikasi Tani Cerdas. Dapat dilihat pada gambar 3.1 dan 3.2 sebagai observasi di Green House IIB Darmajaya dan kebun melon di BPP Lampung.



**Gambar 3.1** Hasil Observasi pada Green House IIB Darmajaya



**Gambar 3.2** Hasil Observasi pada Kebun Melon di BPP

Pada gambar 3.1 dapat dilihat tanaman pada *green house* IIB Darmajaya mengalami banyak kerusakan, sebesar 80% karena hama dan penyakit. Sedangkan gambar 3.2, tanaman buah melon tampak sehat dan segar.

### **3.1.2. Data Collection**

Pada tahap data *collection* dilakukan pengumpulan data terkait gambar penyakit daun pada tanaman melon. Pengumpulan data di lakukan selama 3 minggu pada dua lokasi yang berbeda yakni di *Green House* IIB Darmajaya dengan jumlah 200 tanaman melon dan BPP Provinsi Lampung dengan jumlah 45 tanaman melon. Pengumpulan data tersebut berhasil di kumpulkan sebanyak 440 gambar data mentah di mana data gambar tersebut terdiri dari gambar tanaman melon sehat dan tanaman melon yang sakit.

### **3.1.3. Data Cleaning**

Pada fase *Cleaning* data yang sudah di kumpulkan akan di bersihkan mulai dari pemilihan gambar yang akan di gunakan, pengecilan ukuran *file*, standarisasi ukuran dari ukuran gambar 2992 x 2992 menjadi 1250 x 1250. Hal ini dilakukan agar mengurangi ukuran dari gambar sehingga mempercepat proses unggah dan latih.

### **3.1.4. Data Labeling**

Proses *Labeling* di lakukan secara manual dengan mengelompokkan jenis gambar berdasarkan jenis penyakit yang di alami sesuai dengan tabel tanaman Melon Proses pelabelan juga di bantu oleh BPP Lampung dan mahasiswa ITERA.

### **3.1.5. Feature Engineering**

Proses *Feature Engineering* dilakukan dengan memecah data menjadi data *training* dan testing. Untuk data *training* berjumlah 80% dari keseluruhan data sebagai data *training* dan 20% dari keseluruhan dari data testing. Pada tahap *feature engineering* juga melakukan rancangan model arsitektur *Convolutional Neural Network* yang nantinya akan di latih.

### **3.1.6. Model Training**

*Model training* dilakukan untuk melatih model CNN yang sudah dibuat. *Training* di lakukan agar model dapat bekerja dengan optimal dimana pada proses *training* data *training* akan di bagi menjadi dua bagian yakni *training set* dan *validation set*.

### **3.1.7. Model Evaluation**

*Model evaluation* pada tahapan ini akan mengkonfirmasi bahwa suatu model dapat memprediksi suatu data baru dengan cukup baik. Data yang akan di masukkan ke dalam evaluasi ini adalah data testing di mana data testing memiliki 20% dari seluruh gambar atau 20% x 250 gambar = 80 gambar yang

dimiliki terdiri dari 5 kelas (sehat, layu *fusarium*, kutu daun, ulat daun, dan objek yang tidak berkaitan) seperti pada tabel 2. Setelah model selesai memprediksi data langkah selanjutnya adalah menggunakan dua jenis *evaluation matrix* untuk melihat performa dari model yang sudah di buat. Jenis matrix tersebut adalah *matrix accuracy* dan *confusion matrix*.

### **3.1.8. Production**

Pada tahap ini akan memiliki 3 tahap yakni :

- *Deployment* : model akan diintegrasikan langsung pada aplikasi android Tani Cerdas agar dapat mendeteksi penyakit daun tanaman melon
- *Monitoring* : pada saat model sudah diintegrasikan maka akan dilakukan monitoring yang bertujuan untuk mengetahui kinerja model terhadap aplikasi yang sudah di bangun.
- *Updating* : pembaruan akan terjadi ketika ditemukan penyakit baru atau penambahan fitur terkait aplikasi Tani Cerdas khususnya pada fitur deteksi penyakit. Proses pembaruan dilakukan dengan cara memasukkan data penyakit baru kemudian dilatih, setelah proses latih selesai maka model akan diintegrasikan ke dalam aplikasi Tani Cerdas.

### **3.2. Analisa Permasalahan Sistem Berjalan**

Berdasarkan model *requirement* yang telah dilakukan pada tahapan sebelumnya (Bab 3 sub 3.1) di dapati permasalahan pada pengelolaan kebun di Green House IIB Darmajaya adalah pendeteksian penyakit yang masih mengandalkan konsultan dari Politeknik Negeri Lampung. Hal ini tentu memberikan dampak lambatnya deteksi penyakit dan penanganan sehingga mengakibatkan 80% tanaman terkena penyakit.

Habibi Garden adalah salah satu aplikasi yang bergerak di bidang pertanian dan di luncurkan pada tahun 2017. Aplikasi Habibi Garden bertujuan untuk membantu petani dalam proses budidaya tanaman agar lebih efisien yaitu dengan cara menggunakan aplikasi berbasis cerdas untuk membantu melakukan *monitoring* terhadap lahan. Sehingga dengan bantuan sensor dan aplikasi tersebut petani dapat

memantau kondisi lahan dengan menggunakan *smartphone* tanpa perlu melakukan kunjungan setiap hari.

Aplikasi Habibi Garden memiliki banyak fitur salah satunya fitur sistem pakar untuk mendeteksi penyakit pada tanaman. Fitur tersebut dapat mendeteksi penyakit berdasarkan gejala yang di pilih oleh pengguna. Namun pada fitur deteksi penyakit yang terdapat pada aplikasi Habibi Garden hanya dapat mendeteksi 1 jenis tanaman saja, yaitu kentang, dan tidak ada fitur untuk mendeteksi penyakit tanaman melalui gambar. Saat ini, aplikasi ini dipergunakan sebagai aplikasi utama dalam manajemen *green house* di IIB Darmajaya untuk sementara selama Aplikasi Tani Cerdas masih dalam pengembangan.

Tani Cerdas adalah aplikasi yang dikembangkan di IBI Darmajaya yang digunakan untuk memonitoring *smart farming IBI darmajaya* belum memiliki fitur untuk deteksi penyakit tanaman. Fitur deteksi penyakit pada aplikasi Tani Cerdas akan di kembangkan dengan tujuan mengetahui penyakit daun pada tanaman melon serta hasil yang di berikan berupa jenis penyakit, gejala serta penanggulangan dari setiap penyakit

*Plantix* merupakan aplikasi diagnosis tanaman yang di kembangkan oleh *Progressive Environmental & Argicultural Technologies (PEAT)*, yang ada di Berlin German. Berkerja sama dengan ICRISAT dan *Acharya N G Ranga Agricultural University*. Aplikasi *Plantix* mencakup 30 tanaman dan dapat mendeteksi lebih dari 400 kerusakan tanaman. *Plantix* mendeteksi kerusakan tanaman berdasarkan foto yang di ambil oleh petani. Foto yang sudah di ambil akan di analisa menggunakan *artificial intelligence* dan akan dikembalikan kepada petani berupa hasil penyakit dari tanaman tersebut. Hasil yang di berikan berupa informasi tentang gejala, pemicu, bahan kimia serta perawatan yang disediakan oleh aplikasi.

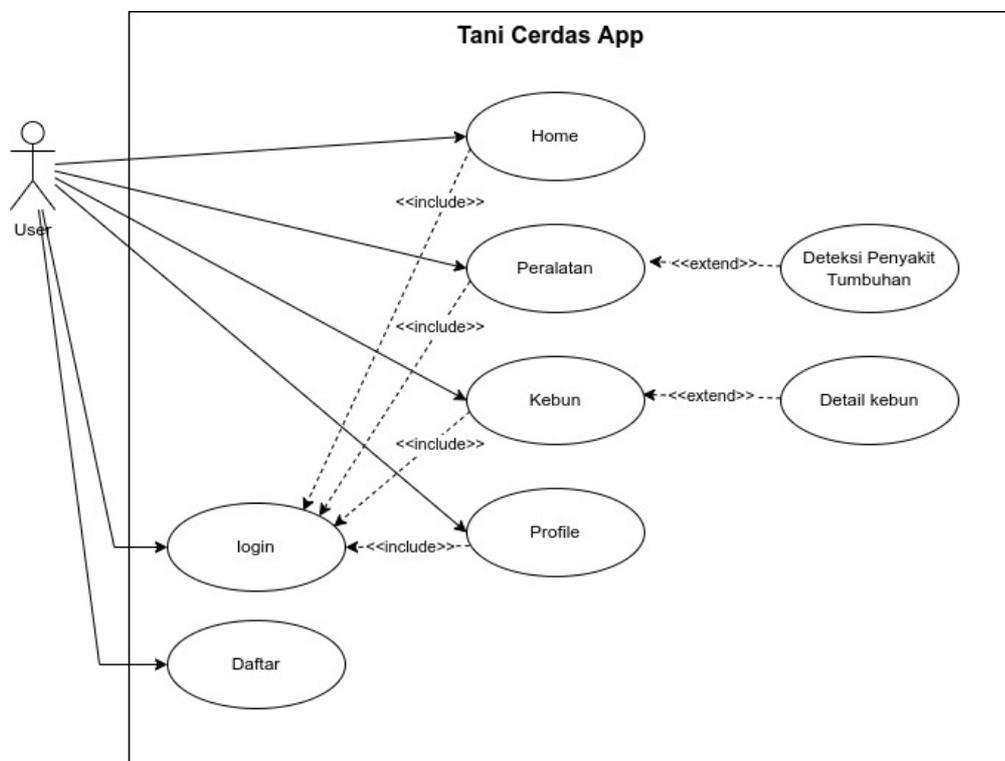
*Plantix* banyak di gunakan oleh Petani modern untuk mendeteksi penyakit. Namun aplikasi ini belum terintegrasi dengan data petani sehingga sulit unuk melakukan pemindaian sebagai rekam jejak penyakit pada suatu lahan yang dapat digunakan sebagai rujukan untuk perbaikan lahan ke depannya.

### 3.3. Desain Sistem

Berikut adalah desain sistem aplikasi Tani Cerdas dan perancangan fitur deteksi penyakit tumbuhan mulai dari *use case* diagram, *activity* diagram, *sequence* diagram sampai rancangan tampilan dari aplikasi.

#### 3.3.1. Use Case

Berikut adalah diagram *use case* dari aplikasi Tani Cerdas dapat dilihat pada Gambar 3.1.



**Gambar 3.3** Use Case Diagram Aplikasi Tani Cerdas

Pada diagram di atas terdapat dua bagian yakni aktor dan *use case*. *Actor* atau petani dapat mengakses *login*, daftar peralatan *home*, kebun *profile*. Untuk deskripsi dari fungsionalitas *use case* dapat di lihat pada tabel 3.4

**Tabel 3.3** Deskripsi *Actor* dan *Use Case*

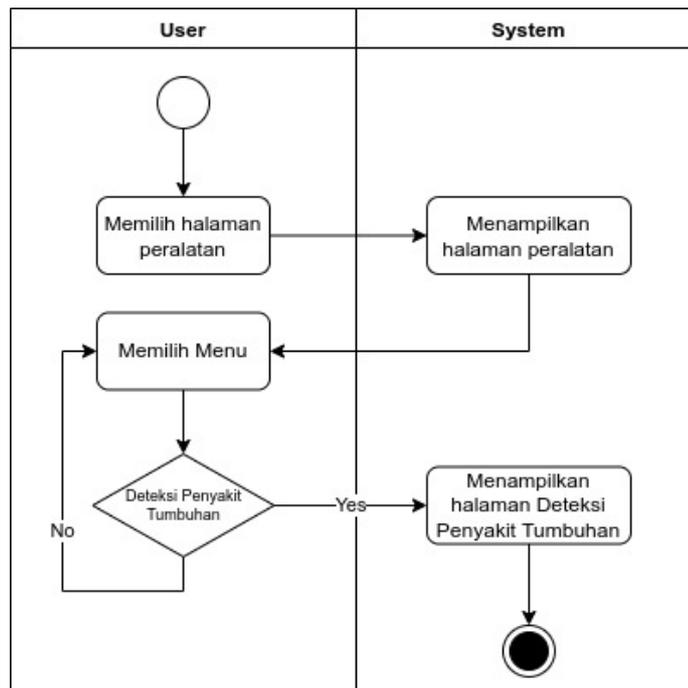
No	Use case	Deskripsi
1.	Login	Merupakan proses di mana aktor dapat memasukkan <i>username</i> dan <i>password</i> untuk masuk ke tampilan utama.

**Tabel 3.3** Deskripsi Actor dan Use Case (Lanjutan)

No	Use case	Deskripsi
2.	Daftar	Merupakan proses di mana aktor baru menggunakan aplikasi dan diharuskan untuk mendaftar dan mengisi identitas diri.
3.	Home	Proses di mana aktor berhasil melakukan <i>login</i> atau daftar.
4.	Peralatan	Tampil peralatan jika aktor memilih peralatan.
5.	Kebun	Tampil kebun jika aktor memilih kebun
6.	Profile	Tampil <i>profile</i> jika aktor memilih <i>profile</i> .
7.	Deteksi penyakit tumbuhan	Deteksi penyakit tumbuhan akan di tampilkan Ketika aktor memilih menu peralatan.
8.	Detail Kebun	Detail kebun akan di tampilkan Ketika aktor memilih menu kebun.

### 3.3.2. Activity Diagram

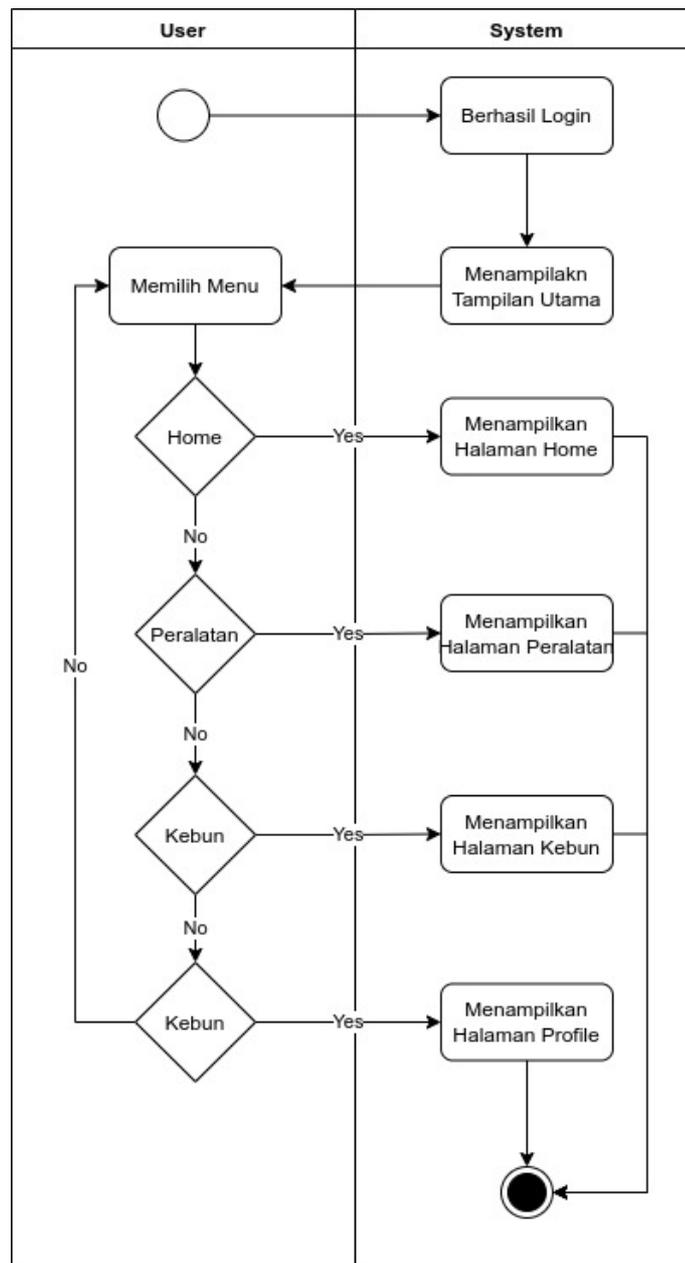
#### 1. Activity Diagram Peralatan



**Gambar 3.4** Activity Diagram Peralatan

Pada activity diagram peralatan system akan mengarahkan user ke dalam halaman peralatan kemudian menampilkan menu deteksi penyakit tumbuhan, dan Ketika user memilih menu tersebut user akan di arahkan ke halaman deteksi penyakit tumbuhan

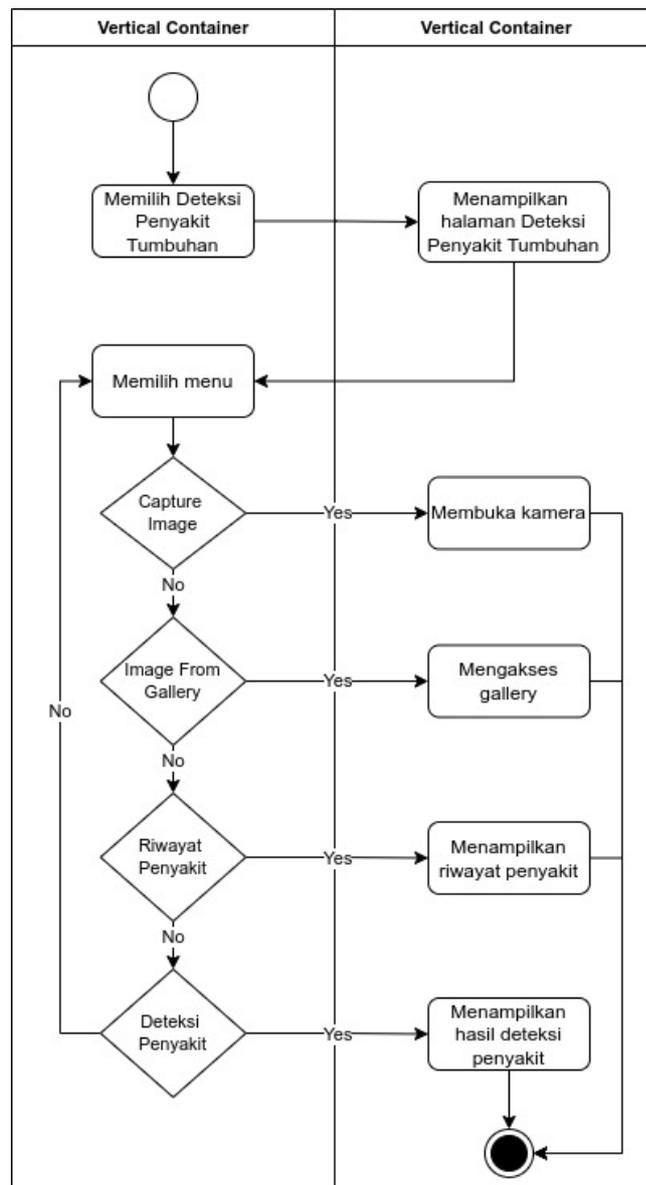
## 2. Activity Diagram Menu Utama



Gambar 3.5 Activity Diagram Menu Utama

Pada *activity* diagram menu utama *user* dapat memilih menu yang tersedia yaitu menu *home*. Peralatan kebun dan *profile*. Ketika *user* memilih salah satu menu tersebut *user* akan di alihkan ke halaman atau *activity* yang bersangkutan.

### 3. Activity Diagram Deteksi Penyakit Tumbuhan



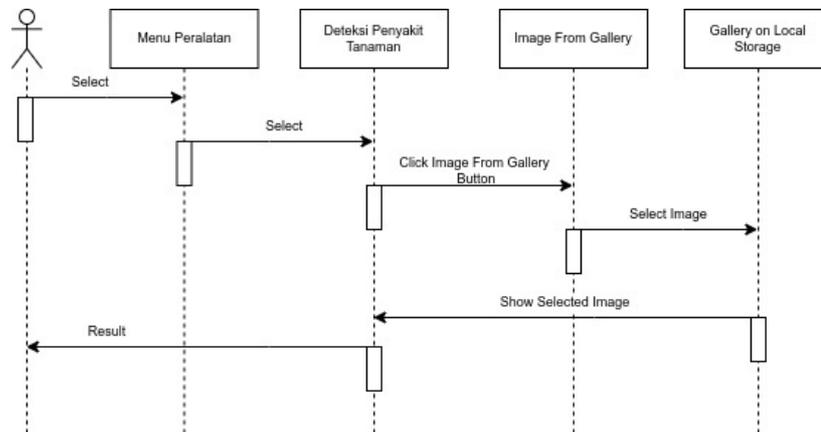
**Gambar 3.6** Activity Diagram Deteksi Penyakit Tumbuhan

Pada *activity* diagram deteksi penyakit tumbuhan memiliki menu *capture image*, *image from gallery*, riwayat penyakit dan deteksi

penyakit. Ketika *user* memilih *capture image* maka *system* akan membuka kamera, Jika *user* memilih menu *image from gallery* maka sistem akan membuka galeri pada ponsel android. Saat *user* memilih riwayat penyakit maka sistem akan menampilkan riwayat penyakit yang sudah di deteksi oleh sistem. Apabila *user* memilih menu deteksi penyakit maka akan di tampilkan hasil deteksi penyakit yang telah di klasifikasikan.

### 3.3.3. Sequence Diagram

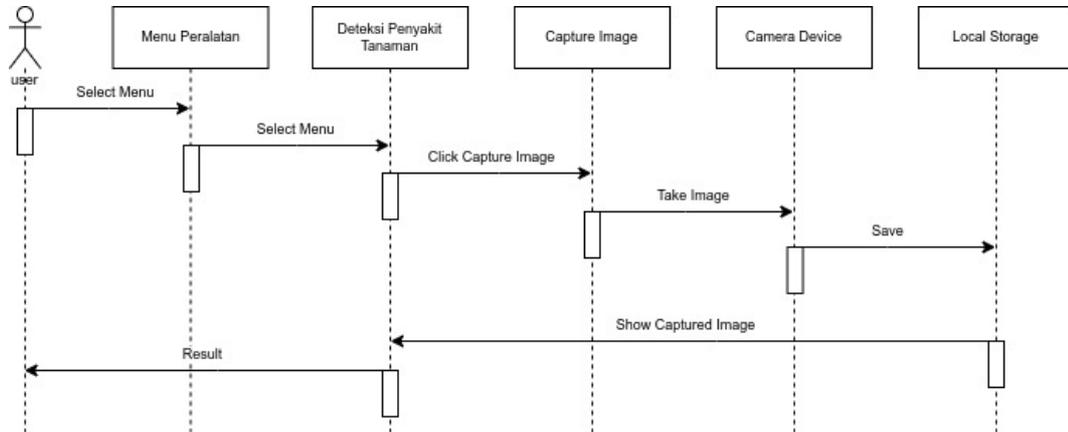
#### 1. Sequence Diagram image from Gallery



Gambar 3.7 Sequence Diagram Image From Gallery

Sequence diagram di atas mendeskripsikan proses pengambilan gambar dari galeri pengguna. Ketika pengguna memilih menu *image from gallery* maka *system* akan menampilkan gambar yang sudah di simpan sebelumnya di galeri kemudian Ketika *user* sudah memilih gambar yang di inginkan maka sistem akan mengembalikan gambar yang sudah di pilih dan di tampilkan ke halaman deteksi penyakit tanaman.

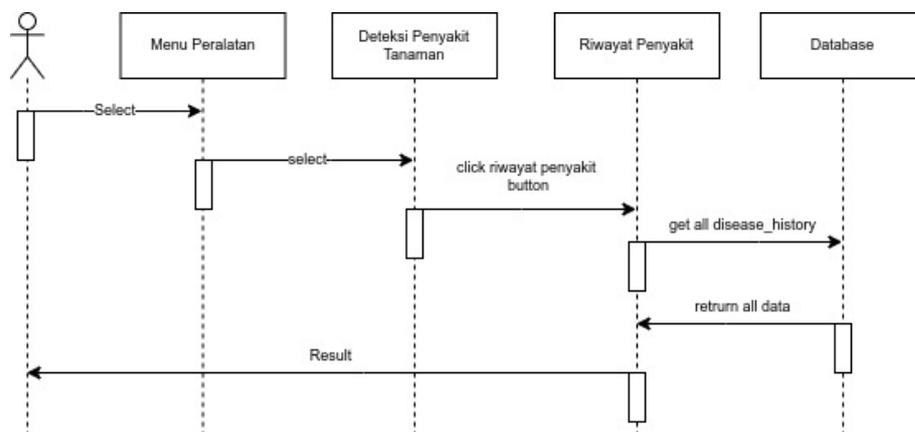
## 2. Sequence Diagram Capture Image



Gambar 3.8 Sequence Diagram Capture Image

Pada *sequence* diagram di atas mendeskripsikan cara sistem membuka kamera pada *smartphone*. Ketika *user* memilih *capture image* maka sistem akan mengakses kamera dari perangkat *user* kemudian *user* bisa mengambil gambar menggunakan kamera. Setelah itu kamera akan menyimpan hasil gambar tersebut ke dalam *local storage* secara sementara, ketika sukses menyimpan ke dalam *local storage* maka sistem akan menampilkan gambar tersebut kepada *user*.

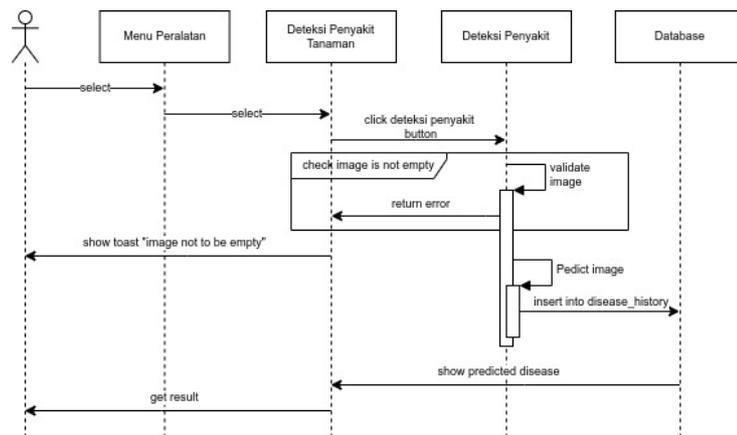
## 3. Sequence Diagram Riwayat Penyakit



Gambar 3.9 Sequence Diagram Riwayat Penyakit

Pada *sequence* diagram Riwayat penyakit *system* akan mengembalikan seluruh data dalam *database* ketika *user* memilih menu Riwayat penyakit. data tersebut berupa gambar penyakit, jenis penyakit dan tanggal pengambilan gambar.

#### 4. *Sequence Diagram* Deteksi Penyakit



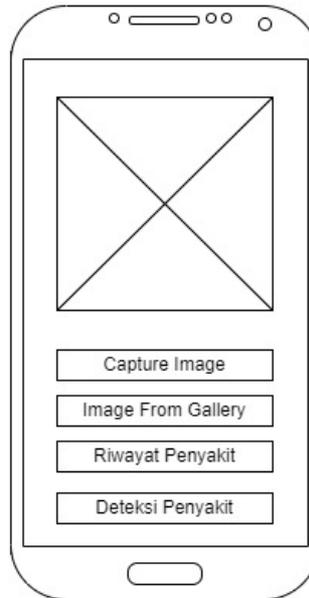
Gambar 3.10 *Sequence Diagram* Deteksi Penyakit

Pada *sequence* diagram deteksi penyakit *system* akan mendeteksi penyakit dari gambar yang diberikan oleh *user*. Sebelum *system* mendeteksi sebuah gambar *system* akan memvalidasi apakah gambar sudah di masukkan oleh *user* atau belum. Jika belum maka *user* akan di berikan pesan ‘*image not to be empty*’, sedangkan jika terdapat gambar maka akan melakukan klasifikasi gambar, kemudian melakukan penyimpanan gambar ke dalam *database* setelah sukses menyimpan data ke dalam *database* maka akan di tampilkan jenis penyakit dan cara penanggulangannya ke *user*.

### 3.3.4. Rancangan *Inteface*

Berikut adalah rancangan interface untuk fitur tani cerdas :

#### 1. Rancangan Tampilan Fitur Deteksi Penyakit Tanaman

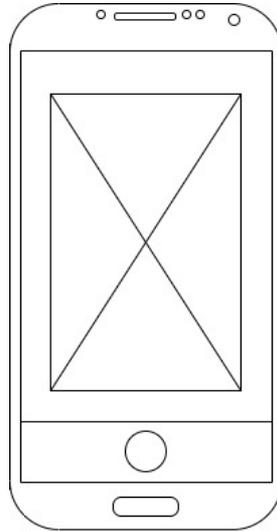


**Gambar 3.11** Rancangan Tampilan Fitur Deteksi Penyakit

Pada rancangan tampilan fitur deteksi penyakit memiliki beberapa tombol yaitu *capture image*, *image from gallery*, riwayat penyakit deteksi penyakit. pada setiap tombol memiliki fungsi yang berbeda-beda yaitu:

- *Capture image* tombol ini akan di memunculkan kamera.
- *Image from gallery* tombol ini akan membuka galeri *user* untuk mengambil gambar.
- Riwayat penyakit tombol ini akan memberikan data riwayat penyakit tanaman melon.
- Deteksi penyakit tombol ini akan mendeteksi penyakit dan memberikan solusi pada setiap penyakit yang berhasil di deteksi.

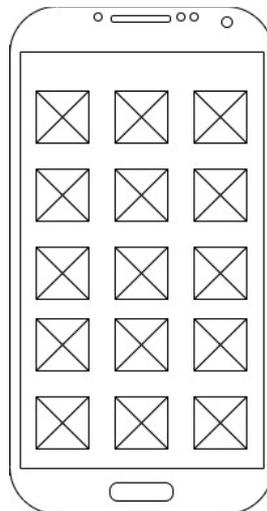
## 2. Rancangan Tampilan Kamera



**Gambar 3.12** Rancangan Tampilan Kamera

Pada rancangan ini akan di tampilkan *user interface* dari kamera Ketika pengguna memilih tombol *capture image*.

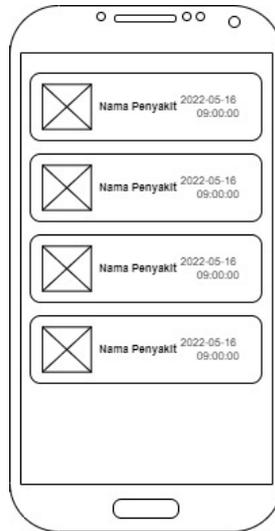
## 3. Rancangan Tampilan Galeri



**Gambar 3.13** Rancangan Tampilan Galeri

Pada rancangan ini akan menampilkan seluruh gambar / foto yang di miliki pengguna yang berada dalam folder galeri. Pengguna dapat memilih gambar yang ingin di deteksi.

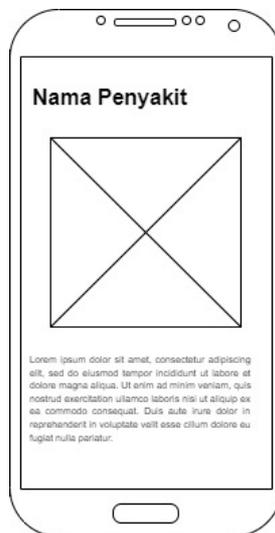
#### 4. Rancangan Riwayat Penyakit



**Gambar 3.14** Rancangan Riwayat Penyakit

Pada rancangan ini akan di tampilkan seluruh data penyakit yang pernah di deteksi oleh pengguna.

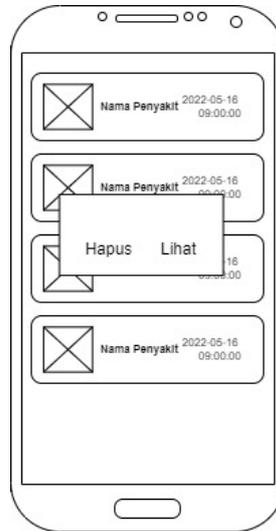
#### 5. Rancangan Tampilan Deteksi Penyakit



**Gambar 3.15** Rancangan Tampilan Deteksi Penyakit

Pada rancangan ini akan di tampilkan nama penyakit, gambar dan solusi ketika penyakit daun tanaman melon di temukan.

## 6. Rancangan Tampilan Lihat dan Hapus pada Riwayat Penyakit



**Gambar 3.16** Rancangan Tampilan Lihat dan Hapus

Tampilan ini kan keluar ketika pengguna memilih salah satu penyakit pada tampilan riwayat penyakit. Ketika pengguna memilih hapus maka data yang ada di riwayat penyakit akan di hapus, jika pengguna memilih lihat maka hasil diagnosa penyakit akan di tampilkan.