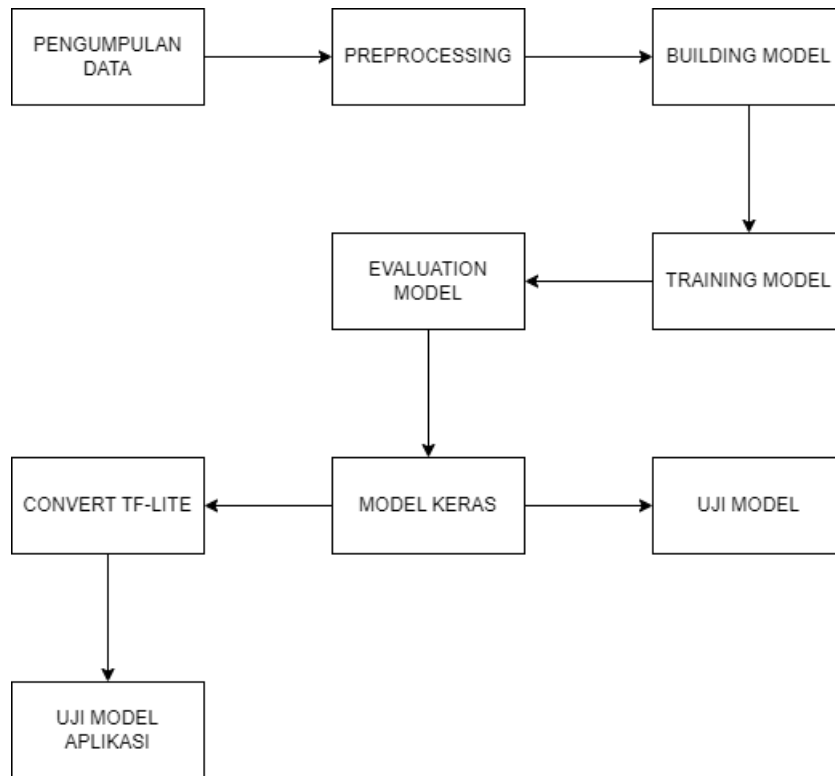


### BAB III METODE PENELITIAN

Metode penelitian ini melibatkan beberapa tahapan yang dijelaskan melalui alur berikut.



Gambar 3.8 Alur Metode

Flowchart ini menjelaskan alur pengembangan model deteksi penyakit tanaman selada, mulai dari pengumpulan data, preprocessing, training, hingga evaluasi dan uji aplikasi untuk memastikan model bekerja optimal di perangkat Android.

#### 3.1 Metode Pengumpulan Data

Pada tahap pengumpulan data, beberapa teknik digunakan untuk mendapatkan informasi tentang tujuan penelitian ini, seperti:

a. Wawancara

Wawancara dilakukan kepada petani hidroponik selada di Sahabat Hidroponik Lampung, tepatnya di Nunyai, Rajabasa, Kota Bandar Lampung, untuk mendapatkan informasi tentang masalah yang dihadapi petani selada hidroponik dan untuk mendapatkan data berupa gambar daun selada yang terlihat terkena penyakit. Untuk penelitian ini, petani diwawancarai untuk

membuat aplikasi deteksi penyakit selada sehingga petani selada dapat melakukan penanganan yang lebih cepat dan efisien untuk tanamannya dan mengatasi gejala awal penyakit.

b. Observasi

Observasi dilakukan dengan memeriksa daun selada yang terkena penyakit untuk mengidentifikasi penyakit yang menyerang tanaman selada. Proses ini mencakup pengamatan langsung terhadap gejala visual seperti perubahan warna, bentuk, dan tekstur pada daun, yang kemudian dibandingkan dengan data penyakit tanaman selada yang sudah ada. Hasil observasi ini akan digunakan sebagai dasar untuk menentukan metode penanganan yang tepat serta pengembangan model deteksi penyakit pada tanaman selada.

c. Pengumpulan Data

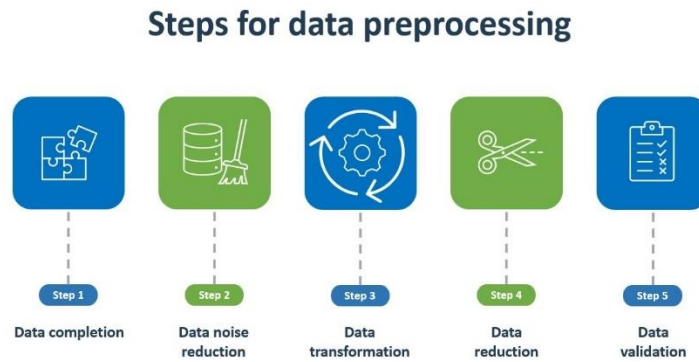
Pengumpulan data penyakit pada tanaman selada dilakukan dengan menggunakan Kaggle untuk mencari beberapa dataset yang terkait dengan penyakit tanaman selada dan mengambil beberapa sampel di lokasi penelitian, yang akan diolah melalui proses train data dalam penelitian ini.

d. Studi Pustaka

Beberapa referensi dari jurnal, artikel ilmiah, dan makalah sebelumnya yang relevan dengan penelitian ini disebutkan dalam studi pustaka.

### **3.1.1 Pre-Processing**

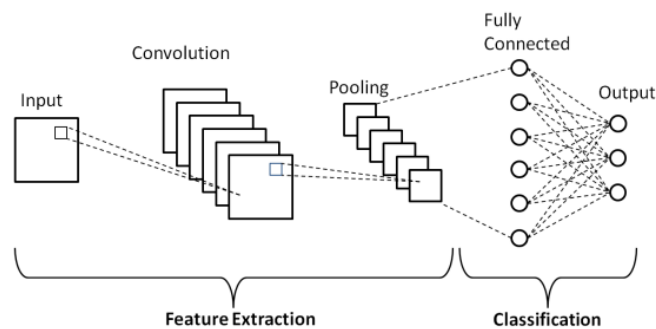
Sebelum dilakukan pelatihan, dataset akan dipersiapkan terlebih dahulu dalam format yang sesuai untuk meningkatkan kinerja model. Selanjutnya, gambar-gambar dengan nilai piksel antara 0 hingga 255 akan dinormalisasi ke dalam rentang 0 hingga 1. Hal ini bertujuan untuk mempermudah proses pembelajaran serta membantu model dalam mencapai konvergensi dengan lebih efisien.



Gambar 3.9 proses *preprocessing*

### 3.1.2 Building Model

Dengan memanfaatkan arsitektur CNN, penelitian ini berfokus pada pencapaian keseimbangan antara akurasi yang tinggi dan nilai *loss* yang rendah. Arsitektur ini menawarkan kemampuan adaptif dalam mempelajari data, mengurangi risiko *overfitting*, serta mendukung transfer learning untuk meningkatkan performa model. Penempatan lapisan konvolusi, pengurangan dimensi melalui *pooling*, dan penerapan teknik regularisasi memungkinkan model memahami pola yang kompleks tanpa mengurangi kemampuan generalisasi.



### 3.1.3 Training Model

*Training model* melibatkan pemrosesan data melalui jaringan untuk menghasilkan prediksi, menghitung selisih antara prediksi dan label dengan fungsi *loss*, dan menggunakan *backpropagation* serta *optimizer* untuk memperbarui bobot model. Proses ini diulang selama beberapa *epoch* untuk meningkatkan akurasi hingga model mencapai performa optimal.

### 3.1.4 Evaluation Model

Kode ini melakukan evaluasi model pada data validasi yang telah disediakan, memberikan metrik yang menunjukkan seberapa baik model bekerja dan menggunakan hasil ini untuk memahami kinerja model dan apakah perlu dilakukan tuning lebih lanjut atau perbaikan.

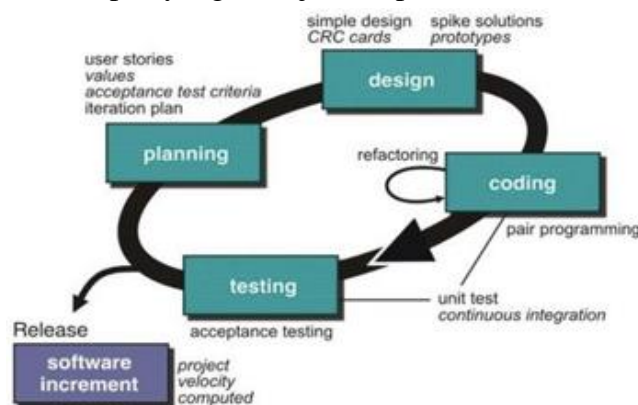
### 3.1.5 Convert TF-Lite

Proses ini berguna untuk mengonversi model Keras ke format yang lebih ringan dan efisien untuk digunakan dalam aplikasi *mobile*. Dengan menggunakan TensorFlow Lite, aplikasi dapat menjalankan inferensi model pada perangkat dengan performa yang lebih baik dan latensi yang lebih rendah

## 3.2 Metode Pengembangan Sistem

### 3.3.1 Metode Extreme Programming

*Extreme Programming* (XP) adalah model pengembangan perangkat lunak yang bertujuan menyederhanakan tahap-tahap pengembangan sistem agar menjadi lebih efisien, adaptif, dan fleksibel[28]. Secara umum, penelitian ini terdiri dari tahapan yang ditunjukkan pada Gambar 3.15



Gambar 3.11 Alur Extreme Programming[29]

#### a. *Planning*/Identifikasi masalah

Pada tahap perencanaan, pengguna diajukan pertanyaan mengenai kebutuhan mereka untuk memahami fitur dan fungsi yang akan diterapkan dalam sistem.

#### b. *Design*/Perancangan

Pada tahap ini, dibuat spesifikasi mengenai arsitektur program, antarmuka, dan komponen pendukung. *Use case diagram*, *activity*

*diagram*, dan *sequence diagram* digunakan untuk menggambarkan proses desain.

c. *Coding/Implementasi*

Pada tahap ini, desain yang telah dibuat diterapkan dilakukan menjadi sistem informasi berbasis aplikasi Android menggunakan Kotlin. Proses ini memanfaatkan framework Android SDK yang dilengkapi dengan pustaka-pustaka dasar, sehingga pengembangan aplikasi menjadi lebih mudah dan cepat tanpa harus melakukan pengkodean secara *native* dari awal. Selain itu, untuk mendukung efisiensi, fitur seperti *View Binding* dan komponen-komponen lainnya dari *Android Jetpack* juga dimanfaatkan dalam pembuatan sistem ini.

d. *Testing/Pengujian*

Tahap ini merupakan proses pengujian pada aplikasi yang telah dikembangkan. Pada tahap ini, fokus utama adalah memastikan bahwa fitur-fitur yang telah dirancang berfungsi dengan baik, bebas dari kesalahan, dan sesuai dengan spesifikasi yang diinginkan.

### 3.3 Desain Sistem

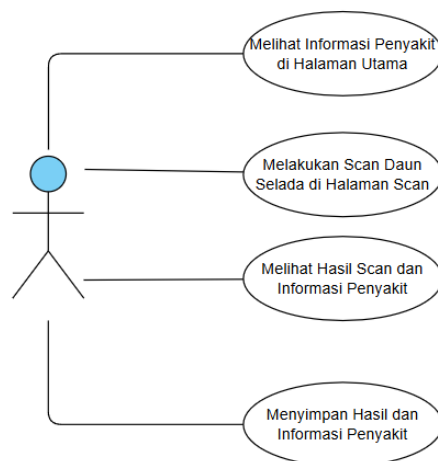
#### 3.3.1 Use Case Diagram

Sistem yang berfungsi untuk mendeteksi penyakit pada daun tanaman selada digambarkan di bawah ini sebagai *Use Case Diagram*.

No	Aktor	Deskripsi
1	Petani	Petani selada adalah pengguna utama sistem deteksi penyakit daun tanaman selada. Mereka memanfaatkan aplikasi untuk memantau kesehatan tanaman selada mereka dengan mengambil atau mengunggah foto daun selada yang dicurigai terkena penyakit. Petani mendapatkan hasil deteksi penyakit secara instan, informasi terkait penyakit, serta langkah-langkah penanganan untuk menjaga kesehatan tanaman mereka.

Tabel 3.2 Use Case Diagram

Berikut adalah *Use Case Diagram* pada sistem deteksi penyakit.

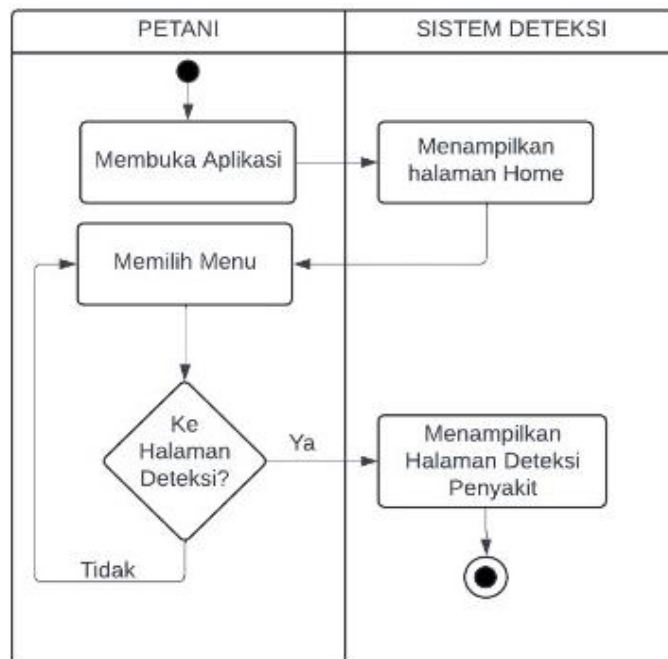


Gambar 3.12 Use Case Diagram Sistem Deteksi Penyakit

### 3.3.2 Activity Diagram

#### 1. Activity Diagram Home

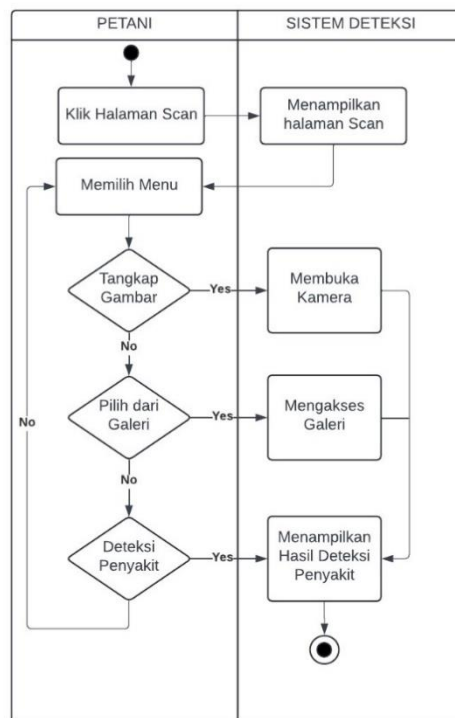
Diagram ini menggambarkan alur interaksi petani dengan sistem deteksi penyakit. Petani membuka aplikasi, memilih menu, dan jika menuju halaman deteksi penyakit, sistem akan menampilkannya.



Gambar 3.13 Activity Diagram Home

#### 2. Activity Diagram Halaman Scan

Diagram ini menunjukkan proses interaksi antara petani dan sistem deteksi penyakit tanaman. Petani memulai dengan membuka halaman scan, mengaktifkan kamera, dan mengambil gambar tanaman. Sistem kemudian menganalisis gambar tersebut, menampilkan hasil diagnosis beserta solusi, dan memberi opsi kepada petani untuk menyimpan hasilnya.



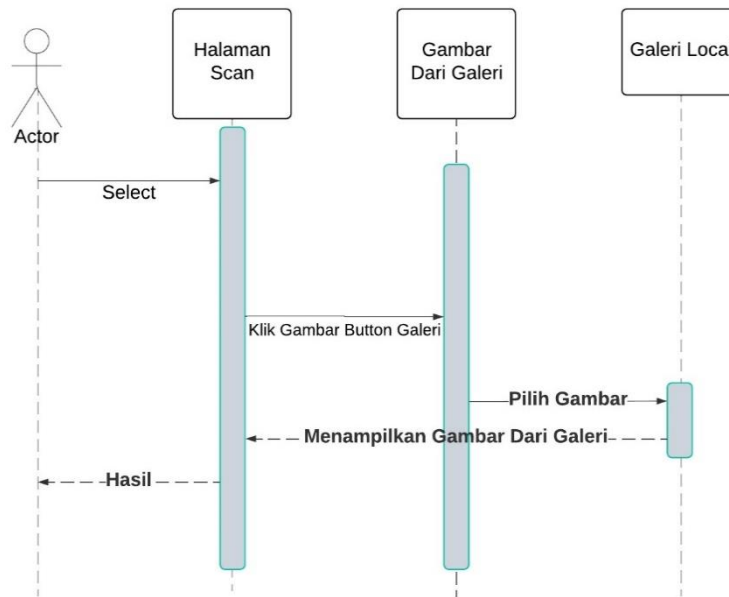
Gambar 3.14 Activity Diagram Halaman Scan

### 3.4.3 Sequence Diagram

#### 1. Sequence Diagram Pilih Dari Galeri

Proses pengambilan gambar dari galeri pengguna digambarkan dalam urutan diagram di bawah. Ketika pengguna memilih menu gambar dari galeri, sistem akan menampilkan gambar yang telah disimpan sebelumnya di galeri. Kemudian, ketika pengguna memilih gambar yang diinginkan, sistem akan mengembalikannya dan menampilkannya di halaman deteksi penyakit tanaman.

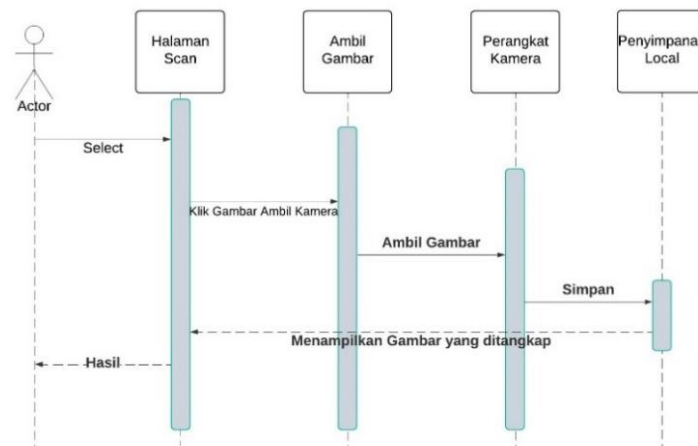




Gambar 3.15 Sequence Diagram Gambar dari Galeri

## 2. Sequence Diagram Tangkap Gambar

Sequence diagram di bawah menunjukkan cara sistem mengaktifkan kamera smartphone. Saat user memilih untuk mengambil gambar, sistem akan mengakses kamera di perangkat user. Setelah itu, kamera akan menyimpan gambar tersebut ke dalam penyimpanan lokal secara sementara, dan ketika penyimpanan lokal berhasil, sistem akan menampilkan gambar kepada user.



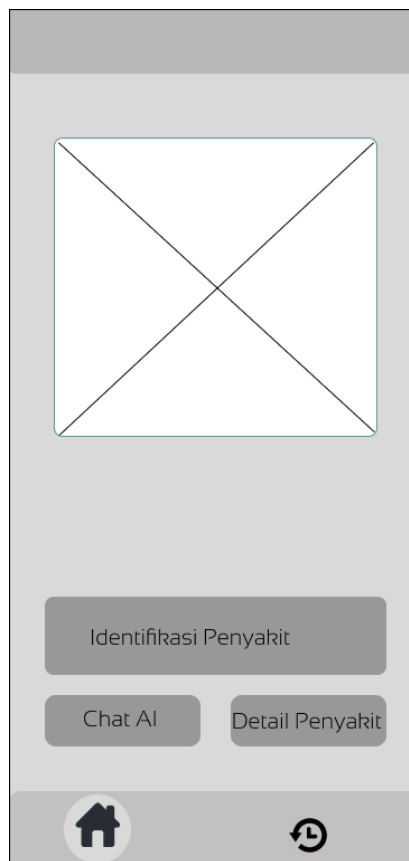
Gambar 3.16 Sequence Diagram Tangkap Gambar

### 3.4 Tampilan Aplikasi

#### 3.4.1 Tampilan Home

Pada rancangan tampilan fitur deteksi penyakit, terdapat beberapa tombol dengan fungsi sebagai berikut:

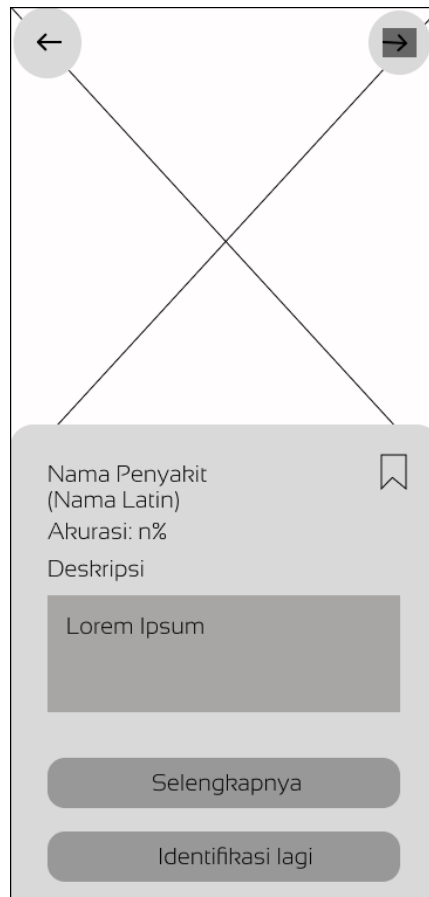
- Identifikasi Penyakit: Tombol ini akan membuka kamera untuk mengambil gambar secara langsung.
- Chat dari AI: Tombol ini akan pergi ke halaman chat dengan AI
- Detail Penyakit: Berisi bebragai informasi terkait penyakit selada yang tersedia di aplikasi ini.



Gambar 3.17 Tampilan Halaman Home

### 3.4.2 Tampilan Halaman Hasil Scan

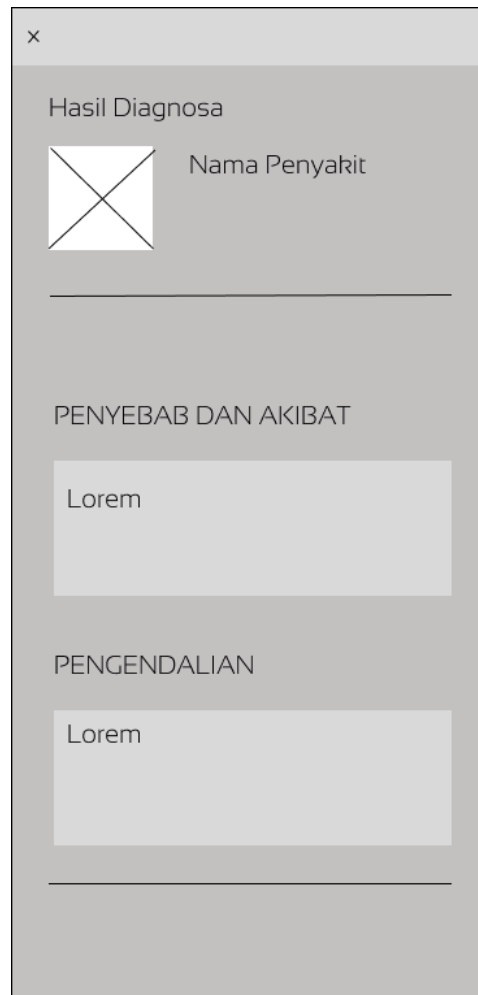
Pada halaman ini, hasil scan akan ditampilkan dan berisi akurasi dan deskripsi singkat dari penyakit yang dideteksi dan terdapat button selengkapnya untuk melihat detail penyakit.



Gambar 3.18 Tampilan Halaman hasil Scan

### 3.4.3 Tampilan Detail Penyakit

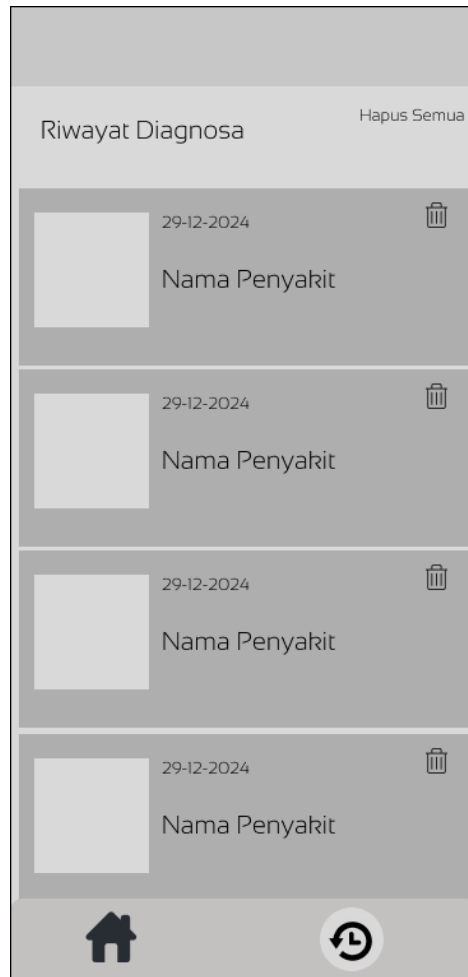
Halaman ini merupakan detail dan hasil dari scan, halaman ini memuat nama penyakit, penyebab penyakit, dan solusi pengobatan dari penyakit tersebut.



Gambar 3.19 Interface Hasil Scan

### 3.4.4 Tampilan halaman history

Halaman Riwayat Diagnosa menampilkan daftar diagnosa beserta tanggalnya, dengan opsi untuk menghapus entri secara individu atau seluruh riwayat



Gambar 3.20 Gambar riwayat hasil