

**PENGEMBANGAN APLIKASI PERANGKAT BERGERAK
IDENTIFIKASI PENYAKIT DAUN JERUK BERBASIS ANDROID
DENGAN MEMANFAATKAN VIZE AI**

(Studi Kasus : Balai Penelitian Tanaman Jeruk dan Buah Subtropika)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Sisco Jupiyandi
NIM: 155150201111236



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PERSETUJUAN

PENGEMBANGAN APLIKASI PERANGKAT BERGERAK IDENTIFIKASI PENYAKIT
DAUN JERUK BERBASIS ANDROID DENGAN MEMANFAATKAN VIZE AI

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Sisco Jupiyandi
NIM: 155150201111236

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Agi Putra Kharisma, S.T., M.T.
NIK: 201304 860430 1 001

Dr. Ir. Anang Triwiratno, M.P.
NIP: 19670107 199103 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 12 Desember 2018

Sisco Jupiyandi

NIM: 155150201111236

KATA PENGANTAR

Segala puji dan syukur bagi Allah SWT atas limpahan rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi yang berjudul “Pengembangan Aplikasi Perangkat Bergerak Identifikasi Penyakit Daun Jeruk Berbasis Android dengan Memanfaatkan Vize AI”. Hasil dari penelitian ini penulis persembahkan kepada pihak-pihak yang memberikan bantuan, bimbingan, saran dan doa dalam pelaksanaan skripsi ini. Maka dari pada itu, penulis berterima kasih kepada :

1. Kepada orang tua penulis yang penulis cintai yaitu Bapak Mardi dan Ibu Waliyah beserta keluarga besar yang selalu memberikan dukungan, doa, motivasi dan semangat.
2. Bapak Agi Putra Kharisma, S.T., M.T. sebagai dosen pembimbing I dan Bapak Dr. Ir. Anang Triwiratno, M.P. sebagai dosen pembimbing II atas segala bimbingan, masukan, dan saran yang bermanfaat bagi penulis selama proses penelitian skripsi ini.
3. Seluruh Bapak dan Ibu Dosen Fakultas Ilmu Komputer, Universitas Brawijaya atas ketersedianya membimbing dan memberikan ilmu yang bermanfaat kepada penulis.
4. Seluruh Keluarga Besar Basic Computing Community FILKOM UB yang telah memberikan ilmu, dukungan, pengalaman dan semangat dalam berorganisasi.
5. Ofi Eka Novyanti yang telah memberikan ketulusan dan motivasi bagi penulis sehingga dapat menyelesaikan penelitian skripsi ini.
6. Para Peneliti dan Kepala Kebun di Balai Penelitian Tanaman Jeruk dan Buah Subtropika (Balitjestro) yang telah membantu dalam menyelesaikan penelitian skripsi ini.
7. Semua pihak yang telah membantu kelancaran penelitian skripsi ini serta memberikan doa yang tidak dapat disebutkan penulis satu persatu.

Penulis peneliti merasa bahwa penelitian ini belum bisa dikatakan sempurna. Untuk itu, demi kesempurnaan penelitian skripsi ini, penulis mengharapkan saran dan kritik sebagai perbaikan untuk penelitian selanjutnya. Semoga laporan skripsi ini bermanfaat bagi semua pihak.

Malang, 12 Desember 2018

Penulis
sjupiyandi@gmail.com

ABSTRAK

Tanaman jeruk merupakan salah satu tanaman buah yang penting dan diperhatikan khusus oleh pemerintah Indonesia. Berdasarkan penelitian yang dilakukan Kementerian Pertanian Indonesia, tanaman jeruk mengalami perkembangan yang cukup pesat setiap tahunnya baik dalam hal luas lahan pertaniannya, jumlah produksi serta jumlah permintaan pasar yang terus meningkat. Walaupun hasil produksi tanaman jeruk terus meningkat namun pada kenyataanya di lapangan para petani jeruk memiliki banyak kendala, salah satunya yaitu kurangnya pengetahuan tentang kualitas tanaman jeruk. Vize AI adalah *web service* yang menyediakan API untuk pengenalan gambar yang dapat dilatih untuk mengenali dan mengklasifikasikan citra atau gambar secara *custom*. Penulis memanfaatkan peluang untuk mengembangkan aplikasi berbasis *mobile* dengan OS Android untuk mengidentifikasi penyakit pada citra daun jeruk yang dibagi atas 3 penyakit yaitu *Downy Mildew*, Cendawan Jelaga dan *Citrus Vein Phloem Degeneration* (CVPD) dengan memanfaatkan Vize AI. Aplikasi yang telah dibuat penulis terdapat fitur-fitur yang disesuaikan dengan kebutuhan pengguna, dan diimplementasikan dengan bahasa pemrograman Java. Setelah dilakukan pengujian akurasi, hasil akurasi aplikasi ini dalam mendeteksi penyakit daun jeruk memiliki akurasi yang sangat akurat dengan persentase 100% dan untuk akurasi perhitungan tingkat keparahan memiliki rata-rata akurasi 81,25% yang dapat dibilang cukup baik. Namun hasil tersebut masih bergantung pada jenis kualitas gambarnya itu sendiri.

Kata kunci: Identifikasi, Penyakit Daun, Jeruk, Android, Vize AI

ABSTRACT

Citrus plants are one of the important plants that get a special consideration by the Indonesian government. Based on research conducted by the Indonesian Ministry of Agriculture, citrus plants have rapid experienced development every year both in terms of the area of agricultural, the amount of production, and the increasing number of market demands. Even though the production of citrus plants continues to increase, in reality the citrus farmers have many obstacles in the field, one of them is the lack of knowledge about the quality of citrus plants Vize AI is a web service that provides an API for image recognition that can be trained to recognize and classify any kinds of images including a custom image. The author takes advantage of the opportunity to develop mobile-based applications with Android OS that can identify diseases from image of citrus leaves into 3 diseases, namely Downy Mildew, Cendawan Jelaga, and Citrus Vein Phloem Degeneration (CVPD) by using Vize AI. Applications that have been made by the author have some features that meet the user's need and implemented in Java programming language. After accuracy test is done, the results of the accuracy of this application in detecting citrus leaf's disease is highly accurate result with a percentage of 100% and accuracy of calculation of severity has an average accuracy of 81.25% which can be considered as good result. But these results still depend on the type of image quality itself.

Keywords: Identification, Leaf Disease, Orange, Android, Vize AI

DAFTAR ISI

PERSETUJUANii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR KODE PROGRAM	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Jeruk Keprok Batu 55	5
2.3 Penyakit Daun Jeruk	6
2.3.1 <i>Citrus Vein Phloem Degeneration (CVPD)</i>	6
2.3.2 Cendawan Jelaga	7
2.3.3 <i>Downy Mildew</i>	8
2.4 Android	8
2.5 Vize AI	9
2.6 <i>SQLite</i>	9
2.7 <i>System Development Life Cycle Waterfall</i>	9
2.8 Pengujian	10
2.8.1 Pengujian Fungsional	10
2.8.2 Pengujian Akurasi.....	10

2.8.3 Pengujian Usabilitas	10
2.9 <i>Standardized User Experience Percentile Rank Questionnaire for Mobile Apps (SUPR-Qm)</i>	10
BAB 3 METODOLOGI	12
3.1 Studi Literatur	12
3.2 Pengumpulan Data	13
3.3 Analisis Kebutuhan	13
3.4 Perancangan Sistem.....	13
3.5 Implementasi	13
3.6 Pengujian Sistem.....	13
3.7 Kesimpulan dan Saran	14
BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN	15
4.1 Analisis Kebutuhan	15
4.1.1 Gambaran Umum Sistem	15
4.1.2 Analisis Kebutuhan Perangkat Lunak.....	16
4.2 Perancangan	17
4.2.1 Diagram <i>Unified Modeling Language (UML)</i>	17
4.2.2 Perancangan Basis Data	29
4.2.3 Perancangan Algoritme.....	29
4.2.4 Perancangan Antarmuka.....	31
BAB 5 IMPLEMENTASI	36
5.1 Batasan Implementasi	36
5.2 Spesifikasi Sistem	36
5.2.1 Spesifikasi Perangkat Keras.....	36
5.2.2 Spesifikasi Perangkat Lunak	37
5.3 Implementasi Basis Data.....	37
5.4 Implementasi Vize Ai	37
5.4.1 <i>Define</i>	38
5.4.2 <i>Train</i>	39
5.4.3 <i>Recognize</i>	39
5.5 Implementasi Kode Program	40
5.5.1 Implementasi Kode Program Mendeteksi Penyakit	40

5.5.2 Implementasi Kode Program Menghitung Tingkat Keparahan ..	42
5.6 Implementasi Antarmuka Pengguna	47
BAB 6 PENGUJIAN	50
6.1 Pengujian Fungsional	50
6.2 Pengujian Akurasi	56
6.2.1 Hasil Pengujian Akurasi	56
6.3 Pengujian Usabilitas.....	64
6.3.1 Hasil Pengujian Usabilitas	66
6.4 Analisis Hasil Pengujian.....	69
6.4.1 Analisis Hasil Pengujian Fungsional	69
6.4.2 Analisis Hasil Pengujian Akurasi	69
6.4.3 Analisis Hasil Pengujian Usabilitas	70
BAB 7 PENUTUP	71
7.1 Kesimpulan.....	71
7.2 Saran	72
Daftar PUSTAKA	73
LAMPIRAN A DATA LATIH GAMBAR DAUN JERUK	75
LAMPIRAN B DATA UJI GAMBAR DAUN JERUK.....	99

DAFTAR TABEL

Tabel 1.1 Produksi Buah Jeruk di Indonesia Tahun 2012 - 2016	1
Tabel 2.1 Kajian Pustaka Berkaitan dengan Kasus.....	5
Tabel 4.1 Aktor Sistem	16
Tabel 4.2 Kebutuhan Fungsional Sistem	16
Tabel 4.3 <i>Use Case Scenario</i> Mengambil Gambar	18
Tabel 4.4 <i>Use Case Scenario</i> Crop Gambar	18
Tabel 4.5 <i>Use Case Scenario</i> Mendeteksi Penyakit dan Tingkat Keparahannya...	19
Tabel 4.6 <i>Use Case Scenario</i> Mengetahui Informasi Cara Pengendalian	19
Tabel 4.7 <i>Use Case Scenario</i> Mengetahui Riwayat Gambar Yang Telah Dideteksi	20
Tabel 4.8 <i>Use Case Scenario</i> Mengetahui Informasi Penyakit Daun Jeruk.....	20
Tabel 4.9 <i>Pseudocode</i> Algoritme Mendeteksi Penyakit.....	29
Tabel 4.10 <i>Pseudocode</i> Algoritme Menghitung Tingkat Keparahan.....	30
Tabel 5.1 Spesifikasi Perangkat Laptop ASUS A456UR	36
Tabel 5.2 Spesifikasi Perangkat Android	36
Tabel 5.3 Spesifikasi Perangkat Lunak Laptop ASUS A455UR.....	37
Tabel 5.4 Spesifikasi Perangkat Lunak Android	37
Tabel 6.1 Pengujian Fungsional Sistem.....	50
Tabel 6.2 Hasil Pengujian Akurasi Data Latih	57
Tabel 6.3 Hasil Pengujian Akurasi Data Uji	61
Tabel 6.4 <i>Task Scenario</i> Pengujian	64
Tabel 6.5 Skor Skala Likert dari Setiap Pertanyaan.....	65
Tabel 6.6 Kuesioner SUPR-Qm	65
Tabel 6.7 Responden Pengujian Usabilitas	66
Tabel 6.8 <i>Task Completion Rate</i>	67
Tabel 6.9 Hasil Pengujian Kuisioner SUPR-Qm	67
Tabel 6.10 Saran dan Umpan Balik Responden	69

DAFTAR GAMBAR

Gambar 1.1 Pengguna <i>Smartphone</i> di Indonesia Tahun 2013 - 2018	2
Gambar 2.1 Detail Jeruk Keprok Batu 55	6
Gambar 2.2 Daun Berpenyakit CVPD	7
Gambar 2.3 Daun Berpenyakit Cendawan Jelaga	7
Gambar 2.4 Daun Berpenyakit <i>Downy Mildew</i>	8
Gambar 2.5 Tahapan Model <i>Waterfall</i>	9
Gambar 2.6 Pertanyaan Pada Kuesioner SUPR-Qm.....	11
Gambar 2.7 Kategori Nilai <i>Usability</i>	11
Gambar 3.1 Metode Pengembangan Sistem	12
Gambar 4.1 Gambaran Penggunaan Sistem	15
Gambar 4.2 <i>Use Case Diagram</i>	17
Gambar 4.3 <i>Sequence Diagram</i> Mengambil Gambar	21
Gambar 4.4 <i>Sequence Diagram</i> Crop Gambar	22
Gambar 4.5 <i>Sequence Diagram</i> Mendeteksi Penyakit dan Tingkat Keparahannya	23
Gambar 4.6 <i>Sequence Diagram</i> Mengetahui Informasi Cara Pengendalian	24
Gambar 4.7 <i>Sequence Diagram</i> Mengetahui Riwayat Gambar Yang Telah Dideteksi	24
Gambar 4.8 <i>Sequence Diagram</i> Mengetahui Informasi Penyakit Daun Jeruk.....	25
Gambar 4.9 <i>Class Diagram</i>	26
Gambar 4.10 <i>Activity Diagram</i>	28
Gambar 4.11 ERD Aplikasi LEAD	29
Gambar 4.12 <i>Screenflow</i> Mendeteksi Penyakit	31
Gambar 4.13 <i>Screenflow</i> Melihat Daftar Riwayat	32
Gambar 4.14 <i>Sreenflow</i> Melihat Daftar Penyakit	32
Gambar 4.15 Antarmuka Halaman Awal	33
Gambar 4.16 Antarmuka Halaman Mengambil Gambar	33
Gambar 4.17 Antarmuka Halaman Detail Gambar	34
Gambar 4.18 Antarmuka Halaman Hasil Deteksi.....	34
Gambar 4.19 Antarmuka Halaman Daftar Riwayat	35

Gambar 4.20 Antarmuka Halaman Daftar Penyakit	35
Gambar 5.1 Implementasi <i>Physical Data Model</i> (PDM)	37
Gambar 5.2 Tahapan Implementasi Vize AI.....	38
Gambar 5.3 Implementasi Tahap <i>Define</i>	38
Gambar 5.4 Implementasi Tahap <i>Train</i>	39
Gambar 5.5 Implementasi Tahap <i>Recognize</i>	39
Gambar 5.6 Halaman Awal	47
Gambar 5.7 Halaman Mengambil Gambar	47
Gambar 5.8 Halaman Detail Gambar	48
Gambar 5.9 Halaman Hasil Deteksi.....	48
Gambar 5.10 Halaman Daftar Riwayat	49
Gambar 5.11 Halaman Daftar Penyakit	49

DAFTAR KODE PROGRAM

Kode Program 5.1 Implementasi Mendeteksi Penyakit	40
Kode Program 5.2 Implementasi Mengitung Tingkat Keparahan.....	42

BAB 1 PENDAHULUAN

1.1 Latar belakang

Tanaman jeruk merupakan salah satu tanaman buah yang penting dan diperhatikan khusus oleh pemerintah Indonesia. Untuk meningkatkan produksi buah dan kualitas tanaman jeruk, pemerintah Indonesia melalui Kementerian Pertanian (Kementan) berkomitmen dengan membentuk Balai Penelitian Tanaman Jeruk dan Buah Subtropika (Balitjestro). Balitjestro memiliki kebijakan strategis yaitu dengan menciptakan inovasi pertanian yang diimplementasikan melalui pemerintah daerah dan *stakeholder* terkait guna meningkatkan produksi dan kualitas tanaman jeruk (Balitjestro, 2017).

Berdasarkan penelitian yang dilakukan Kementerian Pertanian Indonesia, tanaman jeruk merupakan tanaman buah tahunan yang sudah banyak dibudidayakan di Indonesia yaitu sekitar 70-80% dan mengalami perkembangan yang cukup pesat setiap tahunnya baik dalam hal luas lahan pertaniannya, jumlah produksi serta jumlah permintaan pasar yang terus meningkat (Rizal, et al., 2011). Produksi buah jeruk setiap tahunnya selalu mengalami pengingkatan yang dijelaskan pada Tabel 1.1. Pada tahun 2016 jumlah produksi jeruk yaitu 2.014.206 Ton yang meningkat sekitar 15,47% dibandingkan tahun sebelumnya. Jumlah ini perlu diperhatikan oleh Kementerian Pertanian Indonesia agar pemerintah dapat menciptakan kebijakan strategis yang dapat membantu meningkatkan produksi dan kualitas tanaman jeruk.

Tabel 1.1 Produksi Buah Jeruk di Indonesia Tahun 2012 - 2016

No	Tahun	Produksi (Ton)
1	2012	1.498.394
2	2013	1.548.394
3	2014	1.785.256
4	2015	1.744.330
5	2016	2.014.206

Sumber : Statistik Pertanian (Kementan, 2017)

Walaupun hasil produksi tanaman jeruk terus meningkat namun pada kenyataannya di lapangan para petani jeruk memiliki banyak kendala, salah satunya yaitu kurangnya pengetahuan tentang kualitas tanaman jeruk sehingga menyebabkan rendahnya produktifitas buah jeruk. Penyakit pada daun jeruk merupakan salah satu faktor yang dapat mempengaruhi kualitas tanaman jeruk. Daun pada tanaman berfungsi sebagai tempat pengolahan zat makanan atau yang disebut fotosintesis. Jika daun terserang penyakit maka tumbuhan tidak mampu mengolah zat makanan sehingga pertumbuhan tanaman terhambat dan kualitas buah yang dihasilkan menjadi rendah serta dapat membuat tanaman layu atau mati. Adapun penyakit yang menyerang daun jeruk antara lain adalah *Downy Mildew*, Cendawan Jelaga dan *Citrus Vein Phloem Degeneration* (CVPD) (Triwiratno, 2018).

Para petani jeruk saat ini mengidentifikasi penyakit pada tanaman jeruk adalah dengan cara melihat langsung sehingga sangat rawan terjadi kesalahan. Petani jeruk perlu diberi pengetahuan lebih untuk mengidentifikasi penyakit pada tanaman sehingga tidak terjadi sesuatu yang dapat mengurangi kualitas tanaman jeruk. Agar kekhawatiran petani jeruk dapat teratasi diperlukan solusi untuk mengidentifikasi penyakit tanaman jeruk secara otomatis (Triwiratno, 2018).

Seiring dengan banyaknya inovasi yang dikembangkan pada saat ini, teknologi memiliki banyak kemajuan terutama di bidang *machine learning* dalam pengenalan citra digital. *Machine learning* dalam pengenalan citra digital dapat digunakan manusia dalam memroses input citra gambar yang nantinya diberi perlakuan khusus menggunakan algoritme tertentu untuk mendapatkan suatu data yang dibutuhkan. Terdapat sebuah *web service* yang digunakan untuk mengenali suatu citra dan mengklasifikasikannya yaitu Vize AI. Dengan adanya Vize AI memungkinkan untuk menerapkan kecerdasan buatan yang mutakhir ke dalam aplikasi hanya dengan menggunakan API yang terhubung dengan internet (Vize, 2018).

Saat ini internet sangat mudah diakses dimanapun dan kapanpun untuk mendapatkan informasi yang kita inginkan melalui *smartphone*. Dengan kemudahan akses internet tersebut memungkinkan meningkatnya pengguna *smartphone*. Pengguna *smartphone* di Indonesia terus meningkat, tercatat sekitar 103 juta pada tahun 2018 mengingkat sebanyak 16,4 juta dibandingkan tahun sebelumnya yang dijelaskan di Gambar 1.1 dan menurut data statistik di Indonesia, sekitar 80% nya adalah OS Android.



Gambar 1.1 Pengguna *Smartphone* di Indonesia Tahun 2013 - 2018

Sumber : (Techinasia, 2018)

Penulis memanfaatkan peluang untuk mengembangkan aplikasi berbasis *mobile* dengan OS Android untuk mengidentifikasi penyakit pada citra daun jeruk yang dibagi atas 3 penyakit yaitu *Downy Mildew*, Cendawan Jelaga dan *Citrus Vein Phloem Degeneration* (CVPD) dengan memanfaatkan Vize AI. Aplikasi yang akan dibuat oleh penulis adalah aplikasi untuk membantu petani jeruk mengidentifikasi penyakit daun jeruk dan tingkat keparahannya serta memberikan pengetahuan cara pengendalian yang baik dan benar sesuai penyakit yang menyerang daun jeruk tersebut. Berdasarkan latar belakang tersebut maka penulis memberi judul pada penelitian ini yaitu “Pengembangan Aplikasi Perangkat Bergerak Identifikasi Penyakit Daun Jeruk Berbasis Android dengan Memanfaatkan Vize AI”.

1.2 Rumusan masalah

Berdasarkan latar belakang permasalahan tersebut, maka penulis dapat merumuskan masalah adalah sebagai berikut :

1. Bagaimana kebutuhan fungsional dan non-fungsionalitas dari aplikasi identifikasi penyakit daun jeruk?
2. Bagaimana implementasi Vize AI untuk kasus identifikasi penyakit daun jeruk pada aplikasi perangkat bergerak berbasis android?
3. Bagaimana tingkat akurasi pada aplikasi identifikasi penyakit daun jeruk?

1.3 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut :

1. Mengetahui kebutuhan fungsional dan non-fungsionalitas dari aplikasi identifikasi penyakit daun jeruk.
2. Mengimplementasikan Vize AI untuk kasus identifikasi penyakit daun jeruk pada aplikasi perangkat bergerak berbasis android.
3. Mengetahui tingkat akurasi pada aplikasi identifikasi penyakit daun jeruk.

1.4 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut :

1. Menyediakan aplikasi yang dapat membantu peneliti Balai Penelitian Tanaman Jeruk dan Buah Subtropika (Balitjestro) dan membuat petani jeruk untuk mengenali penyakit daun jeruk dan cara pengendaliannya.
2. Dengan mengehui kinerja dari aplikasi pada proses identifikasi penyakit daun jeruk diharapkan dapat digunakan sebagai bahan referensi atau rujukan untuk pengembangan selanjutnya.

1.5 Batasan masalah

Penelitian ini memiliki beberapa batasan dalam pengembangannya, diantaranya :

1. Jenis jeruk yang dipakai keprok batu 55.

2. Penelitian ini difokuskan untuk mengidentifikasi penyakit daun jeruk dengan menggunakan Vize AI terhadap penyakit *Downy Mildew*, Cendawan Jelaga dan CVDP.
3. Pengambilan gambar daun menggunakan kamera OPPO A37f 8 MP dan menggunakan alat bantu *background cover* warna merah.

1.6 Sistematika pembahasan

Sistematika penyusunan dokumen skripsi ini dibagi menjadi beberapa bab, sebagai berikut :

BAB I Pendahuluan

Bagian ini menjelaskan tentang latar belakang permasalahan, rumusah masalah, tujuan dari penelitian, manfaat dari penelitian, batasan penelitian, serta sistematika pembahasan skripsi ini.

BAB II Landasan Kepustakaan

Bagian ini menjelaskan tentang kajian-kajian kepustakaan yang terkait yang digunakan untuk mengidentifikasi penyakit daun jeruk yang dijadikan referensi dalam pembuatan skripsi ini.

BAB III Metodologi

Bagian ini menjelaskan tentang alur kerja penelitian yang dilakukan dan proses menyelesaikan masalah penelitian.

BAB IV Analisis Kebutuhan & Perancangan

Bagian ini menjelaskan tentang segala hal yang terkait seputar proses penggalian kebutuhan serta berisikan tentang hal-hal yang akan berkaitan dengan perancangan sistem berdasarkan data yang telah didapat di tahap sebelumnya.

BAB V Implementasi

Bagian ini menjelaskan tentang proses yang berkaitan dengan implementasi pengembangan sistem berdasarkan pemodelan yang sudah dilakukan di tahap sebelumnya.

BAB VI Pengujian

Bagian ini menjelaskan tentang hal-hal yang berkaitan dengan pengujian sistem yang dilakukan oleh peneliti untuk menganalisis hasil yang telah didapat.

BAB VII Penutup

Bagian ini menjelaskan tentang kesimpulan yang telah diperoleh dari penelitian yang telah dilakukan serta memuat saran untuk pengembangan penelitian selanjutnya agar lebih baik.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisikan landasan kepustakaan yang terdiri dari kajian pustaka dan teori-teori dasar yang berkaitan dengan penelitian. Kajian pustaka berisi penelitian yang dilakukan sebelumnya yang berkaitan dengan kasus dan juga terdapat teori yang berkaitan antara lain tanaman Jeruk Keprok Batu 55, penyakit daun jeruk, android, Vize AI, konsep *waterfall*, SQLite dan teknik pengujian yang digunakan.

2.1 Kajian Pustaka

Terdapat penelitian yang dilakukan sebelumnya yang berkaitan dengan kasus identifikasi penyakit daun jeruk yang akan dijelaskan di tabel 2.1.

Tabel 2.1 Kajian Pustaka Berkaitan dengan Kasus

No.	Judul	Sumber	Hasil
1.	Implementasi Metode <i>K-Nearest Neighor</i> untuk Identifikasi Penyakit Tanaman Jeruk Keprok Berdasarkan Citra Daun	(Priambodo, et al., 2015)	Berdasarkan penelitian tersebut dihasilkan akurasi sekitar 96,67% yaitu dapat mengidentifikasi 29 dari 30 data yang ada.
2.	Implementasi Algoritma <i>Multilevel Thresholding</i> Menggunakan <i>Otsu</i> sebagai <i>Preprocessing</i> Data Citra Daun pada Proses Identifikasi Penyakit Tanaman Jeruk	(Rizal, et al., 2017)	Berdasarkan penelitian tersebut dihasilkan akurasi tertinggi sebesar 92% pada data defisiensi Zn, 80% pada defisiensi Mg, dan 80% pada defisiensi Mg dan Zn.
3.	Implementasi Algoritma <i>K-Means</i> sebagai Metode Segmentasi Citra dalam Identifikasi Penyakit Daun Jeruk	(Febrinanto, et al., 2018)	Berdasarkan penelitian tersebut dihasilkan akurasi mengidentifikasi mencapai 99,17%

2.2 Jeruk Keprok Batu 55

Jeruk Keprok Batu 55 masuk dalam salah satu kelompok jenis jeruk yaitu jeruk mandarin. Masyarakat pada umumnya belum banyak mengetahui bahwa jeruk keprok ini lazim disebut sebagai jeruk mandarin. Walaupun dikategorikan sama, namun rasa dan kualitas jeruk keprok batu 55 lebih baik jika dibandingkan dengan jeruk impor. Hal tersebut tentu menjadi potensi yang bagus bagi petani jeruk jika pengelolaan dan budidaya jeruk dilakukan dengan baik dan serius.

Peluang tersebut dapat diyakini sebagai salah satu langkah yang dapat dilakukan untuk menyejahterakan petani jeruk di Indonesia (Setiono, 2015). Adapun detailnya dijelaskan pada Gambar 2.1



Gambar 2.1 Detail Jeruk Keprok Batu 55

Sumber : (Balitjestro, 2018)

2.3 Penyakit Daun Jeruk

2.3.1 *Citrus Vein Phloem Degeneration (CVPD)*

Salah satu jenis penyakit pada jeruk ini disebabkan oleh bakteri gram negatif yang dikategorikan ke dalam *Proteobacteria* (Meitayani, et al., 2014). Penyakit berbahaya ini dapat diketahui dengan melihat beberapa gejala yang dikelompokkan menjadi dua yaitu gejala luar dan gejala dalam. Karakteristik kedua gejala tersebut tentu berbeda. Pada gejala luar, daun pada jeruk dewasa berwarna kekuningan karena kekurangan unsur hara seperti Zinc, Mn, dan zat besi. Selain itu, warna tulang daun terlihat lebih hijau dan terasa halus. Sedangkan pada gejala dalam, penampakan floem akan terlihat lebih tebal pada tangkai daun yang diiris secara melintang .

Penyakit ini memberikan dampak negatif yang signifikan. Jika tidak dilakukan pencegahan secara menyeluruh dapat mengurangi produktifitas tanaman jeruk seperti ukuran buah yang kecil dan tidak berkembang, serta keguguran buah. Buah yang tidak gugur pun memiliki kualitas yang rendah. Selain itu, CVPD dapat menyebabkan klorosis, dimana tanaman jeruk tidak dapat melakukan proses fotosintesis dengan baik. Hal ini disebabkan karena daun sebagai komponen utama tidak mampu menyalurkan makanan untuk bagian tanaman yang lain. Terhambatnya proses fotosintesis mengakibatkan tanaman menjadi gersang, layu,

lalu mati (Wahyuningsih, 2009). Adapun gambar daunnya ditunjukan pada Gambar 2.2.



Gambar 2.2 Daun Berpenyakit CVPD

2.3.2 Cendawan Jelaga

Jamur *Capnodium citri* pada tanaman jeruk dapat menyebabkan penyakit cendawan jelaga. Pada umumnya penyakit ini dikenal dengan sebutan penyakit embun jelaga. Gejala penyakit ini adalah daun, ranting, dan buah terlihat diselimuti jamur berwarna hitam. Penyakit cendawan Jelaga dapat menyebar ketika jamur terkelupas kemudian terbawa oleh angin dan menempel pada tanaman sehat. Keberadaan penyakit ini dapat mengakibatkan panen menjadi terhambat karena buah terlambat matang dan buah yang dihasilkan memiliki ukuran yang kecil. Cepatnya perkembang biakan jamur *Capnodium citri* didukung oleh adanya hama kutu daun yang mensekresi embun madu. Hama kutu daun inilah yang menjadi inang bagi pertumbuhan jamur *Capnodium citri* (Syafril, 2006). Adapun gambar daunnya ditunjukan pada Gambar 2.3.



Gambar 2.3 Daun Berpenyakit Cendawan Jelaga

2.3.3 Downy Mildew

Penyakit yang dikenal dengan sebutan penyakit embun tepung ini disebabkan oleh jamur yang terbentuk seperti lapisan tepung berwarna putih yang menyelimuti permukaan daun pada tanaman jeruk dan menyebabkan daun menjadi kering. Jamur ini biasanya menyerang tanaman jeruk pada fase pertunasannya dan pada saat pertumbuhan daun muda. Keberadaan penyakit ini juga dapat menyebabkan keguguran pada buah. Penyakit *Downy Mildew* terjadi pada cuaca lembab yang diselingi paparan sinar matahari untuk beberapa jam di musim hujan.

Pencegahan penyakit ini dapat dilakukan dengan memotong bakal cabang yang baru tumbuh, menyemprot larutan fungisida dengan komposisi *Propineb*, *Siprokonozal*, dan *Benomil* pada fase tunas dan disemprot ulang saat fase pertumbuhan daun muda (Syafril, 2006). Adapun gambar daunnya ditunjukkan pada Gambar 2.4.



Gambar 2.4 Daun Berpenyakit *Downy Mildew*

2.4 Android

Android merupakan suatu platform pemrograman atau sistem operasi seluler berbasis *kernel Linux* yang dikembangkan oleh Google yang menjalankan miliaran perangkat yaitu *smartphone*, tablet, jam tangan, TV dan lain sebagainya. Android menyediakan SDK bagi *developer* secara terbuka dan bebas untuk dikembangkan, hal inilah yang membuat perkembangan Android semakin pesat. Android dapat digunakan pada perangkat bergerak yang beraneka ragam, sehingga memudahkan pengguna untuk berinteraksi secara kontinu. Android pula memiliki banyak versi dengan nama-nama yang unik yaitu menggunakan nama makanan, antara lain : Kitkat, Lollipop, Marshmallow, Nougat, Oreo dan yang baru di android yaitu Android Pie yang lebih cepat, lebih cerdas dan nyaman digunakan pengguna (Developers, 2018).

2.5 Vize AI

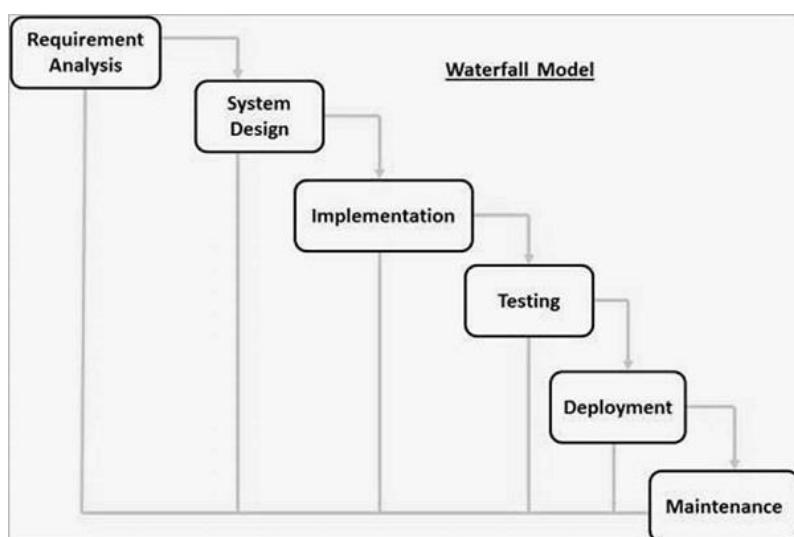
Vize AI adalah *web service* yang menyediakan API untuk pengenalan gambar yang dapat dilatih untuk mengenali dan mengklasifikasikan citra atau gambar secara *custom*. Vize menyediakan antarmuka pengguna yang sederhana yaitu seperti mengelola akun, mengunggah gambar, dan melatih model baru. Untuk mengklasifikasi citra Vize menggunakan *machine learning* dengan langkah awal yaitu pengguna mengelompokkan gambar sesuai kategori, lalu dari gambar tersebut akan dilakukan *training*, pengguna hanya mengirimkan gambar dan mendapatkan hasil yang sangat akurat, mudah dan sangat *scalable* (Vize, 2018).

2.6 SQLite

SQLite adalah salah satu media penyimpanan data yang paling banyak digunakan di dunia dengan ukuran pustaka kode yang relatif kecil. SQLite mengimplementasikan server *database* SQL secara mandiri, tanpa server dan transaksional. Namun, SQLite hanya mendukung beberapa tipe data saja, yaitu *text*, *integer* dan *real*. Apabila ingin menyimpan data selain dari tipe data diatas, maka harus dilakukan konversi tipe data sebelum dilakukan penyimpanan data (SQLite, 2018).

2.7 System Development Life Cycle Waterfall

Waterfall merupakan pendekatan SDLC tertua yang digunakan dalam pengembangan perangkat lunak. *Waterfall* adalah model yang dianggap kuno yaitu pertama kali muncul sekitar tahun 1970 namun model ini adalah model yang paling banyak digunakan dalam pengembangan perangkat lunak. Model ini bersifat serial dan tidak boleh meloncat ketahap selanjutnya, jadi harus melalui tahap sesuai urutannya yaitu mulai dari analisis kebutuhan, desain, implementasi, pengujian, *deployment* dan *maintenance* yang ditujukan pada Gambar 2.5 (Sommerville, 2011).



Gambar 2.5 Tahapan Model Waterfall

Sumber : (Point, 2018)

2.8 Pengujian

2.8.1 Pengujian Fungsional

Pengujian fungsional adalah jenis pengujian yang berfokus pada keluaran hasil yang dilakukan sesuai dengan kasus uji pada masing-masing kebutuhan fungsional. Hasil dari pengujian fungsionalitas mengetahui apakah ada *error* atau fungsi yang tidak bisa berjalan sesuai dan dipresentasikan berupa tabel yang menjelaskan apakah kebutuhan fungsional telah terpenuhi oleh sistem valid atau tidak valid.

2.8.2 Pengujian Akurasi

Pengujian akurasi adalah jenis pengujian yang digunakan untuk mengukur seberapa akurat hasil dalam mengidentifikasi penyakit daun jeruk dari aplikasi yaitu dengan mencoba aplikasi secara langsung dengan mengujinya dengan data latih dan data uji yang ada.

2.8.3 Pengujian Usabilitas

Pengujian usabilitas adalah jenis pengujian dengan melihat cara pengguna berinteraksi dengan sistem yang telah dibangun. Pengujian ini memerlukan sejumlah responden untuk menguji aplikasi secara langsung guna mengetahui apakah aplikasi yang telah dibangun mudah digunakan atau tidak. Hasil dari pengujian ini nantinya didapatkan persentase tingkat kepuasan pengguna pengguna. Pelaksanaan pengujian usabilitas ini dapat dilakukan dengan 5 orang responden (Nielsen, 2004).

2.9 Standardized User Experience Percentile Rank Questionnaire for Mobile Apps (SUPR-Qm)

SUPR-Qm merupakan kuesioner yang difokuskan untuk pengujian usabilitas dalam aplikasi perangkat bergerak dan merupakan pengembangan dari versi kuesioner SUPR-Q. SUPR-Q sendiri merupakan kuesioner yang mengukur kualitas pengalaman pengguna dari sebuah situs web hanya dengan menggunakan 8 pertanyaan saja yang mencakup 4 aspek, yaitu *usability* yang merupakan aspek yang mencakup kemudahan dalam menggunakan sebuah website, *credibility* yang merupakan aspek yang mencakup kepercayaan, nilai dan kenyamanan dalam menggunakan sebuah website, *loyalty* yang merupakan aspek yang mencakup kesetiaan pengguna dalam menggunakan website dan *appearance* yang merupakan aspek yang mencakup tampilan yang baik dari sebuah website (MeasuringU, 2018).

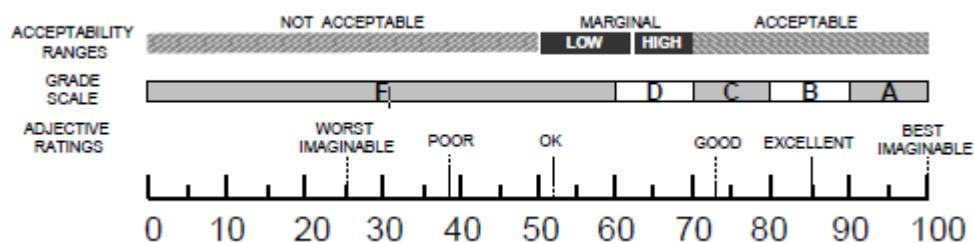
SUPR-Qm sendiri memiliki 16 buah pertanyaan dalam mengukur kualitas pengalaman pengguna dalam sebuah aplikasi perangkat bergerak atau *mobile* dan dari 16 buah pertanyaan tersebut dinilai menggunakan skala likert dalam rentang 1 sampai dengan 5. Adapun 16 buah pertanyaan tersebut ditunjukkan pada Gambar 2.6.

Item	Logit Position	Full Item Wording
CantLiveWo	1.55	I can't live without the app on my phone.
AppBest	1.50	The app is the best app I've ever used.
CantImagineBetter	0.88	I can't imagine a better app than this one.
NeverDelete	0.70	I would never delete the app.
EveryoneHave	0.50	Everyone should have the app.
Discover	0.32	I like discovering new features on the app.
AllEverWant	0.05	The app has all the features and functions you could ever want.
UseFreq	0.03	I like to use the app frequently.
Delightful	-0.04	The app is delightful.
Integrates	-0.04	This app integrates well with the other features of my mobile phone.
DefFuture	-0.30	I will definitely use this app many times in the future.
FindInfo	-0.59	The design of this app makes it easy for me to find the information I'm looking for.
AppAttractive	-0.71	I find the app to be attractive.
AppMeetsNeeds	-0.79	The app's capabilities meet my requirements.
EasyNav	-1.44	It is easy to navigate within the app.
Easy	-1.63	The app is easy to use.

Gambar 2.6 Pertanyaan Pada Kuesioner SUPR-Qm

Sumber : (Sauro & Zarolia, 2017)

Pada Gambar 2.6 merupakan 16 pertanyaan SUPR-Qm yang digunakan untuk mengukur kualitas pengalaman pengguna dalam menggunakan aplikasi perangkat bergerak. Nantinya didapatkan nilai SUPR-Qm dan dikenversikan ke dalam kategori nilai *usability* yang ditunjukkan pada Gambar 2.7.

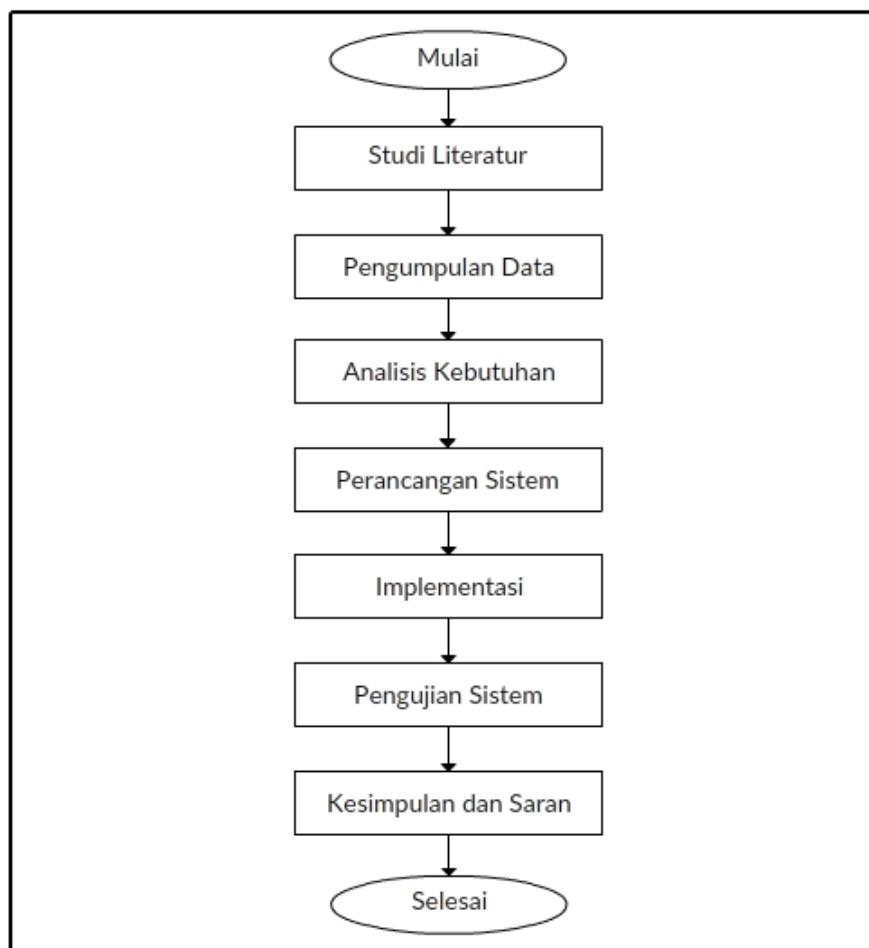


Gambar 2.7 Kategori Nilai *Usability*

Sumber : (Bangor, et al., 2009)

BAB 3 METODOLOGI

Metodologi penelitian menjelaskan tahapan yang dilakukan peneliti dalam mengembangkan perangkat lunak. Penelitian ini menggunakan metode *System Development Life Cycle Waterfall* dengan alur metode digambarkan pada Gambar 3.1.



Gambar 3.1 Metode Pengembangan Sistem

Pada Gambar 3.1 dijelaskan bahwa dalam penelitian ini terdapat 7 tahapan antara lain studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, pengujian sistem serta kesimpulan dan saran.

3.1 Studi Literatur

Pada tahap ini digunakan dengan mempelajari literatur dari beberapa bidang ilmu yang berhubungan untuk medapatkan informasi tambahan yang digunakan sebagai acuan dalam penelitian yaitu meliputi :

1. Penyakit daun jeruk antara lain *Downy Mildew*, Cendawan Jelaga dan *Citrus Vein Phloem Degeneration* (CVPD)

2. *SDLC Waterfall*
3. Pemrograman Android Java
4. Konsep Vize AI untuk proses identifikasi penyakit daun jeruk
5. Pengujian fungsional, pengujian akurasi dan pengujian usabilitas untuk proses pengujian aplikasi

3.2 Pengumpulan Data

Pada tahap ini proses pengumpulan data untuk penelitian ini berasal dari Balai Penelitian Tanaman Jeruk dan Buah Subtropika (Balitjestro) yang berada di daerah Junrejo Kota Batu. Didapat sebanyak 120 data latih yang dikumpulkan. Dengan rincian 30 gambar daun dari setiap penyakit dan ditambah 30 data daun yang sehat. Proses pengambilan data daun menggunakan kamera OPPO A37f 8 MP dan menggunakan alat bantu *background cover* warna merah dengan pencahayaan yang merata.

3.3 Analisis Kebutuhan

Pada tahap ini dilakukan proses penggalian kebutuhan dengan cara dilakukan wawancara kepada Peneliti Madya di bidang Hama dan Penyakit Tanaman Jeruk serta melakukan observasi pada Balai Penelitian Tanaman Jeruk dan Buah Subtropika (Balitjestro) yang terletak di daerah Junrejo Kota Batu yang digunakan untuk menggali semua kebutuhan dari pengembangan perangkat lunak. Kebutuhan yang nantinya didapat meliputi kebutuhan fungsional maupun kebutuhan non-fungsionalitas.

3.4 Perancangan Sistem

Tahap ini akan dilakukan setelah proses analisis kebutuhan selesai. Pada tahap ini dilakukan pemodelan sistem menggunakan diagram *Unified Modeling Language* (UML), yaitu *use case diagram*, *use case scenario*, *sequence diagram*, *class diagram* dan *activity diagram* serta dilakukan perancangan basis data, perancangan algoritme dan perancangan antarmuka.

3.5 Implementasi

Pada tahap ini merupakan tahap pembangunan program dengan memperhatikan proses perancangan sebelumnya dan tahap ini pula akan menjelaskan detail implementasi Vize AI untuk pengenalan citra dari penyakit daun jeruk dan juga implementasi perhitungan algoritme untuk mengkategorikan tingkat keparahan penyakit daun jeruk. Aplikasi ini akan diimplementasikan (ditulis) dengan menggunakan bahasa pemrograman Android Java.

3.6 Pengujian Sistem

Tahap ini dilakukan setelah tahap implementasi selesai, lalu dilakukan tahap pengujian yang bertujuan untuk mengetahui apakah aplikasi sesuai dengan

kebutuhan perangkat lunak serta digunakan untuk menemukan kesalahan pada aplikasi dan memperbaikinya. Terdapat tiga buah pengujian, pengujian fungsional untuk mengetahui apakah semua kebutuhan telah terpenuhi, juga terdapat pengujian akurasi yang digunakan untuk mengukur seberapa akurat hasil dalam mengidentifikasi penyakit daun jeruk, dan terdapat pula pengujian usabilitas untuk mengetahui tingkat kepuasan pengguna dalam menggunakan aplikasi dengan menggunakan kuesioner SUPR-Qm.

3.7 Kesimpulan dan Saran

Pada tahap ini berisikan kesimpulan yang diambil setelah semua tahapan selesai dilakukan sesuai rumusan masalah yang ditentukan sebelumnya, lalu dilanjutkan penulisan saran yang digunakan untuk memperbaiki kesalahan yang terjadi dan menyempurnakan penulisan serta memberikan pertimbangan untuk penelitian selanjutnya.

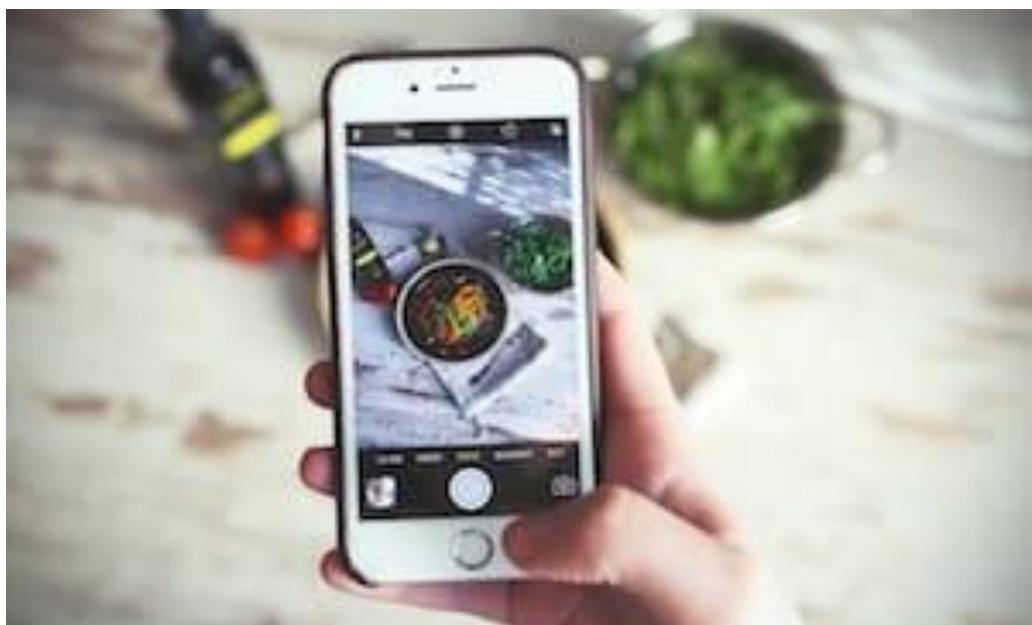
BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN

Pada bab ini berisikan penjelasan tentang segala hal yang terkait seputar proses penggalian kebutuhan sistem serta berisikan tentang hal-hal yang akan berkaitan dengan perancangan sistem berdasarkan data yang telah didapat di tahap sebelumnya.

4.1 Analisis Kebutuhan

4.1.1 Gambaran Umum Sistem

Dalam penelitian ini, penulis membangun sebuah aplikasi identifikasi penyakit daun jeruk berbasis android. Tujuan dari dikembangkannya aplikasi ini yaitu untuk membantu para petani jeruk guna mengenali jenis penyakit pada daun jeruk secara otomatis menggunakan kamera atau gambar dari galeri yang ada di *smartphone* seperti pada Gambar 4.1. Aplikasi ini juga dapat mendeteksi tingkat keparahan dari penyakit daun jeruk yang menyerangnya berupa persentase yang dapat dikelompokkan menjadi 4 tingkatan yaitu rendah, sedang, tinggi dan sangat tinggi. Untuk mendeteksi penyakit daun jeruk melalui gambar, aplikasi ini memanfaatkan sebuah *web service* yang telah menyediakan API untuk proses pengenalan gambar yang dapat dilatih untuk mengenali dan mengklasifikasikan gambar secara *custom* yaitu Vize AI. Setelah gambar telah berhasil mendeteksi penyakit pada daun jeruk dan tingkat keparahannya, aplikasi ini akan menampilkan cara pengendalian berdasarkan jenis penyakit yang menyerangnya. Dalam aplikasi ini terdapat pula fitur Riwayat yaitu berisi daftar riwayat dari gambar-gambar daun yang telah dideteksi sebelumnya.



Gambar 4.1 Gambaran Penggunaan Sistem

Sumber : (Vize, 2018)

4.1.2 Analisis Kebutuhan Perangkat Lunak

4.1.2.1 Identifikasi Aktor

Aktor adalah seseorang ataupun mesin yang berinteraksi langsung dengan sistem untuk melakukan suatu pekerjaan tertentu. Adapun aktor yang berhubungan dengan sistem ini ditunjukkan pada Tabel 4.1.

Tabel 4.1 Aktor Sistem

Aktor	Deskripsi
Pengguna	Merupakan orang yang berinteraksi langsung dengan aplikasi dan ingin mendeteksi apakah daun pada tanaman jeruknya terserang penyakit atau tidak. Pengguna aplikasi dapat seorang petani ataupun peneliti tanaman jeruk.

4.1.2.2 Kebutuhan Fungsional Sistem

Kebutuhan fungsional sistem merupakan suatu kebutuhan yang harus disediakan oleh sistem, termasuk bagaimana suatu sistem harus merespon terhadap input dan situasi tertentu serta harus dapat menyelesaikan permasalahan yang telah dirumuskan sebelumnya (Sommerville, 2011).

Kebutuhan fungsional pula harus memiliki suatu kode untuk memudahkan dalam pengidentifikasiannya guna menjaga konsistensi kebutuhan sampai proses pengujian selesai dilakukan. Pengkodean pada tiap kebutuhan tersebut yakni SRS-LEAD-XXX. SRS merupakan singkatan dari *System Requirement Specification*, LEAD merupakan singkatan dari nama aplikasi ini yaitu Leaf Disease Detector, dan XXX merupakan nomor dari kebutuhannya. Pada sistem ini terdapat 6 buah kebutuhan fungsional sistem yang dijelaskan pada Tabel 4.2.

Tabel 4.2 Kebutuhan Fungsional Sistem

No	Kode Kebutuhan Sistem	Deskripsi Kebutuhan
1	SRS-LEAD-001	Sistem harus mampu menyediakan mekanisme untuk mengambil gambar menggunakan kamera dan galeri.
2	SRS-LEAD-002	Sistem harus mampu menyediakan fungsi untuk memotong gambar.
3	SRS-LEAD-003	Sistem harus mampu menyediakan fungsi untuk mendeteksi penyakit dan menentukan tingkat keparahan dari penyakit yang diderita.
4	SRS-LEAD -004	Sistem harus mampu memberikan informasi tentang cara pengendalian berdasarkan penyakit daun jeruk yang menyerangnya.

5	SRS-LEAD-005	Sistem harus mampu menyediakan fitur riwayat dari gambar-gambar daun yang telah dideteksi sebelumnya.
6	SRS-LEAD-006	Sistem harus mampu menyediakan informasi tentang penyakit daun jeruk.

4.1.2.3 Kebutuhan Non-Fungsionalitas

Kebutuhan non-fungsionalitas merupakan suatu kebutuhan yang berfokus pada perilaku yang dimiliki oleh sistem. Kebutuhan non-fungsionalitas juga sering didefinisikan sebagai suatu fungsi yang ditawarkan atau suatu batasan layanan dari suatu sistem (Sommerville, 2011). Pada sistem ini terdapat satu kebutuhan non-fungsionalitas yaitu *Usability* yang merupakan kemudahan dalam penggunaan sistem.

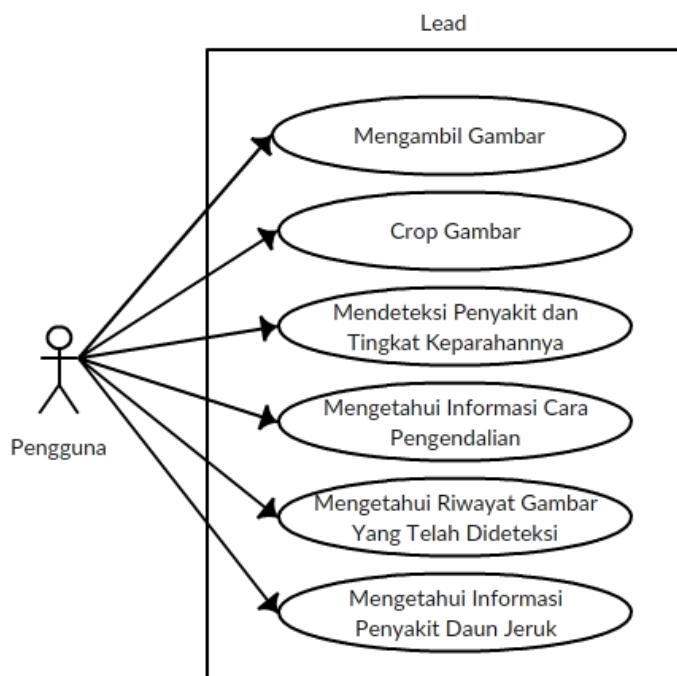
4.2 Perancangan

4.2.1 Diagram Unified Modeling Language (UML)

Untuk mempermudah pemahaman pada perancangan sistem, diperlukan pemodelan menggunakan diagram UML yang terdiri dari *use case diagram*, *use case scenario*, *sequence diagram*, *class diagram* dan *activity diagram*.

4.2.1.1 Use Case Diagram

Use case diagram terdiri dari sejumlah aksi yang dilakukan pengguna terhadap sistem untuk mencapai tujuan tertentu. *Use case diagram* pada sistem ini dijelaskan pada Gambar 4.2.



Gambar 4.2 Use Case Diagram

Pada Gambar 4.2 dijelaskan bahwa sistem ini terdapat 6 buah aksi yang dapat dilakukan pengguna terhadap sistem untuk mencapai tujuan tertentu. Enam buah aksi tersebut antara lain mengambil gambar, *crop* gambar, mendeteksi penyakit dan tingkat keparahannya, mengetahui informasi cara pengendalian, mengetahui riwayat gambar yang telah dideteksi dan mengetahui informasi penyakit daun jeruk.

4.2.1.2 Use Case Scenario

Setelah menggambarkan *use case diagram*, maka langkah selanjutnya yaitu menjabarkan secara detail dari tiap-tiap *use case* dengan membuat *use case scenario*. Pada *use case scenario* ini terdapat beberapa bagian, antara lain nama *use case*, kode kebutuhan, aktor, tujuan, *pre condition* (kondisi yang harus terpenuhi sebelum *main flow* dijalankan), *main flow* (alur pelaksanaan *use case* hingga mencapai tujuan yang diharapkan), *alternative flow* (alur pelaksanaan ketika dalam pelaksanaan *main flow* terjadi skenario yang tidak dideklarasikan) dan *post condition* (kondisi akhir setelah *main flow* selesai dijalankan). *Use case scenario* pada sistem ini dijelaskan pada Tabel 4.3 sampai Tabel 4.8.

Tabel 4.3 Use Case Scenario Mengambil Gambar

Nama <i>Use Case</i>	Mengambil Gambar
Kode Kebutuhan	SRS-LEAD-001
Aktor	Pengguna
Tujuan	Pengguna dapat mengambil gambar daun menggunakan kamera atau galeri
<i>Pre Condition</i>	Pengguna sudah berada pada halaman deteksi
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih untuk mengambil gambar melalui kamera atau galeri 2. Sistem akan memproses pengambilan gambar sesuai apa yang dipilih pengguna
<i>Post Condition</i>	Gambar berhasil diambil dan siap untuk dideteksi
<i>Alternative Flow</i>	-

Tabel 4.4 Use Case Scenario Crop Gambar

Nama <i>Use Case</i>	<i>Crop</i> Gambar
Kode Kebutuhan	SRS-LEAD-002
Aktor	Pengguna
Tujuan	Pengguna mendapatkan gambar yang ia inginkan

<i>Pre Condition</i>	Pengguna sudah memilih gambar daun dari kamera atau galeri
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna mengatur gambar yang ingin ia potong 2. Sistem melakukan <i>cropping</i> dan menampilkan hasilnya ke layar
<i>Post Condition</i>	Pengguna mendapatkan gambar yang ia inginkan dan siap untuk dideteksi
<i>Alternative Flow</i>	-

Tabel 4.5 Use Case Scenario Mendeteksi Penyakit dan Tingkat Keparahannya

Nama Use Case	Mendeteksi Penyakit dan Tingkat Keparahannya
Kode Kebutuhan	SRS-LEAD-003
Aktor	Pengguna
Tujuan	Pengguna memperoleh informasi tentang penyakit yang menyerang daunnya dan tingkat keparahannya
<i>Pre Condition</i>	Pengguna telah memilih gambar daun yang akan dideteksi
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol centang untuk mengirim gambar 2. Sistem melakukan pendekripsi penyakit beserta menghitung tingkat keparahan penyakitnya dan menampilkan informasi tersebut ke dalam layar
<i>Post Condition</i>	Pengguna mengetahui jenis penyakit yang menyerang daunnya dan persentase tingkat keparahan penyakit daunnya dan mengetahui 4 gambar yaitu gambar <i>input</i> , gambar daun (<i>tanpa background</i>), gambar daun yang sehat, dan gambar daun berpenyakit.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika <i>smartphone</i> pengguna tidak terkoneksi internet, maka sistem akan menampilkan pemberitahuan bahwa <i>smartphone</i> tidak terkoneksi internet. 2. Jika sistem gagal mengirim gambar maka sistem akan menginstruksikan pengguna untuk melakukan pengiriman ulang gambar.

Tabel 4.6 Use Case Scenario Mengetahui Informasi Cara Pengendalian

Nama Use Case	Mengetahui Informasi Cara Pengendalian
Kode Kebutuhan	SRS-LEAD-004

Aktor	Pengguna
Tujuan	Pengguna mengetahui informasi cara pengendalian berdasarkan penyakit daun dan tingkat keparahannya
<i>Pre Condition</i>	Sistem telah mengetahui jenis penyakit daun dan tingkat keparahannya
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “Cara Pengendalian” untuk menampilkan cara pengendaliannya 2. Sistem menampilkan informasi cara pengendaliannya berdasarkan penyakit dan tingkat keparahan
<i>Post Condition</i>	Pengguna mengetahui informasi cara pengendalian berdasarkan penyakit daun dan tingkat keparahannya
<i>Alternative Flow</i>	-

Tabel 4.7 Use Case Scenario Mengetahui Riwayat Gambar Yang Telah Dideteksi

Nama Use Case	Mengetahui Riwayat Gambar Yang Telah Dideteksi
Kode Kebutuhan	SRS-LEAD-005
Aktor	Pengguna
Tujuan	Pengguna mengetahui daftar riwayat gambar yang telah dideteksi sebelumnya
<i>Pre Condition</i>	Pengguna sudah berada pada halaman utama aplikasi
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “Riwayat” untuk menampilkan riwayat gambar yang telah dideteksi 2. Sistem menampilkan daftar riwayat
<i>Post Condition</i>	Pengguna mengetahui daftar riwayat gambar yang telah dideteksi sebelumnya
<i>Alternative Flow</i>	Jika riwayat gambar tidak tersedia, maka sistem akan memberikan pemberitahuan bahwa riwayat masih kosong

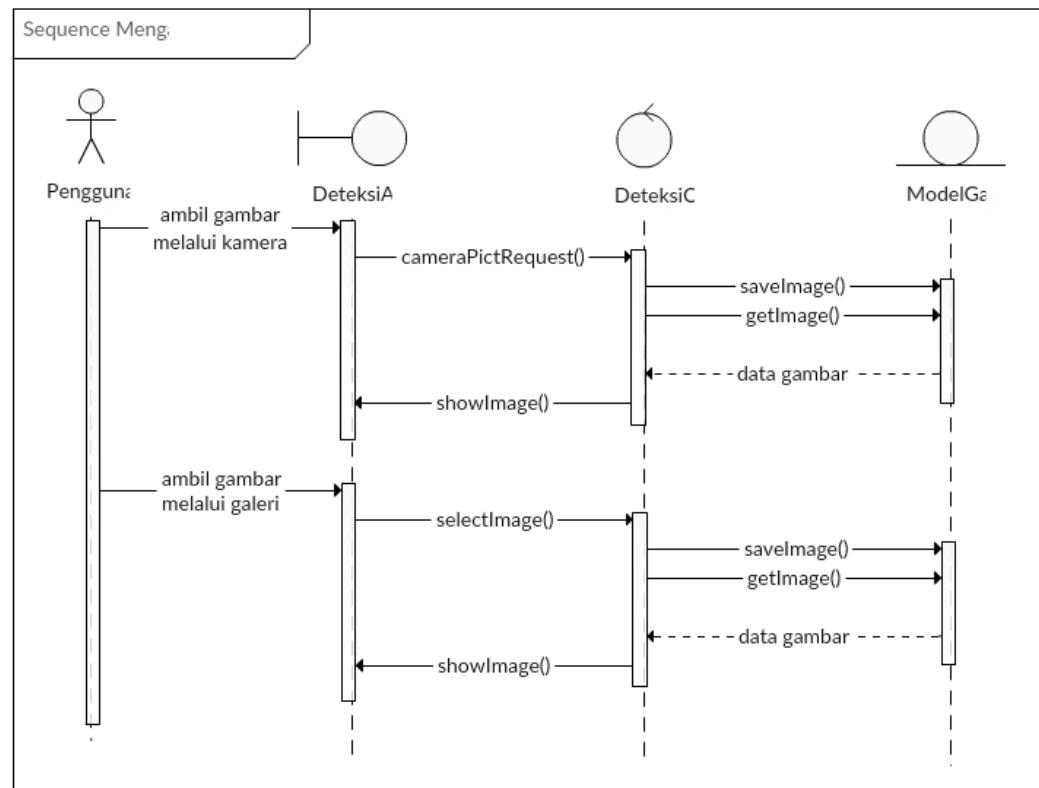
Tabel 4.8 Use Case Scenario Mengetahui Informasi Penyakit Daun Jeruk

Nama Use Case	Mengetahui Informasi Penyakit Daun Jeruk
Kode Kebutuhan	SRS-LEAD-006
Aktor	Pengguna
Tujuan	Pengguna dapat mengetahui informasi tentang jenis-jenis penyakit pada daun jeruk

<i>Pre Condition</i>	Pengguna sudah berada pada halaman utama aplikasi
<i>Main Flow</i>	1. Pengguna menekan tombol “Daftar Penyakit” 2. Sistem menampilkan daftar penyakit pada daun jeruk
<i>Post Condition</i>	Pengguna dapat mengetahui informasi tentang jenis-jenis penyakit pada daun jeruk
<i>Alternative Flow</i>	-

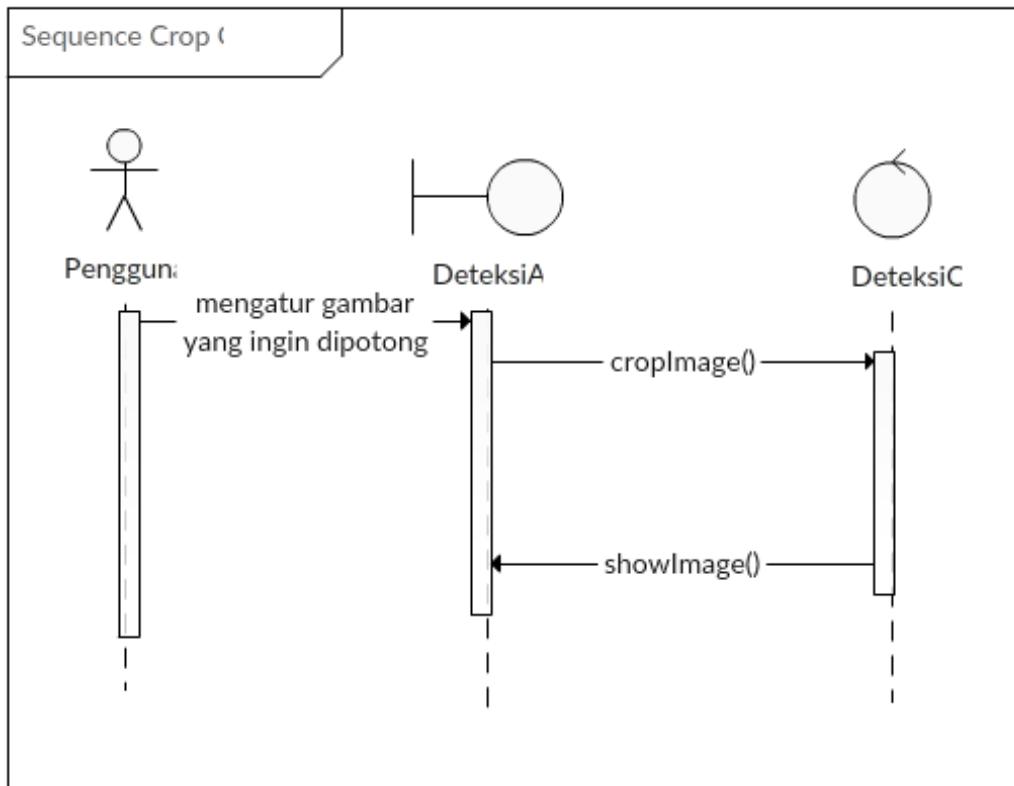
4.2.1.3 Sequence Diagram

Setelah membuat *use case scenario*, dilakukan pembuatan *sequence diagram*. *Sequence diagram* merupakan penjelasan mengenai urutan proses untuk mencapai tujuan tertentu yang ada pada kebutuhan dan menjelaskan mengenai urutan proses yang terjadi dalam mencapai kebutuhan sistem. Hasil dari identifikasi kebutuhan dan *use case scenario* dipresentasikan kedalam objek-objek yang ada pada *sequence diagram*. Untuk satu *sequence diagram* menggambarkan tiap satu *use case*. Pada sistem ini terdapat 6 buah *sequence diagram* antara lain *sequence mengambil gambar*, *sequence crop gambar*, *sequence mendeteksi penyakit* dan tingkat keparahannya, *sequence mengetahui informasi cara penanggulangan*, *sequence mengetahui riwayat gambar* yang telah dideteksi, dan *sequence mengetahui informasi daun jeruk* yang dijelaskan pada Gambar 4.3 sampai 4.8.



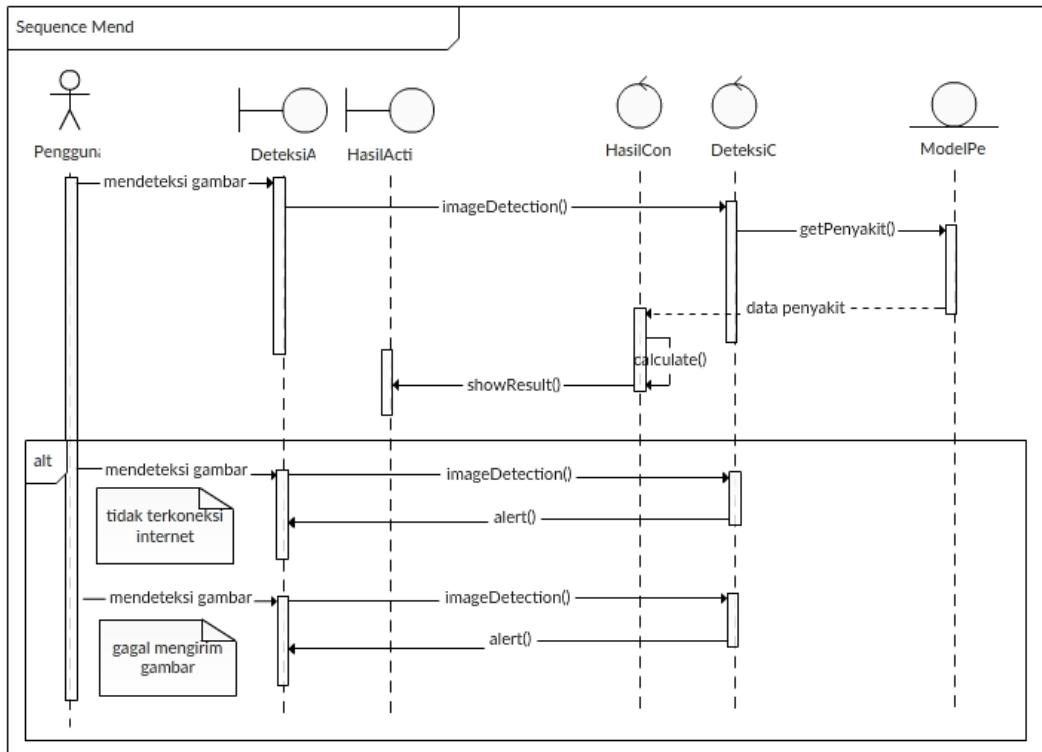
Gambar 4.3 *Sequence Diagram* Mengambil Gambar

Pada Gambar 4.3 merupakan *sequence diagram* mengambil gambar, pengguna mengambil gambar melalui kamera atau galeri. Jika pengguna mengambil gambar melalui kamera, maka sistem akan memanggil *method* cameraPictRequest() yang ada pada DeteksiController. Lalu gambar pun disimpan dengan memanggil *method* saveImage() dan sistem akan meminta kembali gambar tersebut dengan memanggil *method* getImage() yang ada pada ModelGambar dan akan mengembalikan data berupa gambarnya. Terakhir sistem memanggil *method* showImage() untuk menampilkan gambar yang diambil ke layar. Sedangkan jika pengguna mengambil gambar melalui galeri, maka sistem akan memanggil *method* selectImage() yang ada pada DeteksiController. Lalu gambar pun disimpan dengan memanggil *method* saveImage() dan sistem akan meminta kembali gambar tersebut dengan memanggil *method* getImage() yang ada pada ModelGambar dan akan mengembalikan data berupa gambarnya. Terakhir sistem memanggil *method* showImage() untuk menampilkan gambar yang diambil ke layar.



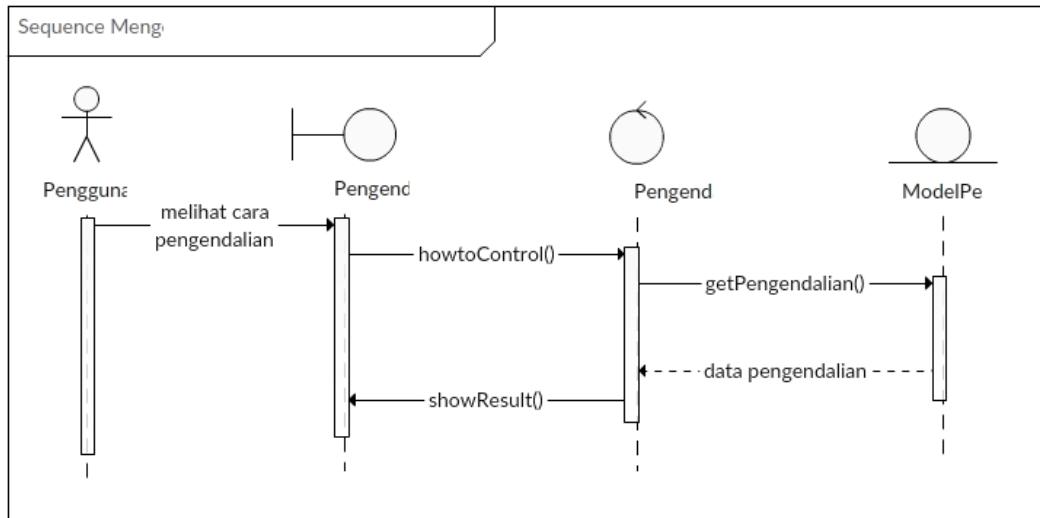
Gambar 4.4 Sequence Diagram Crop Gambar

Pada Gambar 4.4 merupakan *sequence diagram* *crop* gambar, pengguna mengatur gambar yang ingin dipotong. Setelah pengguna selesai mengatur gambar yang ingin dipotong lalu sistem akan memanggil *method* cropImage() yang ada pada DeteksiController untuk melakukan proses pemotongan gambar dan terakhir sistem akan memanggil *method* showImage() untuk menampilkan gambar yang berhasil dipotong dan gambar siap untuk dideteksi.



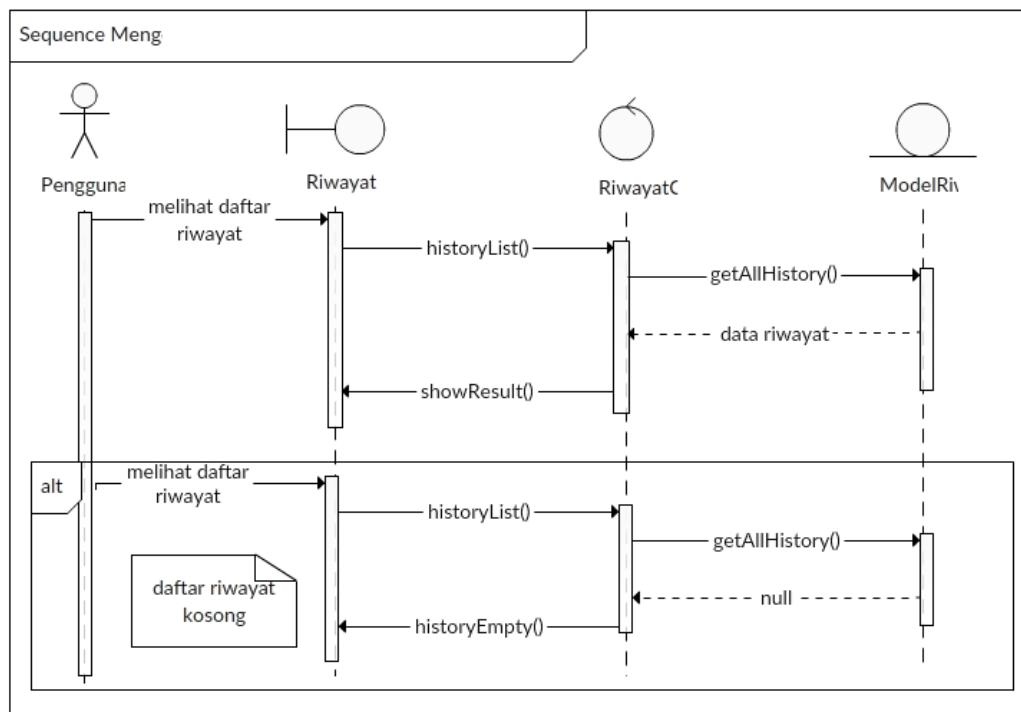
Gambar 4.5 Sequence Diagram Mendeteksi Penyakit dan Tingkat Keparahannya

Pada Gambar 4.5 merupakan *sequence diagram* mendeteksi penyakit dan tingkat keparahannya, pengguna mengambil gambar yang telah dipilih sebelumnya. Sistem akan memanggil *method* `ImageDetection()` yang ada pada *DeteksiController* untuk melakukan pendekripsi penyakit. Gambar akan dikirimkan ke *web service* Vize AI untuk mengetahui jenis penyakit yang menyerang daun tersebut. Setelah didapatkan jenis penyakit daunnya maka sistem akan mengambil data penyakit melalui *method* `getPenyakit()` pada *ModelPenyakit* dan sistem akan mengembalikan data penyakitnya. Setelah diketahui jenis penyakit daunnya maka sistem akan menghitung persentase tingkat keparahan penyakit pada daunnya dengan menggunakan *method* `calculate()` yang ada pada *HasilController*. Terakhir setelah jenis penyakit dan persentase tingkat keparahannya maka sistem akan menampilkan hasilnya ke layar melalui *method* `showResult()`. Pada *sequence diagram* ini pula terdapat 2 *alternative scenario* yaitu kondisi jika *smartphone* tidak terkoneksi internet dan kondisi jika *smartphone* gagal mengirimkan gambar. Pada *alternative scenario* dengan kondisi *smartphone* tidak terkoneksi dengan internet, sistem akan melakukan pengecekan apakah *smartphone* terkoneksi dengan internet atau tidak, jika tidak sistem akan berhenti melakukan proses pada *method* `ImageDetection()` dan memanggil *method* `alert()` untuk menampilkan pemberitahuan pengguna bahwa *smartphone* tidak terkoneksi internet. Sedangkan pada *alternative scenario* dengan kondisi *smartphone* gagal mengirimkan gambar, maka sistem akan berhenti melakukan pendekripsi gambar dan menampilkan pemberitahuan untuk mendekripsi ulang gambarnya.



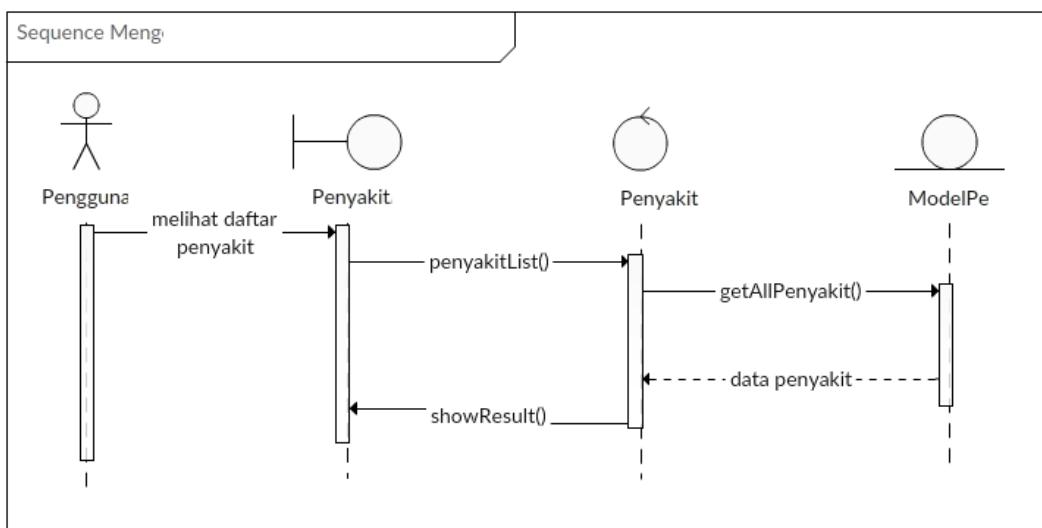
Gambar 4.6 Sequence Diagram Mengetahui Informasi Cara Pengendalian

Pada Gambar 4.6 merupakan *sequence diagram* mengetahui informasi cara pengendalian, pengguna ingin mengetahui cara pengendalian berdasarkan penyakitnya. Kemudian sistem memanggil *method* *howtoControl()* pada PengendalianController untuk mengetahui cara pengendalian penyakit berdasarkan penyakit dan tingkat keparahan. Lalu sistem memanggil *method* *getPengendalian()* untuk mendapatkan informasi cara pengendalian. Setelah didapatkan informasi cara pengendaliannya maka selanjutnya sistem menampilkan informasi tersebut melalui *method* *showResult()*.



Gambar 4.7 Sequence Diagram Mengetahui Riwayat Gambar Yang Telah Dideteksi

Pada Gambar 4.7 merupakan *sequence diagram* mengetahui riwayat gambar yang telah dideteksi, dimana pengguna ingin mengetahui daftar riwayat yang telah dideteksi sebelumnya. Kemudian sistem memanggil *method* `historyList()` yang ada pada `RiwayatController` untuk mengetahui daftar riwayat pendekripsi. Lalu sistem memanggil *method* `getAllHistory()` untuk mendapatkan daftar riwayat. Setelah didapatkan daftar riwayatnya maka selanjutnya sistem menampilkan daftar riwayat tersebut ke layar melalui *method* `showResult()`. Pada sequence ini terdapat satu *alternative scenario* yaitu jika daftar riwayat kosong. Kondisi tersebut terjadi jika pada saat sistem mengambil data riwayat pendekripsi, ModelPenyakit mengembalikan nilai `null` lalu sistem akan memanggil *method* `historyEmpty()` yang akan memberitahukan pengguna bahwa daftar riwayat kosong.

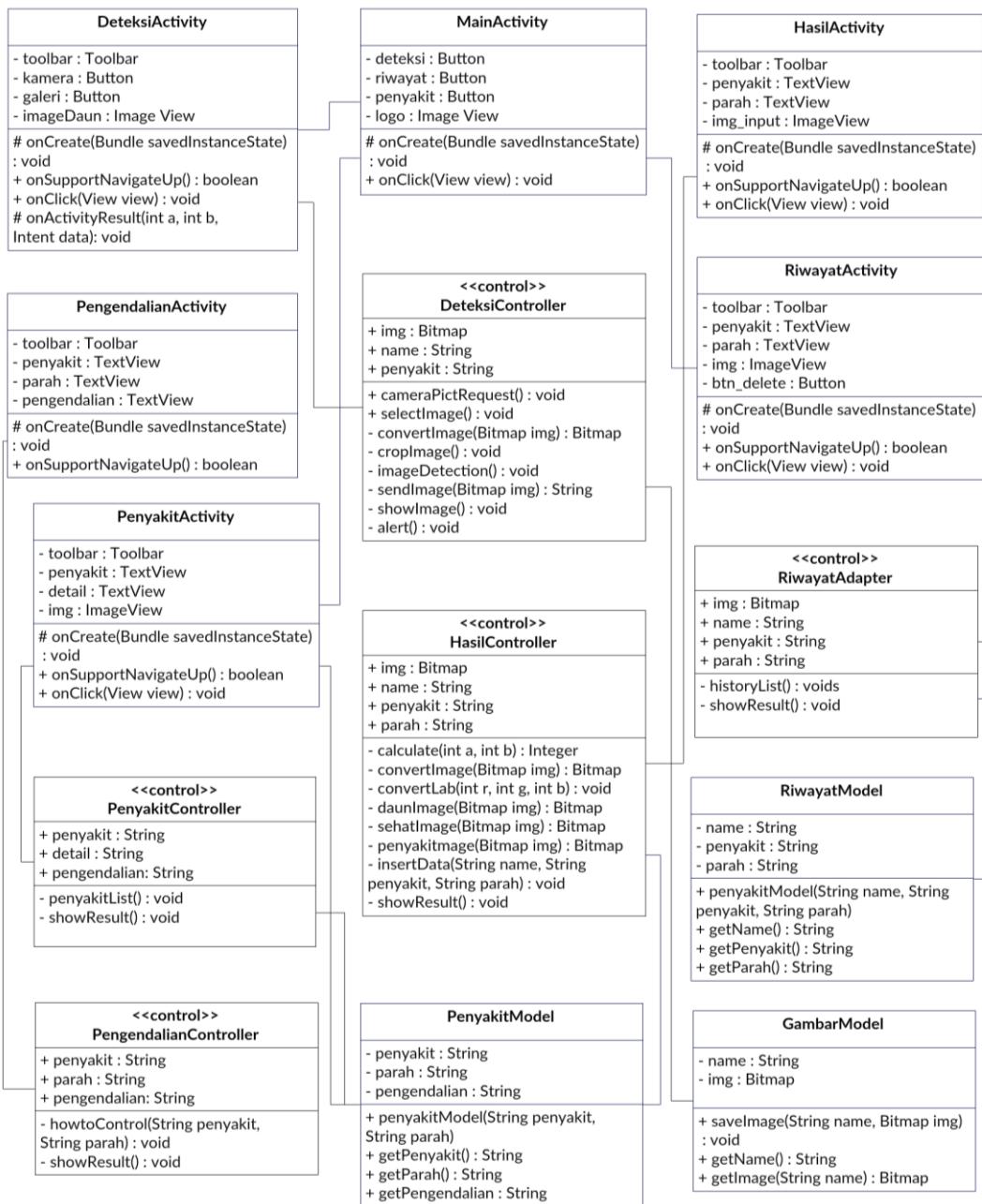


Gambar 4.8 Sequence Diagram Mengetahui Informasi Penyakit Daun Jeruk

Pada Gambar 4.8 merupakan *sequence diagram* mengetahui informasi penyakit daun jeruk, dimana pengguna ingin mengetahui informasi penyakit daun jeruk. Kemudian sistem memanggil *method* `penyakitList()` yang ada pada `PenyakitController` untuk mengetahui daftar penyakit daun jeruk. Lalu sistem memanggil *method* `getAllPenyakit()` untuk mendapatkan daftar penyakit dan sistem mengembalikan berupa data penyakit. Setelah didapatkan daftar penyakit maka selanjutnya sistem menampilkan daftar tersebut melalui *method* `showResult()`.

4.2.1.4 Class Diagram

Setelah membuat *sequence diagram* selanjutnya dilakukan pembuatan *class diagram*. *Class diagram* merupakan perancangan dari kelas-kelas yang dapat menggambarkan struktur sistem. Kelas-kelas tersebut saling berelasi yang digambarkan dalam bentuk tabel-tabel yang berisikan atribut dan operasi yang dapat dilakukan dari sebuah kelas. *Class diagram* pada sistem ini dijelaskan pada Gambar 4.8.



Gambar 4.9 Class Diagram

Pada Gambar 4.9 merupakan *class diagram* pada sistem ini. Pada *class diagram* tersebut terdiri dari beberapa bagian seperti *Activity*, *Controller* dan *Model*.

- **Activity**

`MainActivity`, *class* yang merupakan tampilan utama yang berisikan 3 buah *Button* yang nantinya terhubung ke *activity* selanjutnya.

`DeteksiActivity`, *class* yang dimana pengguna memilih untuk mengambil gambar melalui kamera atau galeri yang berisikan 2 buah *Button* dan 1 buah *ImageView* untuk menampilkan gambar yang dipilih.

HasilActivity, *class* yang dimana pengguna dapat melihat hasil pendektsian yang di dalamnya terdapat 2 buah *TextView* yang berisi hasil pendektsian penyakit dan tingkat keparahan.

PengendalianActivity, *class* yang dimana pengguna dapat melihat cara pengendalian berdasarkan penyakit dan tingkat keparahan yang berisikan 3 buah *TextView* yaitu penyakit, tingkat keparahan dan cara pengendalian.

RiwayatActivity, *class* yang dimana pengguna dapat melihat daftar riwayat yang telah dideteksi sebelumnya. Pada *class* ini berisikan 1 buah *ImageView* dan 2 *TextView* yaitu jenis penyakit dan tingkat keparahan serta terdapat 1 *Button* sebagai tombol untuk menghapus riwayat.

PenyakitActivity, *class* yang dimana pengguna dapat melihat daftar penyakit daun jeruk. Pada *class* ini berisikan 1 buah *ImageView* dan 2 *TextView* yaitu jenis penyakit dan detail informasi dari penyakitnya.

- *Controller dan Adapter*

DeteksiController, *class* yang berisikan fungsi-fungsi dimana dapat melakukan pengambilan gambar melalui kamera atau galeri dan disimpan ke dalam GambarModel, lalu mengambil gambar dan menampilkannya, meng-crop gambar, melakukan *convert* pada ukuran gambar dan melakukan pendektsian gambar.

HasilController, *class* yang berisikan fungsi-fungsi dimana dapat mengambil data dari PenyakitModel berupa jenis penyakitnya lalu melakukan perhitungan persentase tingkat keparahan penyakit dan mengirimkan hasilnya ke HasilActivity.

PengendalianController, *class* yang dimana dapat mengambil data cara pengendalian dari PengendalianModel berdasarkan jenis penyakit dan tingkat keparahannya lalu menampungnya dan didapatkan hasil berupa jenis penyakit, tingkat keparahan dan cara pengendalian.

RiwayatAdapter, *class* yang dimana dapat menampung seluruh data riwayat pendektsian penyakit dan meng-handle riwayat tersebut yang berisikan list dari RiwayatModel.

PenyakitController, *class* yang dimana dapat mengambil data jenis penyakit, tingkat keparahan dan cara pengendalian dari PenyakitModel lalu menampungnya dan didapatkan hasil berupa list penyakit dan detail dari penyakit tersebut.

- *Model*

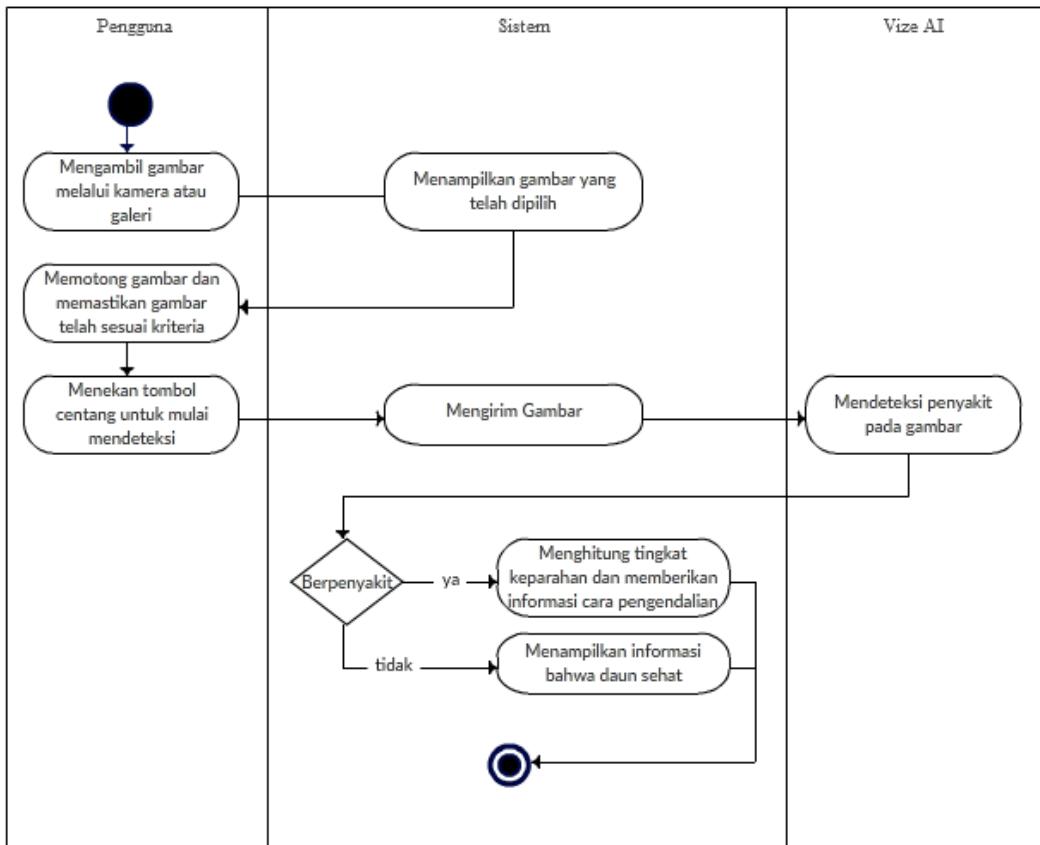
GambarModel, *class* yang menampung dapat data yang berisikan name *String* dan img *String*.

PenyakitModel, *class* yang menampung dapat data yang berisikan penyakit *String*, parah *String*, pengendalian *String*.

RiwayatModel, *class* yang menampung dapat data yang berisikan name *String*, penyakit *String* dan parah *String*.

4.2.1.5 Activity Diagram

Activity diagram menjelaskan aktifitas yang terjadi antara pengguna, sistem dan Vize AI untuk menjalankan sistem ini. *Activity diagram* pada sistem ini dijelaskan pada Gambar 4.10.

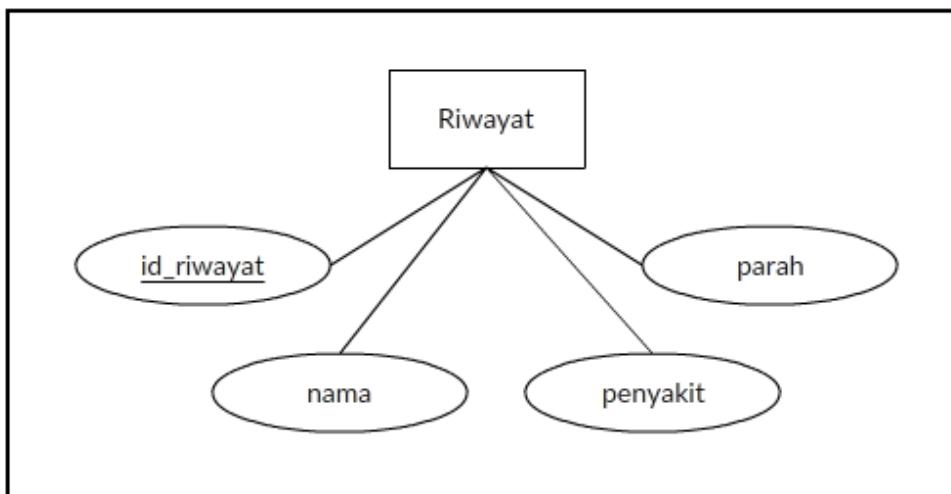


Gambar 4.10 *Activity Diagram*

Pada Gambar 4.10 dijelaskan aktifitas antara pengguna, sistem dan Vize AI di dalam sistem ini yang terdapat 8 aktifitas atau *state* dan 1 seleksi kondisi. Dimulai dari *initial state* lalu menjalankan aktifitas yang dilakukan pengguna yaitu mengambil gambar melalui kamera atau galeri lalu menuju aktifitas yang ada pada sistem dengan menampilkan gambar yang telah dipilih. Kemudian pengguna melakukan aktifitas memotong gambar dan memastikan gambar telah sesuai kriteria dan pengguna menekan tombol centang untuk memulai pendekripsi. Kemudian dilanjutkan aktifitas oleh sistem dengan mengirimkan gambar ke *web service* Vize AI. Vize AI melakukan aktifitas pendekripsi penyakit pada gambar dan mengembalikan data penyakit ke sistem untuk diproses. Jika gambar daun terdeteksi berpenyakit maka sistem akan melakukan aktifitas untuk menghitung tingkat keparahan dan memberikan informasi cara pengendalian. Sedangkan jika gambar daun tidak terdeteksi berpenyakit maka sistem akan melakukan aktifitas menampilkan infomasi bahwa daun sehat. Dan terakhir terdapat *final state* yang merupakan akhir dari aktifitas.

4.2.2 Perancangan Basis Data

Setelah dilakukan pemodelan menggunakan diagram UML, maka selanjutnya yaitu merancang basis data yang digambarkan dalam diagram ERD. Perancangan basis data diperlukan untuk mengontrol data yang keluar dan masuk dari sistem agar lebih terstruktur. merupakan tempat penyimpanan data yang dibutuhkan oleh sistem. Perancangan basis data dijelaskan pada Gambar 4.3.



Gambar 4.11 ERD Aplikasi LEAD

Pada Gambar 4.11 dijelaskan *Entity Relationship Diagram* (ERD) dari sistem ini yang terdapat 1 buah tabel yaitu tabel riwayat yang berguna untuk menyimpan riwayat dari pendekripsi daun jeruk. Tabel riwayat berisi 4 buah atribut yaitu id_riwayat yang merupakan *primary key* dari tabel, nama merupakan nama file gambarnya, penyakit merupakan nama penyakit yang menyerangnya, dan parah merupakan persentase tingkat keparahannya.

4.2.3 Perancangan Algoritme

Setelah dilakukan perancangan basis data, maka selanjutnya yaitu merancang algoritme yang ditulis dalam bentuk *Pseudocode*. Perancangan algoritme dilakukan untuk mempermudah dalam menyelesaikan masalah secara efisien. Pada penelitian ini terdapat 2 perancangan algoritme yaitu perancangan algorima mendekripsi penyakit dan perancangan algoritme menghitung tingkat keparahan penyakit yang dijelaskan pada Tabel 4.9 dan Tabel 4.10

Tabel 4.9 Pseudocode Algoritme Mendekripsi Penyakit

1	function imageDetection()
2	img -> Bitmap
3	penyakit -> String
4	captureImage()
5	resizeImage(widht, height, img)
6	if(img != null)
7	if(isNetworkAvailable())
8	penyakit -> sendImage(img)
9	else
10	alert("not connected")

```

11         endif
12     endif
13 end function
14
15 function String sendImage(Bitmap img -> Bitmap)
16     penyakit -> String
17     if(onResponse)
18         penyakit -> best_label_name
19     else
20         penyakit -> null
21         alert("reset detection")
22     endif
23     return penyakit
24 end function

```

Pada Tabel 4.9 merupakan *pseudocode* algoritme untuk mendeteksi penyakit pada daun jeruk. Sebelum melakukan pendektsian dilakukan pengambilan gambar melalui kamera atau galeri, lalu dilakukan perubahan ukuran gambar. Lalu setelah semua proses selesai, dilakukan pengiriman gambar daun ke *web service* dan langsung mendapatkan hasil berupa penyakit daunnya.

Tabel 4.10 Pseudocode Algoritme Menghitung Tingkat Keparahan

```

1 function calculate()
2     img_input, img_daun, img_sehat, img_penyakit -> Bitmap
3     penyakit -> String
4     persentage, count_penyakit, count_daun -> int
5     resizeImage(widht, height, img_input)
6     img_daun -> daunImage(img_input)
7     img_sehat -> sehatImage(img_daun)
8     img_penyakit -> penyakitImage(img_daun)
9     persentage -> (count_penyakit * 100) / count_daun
10 end function
11
12 function Bitmap daunImage(Bitmap img -> Bitmap)
13     count_daun -> int
14     R, G, B, l, a, b -> int
15     convertLab()
16     loop (i -> 0, i < img.widht, i++)
17         loop (j -> 0, j < img.hight, j++)
18             if(l, a, b -> warna_merah)
19                 R -> 255
20                 G -> 255
21                 B -> 255
22             else
23                 count_daun++
24             endif
25         endloop
26     endloop
27 end function
28
29 function Bitmap sehatImage(Bitmap img -> Bitmap)
30     penyakit -> String
31     count_penyakit -> int
32     R, G, B, l, a, b -> int
33     convertLab()
34     loop (i -> 0, i < img.widht, i++)
35         loop (j -> 0, j < img.hight, j++)
36             if(l, a, b -> warna_penyakit)

```

```

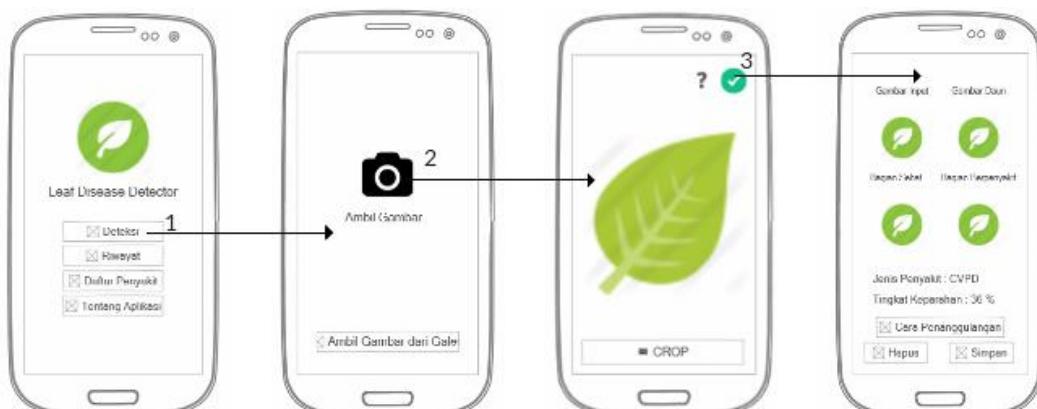
37                     R -> 255
38                     G -> 255
39                     B -> 255
40             count_penyakit++
41         endif
42     endloop
43 endloop
44 end function
45
46 function Bitmap penyakitImage(img -> Bitmap)
47     R, G, B, l, a, b -> int
48     convertLab()
49     loop (i -> 0, i < img.widht, i++)
50         loop (j -> 0, j < img.hight, j++)
51             if(l, a, b != warna_penyakit)
52                 R -> 255
53                 G -> 255
54                 B -> 255
55             endif
56         endloop
57     endloop
58 end function

```

Pada tabel 4.10 merupakan *pseudocode* algoritme untuk menghitung tingkat keparahan penyakit pada daun jeruk. Pada algoritme diatas dilakukan iterasi sejumlah lebar dan tinggi *pixel* dari gambar untuk mengubah gambar menjadi 3 gambar yaitu gambar daun saja, gambar daun sehat, dan gambar daun berpenyakit. Lalu didalam algoritme tersebut juga terdapat penambahan variabel jumlah *pixel* daun jika pada gambar terdeteksi bagian daun dan penambahan variabel jumlah *pixel* penyakit jika gambar terdeteksi bagian berpenyakit.

4.2.4 Perancangan Antarmuka

Pada perancangan antarmuka diperlukan pembuatan *Screenflow* atau diagram alir *screen* dari tiap aktifitas dalam memudahkan proses implementasi agar dapat dengan jelas mengetahui setiap perpindahan halaman. Pada perancangan antarmuka aplikasi ini dibuat menggunakan *tools* online yaitu “NinjaMock”. Pada penelitian ini terdapat 3 *screenflow* yaitu *screenflow* mendeteksi penyakit, *screenflow* melihat daftar riwayat, dan *screenflow* melihat daftar penyakit yang dijelaskan pada Gambar 4.12 sampai Gambar 4.14.



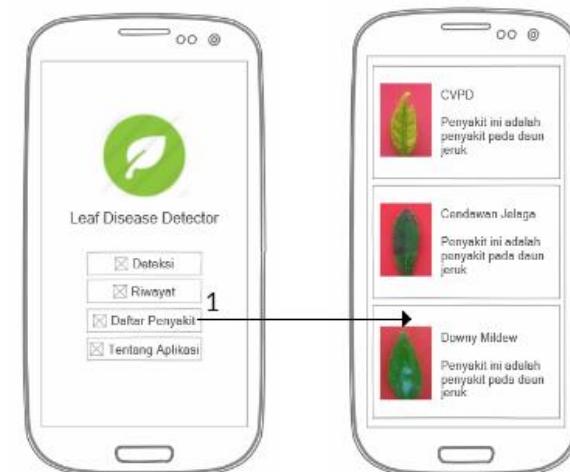
Gambar 4.12 Screenflow Mendeteksi Penyakit

Pada Gambar 4.12 merupakan *screenflow* mendeteksi penyakit yang menjelaskan perpindahan antara halaman awal sampai halaman hasil dari pendeksteksian. Perpindahan pertama yaitu perpindahan dari halaman awal ke halaman mengambil gambar, terjadi ketika kita menekan tombol “Deteksi”. Perpindahan kedua yaitu perpindahan dari halaman mengambil gambar ke halaman hasil gambar, terjadi ketika kita selesai mengambil gambar melalui kamera atau galeri dan gambar langsung di tampilkan pada halaman hasil gambar. Perpindahan ketiga yaitu perpindahan dari hasil gambar ke halaman hasil pendeksteksian, terjadi ketika kita menekan icon cek pada ujung kiri atas halaman hasil gambar.



Gambar 4.13 Screenflow Melihat Daftar Riwayat

Pada Gambar 4.13 merupakan *screenflow* melihat daftar riwayat yang menjelaskan perpindahan antara halaman awal kea halaman daftar riwayat, terjadi ketika kita menekan tombol “Riwayat” pada halaman awal.



Gambar 4.14 Sreenflow Melihat Daftar Penyakit

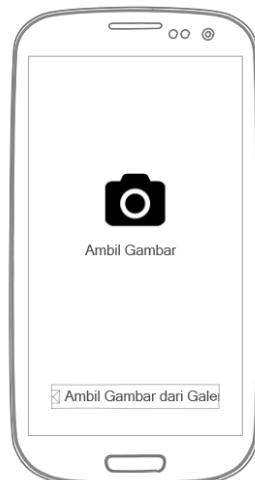
Pada Gambar 4.14 merupakan *screenflow* melihat daftar penyakit yang menjelaskan perpindahan antara halaman awal ke halaman daftar penyakit, terjadi ketika kita menekan tombol “Daftar Penyakit” pada halaman awal.

Perancangan antarmuka dilakukan untuk mempermudah dalam mendapatkan gambaran desain pada proses implementasi dan perancangan antarmuka juga harus disesuaikan dengan kebutuhan yang telah didefinisikan sebelumnya. Pada penelitian ini terdapat 6 perancangan antarmuka yaitu antarmuka halaman awal, antarmuka halaman mengambil gambar, antarmuka detail gambar, antarmuka halaman hasil deteksi, antarmuka halaman daftar riwayat dan antarmuka halaman daftar penyakit yang dijelaskan pada Gambar 4.15 sampai Gambar 4.20.



Gambar 4.15 Antarmuka Halaman Awal

Pada Gambar 4.15 merupakan perancangan antarmuka halaman awal aplikasi yang terdapat 4 buah tombol di dalamannya yaitu deteksi (berisi proses pendektesian penyakit daun jeruk), riwayat (berisi daftar riwayat), daftar penyakit (berisi daftar penyakit daun jeruk) dan deskripsi tentang aplikasi.



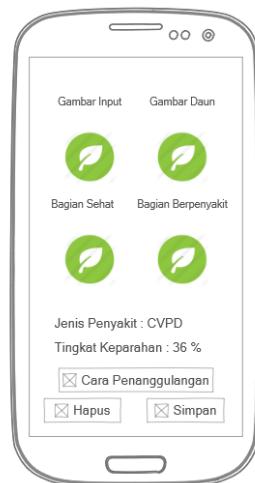
Gambar 4.16 Antarmuka Halaman Mengambil Gambar

Pada Gambar 4.16 merupakan perancangan antarmuka halaman mengambil gambar yang terdiri dari 2 buah pilihan untuk mengambil gambar melalui kamera atau galeri. Jika pengguna memilih ambil gambar melalui kamera maka aplikasi akan membuka kamera sedangkan jika pengguna memilih ambil gambar dari galeri maka aplikasi akan membuka galeri



Gambar 4.17 Antarmuka Halaman Detail Gambar

Pada Gambar 4.17 merupakan perancangan antarmuka halaman detail gambar. Pada halaman antarmuka tersebut berisi gambar dari daun yang telah dipilih. Didalamnya juga terdapat tombol “Crop” yang berfungsi untuk memotong gambar jika gambar tidak sesuai dengan keinginan pengguna. Dan terdapat 2 tombol lain yaitu tombol bantuan yang berisi tentang bantuan sebelum gambar dikirim dan tombol cek berfungsi mengirim gambar untuk dideteksi.



Gambar 4.18 Antarmuka Halaman Hasil Deteksi

Pada Gambar 4.18 perancangan merupakan antarmuka halaman hasil deteksi. Pada halaman antarmuka tersebut terdapat 4 buah detail gambar yaitu gambar *input*, gambar daun, bagian daun yang sehat dan bagian daun yang berpenyakit dan didalamnya terdapat hasil dari deteksi penyakit yaitu jenis penyakit dan persentase tingkat keparahan penyakitnya. Terdapat pula 3 buah tombol yaitu cara pengendalian berdasarkan penyakit yang diderita, simpan riwayat pendeksteksian dan hapus riwayat pendeksteksian.



Gambar 4.19 Antarmuka Halaman Daftar Riwayat

Pada Gambar 4.19 merupakan perancangan antarmuka halaman daftar riwayat. Pada halaman ini berisikan daftar gambar-gambar daun yang telah dideteksi sebelumnya. Detail dari listnya berupa gambar daun, jenis penyakit dan persentase tingkat keparahan serta terdapat pula tombol hapus untuk menghapus daftar riwayat tertentu.



Gambar 4.20 Antarmuka Halaman Daftar Penyakit

Pada Gambar 4.20 merupakan perancangan antarmuka halaman daftar penyakit. Pada antarmuka halaman ini berisikan detail dari 3 jenis penyakit daun jeruk yaitu CVPD, Cendawan Jelaga dan *Downy Mildew*. Jika salah satu daftar penyakit tersebut ditekan maka akan menampilkan detail dari penyakit daun tersebut serta menampilkan detail cara pengendaliannya.

BAB 5 IMPLEMENTASI

Setelah proses perancangan selesai dilakukan, langkah selanjutnya yaitu implementasi sistem. Pada tahap implementasi sistem ini berisikan batasan implementasi, spesifikasi sistem yang terdiri dari spesifikasi perangkat keras dan perangkat lunak, implementasi basis data, implementasi Vize AI, implementasi kode program, dan implementasi antarmuka pengguna.

5.1 Batasan Implementasi

Terdapat beberapa batasan dalam implementasi terkait aplikasi yang dikembangkan, diantaranya :

1. Aplikasi dikembangkan menggunakan *smartphone* Android dengan versi OS minimal API 15.
2. Aplikasi dikembangkan menggunakan *Integrated Development Environment* Android Studio 3.1 dengan menggunakan bahasa pemrograman Java.
3. Aplikasi memanfaatkan Vize AI untuk proses identifikasi daun jeruk.
4. Aplikasi dikembangkan dengan menggunakan retrofit versi 2.3.0 yang digunakan sebagai *Rest Client* pada Android.
5. Aplikasi dikembangkan dengan menggunakan basis data SQLite versi 3.25.3.

5.2 Spesifikasi Sistem

5.2.1 Spesifikasi Perangkat Keras

Perangkat keras yang digunakan untuk mengembangkan aplikasi ini dilakukan di laptop ASUS A456UR dan pengujinya dilakukan di perangkat Android dengan spesifikasi masing-masing dijelaskan dalam Tabel 5.1 dan 5.2

Tabel 5.1 Spesifikasi Perangkat Laptop ASUS A456UR

Komponen	Spesifikasi
Model	ASUS A456UR
Prosesor	Intel Core i5-7200U 2.50 GHz
RAM	8 GB DDR 4
GPU	NVDIA GeForce 930MX

Tabel 5.2 Spesifikasi Perangkat Android

Komponen	Spesifikasi
Model	OPPO A37f
Prosesor	MSM 8916 Quad-core 1.2GHz
RAM	2 GB
Kapasitas Penyimpanan	16 GB
Kamera	8 MP

5.2.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan aplikasi ini menggunakan perangkat berbasis windows dan pada aplikasi perangkat bergeraknya berbasis android dengan spesifikasi masing-masing dijelaskan pada Tabel 5.3 dan 5.4.

Tabel 5.3 Spesifikasi Perangkat Lunak Laptop ASUS A455UR

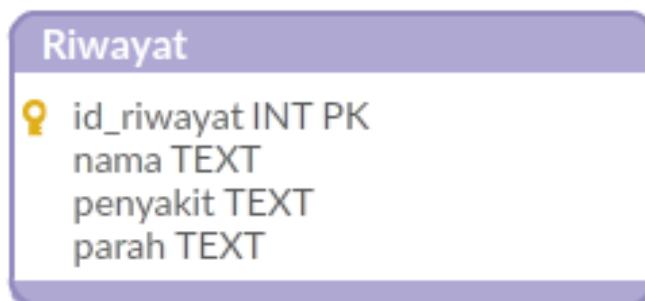
Komponen	Spesifikasi
Sistem Operasi	Windows 10
IDE	Android Studio 3.1
Bahasa Pemrograman	Java 8

Tabel 5.4 Spesifikasi Perangkat Lunak Android

Komponen	Spesifikasi
Sistem Operasi	Android 5.1.1

5.3 Implementasi Basis Data

Implementasi basis data didasarkan pada perancangan basis data yang telah dibuat sebelumnya. Implementasi basis data pada aplikasi ini diimplementasikan menggunakan SQLite yang dijelaskan pada Gambar 5.1.

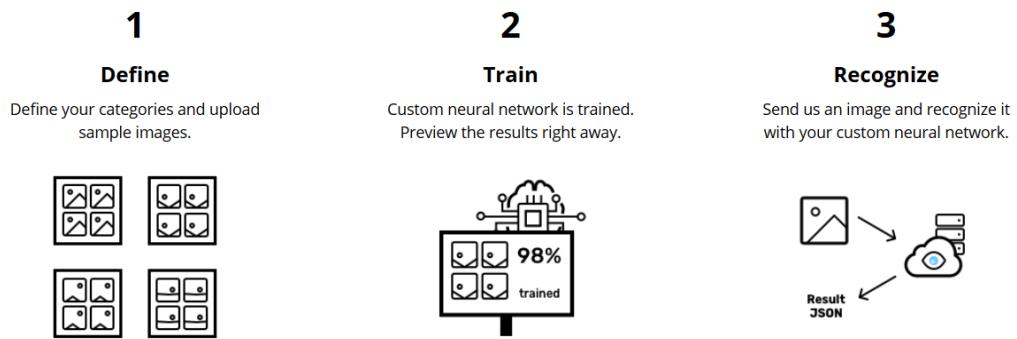


Gambar 5.1 Implementasi *Physical Data Model* (PDM)

Pada Gambar 5.1 merupakan gambar dari *Physical Data Model* (PDM) pada aplikasi ini yang terdapat 1 buah tabel yaitu tabel Riwayat. Tabel riwayat berisikan 4 buah yaitu atribut id_riwayat sebagai *primary key* dengan tipe data *integer*, atribut nama dengan tipe data *text* yang merupakan nama dari gambarnya, penyakit dengan tipe data *text* yang merupakan jenis penyakitnya dan parah dengan tipe data *text* yang merupakan persentase keparahan dari penyakit yang diderita.

5.4 Implementasi Vize Ai

Aplikasi ini memanfaatkan sebuah *web service* untuk proses identifikasi penyakit daun jeruk. Pada implementasinya diperlukan 3 cara yaitu *define*, *train* dan *recognize* yang dijelaskan pada Gambar 5.2.



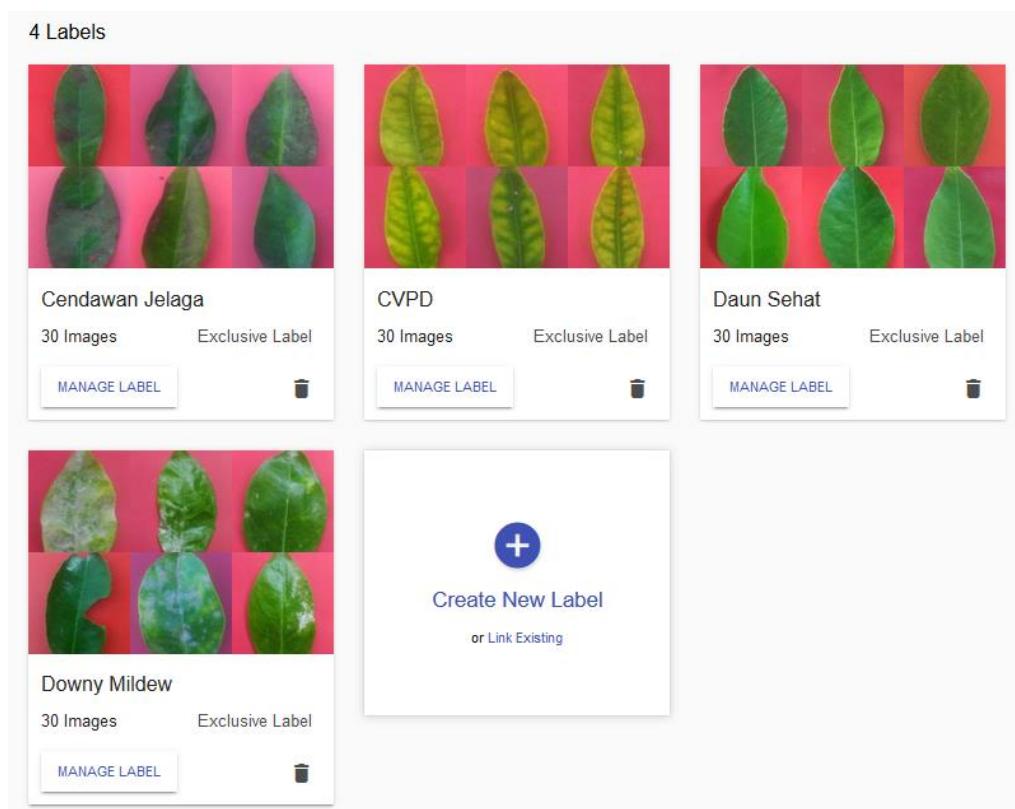
Gambar 5.2 Tahapan Implementasi Vize AI

Sumber : (Vize, 2018)

Pada Gambar 5.2 merupakan 3 tahapan dalam implementasi Vize AI yaitu *define*, *train* dan *recognize*. Pada tahap *define* dilakukan pengelompokan kategori dan meng-*upload* gambar dari sesuai kategorinya. Pada tahap *train* dilakukan proses pelatihan untuk melatih Vize AI mengenali gambar. Pada tahap *recognize* Vize AI dilakukan pengenalan gambar berdasarkan gambar yang telah dikirimkan dengan hasilnya berupa JSON.

5.4.1 Define

Pada tahap *define* dilakukan pengelompokan kategori dan meng-*upload* gambar dari sesuai kategorinya yang dijelaskan pada Gambar 5.3.



Gambar 5.3 Implementasi Tahap Define

Pada Gambar 5.3 merupakan tahapan *define* dalam implementasi Vize AI yaitu dilakukan pengelompokan ke dalam 4 kategori penyakit antara lain Cendawan Jelaga, *Downy Mildew*, CVPD dan Daun Sehat. Lalu dari tiap katergori dimasukkan gambar daun sejumlah 30 gambar sesuai dengan penyakitnya yang nantinya digunakan sebagai data latih untuk proses pengenalan gambar.

5.4.2 Train

Setelah tahap *define* selesai dilakukan maka tahap selanjutnya yaitu tahap *train*. Pada tahap *train* dilakukan proses pelatihan gambar berdasarkan data latih untuk mengenali penyakit daun yang dijelaskan pada Gambar 5.4.

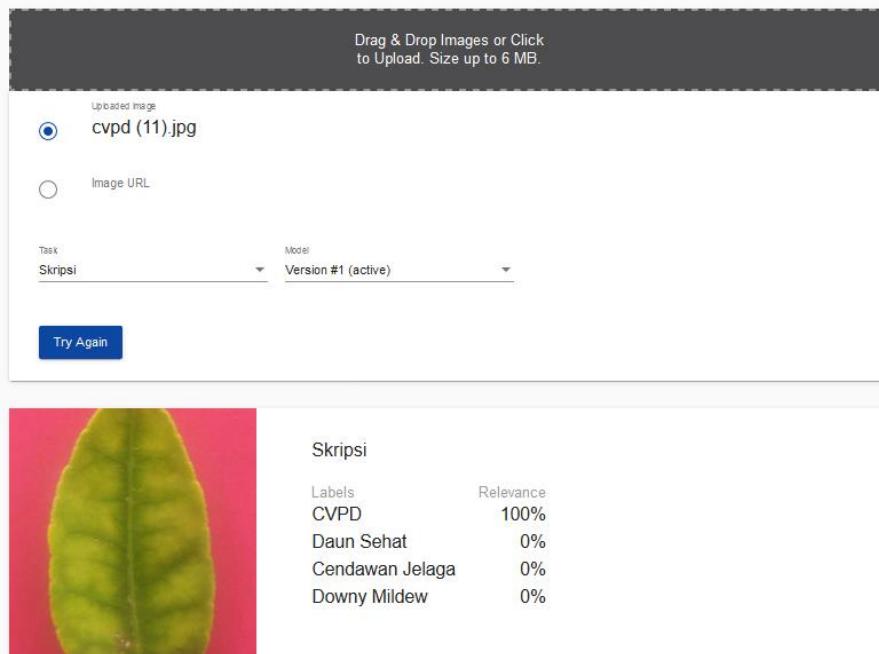


Gambar 5.4 Implementasi Tahap Train

Pada Gambar 5.4 merupakan tahapan *train* dalam implementasi Vize AI yaitu dilakukan proses pelatihan untuk melatih Vize AI mengenali gambar penyakit daun jeruk. Pelatihan telah selesai jika status pada Vize AI yaitu *Trained* yang menandakan bahwa Vize AI siap untuk dilakukan pengenalan gambar.

5.4.3 Recognize

Setelah tahap *train* selesai dilakukan maka tahap selanjutnya yaitu tahap *recognize*. Pada tahap *recognize* dilakukan pengenalan gambar yang dijelaskan pada Gambar 5.5.



Labels	Relevance
CVPD	100%
Daun Sehat	0%
Cendawan Jelaga	0%
Downy Mildew	0%

Gambar 5.5 Implementasi Tahap Recognize

Pada Gambar 5.5 merupakan tahapan *recognize* dalam implementasi Vize AI yaitu dilakukan proses pengenalan gambar dengan cara mengirimkan gambar daun ke *web service* Vize AI. Gambar yang dikirimkan maksimal berukuran 6 MB. Setelah dilakukan proses pengenalan gambar maka hasil yang keluar yaitu persentase dari semua jenis penyakit serta menampilkan hasil persentase terbaik dari jenis penyakit yang menyerang daunnya dan Vize AI juga mengirimkan hasil berupa JSON yang nantinya data JSON tersebut dapat diolah ke dalam aplikasi yang berbeda.

5.5 Implementasi Kode Program

Implementasi kode program didasarkan pada perancangan algoritme telah dibuat sebelumnya. Implementasi kode program pada aplikasi ini terdapat dua implementasi kode program yaitu implementasi kode program mendeteksi penyakit dan implementasi kode program menghitung tingkat keparahan yang diimplementasikan menggunakan Android Studio.

5.5.1 Implementasi Kode Program Mendeteksi Penyakit

Implementasi mendeteksi penyakit pada aplikasi ini memanfaatkan *web service* yaitu Vize AI, yang nantinya akan mengembalikan data berupa penyakit yang menyerang daunnya. Adapun kode program dalam implementasi mendeteksi penyakit dituliskan pada Kode Program 5.1.

Kode Program 5.1 Implementasi Mendeteksi Penyakit

DeteksiActivity.java	
1 public void diseaseDetection(){ 2 final ProgressDialog pd = new 3 ProgressDialog(DeteksiActivity.this); 4 pd.setMessage("Mohon Tunggu"); 5 pd.show(); 6 7 if (isNetworkAvailable(getApplicationContext())) { 8 File file = new File(new 9 File("/sdcard/leafDetector/"),name); 10 RequestBody requestFile = 11 RequestBody.create(MediaType.parse("multipart/form-data"), 12 file); 13 MultipartBody.Part body = 14 MultipartBody.Part.createFormData("image_file", 15 file.getName(), requestFile); 16 RequestBody task = 17 RequestBody.create(okhttp3.MultipartBody.FORM, "32fe9e75- 18 0d04-4f52-8aa9-990019b97607"); 19 20 service = APIUtil.getAPIService(); 21 service.sendImage("Token 22 62ee153fb1d9327e93d84b03eb9daa099f66304d",body,task).enqueue 23 (new Callback<DeteksiModel>() { 24 @Override 25 public void onResponse(Call<DeteksiModel> 26 call, Response<DeteksiModel> response) { 27 28 } 29 }); 30 } 31 } 32 } 33 } 34 } 35 } 36 } 37 } 38 } 39 } 40 } 41 } 42 } 43 } 44 } 45 } 46 } 47 } 48 } 49 } 50 } 51 } 52 } 53 } 54 } 55 } 56 } 57 } 58 } 59 } 60 } 61 } 62 } 63 } 64 } 65 } 66 } 67 } 68 } 69 } 70 } 71 } 72 } 73 } 74 } 75 } 76 } 77 } 78 } 79 } 80 } 81 } 82 } 83 } 84 } 85 } 86 } 87 } 88 } 89 } 90 } 91 } 92 } 93 } 94 } 95 } 96 } 97 } 98 } 99 } 100 } 101 } 102 } 103 } 104 } 105 } 106 } 107 } 108 } 109 } 110 } 111 } 112 } 113 } 114 } 115 } 116 } 117 } 118 } 119 } 120 } 121 } 122 } 123 } 124 } 125 } 126 } 127 } 128 } 129 } 130 } 131 } 132 } 133 } 134 } 135 } 136 } 137 } 138 } 139 } 140 } 141 } 142 } 143 } 144 } 145 } 146 } 147 } 148 } 149 } 150 } 151 } 152 } 153 } 154 } 155 } 156 } 157 } 158 } 159 } 160 } 161 } 162 } 163 } 164 } 165 } 166 } 167 } 168 } 169 } 170 } 171 } 172 } 173 } 174 } 175 } 176 } 177 } 178 } 179 } 180 } 181 } 182 } 183 } 184 } 185 } 186 } 187 } 188 } 189 } 190 } 191 } 192 } 193 } 194 } 195 } 196 } 197 } 198 } 199 } 200 } 201 } 202 } 203 } 204 } 205 } 206 } 207 } 208 } 209 } 210 } 211 } 212 } 213 } 214 } 215 } 216 } 217 } 218 } 219 } 220 } 221 } 222 } 223 } 224 } 225 } 226 } 227 } 228 } 229 } 230 } 231 } 232 } 233 } 234 } 235 } 236 } 237 } 238 } 239 } 240 } 241 } 242 } 243 } 244 } 245 } 246 } 247 } 248 } 249 } 250 } 251 } 252 } 253 } 254 } 255 } 256 } 257 } 258 } 259 } 260 } 261 } 262 } 263 } 264 } 265 } 266 } 267 } 268 } 269 } 270 } 271 } 272 } 273 } 274 } 275 } 276 } 277 } 278 } 279 } 280 } 281 } 282 } 283 } 284 } 285 } 286 } 287 } 288 } 289 } 290 } 291 } 292 } 293 } 294 } 295 } 296 } 297 } 298 } 299 } 300 } 301 } 302 } 303 } 304 } 305 } 306 } 307 } 308 } 309 } 310 } 311 } 312 } 313 } 314 } 315 } 316 } 317 } 318 } 319 } 320 } 321 } 322 } 323 } 324 } 325 } 326 } 327 } 328 } 329 } 330 } 331 } 332 } 333 } 334 } 335 } 336 } 337 } 338 } 339 } 340 } 341 } 342 } 343 } 344 } 345 } 346 } 347 } 348 } 349 } 350 } 351 } 352 } 353 } 354 } 355 } 356 } 357 } 358 } 359 } 360 } 361 } 362 } 363 } 364 } 365 } 366 } 367 } 368 } 369 } 370 } 371 } 372 } 373 } 374 } 375 } 376 } 377 } 378 } 379 } 380 } 381 } 382 } 383 } 384 } 385 } 386 } 387 } 388 } 389 } 390 } 391 } 392 } 393 } 394 } 395 } 396 } 397 } 398 } 399 } 400 } 401 } 402 } 403 } 404 } 405 } 406 } 407 } 408 } 409 } 410 } 411 } 412 } 413 } 414 } 415 } 416 } 417 } 418 } 419 } 420 } 421 } 422 } 423 } 424 } 425 } 426 } 427 } 428 } 429 } 430 } 431 } 432 } 433 } 434 } 435 } 436 } 437 } 438 } 439 } 440 } 441 } 442 } 443 } 444 } 445 } 446 } 447 } 448 } 449 } 450 } 451 } 452 } 453 } 454 } 455 } 456 } 457 } 458 } 459 } 460 } 461 } 462 } 463 } 464 } 465 } 466 } 467 } 468 } 469 } 470 } 471 } 472 } 473 } 474 } 475 } 476 } 477 } 478 } 479 } 480 } 481 } 482 } 483 } 484 } 485 } 486 } 487 } 488 } 489 } 490 } 491 } 492 } 493 } 494 } 495 } 496 } 497 } 498 } 499 } 500 } 501 } 502 } 503 } 504 } 505 } 506 } 507 } 508 } 509 } 510 } 511 } 512 } 513 } 514 } 515 } 516 } 517 } 518 } 519 } 520 } 521 } 522 } 523 } 524 } 525 } 526 } 527 } 528 } 529 } 530 } 531 } 532 } 533 } 534 } 535 } 536 } 537 } 538 } 539 } 540 } 541 } 542 } 543 } 544 } 545 } 546 } 547 } 548 } 549 } 550 } 551 } 552 } 553 } 554 } 555 } 556 } 557 } 558 } 559 } 560 } 561 } 562 } 563 } 564 } 565 } 566 } 567 } 568 } 569 } 570 } 571 } 572 } 573 } 574 } 575 } 576 } 577 } 578 } 579 } 580 } 581 } 582 } 583 } 584 } 585 } 586 } 587 } 588 } 589 } 590 } 591 } 592 } 593 } 594 } 595 } 596 } 597 } 598 } 599 } 600 } 601 } 602 } 603 } 604 } 605 } 606 } 607 } 608 } 609 } 610 } 611 } 612 } 613 } 614 } 615 } 616 } 617 } 618 } 619 } 620 } 621 } 622 } 623 } 624 } 625 } 626 } 627 } 628 } 629 } 630 } 631 } 632 } 633 } 634 } 635 } 636 } 637 } 638 } 639 } 640 } 641 } 642 } 643 } 644 } 645 } 646 } 647 } 648 } 649 } 650 } 651 } 652 } 653 } 654 } 655 } 656 } 657 } 658 } 659 } 660 } 661 } 662 } 663 } 664 } 665 } 666 } 667 } 668 } 669 } 670 } 671 } 672 } 673 } 674 } 675 } 676 } 677 } 678 } 679 } 680 } 681 } 682 } 683 } 684 } 685 } 686 } 687 } 688 } 689 } 690 } 691 } 692 } 693 } 694 } 695 } 696 } 697 } 698 } 699 } 700 } 701 } 702 } 703 } 704 } 705 } 706 } 707 } 708 } 709 } 710 } 711 } 712 } 713 } 714 } 715 } 716 } 717 } 718 } 719 } 720 } 721 } 722 } 723 } 724 } 725 } 726 } 727 } 728 } 729 } 730 } 731 } 732 } 733 } 734 } 735 } 736 } 737 } 738 } 739 } 740 } 741 } 742 } 743 } 744 } 745 } 746 } 747 } 748 } 749 } 750 } 751 } 752 } 753 } 754 } 755 } 756 } 757 } 758 } 759 } 760 } 761 } 762 } 763 } 764 } 765 } 766 } 767 } 768 } 769 } 770 } 771 } 772 } 773 } 774 } 775 } 776 } 777 } 778 } 779 } 780 } 781 } 782 } 783 } 784 } 785 } 786 } 787 } 788 } 789 } 790 } 791 } 792 } 793 } 794 } 795 } 796 } 797 } 798 } 799 } 800 } 801 } 802 } 803 } 804 } 805 } 806 } 807 } 808 } 809 } 810 } 811 } 812 } 813 } 814 } 815 } 816 } 817 } 818 } 819 } 820 } 821 } 822 } 823 } 824 } 825 } 826 } 827 } 828 } 829 } 830 } 831 } 832 } 833 } 834 } 835 } 836 } 837 } 838 } 839 } 840 } 841 } 842 } 843 } 844 } 845 } 846 } 847 } 848 } 849 } 850 } 851 } 852 } 853 } 854 } 855 } 856 } 857 } 858 } 859 } 860 } 861 } 862 } 863 } 864 } 865 } 866 } 867 } 868 } 869 } 870 } 871 } 872 } 873 } 874 } 875 } 876 } 877 } 878 } 879 } 880 } 881 } 882 } 883 } 884 } 885 } 886 } 887 } 888 } 889 } 890 } 891 } 892 } 893 } 894 } 895 } 896 } 897 } 898 } 899 } 900 } 901 } 902 } 903 } 904 } 905 } 906 } 907 } 908 } 909 } 910 } 911 } 912 } 913 } 914 } 915 } 916 } 917 } 918 } 919 } 920 } 921 } 922 } 923 } 924 } 925 } 926 } 927 } 928 } 929 } 930 } 931 } 932 } 933 } 934 } 935 } 936 } 937 } 938 } 939 } 940 } 941 } 942 } 943 } 944 } 945 } 946 } 947 } 948 } 949 } 950 } 951 } 952 } 953 } 954 } 955 } 956 } 957 } 958 } 959 } 960 } 961 } 962 } 963 } 964 } 965 } 966 } 967 } 968 } 969 } 970 } 971 } 972 } 973 } 974 } 975 } 976 } 977 } 978 } 979 } 980 } 981 } 982 } 983 } 984 } 985 } 986 } 987 } 988 } 989 } 990 } 991 } 992 } 993 } 994 } 995 } 996 } 997 } 998 } 999 } 1000 } 1001 } 1002 } 1003 } 1004 } 1005 } 1006 } 1007 } 1008 } 1009 } 1010 } 1011 } 1012 } 1013 } 1014 } 1015 } 1016 } 1017 } 1018 } 1019 } 1020 } 1021 } 1022 } 1023 } 1024 } 1025 } 1026 } 1027 } 1028 } 1029 } 1030 }<br	

```

28                     if (response.isSuccessful()) {
29                         penyakit =
30                             response.body().getBest_label().getLabel_name().toString();
31                         Intent intent = new
32                             Intent(getApplicationContext(), HasilActivity.class);
33                             intent.putExtra("penyakit", penyakit);
34                             startActivity(intent);
35                             pd.dismiss();
36                             finish();
37                     } else {
38                         Log.d("RESPONSE", "GAGAL");
39                     }
40                 }
41
42             @Override
43             public void onFailure(Call<DeteksiModel>
44 call, Throwable t) {
45                 Log.d("RESPONSE", t.getMessage());
46                 pd.dismiss();
47                 AlertDialog.Builder builder1 = new
48                     AlertDialog.Builder(DeteksiActivity.this);
49                     builder1.setMessage("Gagal mendeteksi.
50 \nDeteksi ulang gambar!");
51                     builder1.setCancelable(true);
52                     builder1.setPositiveButton(
53                         "OK",
54                         new
55                         DialogInterface.OnClickListener() {
56                             public void
57                             onClick(DialogInterface dialog, int id) {
58                                 dialog.cancel();
59                             }
60                             });
61                     AlertDialog alert11 =
62                     builder1.create();
63                     alert11.show();
64                 }
65             });
66         } else {
67             pd.dismiss();
68             AlertDialog.Builder builder1 = new
69             AlertDialog.Builder(DeteksiActivity.this);
70             builder1.setMessage("Tidak ada koneksi
71             internet. \nMohon cek koneksi internet anda!");
72             builder1.setCancelable(true);
73             builder1.setPositiveButton(
74                 "OK",
75                 new DialogInterface.OnClickListener() {
76                     public void onClick(DialogInterface
77                     dialog, int id) {
78                         dialog.cancel();
79                     }
80                 });
81             AlertDialog alert11 = builder1.create();
82             alert11.show();
83         }
84     }

```

Pembahasan dari Kode Program 5.1 adalah sebagai berikut :

1. Baris 1 – 6 : Membuat *progress dialog* yang berisikan pesan “Mohon Tunggu” lalu menampilkannya.
2. Baris 7 – 19 : Terdapat seleksi kondisi jika terhubung koneksi maka akan mengirimkan gambar daunnya. Lalu gambar tersebut dibungkus dalam tipe data *requestBody* agar bisa dikirimkan.
3. Baris 20 – 41 : Terdapat *method sendImage()* yang digunakan untuk mengirimkan gambar yang nantinya dikirimkan ke *web service* Vize AI menggunakan *method POST*. Lalu didalamnya terdapat *method onResponse()* yang harus di-*override* ketika menggunakan *library Retrofit*, dimana *method* tersebut dipanggil jika berhasil terhubung dengan *web service*. Jika *response* sukses maka data penyakit daun dapat diambil jika tidak akan menampilkan pemberitahuan gagal.
4. Baris 42 – 65 : Terdapat *method onFailure()* yang harus di-*override* ketika menggunakan *library Retrofit*, dimana *method* tersebut dipanggil jika gagal terhubung dengan *web service* dan akan menampilkan pemberitahuan gagal dan memerintahkan pengguna untuk melakukan pendekripsi ulang.
5. Baris 67 - 84 : Terdapat seleksi kondisi jika tidak terhubung dengan koneksi dan akan menampilkan pemberitahuan untuk mengecek koneksi.

5.5.2 Implementasi Kode Program Menghitung Tingkat Keparahan

Implementasi menghitung tingkat keparahan pada aplikasi ini menggunakan perhitungan per-*pixel* dari gambar yang diambil yang nantinya tiap *pixel* akan di seleksi apakah warnanya sesuai dengan warna penyakit atau tidak dan pada tahap ini juga aplikasi dapat menampilkan berupa 4 buah gambar yaitu gambar *input*, gambar daun, gambar daun sehat dan gambar daun berpenyakit yang dapat membantu pengguna untuk melihat hasil dari perhitungan tingkat keparahan. Adapun kode program dalam implementasi mengitung tingkat keparahan dituliskan pada Kode Program 5.2.

Kode Program 5.2 Implementasi Mengitung Tingkat Keparahan

HasilActivity.java	
1 @Override 2 protected void onCreate(Bundle savedInstanceState) { 3 super.onCreate(savedInstanceState); 4 setContentView(R.layout.activity_hasil); 5 penyakit = getIntent().getStringExtra("penyakit"); 6 b_input = convertImage(b_input); 7 img_input.setImageBitmap(b_input); 8 b_daun = daunImage(b_input); 9 b_sehat = sehatImage(b_daun); 10 b_penyakit = penyakitImage(b_daun); 11 img_daun.setImageBitmap(b_daun); 12 img_sehat.setImageBitmap(b_sehat); 13 img_penyakit.setImageBitmap(b_penyakit); 14 parah = (int) (count_penyakit * 100) / count_daun;	

```

15 }
16
17     public static Bitmap convertImage(Bitmap original) {
18         Bitmap resized =
19             Bitmap.createScaledBitmap(original, 112, 150, true);
20         return resized;
21     }
22
23     public static void convertLab(int red, int green, int
24 blue) {
25         float x = (0.412453f * red) + (0.35758f * green) +
26 (0.180423f * blue);
27         float y = (0.212671f * red) + (0.71516f * green) +
28 (0.072169f * blue);
29         float z = (0.019334f * red) + (0.119193f * green) +
30 (0.950227f * blue);
31         float xNormal = x / 242.36628f;
32         float yNormal = y / 255f;
33         float zNormal = z / 277.63227f;
34         float fx, fy, fz;
35         if (xNormal > 0.008856) {
36             fx = (float) Math.pow(xNormal, 1 / 3.);
37         } else {
38             fx = (float) (7.787 * xNormal + 0.1379);
39         }
40         if (yNormal > 0.008856) {
41             fy = (float) Math.pow(yNormal, 1 / 3.);
42         } else {
43             fy = (float) (7.787 * yNormal + 0.1379);
44         }
45         if (zNormal > 0.008856) {
46             fz = (float) Math.pow(zNormal, 1 / 3.);
47         } else {
48             fz = (float) (7.787 * zNormal + 0.1379);
49         }
50         if (yNormal > 0.008856) {
51             l = 116 * fy - 16;
52         } else {
53             l = (float) 903.3 * yNormal;
54         }
55         a = 500 * (fx - fy);
56         b = 200 * (fy - fz);
57     }
58
59     public static Bitmap daunImage(Bitmap original) {
60         count_daun = 0;
61         Bitmap gambar_daun =
62             Bitmap.createBitmap(original.getWidth(),
63 original.getHeight(), original.getConfig());
64         int colorPixel;
65         int width = original.getWidth();
66         int height = original.getHeight();
67         for (int i = 0; i < width; i++) {
68             for (int j = 0; j < height; j++) {
69                 colorPixel = original.getPixel(i, j);
70                 int A = Color.alpha(colorPixel);
71                 int red = Color.red(colorPixel);
72                 int green = Color.green(colorPixel);
73                 int blue = Color.blue(colorPixel);

```

```

74             l=0; a=0; b=0;
75             convertLab(red,green,blue);
76             if ( (a>=15 && a<=55) && (b>=-20 &&
77             b<=25) ) {
78                 red = 255;
79                 green = red;
80                 blue = red;
81             } else {
82                 count_daun++;
83             }
84             gambar_daun.setPixel(i, j, Color.argb(A,
85             red, green, blue));
86         }
87     }
88     return gambar_daun;
89 }
90
91 public static Bitmap sehatImage(Bitmap original){
92     Bitmap gambar_daun =
93     Bitmap.createBitmap(original.getWidth(),
94     original.getHeight(), original.getConfig());
95     int colorPixel;
96     int widht = original.getWidth();
97     int height = original.getHeight();
98     for (int i = 0; i < widht; i++){
99         for (int j = 0; j < height; j++){
100            colorPixel = original.getPixel(i,j);
101            int A = Color.alpha(colorPixel);
102            int red = Color.red(colorPixel);
103            int green = Color.green(colorPixel);
104            int blue = Color.blue(colorPixel);
105            l=0; a=0; b=0;
106            convertLab(red,green,blue);
107            if (penyakit.equalsIgnoreCase("Downy
108 Mildew") || penyakit.equalsIgnoreCase("Mildew")){
109                if ((a>=-37 && a<=0) && (b>=-26 &&
110 b<=11) && (red!=255)){
111                    red = 255;
112                    green = red;
113                    blue = red;
114                }
115            } else if
116            (penyakit.equalsIgnoreCase("CVPD")){
117                if ((l>=71 && l<=100) && (a>=-28 &&
118 a<=15) && (b>=40 && b<=85) && (red!=255) ){
119                    red = 255;
120                    green = red;
121                    blue = red;
122                }
123            } else if
124            (penyakit.equalsIgnoreCase("Cendawan
125 Jelaga") || penyakit.equalsIgnoreCase("Jelaga")){
126                if ((a>=-18 && a<=9) && (b>=-5 &&
127 b<=29) && (red!=255)){
128                    red = 255;
129                    green = red;
130                    blue = red;
131                }
132            }

```

```

133             gambar_daun.setPixel(i, j, Color.argb(A,
134             red, green, blue));
135         }
136     }
137     return gambar_daun;
138 }
139
140 public static Bitmap penyakitImage(Bitmap original){
141     count_penyakit = 0;
142     Bitmap gambar_daun =
143 Bitmap.createBitmap(original.getWidth(),
144 original.getHeight(), original.getConfig());
145     int colorPixel;
146     int width = original.getWidth();
147     int height = original.getHeight();
148     for (int i = 0; i < width; i++){
149         for (int j = 0; j < height; j++){
150             colorPixel = original.getPixel(i,j);
151             int A = Color.alpha(colorPixel);
152             int red = Color.red(colorPixel);
153             int green = Color.green(colorPixel);
154             int blue = Color.blue(colorPixel);
155             l=0; a=0; b=0;
156             convertLab(red,green,blue);
157             if (penyakit.equalsIgnoreCase("Downy
158 Mildew")) || penyakit.equalsIgnoreCase("Mildew")){
159                 if ((a>=-37 && a<=0) && (b>=-26 &&
160 b<=11) && (red!=255)) {
161                     count_penyakit++;
162                 } else {
163                     red = 255;
164                     green = red;
165                     blue = red;
166                 }
167             } else if
168             (penyakit.equalsIgnoreCase("CVPD")){
169                 if ((l>=71 && l<=100)&&(a>=-28 &&
170 a<=15) && (b>=40 && b<=85) && (red!=255)){
171                     count_penyakit++;
172                 } else {
173                     red = 255;
174                     green = red;
175                     blue = red;
176                 }
177             } else if
178             (penyakit.equalsIgnoreCase("Cendawan
179 Jelaga")) || penyakit.equalsIgnoreCase("Jelaga")){
180                 if ((a>=-18 && a<=9) && (b>=-5 &&
181 b<=29) && (red!=255)){
182                     count_penyakit++;
183                 } else {
184                     red = 255;
185                     green = red;
186                     blue = red;
187                 }
188             }
189         }
190         gambar_daun.setPixel(i, j, Color.argb(A,
191             red, green, blue));
192     }

```

193	}
194	return gambar_daun;
195	}

Pembahasan dari Kode Program 5.2 adalah sebagai berikut :

1. Baris 1 – 16 : Terdapat *method onCreate* yang merupakan *method* yang pertama kali dijalankan oleh *compiler*. Di dalam *method* tersebut dilakukan pemanggilan beberapa *method* yaitu *method convertImage()*, *daunImage()*, *sehatImage()* dan *penyakitImage()* serta dilakukan perhitungan persentase tingkat keparahan penyakit.
2. Baris 17 – 22 : Terdapat *method convertImage()* yang digunakan untuk mengubah ukuran gambar menjadi gambar dengan ukuran yang lebih kecil yaitu dengan lebar 112 *pixel* dan tinggi 150 *pixel*.
3. Baris 23 – 58 : Terdapat *method convertLab()* yang digunakan untuk mengubah ruang warna RGB menjadi ruang warna L*a*b*.
4. Baris 59 – 90 : Terdapat *method daunImage()* yang digunakan untuk menghilangkan *cover* yang berwarna merah pada gambar dan pada *method* ini pula dilakukan perhitungan jumlah *pixel* yang termasuk gambar daun. Di dalamnya dilakukan perulangan sejumlah lebar dan tinggi dari *pixel* gambar dan dari tiap perulangan dilakukan convert ruang warna RGB menjadi ruang warna L*a*b*. Lalu setelah nilai l,a, dan b didapatkan maka dilakukan seleksi kondisi jika warna L*a*b* nya sesuai dengan warna merah maka covernya akan diubah menjadi warna putih dan jika warna L*a*b* nya tidak berwarna merah maka dilakukan penambahan pada variabel *count_daun* dari tiap *pixel*.
5. Baris 91 – 139 : Terdapat *method sehatImage()* yang digunakan untuk menampilkan gambar daun yang sehat saja dengan menghilangkan bagian penyakit pada daun dan pada *method* ini pula dilakukan perhitungan jumlah *pixel* yang termasuk gambar penyakit. Di dalamnya dilakukan perulangan sejumlah lebar dan tinggi dari *pixel* gambar dan dari tiap perulangan dilakukan convert ruang warna RGB menjadi ruang warna L*a*b*. Lalu setelah nilai l,a, dan b didapatkan maka dilakukan seleksi kondisi jika warna L*a*b* nya sesuai dengan warna penyakit maka warna penyakit akan diubah menjadi warna putih serta.
6. Baris 140 – 195 : Terdapat *method penyakitImage()* yang digunakan untuk menampilkan gambar daun yang berpenyakit saja dengan menghilangkan bagian yang sehat pada daun. Di dalamnya dilakukan perulangan sejumlah lebar dan tinggi dari *pixel* gambar dan dari tiap perulangan dilakukan convert ruang warna RGB menjadi ruang warna L*a*b*. Lalu setelah nilai l,a, dan b didapatkan maka dilakukan seleksi kondisi jika warna L*a*b* nya sesuai dengan warna penyakit maka dilakukan penambahan pada variabel *count_penyakit* dari tiap *pixel* namun jika tidak sesuai dengan warna penyakit maka akan diubah menjadi warna putih.

5.6 Implementasi Antarmuka Pengguna

Implementasi antarmuka didasarkan pada perancangan antarmuka yang telah dilakukan sebelumnya. Pada Aplikasi ini terdapat beberapa antarmuka pengguna yaitu antarmuka halaman awal, halaman mengambil gambar, halaman detail gambar, halaman hasil deteksi, halaman daftar riwayat dan halaman daftar penyakit yang dijelaskan pada Gambar 5.6 sampai 5.11.



Gambar 5.6 Halaman Awal

Pada Gambar 5.6 merupakan antarmuka halaman awal aplikasi yang terdapat logo, nama aplikasi dan 4 buah tombol di dalamnya yaitu deteksi (berisi proses pendektesian penyakit daun jeruk), riwayat (berisi daftar riwayat pendektesian), daftar penyakit (berisi daftar penyakit daun jeruk) dan deskripsi tentang aplikasi.



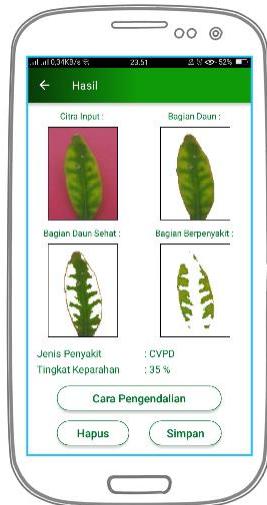
Gambar 5.7 Halaman Mengambil Gambar

Pada Gambar 5.7 merupakan antarmuka halaman mengambil gambar yang terdiri dari 2 buah pilihan untuk mengambil gambar melalui kamera atau galeri. Jika pengguna memilih ambil gambar melalui kamera maka aplikasi akan membuka kamera sedangkan jika pengguna memilih ambil gambar dari galeri maka aplikasi akan membuka galeri



Gambar 5.8 Halaman Detail Gambar

Pada Gambar 5.8 merupakan antarmuka halaman detail gambar. Pada halaman antarmuka tersebut berisi gambar dari daun yang telah dipilih. Didalamnya juga terdapat tombol “Crop” yang berfungsi untuk memotong gambar jika gambar tidak sesuai dengan keinginan pengguna. Dan terdapat 2 tombol lain yaitu tombol bantuan yang berisi tentang bantuan sebelum gambar dikirim dan tombol cek berfungsi mengirim gambar untuk dideteksi.



Gambar 5.9 Halaman Hasil Deteksi

Pada Gambar 5.9 merupakan antarmuka halaman hasil deteksi. Pada halaman antarmuka tersebut terdapat 4 buah detail gambar yaitu gambar *input*, gambar daun, bagian daun yang sehat dan bagian daun yang berpenyakit dan didalamnya terdapat hasil dari deteksi penyakit yaitu jenis penyakit dan persentase tingkat keparahan penyakitnya. Terdapat pula 3 buah tombol yaitu cara pengendalian berdasarkan penyakit yang diderita, simpan riwayat pendekstian dan hapus riwayat pendekstian.



Gambar 5.10 Halaman Daftar Riwayat

Pada Gambar 5.10 merupakan antarmuka halaman daftar riwayat. Pada halaman ini berisikan daftar riwayat gambar daun yang telah dideteksi sebelumnya. Detail dari listnya berupa gambar daun, jenis penyakit dan persentase tingkat keparahan serta terdapat pula tombol hapus untuk menghapus daftar riwayat tertentu.



Gambar 5.11 Halaman Daftar Penyakit

Pada Gambar 5.11 merupakan perancangan antarmuka halaman daftar penyakit. Pada antarmuka halaman ini berisikan detail dari 3 jenis penyakit daun jeruk yaitu CVPD, Cendawan Jelaga dan *Downy Mildew*. Jika salah satu daftar penyakit tersebut ditekan maka akan menampilkan detail dari penyakit daun tersebut serta menampilkan detail cara pengendaliannya.

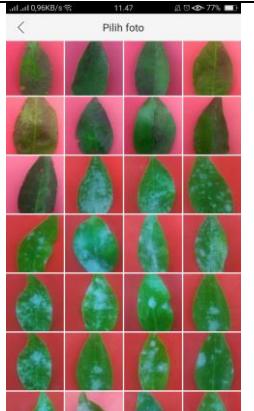
BAB 6 PENGUJIAN

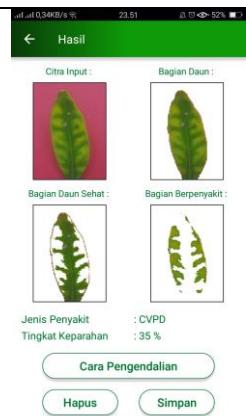
Pada bab ini berisikan pengujian aplikasi yang dibuat apakah aplikasi sudah sesuai dengan kebutuhan pengguna dan apakah aplikasi sudah berjalan dengan baik dan benar. Pengujian pada aplikasi ini terdiri dari 3 pengujian yaitu pengujian fungsional, pengujian akurasi dan pengujian usabilitas.

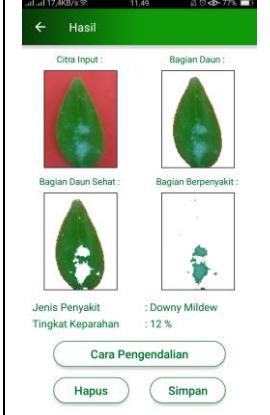
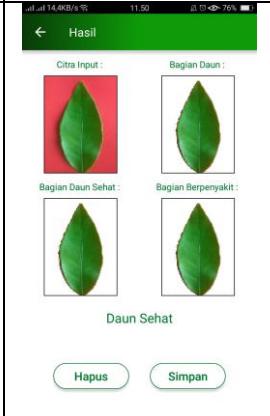
6.1 Pengujian Fungsional

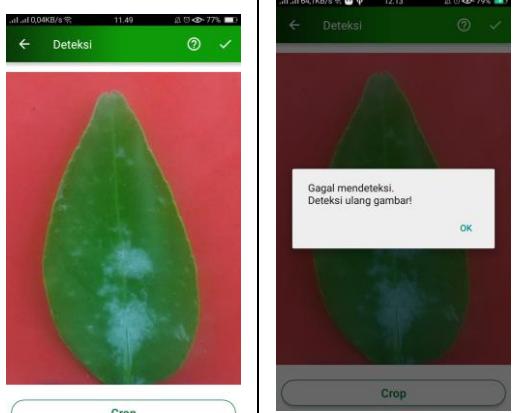
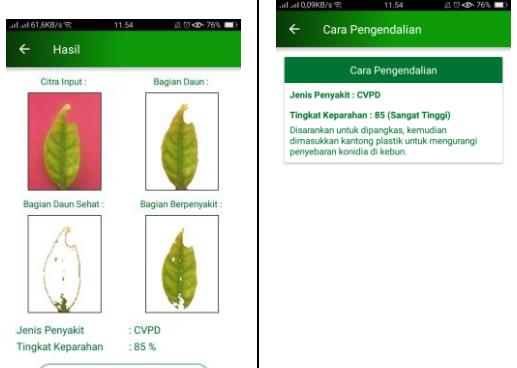
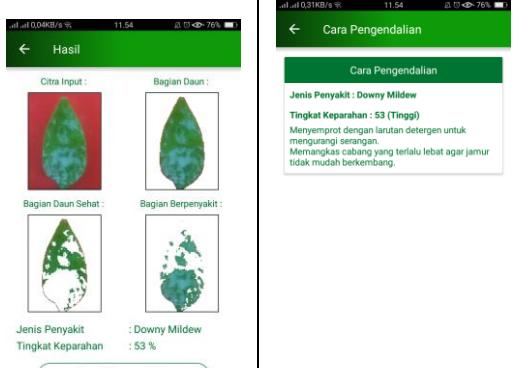
Pengujian fungsional adalah jenis pengujian yang berfokus pada keluaran hasil yang dilakukan sesuai dengan kasus uji pada masing-masing kebutuhan fungsional dengan menggunakan metode *blackbox testing*. Terdapat 6 kasus uji pada pengujian fungsional ini yang dijelaskan pada Tabel 6.1.

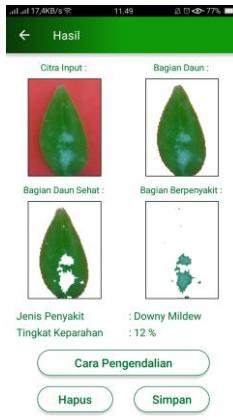
Tabel 6.1 Pengujian Fungsional Sistem

No	Test Case	Target	Test Execution	Expected Result	Validation
1	Mengambil Gambar	Gambar dapat diambil melalui kamera			Valid
		Gambar dapat diambil melalui galeri			Valid

2	Crop Gambar	Gambar berhasil dipotong Mengatur gambar yang ingin dipotong dan menekan tombol "Simpan"	 	Valid
3	Mendeteksi penyakit dan tingkat keparahannya	Gambar terdeteksi penyakit CVPD Mendeteksi daun berpenyakit CVPD	 	Valid
		Gambar terdeteksi penyakit Cendawan Jelaga Mendeteksi daun berpenyakit Cendawan Jelaga	 	Valid

	<p>Gambar terdeteksi penyakit <i>Downy Mildew</i></p> <p>Mendeteksi daun berpenyakit <i>Downy Mildew</i></p>	 <p>Crop</p>  <p>Citra Input : Bagian Daun : Bagian Daun Sehat : Bagian Berpenyakit : Jenis Penyakit : Downy Mildew Tingkat Keparahan : 12 % Cara Pengendalian Hapus Simpan</p>	Valid
	<p>Gambar terdeteksi daun sehat</p> <p>Mendeteksi daun sehat</p>	 <p>Crop</p>  <p>Citra Input : Bagian Daun : Bagian Daun Sehat : Bagian Berpenyakit : Daun Sehat Hapus Simpan</p>	Valid
	<p>Gambar gagal dideteksi karena <i>smartphone</i> tidak terkoneksi internet</p> <p>Mendeteksi gambar dengan <i>smartphone</i> tidak terkoneksi internet</p>	 <p>Crop</p>  <p>Tidak ada koneksi internet. Mohon cek koneksi internet anda! OK</p>	Valid

		Gambar tidak berhasil dikirim		Valid
4	Mengetahui informasi cara pengendalian	Cara pengendalian dengan persentase keparahan antara 76 – 100		Valid
		Cara pengendalian dengan persentase keparahan antara 51 – 75		Valid

		<p>Cara pengendalian dengan persentase keparahan antara 26 – 50</p>	 <p>Penyakit dengan persentase keparahan 29%</p>		Valid
		<p>Cara pengendalian dengan persentase keparahan antara 0 - 25</p>	 <p>Penyakit dengan persentase keparahan 12%</p>		Valid
5	Mengetahui riwayat gambar yang telah terdeteksi	Menampilkan daftar riwayat pendektsian	 <p>Menekan tombol bertuliskan "Daftar Riwayat"</p>	 <p>Menampilkan daftar riwayat pendektsian</p>	Valid

		Menampilkan informasi jika daftar riwayat masih kosong			Valid
6	Mengetahui informasi penyakit daun jeruk	Menekan tombol bertuliskan "Daftar Riwayat"		Menampilkan informasi bahwa riwayat masih kosong	
		Mengetahui detail penyakit daun CVPD			Valid
		Menekan tombol bertuliskan "Baca selengkapnya"			Menampilkan detail penyakit daun CVPD
		Mengetahui detail penyakit daun Cendawan Jelaga			Valid

		<p>Mengetahui detail penyakit daun <i>Downy Mildew</i></p>	 <p>Menekan tombol bertuliskan “Baca selengkapnya”</p>	 <p>Menampilkan detail penyakit daun <i>Downy Mildew</i></p>	Valid
--	--	--	---	---	-------

Berdasarkan pada Tabel 6.1 didapatkan 6 buah *Test Case* yang merupakan kebutuhan fungsional sistem dan dibagi menjadi 18 *Target* yang merupakan *scenario* saat kebutuhan dijalankan. Dari 18 *scenario* tersebut memiliki status valid.

6.2 Pengujian Akurasi

Pengujian akurasi adalah jenis pengujian yang digunakan untuk mengukur seberapa akurat hasil dalam mengidentifikasi penyakit daun jeruk dan tingkat keparahan daun jeruk dari aplikasi ini. Pengujian ini dilakukan dengan mencoba aplikasi secara langsung dengan menggunakan data latih dan data uji yang ada. Untuk data latih diujikan sebanyak 120 gambar daun jeruk yang terdiri 30 gambar daun dari tiap kategori penyakit yang diambil dengan cara dipetik lalu diambil gambarnya dengan pencahayaan yang merata. Sedangkan untuk data uji diujikan sebanyak 60 gambar daun yang ada pada Kebun Jeruk Tlekung Kota Batu yang diambil secara acak dan diambil langsung gambarnya tanpa dipetik.

Adapun perhitungan akurasi pada pengujian ini ditunjukan pada Persamaan 6.1 dibawah ini.

$$Akurasi = \frac{\text{Jumlah pendekatan valid}}{\text{Jumlah pendekatan total}} \times 100\% \quad (6.1)$$

Pendeteksian penyakit dinyatakan valid jika *target output* dan hasil klasifikasi penyakit bernilai sama. Sedangkan tingkat keparahan dinyatakan valid jika daerah berpenyakit sesuai dengan hasil klasifikasi. Proses validitas pada pengujian ini dilakukan oleh Peneliti Hama dan Penyakit Tanaman yang ada di Balitjestro.

6.2.1 Hasil Pengujian Akurasi

Hasil pengujian ini dibagi menjadi 2 yaitu hasil pengujian akurasi dengan menggunakan data latih dan hasil pengujian akurasi dengan menggunakan data uji.

6.2.1.1 Hasil Pengujian Akurasi Menggunakan Data Latih

Pengujian akurasi pada bagian ini menggunakan gambar yang ada pada data latih sejumlah 120 gambar daun yang terdiri dari 30 gambar daun dari 4 kategori daun. Proses pengambilan data daun menggunakan kamera OPPO A37f 8 MP dan menggunakan alat bantu *background cover* warna merah dengan pencahayaan yang merata dan daun yang diambil dipetik terlebih dahulu dari pohonnya. Hasil pengujian akurasi dapat dilihat pada Tabel 6.2.

Tabel 6.2 Hasil Pengujian Akurasi Data Latih

No	Citra Input	Target Ouput	Hasl Pendektsian	Status Pendektsian	Status Tingkat Keparahan
1	Data latih - 1	Daun Sehat	Daun Sehat	Valid	Valid
2	Data latih - 2	Daun Sehat	Daun Sehat	Valid	Valid
3	Data latih - 3	Daun Sehat	Daun Sehat	Valid	Valid
4	Data latih - 4	Daun Sehat	Daun Sehat	Valid	Valid
5	Data latih - 5	Daun Sehat	Daun Sehat	Valid	Valid
6	Data latih - 6	Daun Sehat	Daun Sehat	Valid	Valid
7	Data latih - 7	Daun Sehat	Daun Sehat	Valid	Valid
8	Data latih - 8	Daun Sehat	Daun Sehat	Valid	Valid
9	Data latih - 9	Daun Sehat	Daun Sehat	Valid	Valid
10	Data latih - 10	Daun Sehat	Daun Sehat	Valid	Valid
11	Data latih - 11	Daun Sehat	Daun Sehat	Valid	Valid
12	Data latih - 12	Daun Sehat	Daun Sehat	Valid	Valid
13	Data latih - 13	Daun Sehat	Daun Sehat	Valid	Valid
14	Data latih - 14	Daun Sehat	Daun Sehat	Valid	Valid
15	Data latih - 15	Daun Sehat	Daun Sehat	Valid	Valid
16	Data latih - 16	Daun Sehat	Daun Sehat	Valid	Valid
17	Data latih - 17	Daun Sehat	Daun Sehat	Valid	Valid
18	Data latih - 18	Daun Sehat	Daun Sehat	Valid	Valid
19	Data latih - 19	Daun Sehat	Daun Sehat	Valid	Valid
20	Data latih - 20	Daun Sehat	Daun Sehat	Valid	Valid
21	Data latih - 21	Daun Sehat	Daun Sehat	Valid	Valid
22	Data latih - 22	Daun Sehat	Daun Sehat	Valid	Valid
23	Data latih - 23	Daun Sehat	Daun Sehat	Valid	Valid
24	Data latih - 24	Daun Sehat	Daun Sehat	Valid	Valid
25	Data latih - 25	Daun Sehat	Daun Sehat	Valid	Valid
26	Data latih - 26	Daun Sehat	Daun Sehat	Valid	Valid
27	Data latih - 27	Daun Sehat	Daun Sehat	Valid	Valid
28	Data latih - 28	Daun Sehat	Daun Sehat	Valid	Valid
29	Data latih - 29	Daun Sehat	Daun Sehat	Valid	Valid
30	Data latih - 30	Daun Sehat	Daun Sehat	Valid	Valid
31	Data latih - 31	CVPD	CVPD	Valid	Valid
32	Data latih - 32	CVPD	CVPD	Valid	Tidak Valid

33	Data latih - 33	CVPD	CVPD	Valid	Valid
34	Data latih - 34	CVPD	CVPD	Valid	Valid
35	Data latih - 35	CVPD	CVPD	Valid	Valid
36	Data latih - 36	CVPD	CVPD	Valid	Valid
37	Data latih - 37	CVPD	CVPD	Valid	Valid
38	Data latih - 38	CVPD	CVPD	Valid	Valid
39	Data latih - 39	CVPD	CVPD	Valid	Valid
40	Data latih - 40	CVPD	CVPD	Valid	Valid
41	Data latih - 41	CVPD	CVPD	Valid	Valid
42	Data latih - 42	CVPD	CVPD	Valid	Valid
43	Data latih - 43	CVPD	CVPD	Valid	Valid
44	Data latih - 44	CVPD	CVPD	Valid	Valid
45	Data latih - 45	CVPD	CVPD	Valid	Valid
46	Data latih - 46	CVPD	CVPD	Valid	Valid
47	Data latih - 47	CVPD	CVPD	Valid	Valid
48	Data latih - 48	CVPD	CVPD	Valid	Valid
49	Data latih - 49	CVPD	CVPD	Valid	Valid
50	Data latih - 50	CVPD	CVPD	Valid	Valid
51	Data latih - 51	CVPD	CVPD	Valid	Valid
52	Data latih - 52	CVPD	CVPD	Valid	Valid
53	Data latih - 53	CVPD	CVPD	Valid	Valid
54	Data latih - 54	CVPD	CVPD	Valid	Valid
55	Data latih - 55	CVPD	CVPD	Valid	Valid
56	Data latih - 56	CVPD	CVPD	Valid	Valid
57	Data latih - 57	CVPD	CVPD	Valid	Valid
58	Data latih - 58	CVPD	CVPD	Valid	Valid
59	Data latih - 59	CVPD	CVPD	Valid	Valid
60	Data latih - 60	CVPD	CVPD	Valid	Valid
61	Data latih - 61	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
62	Data latih - 62	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
63	Data latih - 63	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
64	Data latih - 64	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
65	Data latih - 65	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
66	Data latih - 66	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
67	Data latih - 67	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
68	Data latih - 68	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid

69	Data latih - 69	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
70	Data latih - 70	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
71	Data latih - 71	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
72	Data latih - 72	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
73	Data latih - 73	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
74	Data latih - 74	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
75	Data latih - 75	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
76	Data latih - 76	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
77	Data latih - 77	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
78	Data latih - 78	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
79	Data latih - 79	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
80	Data latih - 80	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
81	Data latih - 81	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
82	Data latih - 82	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
83	Data latih - 83	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
84	Data latih - 84	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
85	Data latih - 85	Cendawan Jelaga	Cendawan Jelaga	Valid	Tidak Valid
86	Data latih - 86	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
87	Data latih - 87	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
88	Data latih - 88	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
89	Data latih - 89	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
90	Data latih - 90	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid

91	Data latih - 91	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
92	Data latih - 92	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
93	Data latih - 93	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
94	Data latih - 94	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
95	Data latih - 95	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
96	Data latih - 96	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
97	Data latih - 97	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
98	Data latih - 98	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
99	Data latih - 99	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
100	Data latih - 100	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
101	Data latih - 101	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
102	Data latih - 102	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
103	Data latih - 103	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
104	Data latih - 104	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
105	Data latih - 105	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
106	Data latih - 106	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
107	Data latih - 107	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
108	Data latih - 108	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
109	Data latih - 109	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
110	Data latih - 110	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
111	Data latih - 111	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
112	Data latih - 112	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid

113	Data latih - 113	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
114	Data latih - 114	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
115	Data latih - 115	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
116	Data latih - 116	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
117	Data latih - 117	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
118	Data latih - 118	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
119	Data latih - 119	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
120	Data latih - 120	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
Total Akhir				120	118

Pada Tabel 6.2 merupakan hasil akurasi pendekripsi penyakit dan tingkat keparahan menggunakan data latih. Jumlah status valid pada pendekripsi penyakit berjumlah 120 sedangkan jumlah status valid pada tingkat keparahan berjumlah 118. Terdapat 2 data latih dengan status tingkat keparahan Tidak Valid dikarenakan penentuan tingkat keparahan pada gambar daun tidak sesuai. Sehingga dapat disimpulkan bahwa aplikasi dapat mengidentifikasi penyakit pada daun jeruk secara akurat dengan akurasi 100%. Namun pada pendekripsi tingkat keparahan aplikasi memiliki akurasi 97,5%.

6.2.1.2 Hasil Pengujian Akurasi Menggunakan Data Uji

Pengujian akurasi pada bagian ini menggunakan gambar yang ada pada data uji sejumlah 60 gambar daun yang diambil secara acak di Kebun Jeruk Tlekung Kota Batu. Proses pengambilan data daun menggunakan kamera OPPO A37f 8 MP dan menggunakan alat bantu *background cover* warna merah dengan pencahayaan yang merata dan daun yang diambil tanpa dipotong terlebih dahulu dari pohonnya. Hasil pengujian akurasi dapat dilihat pada Tabel 6.3.

Tabel 6.3 Hasil Pengujian Akurasi Data Uji

No	Citra Input	Target Ouput	Hasl Pendekripsi	Status Pendekripsi	Status Tingkat Keparahan
1	Data uji - 1	CVPD	CVPD	Valid	Valid
2	Data uji - 2	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
3	Data uji - 3	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
4	Data uji - 4	Daun Sehat	Daun Sehat	Valid	Valid

5	Data uji - 5	Cendawan Jelaga	Cendawan Jelaga	Valid	Tidak Valid
6	Data uji - 6	Cendawan Jelaga	Cendawan Jelaga	Valid	Tidak Valid
7	Data uji - 7	Cendawan Jelaga	Cendawan Jelaga	Valid	Tidak Valid
8	Data uji - 8	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
9	Data uji - 9	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
10	Data uji - 10	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
11	Data uji - 11	Cendawan Jelaga	Cendawan Jelaga	Valid	Tidak Valid
12	Data uji - 12	Cendawan Jelaga	Cendawan Jelaga	Valid	Valid
13	Data uji - 13	Cendawan Jelaga	Cendawan Jelaga	Valid	Tidak Valid
14	Data uji - 14	Cendawan Jelaga	Cendawan Jelaga	Valid	Tidak Valid
15	Data uji - 15	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
16	Data uji - 16	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
17	Data uji - 17	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
18	Data uji - 18	Daun Sehat	Daun Sehat	Valid	Valid
19	Data uji - 19	Daun Sehat	Daun Sehat	Valid	Valid
20	Data uji - 20	Daun Sehat	Daun Sehat	Valid	Valid
21	Data uji - 21	Daun Sehat	Daun Sehat	Valid	Valid
22	Data uji - 22	Daun Sehat	Daun Sehat	Valid	Valid
23	Data uji - 23	Daun Sehat	Daun Sehat	Valid	Valid
24	Data uji - 24	Daun Sehat	Daun Sehat	Valid	Valid
25	Data uji - 25	Daun Sehat	Daun Sehat	Valid	Valid
26	Data uji - 26	Daun Sehat	Daun Sehat	Valid	Valid
27	Data uji - 27	Daun Sehat	Daun Sehat	Valid	Valid
28	Data uji - 28	Daun Sehat	Daun Sehat	Valid	Valid
29	Data uji - 29	Daun Sehat	Daun Sehat	Valid	Valid
30	Data uji - 30	Daun Sehat	Daun Sehat	Valid	Valid
31	Data uji - 31	Daun Sehat	Daun Sehat	Valid	Valid
32	Data uji - 32	Daun Sehat	Daun Sehat	Valid	Valid
33	Data uji - 33	Daun Sehat	Daun Sehat	Valid	Valid
34	Data uji - 34	Daun Sehat	Daun Sehat	Valid	Valid

35	Data uji - 35	Cendawan Jelaga	Cendawan Jelaga	Valid	Tidak Valid
36	Data uji - 36	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
37	Data uji - 37	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
38	Data uji - 38	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
39	Data uji - 39	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Tidak Valid
40	Data uji - 40	CVPD	CVPD	Valid	Tidak Valid
41	Data uji - 41	CVPD	CVPD	Valid	Valid
42	Data uji - 42	CVPD	CVPD	Valid	Tidak Valid
43	Data uji - 43	CVPD	CVPD	Valid	Tidak Valid
44	Data uji - 44	CVPD	CVPD	Valid	Tidak Valid
45	Data uji - 45	CVPD	CVPD	Valid	Tidak Valid
46	Data uji - 46	CVPD	CVPD	Valid	Valid
47	Data uji - 47	CVPD	CVPD	Valid	Valid
48	Data uji - 48	CVPD	CVPD	Valid	Tidak Valid
49	Data uji - 49	CVPD	CVPD	Valid	Tidak Valid
50	Data uji - 50	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Tidak Valid
51	Data uji - 51	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
52	Data uji - 52	Cendawan Jelaga	Cendawan Jelaga	Valid	Tidak Valid
53	Data uji - 53	Cendawan Jelaga	Cendawan Jelaga	Valid	Tidak Valid
54	Data uji - 54	Cendawan Jelaga	Cendawan Jelaga	Valid	Tidak Valid
55	Data uji - 55	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Tidak Valid
56	Data uji - 56	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
57	Data uji - 57	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
58	Data uji - 58	Daun Sehat	Daun Sehat	Valid	Valid
59	Data uji - 59	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Tidak Valid
60	Data uji - 60	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Valid	Valid
Total Akhir				60	39

Pada Tabel 6.3 merupakan hasil akurasi pendekripsi penyakit dan tingkat keparahan menggunakan data uji. Jumlah status valid pada pendekripsi penyakit berjumlah 60 sedangkan jumlah status valid pada tingkat keparahan berjumlah 39. Terdapat 21 data latih dengan status tingkat keparahan Tidak Valid dikarenakan penentuan tingkat keparahan pada gambar daun tidak sesuai. Hal ini bisa dikarenakan pencahayaan yang tidak merata pada saat pengambilan gambar dan bisa dikarenakan kualitas gambar yang kurang baik. Berdasarkan hasil pada Tabel 6.3 dapat disimpulkan bahwa aplikasi dapat mengidentifikasi penyakit pada daun jeruk secara akurat dengan akurasi 100%. Namun, pada perhitungan tingkat keparahan aplikasi memiliki akurasi 65%.

6.3 Pengujian Usabilitas

Pengujian usabilitas adalah jenis pengujian dengan melihat cara pengguna berinteraksi dengan sistem yang telah dibangun. Pengujian ini memerlukan sejumlah responden untuk menguji aplikasi secara langsung guna mengetahui apakah aplikasi yang telah dibangun mudah digunakan atau tidak. Pada pengujian ini terdapat *task scenario* pengujian dimana responden diminta untuk melakukan proses-proses untuk menjalankan aplikasi. Pada pengujian ini terdapat 7 *task scenario* yang harus dilakukan responden. Adapun *task scenario* tersebut ditunjukkan pada Tabel 6.4.

Tabel 6.4 Task Scenario Pengujian

No	<i>Task Scenario</i>
1	Mulailah aplikasi dengan mengambil gambar melalui kamera atau galeri
2	Jika gambar sudah selesai dipilih, maka lakukanlah pemotongan gambar.
3	Lakukanlah pendekripsi gambar dengan menekan <i>icon</i> centang
4	Setelah aplikasi mendapatkan hasil pendekripsi, lihatlah informasi cara pengendaliannya.
5	Simpanlah gambar pendekripsi ke dalam daftar riwayat
6	Jika daftar pendekripsi telah muncul, hapuslah salah satu riwayat pendekripsi dan kembali ke menu utama
7	Setelah berada di menu utama lihatlah daftar penyakit daun jeruk

Pada Tabel 6.4 merupakan 7 buah *task scenario* yang harus dilakukan responden secara berurutan. Pada tahap ini juga responden diminta untuk mengungkapkan apa saja yang dialami dan dirasakan responden secara langsung (bukan bertanya) kepada moderator dalam menjalankan 7 buah *task scenario* yang ada.

Kemudian responden diminta untuk mengisikan kuesioner sebagai penilaian tentang usabilitas pada aplikasi ini. Pengujian pada penelitian ini menggunakan kuesioner SUPR-Qm yang terdiri 16 buah pertanyaan yang telah didefinisikan pada bab landasan kepustakaan. Adapun 16 buah pertanyaan tersebut ditunjukkan pada

Tabel 6.6. Kuesioner ini nantinya diisikan oleh responden dimana setiap pertanyaan menggunakan skala likert dalam rentang 1 sampai 5 yang dijelaskan pada Tabel 6.5.

Tabel 6.5 Skor Skala Likert dari Setiap Pertanyaan

Skor	Keterangan
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Netral
4	Setuju
5	Sangat Setuju

Tabel 6.6 Kuesioner SUPR-Qm

No	Pertanyaan
1	Saya merasa bahwa aplikasi ini penting untuk saya
2	Saya merasa bahwa aplikasi ini merupakan aplikasi Identifikasi Penyakit Daun Jeruk terbaik yang pernah saya gunakan
3	Saya tidak bisa membayangkan ada aplikasi yang sebaik ini
4	Saya merasa bahwa saya tidak akan menghapus aplikasi ini dari <i>smartphone</i> saya
5	Saya merasa bahwa seluruh petani jeruk harus menggunakan aplikasi Identifikasi Penyakit Daun Jeruk ini
6	Saya merasa bahwa saya tertarik untuk menggunakan fitur-fitur yang ada pada aplikasi ini
7	Saya berpikir bahwa aplikasi Identifikasi Penyakit Daun Jeruk ini memiliki fitur-fitur yang saya inginkan
8	Saya berpikir bahwa saya akan sering menggunakan aplikasi ini pada saat mendeteksi penyakit daun jeruk
9	Saya berpikir aplikasi ini sangat menyenangkan
10	Saya berpikir bahwa aplikasi ini terintegrasi dengan baik dengan fitur-fitur lain dari ponsel saya (misalnya membuka kamera dan galeri)
11	Saya merasa bahwa saya mungkin akan menggunakan aplikasi Identifikasi Penyakit Daun Jeruk ini di masa yang akan datang
12	Saya merasa bahwa desain tampilan dari aplikasi Identifikasi Penyakit Daun Jeruk ini dapat memudahkan saya mendapatkan informasi yang saya inginkan

13	Saya merasa bahwa aplikasi ini sangat menarik
14	Saya merasa bahwa aplikasi ini sesuai dengan apa yang saya inginkan
15	Saya merasa bahwa aplikasi ini memiliki panduan yang lengkap untuk menjalankan setiap fiturnya
16	Saya merasa bahwa aplikasi ini sangat mudah digunakan

Pada Tabel 6.6 merupakan 16 buah pertanyaan dengan menggunakan kuesioner SUPR-Qm yang nantinya didapatkan hasil secara kuantitatif terhadap kemudahan penggunaan aplikasi yang dikembangkan.

Adapun rumus perhitungan nilai kuesioner SUPR-Qm ialah sama dengan rumus perhitungan nilai SUPR-Q yang ditunjukkan pada Persamaan 6.2

$$\text{Nilai SUPR} - Q = \frac{\text{Jumlah Nilai Diperoleh}}{\text{Jumlah Nilai Maksimal}} \times 100\% \quad (6.2)$$

Pada Persamaan 6.2 adalah rumus perhitungan nilai SUPR-Q yang digunakan juga dalam perhitungan nilai SUPR-Qm. Berdasarkan nilai SUPR-Qm tersebut nantinya konversikan ke dalam kategori penilaian usability.

6.3.1 Hasil Pengujian Usabilitas

Pada pengujian usabilitas pada aplikasi ini dilakukan dengan metode *Individual Interview* kepada 5 orang responden yang berasal dari Balai Penelitian Tanaman Jeruk dan Buah Subtropika (Balitjestro) yang merupakan calon pengguna aplikasi identifikasi penyakit daun jeruk ini. Adapun 5 orang responden tersebut dijelaskan pada Tabel 6.7.

Tabel 6.7 Responden Pengujian Usabilitas

No	Nama	Pekerjaan
1	Anang Triwiratno	Peneliti Madya di bidang Hama dan Penyakit Tanaman
2	Sri Widyaningsih	Peneliti Muda
3	Rudi Cahyo Wicaksono	Ahli Peneliti Pertama
4	Dwi Agung	Kepala Kebun Kliran
5	Ady Cahyono	Kepala Kebun Banaran

Pada Tabel 6.7 merupakan para responden yang membantu dalam pengujian usabilitas pada aplikasi ini yang terdiri dari 3 orang peneliti dan 2 orang kepala kebun.

Lalu responden diminta untuk mengoperasikan aplikasi berdasarkan *task scenario* yang dijelaskan pada Tabel 6.4 dan moderator mengamati perilaku responden, apakah responden dapat menyelesaikan *task scenario* yang diberikan atau tidak. Adapun hasil dari pengamatan tersebut dijelaskan pada Tabel 6.8.

Tabel 6.8 Task Completion Rate

	User 1	User 2	User 3	User 4	User 5	Completion Rate
Task 1	✓	✓	✓	✓	✓	100%
Task 2	✓	✓	✓	✓	✓	100%
Task 3	✓	✓	✓	✓	✓	100%
Task 4	✓	✓	✓	✓	✓	100%
Task 5	✓	✓	✓	✓	✓	100%
Task 6	✓	✓	✓	✓	✓	100%
Task 7	✓	✓	✓	✓	✓	100%

Pada Tabel 6.8 dapat disimpulkan bahwa semua responden dapat menyelesaikan ketujuh *task scenario* yang diberikan. Hal ini ditunjukkan pada kolom *Completion Rate* yang menunjukkan hasil 100% dari semua *task scenario*.

Setelah responden menyelesaikan *task scenario* yang diberikan, selanjutnya responden diminta untuk mengisi kuesioner SUPR-Qm yang telah didefinisikan sebelumnya. Adapun hasil pengujian kuisioner SUPR-Qm ditunjukkan pada Tabel 6.9.

Tabel 6.9 Hasil Pengujian Kuisioner SUPR-Qm

No	Pertanyaan	Skor					Total Skor
		1	2	3	4	5	
1	Saya merasa bahwa aplikasi ini penting untuk saya	0	1	2	1	1	17
2	Saya merasa bahwa aplikasi ini merupakan aplikasi Identifikasi Penyakit Daun Jeruk terbaik yang pernah saya gunakan	0	1	2	2	0	17
3	Saya tidak bisa membayangkan ada aplikasi yang sebaik ini	0	0	3	2	0	17
4	Saya merasa bahwa saya tidak akan menghapus aplikasi ini dari smartphone saya	0	1	1	2	1	18
5	Saya merasa bahwa seluruh petani jeruk harus menggunakan aplikasi Identifikasi Penyakit Daun Jeruk ini	0	1	1	2	1	18
6	Saya merasa bahwa saya tertarik untuk menggunakan fitur-fitur yang ada pada aplikasi ini	0	0	3	2	0	17

7	Saya berpikir bahwa aplikasi Identifikasi Penyakit Daun Jeruk ini memiliki fitur-fitur yang saya inginkan	0	2	0	2	1	17
8	Saya berpikir bahwa saya akan sering menggunakan aplikasi ini pada saat mendeteksi penyakit daun jeruk	0	1	1	3	0	17
9	Saya berpikir aplikasi ini sangat menyenangkan	0	0	4	0	1	17
10	Saya berpikir bahwa aplikasi ini terintegrasi dengan baik dengan fitur-fitur lain dari ponsel saya (misalnya membuka kamera dan galeri)	0	0	4	1	0	16
11	Saya merasa bahwa saya mungkin akan menggunakan aplikasi Identifikasi Penyakit Daun Jeruk ini di masa yang akan datang	0	0	2	3	0	18
12	Saya merasa bahwa desain tampilan dari aplikasi Identifikasi Penyakit Daun Jeruk ini dapat memudahkan saya mendapatkan informasi yang saya inginkan	0	0	2	3	0	18
13	Saya merasa bahwa aplikasi ini sangat menarik	0	0	2	2	1	19
14	Saya merasa bahwa aplikasi ini sesuai dengan apa yang saya inginkan	0	0	3	1	1	18
15	Saya merasa bahwa aplikasi ini memiliki panduan yang lengkap untuk menjalankan setiap fiturnya	0	0	1	3	1	20
16	Saya merasa bahwa aplikasi ini sangat mudah digunakan	0	0	0	2	3	23
Total Akhir							286
Total Nilai Maksimum							400
Nilai SUPR-Qm							71.5%

Tabel 6.9 menunjukkan penilaian tertinggi ditunjukkan oleh *task scenario* nomor 16 yaitu dengan skor 23, dan penilaian terendah ditunjukkan oleh *scenario* nomor 10 yaitu dengan skor 16. Pada tabel juga menunjukkan hasil dari pengujian dengan total skor akhir berjumlah 286 dan untuk total nilai maksimumnya berjumlah 400. Dengan menggunakan rumus perhitungan nilai SUPR-Qm didapatkan nilai SUPR-Qm yaitu 71.5%.

Kemudian responden juga diminta untuk memberikan saran dan umpan balik untuk aplikasi ini yang ditunjukan pada Tabel 6.10.

Tabel 6.10 Saran dan Umpan Balik Responden

Responden	Saran dan Umpan Balik
1	Aplikasi ini sangat mudah digunakan oleh pengguna dari berbagai kalangan termasuk petani awam asalkan punya <i>smartphone</i>
2	Untuk penyakit CVPD bisa rancu dengan gejala penyakit yang lain atau defisiensi unsur hara sehingga harusnya jangan hanya bagian daun yang digunakan tetapi bagian tanaman yang lain
3	Perlu dilengkapi untuk jenis penyakit yang lain
4	Ruang lingkup aplikasi sebaiknya ditambah dengan jenis penyakit lain, aplikasi sebaiknya bisa dikontrol oleh <i>programmer</i> agar tidak terjadi hal yang tidak diinginkan, menambahkan fitur mengetahui jumlah pengguna dan lokasi penggunanya
5	Aplikasi ini sangat membantu para petani , namun pendekripsi jenis penyakitnya perlu ditambah lagi

6.4 Analisis Hasil Pengujian

Analisis hasil pengujian pada aplikasi ini dibagi menjadi 3 pengujian yaitu pengujian fungsional, pengujian akurasi dan pengujian usabilitas.

6.4.1 Analisis Hasil Pengujian Fungsional

Hasil dari pengujian fungsional pada aplikasi ini ditunjukkan pada Tabel 6.1 yang terdapat 6 buah *Test Case* yang merupakan kebutuhan fungsional sistem dan dibagi menjadi 18 *Target* yang merupakan *scenario* saat kebutuhan dijalankan. Dari 18 *Target* tersebut memiliki status valid semua, yang berarti sistem dapat beroperasi dengan baik.

6.4.2 Analisis Hasil Pengujian Akurasi

Hasil dari pengujian akurasi aplikasi ini dibagi menjadi 2 bagian yaitu pengujian akurasi menggunakan data latih dan data uji. Berdasarkan hasil pengujian didapatkan persentase 100% untuk pengujian akurasi pendekripsi penyakit menggunakan data latih dan data uji. Sedangkan untuk pengujian akurasi perhitungan tingkat keparahan didapatkan persentase 97,5% menggunakan data latih dan 65% menggunakan data uji. Sehingga dapat disimpulkan bahwa hasil akurasi aplikasi untuk mendekripsi penyakit daun jeruk memiliki akurasi yang sangat akurat dengan persentase 100% dan untuk akurasi perhitungan tingkat keparahan memiliki rata-rata akurasi 81,25% yang dapat dibilang cukup baik. Namun hasil tersebut masih bergantung pada jenis kualitas gambarnya itu sendiri. Sehingga dapat disimpulkan bahwa aplikasi ini dapat menghasilkan akurasi yang baik jika gambar yang diambil memiliki kualitas yang baik juga.

6.4.3 Analisis Hasil Pengujian Usabilitas

Hasil dari pengujian usabilitas pada aplikasi ini didapatkan hasil nilai *usability* dari SUPR-Qm yaitu 71.5%. Nilai tersebut apabila dikonversikan ke dalam skala konversi kategori nilai *usability* maka akan masuk ke dalam kategori nilai C yaitu bernilai baik. Sehingga dapat disimpulkan aplikasi Identifikasi Penyakit Daun Jeruk ini dinilai baik dan dapat diterima oleh para peneliti tanaman jeruk dan petani jeruk di Balai Penelitian Tanaman Jeruk dan Buah Subtropika (Balitjestro).

BAB 7 PENUTUP

Pada bagian ini berisikan kesimpulan dan saran terhadap penelitian pengembangan aplikasi perangkat bergerak identifikasi penyakit daun jeruk berbasis android dengan memanfaatkan Vize AI.

7.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, perancangan, implementasi dan pengujian pada penelitian pengembangan aplikasi perangkat bergerak identifikasi penyakit daun jeruk berbasis android dengan memanfaatkan Vize AI didapatkan 3 kesimpulan, antara lain :

1. Berdasarkan proses penggalian kebutuhan dengan cara dilakukan wawancara kepada Peneliti Madya di bidang Hama dan Penyakit Tanaman Jeruk serta melakukan observasi pada Balai Penelitian Tanaman Jeruk dan Buah Subtropika (Balitjestro) yang terletak di daerah Junrejo Kota Batu didapatkan 6 buah kebutuhan fungsional dan 1 buah kebutuhan non-fungsionalitas. Kebutuhan fungsional tersebut antara lain mengambil gambar, *crop* gambar, mendeteksi penyakit dan tingkat keparahannya, mengetahui informasi cacra pengendalian, mengetahui riwayat gambar yang telah dideteksi dan mengetahui informasi penyakit daun jeruk. Dan terdapat 1 buah kebutuhan non-fungsionalitas yaitu *Usability* yang merupakan kemuadian dalam penggunaan aplikasi.
2. Untuk melakukan implementasi Vize AI sebagai *web service* untuk mengidentifikasi penyakit pada daun jeruk diperlukan 3 cara yaitu *define*, *train* dan *recognize*. Pada tahap *define* dilakukan pengelompokan ke dalam 4 kategori penyakit antara lain Cendawan Jelaga, *Downy Mildew*, CVPD dan Daun Sehat. Lalu dari tiap katergori dimasukkan gambar daun sejumlah 30 gambar sesuai dengan penyakitnya yang nantinya digunakan sebagai data latih untuk proses pengenalan gambar. Pada tahap *train* dilakukan proses pelatihan untuk melatih Vize AI mengenali gambar penyakit daun jeruk. Pada tahap *recognize* dilakukan proses pengenalan gambar dengan cara mengirimkan gambar daun ke *web service* Vize AI. Setelah dilakukan proses pengenalan gambar maka hasil yang keluar yaitu persentase dari semua jenis penyakit serta menampilkan hasil persentase terbaik dari jenis penyakit yang menyerang daunnya dan Vize AI juga mengirimkan hasil berupa JSON yang nantinya data JSON tersebut dapat diolah ke dalam aplikasi yang berbeda.
3. Hasil dari pengujian akurasi aplikasi ini dibagi menjadi 2 bagian yaitu pengujian akurasi menggunakan data latih dan data uji. Berdasarkan hasil pengujian didapatkan persentase 100% untuk pengujian akurasi pendekripsi penyakit menggunakan data latih dan data uji. Sedangkan untuk pengujian akurasi perhitungan tingkat keparahan didapatkan persentase 97,5% menggunakan data latih dan 65% menggunakan data uji. Sehingga dapat disimpulkan bahwa hasil akurasi aplikasi untuk mendekripsi penyakit daun

jeruk memiliki akurasi yang sangat akurat dengan persentase 100% dan untuk akurasi perhitungan tingkat keparahan memiliki rata-rata akurasi 81,25% yang dapat dibilang cukup baik. Namun hasil tersebut masih bergantung pada jenis kualitas gambarnya itu sendiri. Sehingga dapat disimpulkan bahwa aplikasi ini dapat menghasilkan akurasi yang baik jika gambar yang diambil memiliki kualitas yang baik juga.

7.2 Saran

Berdasarkan penelitian yang telah dilakukankan, peneliti merasa bahwa penelitian ini belum bisa dikatakan sempurna. Untuk itu, terdapat beberapa saran yang dapat dikembangkan untuk penelitian selanjutnya, antara lain :

1. Perlu ditambahkan jenis penyakit yang lain selain dari penyakit CVPD, Cendawan Jelaga dan *Downy Mildew*.
2. Perlu dilakukan penelitian lebih lanjut untuk mendeteksi penyakit tanaman jeruk tidak hanya dari gambar daunnya saja.
3. Perlu diujikan lebih lanjut terkait kualitas gambar yang baik yang dapat terdeteksi secara maksimal oleh aplikasi.
4. Pelu dilakukan penelitian lebih lanjut untuk mengatur pencahayaan pada saat pengambilan gambar.

DAFTAR PUSTAKA

- Balitjestro, 2017. *Laporan Kinerja*. Malang: Balai Penelitian Tanaman Jeruk dan Buah Subtropika.
- Balitjestro, 2018. *Balitjestro - Kementrian Pertanian*. [Online] Tersedia di: <http://balitjestro.litbang.pertanian.go.id/wp-content/uploads/2017/08/batu-55.jpg> [Diakses 27 Agustus 2018].
- Bangor, A., Kortum, P. & Miller, J., 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale.
- Developers, G., 2018. *About the platform*. [Online] Tersedia di: <https://developer.android.com/about/> [Diakses 26 Agustus 2018].
- Febrinanto, F. G., Dewi, C. & Triwiratno, A., 2018. Implementasi Algoritme K-Means Sebagai Metode Segmentasi Citra Dalam Identifikasi Penyakit Daun Jeruk.
- Kementan, 2017. *Statistik Pertanian*. s.l.: Pusat Data dan Sistem Informasi Pertanian Kementerian Pertanian Republik Indonesia.
- MeasuringU, 2018. *SUPR-Q Full License*. [Online] Tersedia di: <https://measuringu.com/product/suprq/> [Diakses 06 Desember 2018].
- Meitayani, N. P. S., Adiartayasa, W. & Wijaya, I. N., 2014. Deteksi Penyakit Citrus Vein Phloem Degeneration (CPVD) dengan Teknik Polymerase Chain Reaction (PCR) pada Tanaman Jeruk di Bali. *E-Jurnal Agroekoteknologi*, 3(2), pp. 70-79.
- Nielsen, J., 2004. *Card Sorting: How Many Users to Test*. [Online] Tersedia di: <https://www.nngroup.com/articles/card-sorting-how-many-users-to-test/> [Diakses 17 Oktober 2018].
- Point, T., 2018. *SDLC - Waterfall Model*. [Online] Tersedia di: https://www.tutorialspoint.com/sdlc/images/sdlc_waterfall_model.jpg [Diakses 15 Oktober 2018].
- Priambodo, A., Dewi, C. & Triwiratno, A., 2015. Implementasi Metode K-Nearest Neighbour Untuk Identifikasi Penyakit Tanaman Jeruk Keprok Berdasarkan Citra Daun.
- Rizal, A. Y., Dewi, C. & Widodo, A. W., 2017. Implementasi Algoritma Multilevel Thresholding Menggunakan Otsu Sebagai Preprocessing Data Citra Daun Pada Proses Identifikasi Penyakit Tanaman Jeruk.
- Rizal, P., Bachrian, W. & Retno, 2011. *Budidaya Jeruk Bebas Penyakit*. Kalimantan Timur: Balai Pengkajian Teknologi Pertanian.

- Sauro, J. & Zarolia, P., 2017. SUPR-Qm: A Questionnaire to Measure the Mobile App User Experience.
- Setiono, 2015. *Inovasi Jeruk Keprok Batu* 55. Jakarta: IAARD Press.
- Sommerville, I., 2011. *Software Engineering , 7th Edition*. United State of America: Addison-Wesley Publishing Company Inc.
- SQLite, 2018. *About SQLite*. [Online] Tersedia di: <https://www.sqlite.org/about.html> [Diakses 26 Oktober 2018].
- Sukamto, R. A. & Shalahuddin, M., 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung : Informatika.
- Syafril, 2006. *Jenis Hama dan Penyakit Penting Menyerang Jeruk Koto Tinggi*. Padang: Balai Pengkajian Teknologi Pertanian Sumatra Barat.
- Techinasia, 2018. *Jumlah Pengguna Smartphone 2018*. [Online] Tersedia di: <https://d26bwijyd9l0e3m.cloudfront.net/wp-content/uploads/2014/12/Indonesia-to-be-worlds-fourth-largest-smartphone-by-2018-surpass-100-million-users-chart1.jpg> [Diakses 26 September 2018].
- Triwiratno, A., 2018. *Macam dan Cara Mengidentifikasi Penyakit Daun Jeruk* [Wawancara] (26 Agustus 2018).
- Vize, 2018. *Custom Image Recognition API*. [Online] Tersedia di: <https://vize.ai/i/u-custom5.jpg> [Diakses 03 Oktober 2018].
- Vize, 2018. *Recognize and automate your images*. [Online] Tersedia di: <https://vize.ai> [Diakses 26 Agustus 2018].
- Wahyuningsih, E., 2009. CVPD pada Jeruk (*Citrus spp*) dan Upaya Pengendaliannya. *Vis Vitalis*, 2(2), pp. 65-73.

LAMPIRAN A DATA LATIH GAMBAR DAUN JERUK

No	Gambar Daun	Jenis Penyakit
1		Daun Sehat
2		Daun Sehat
3		Daun Sehat
4		Daun Sehat
5		Daun Sehat

6		Daun Sehat
7		Daun Sehat
8		Daun Sehat
9		Daun Sehat
10		Daun Sehat

11		Daun Sehat
12		Daun Sehat
13		Daun Sehat
14		Daun Sehat
15		Daun Sehat

16		Daun Sehat
17		Daun Sehat
18		Daun Sehat
19		Daun Sehat
20		Daun Sehat

21				Daun Sehat
22				Daun Sehat
23				Daun Sehat
24				Daun Sehat
25				Daun Sehat

26		Daun Sehat
27		Daun Sehat
28		Daun Sehat
29		Daun Sehat
30		Daun Sehat

31		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
32		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
33		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
34		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
35		<i>Citrus Vein Phloem Degeneration (CVPD)</i>

36		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
37		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
38		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
39		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
40		<i>Citrus Vein Phloem Degeneration (CVPD)</i>

41		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
42		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
43		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
44		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
45		<i>Citrus Vein Phloem Degeneration (CVPD)</i>

46		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
47		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
48		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
49		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
50		<i>Citrus Vein Phloem Degeneration (CVPD)</i>

51		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
52		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
53		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
54		<i>Citrus Vein Phloem Degeneration (CVPD)</i>
55		<i>Citrus Vein Phloem Degeneration (CVPD)</i>

56			<i>Citrus Vein Phloem Degeneration (CVPD)</i>
57			<i>Citrus Vein Phloem Degeneration (CVPD)</i>
58			<i>Citrus Vein Phloem Degeneration (CVPD)</i>
59			<i>Citrus Vein Phloem Degeneration (CVPD)</i>
60			<i>Citrus Vein Phloem Degeneration (CVPD)</i>

61		Cendawan Jelaga
62		Cendawan Jelaga
63		Cendawan Jelaga
64		Cendawan Jelaga
65		Cendawan Jelaga

66		Cendawan Jelaga
67		Cendawan Jelaga
68		Cendawan Jelaga
69		Cendawan Jelaga
70		Cendawan Jelaga

71		Cendawan Jelaga
72		Cendawan Jelaga
73		Cendawan Jelaga
74		Cendawan Jelaga
75		Cendawan Jelaga

76		Cendawan Jelaga
77		Cendawan Jelaga
78		Cendawan Jelaga
79		Cendawan Jelaga
80		Cendawan Jelaga

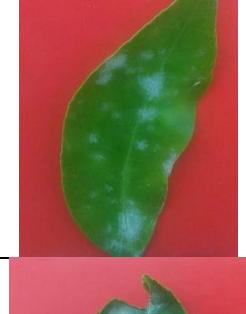
81		Cendawan Jelaga
82		Cendawan Jelaga
83		Cendawan Jelaga
84		Cendawan Jelaga
85		Cendawan Jelaga

86		 A green leaf specimen with a prominent dark purple spot near the base, set against a red background.	Cendawan Jelaga
87		 A green leaf specimen with a prominent dark purple spot near the base, set against a pink background.	Cendawan Jelaga
88		 A green leaf specimen with a prominent dark purple spot near the base, set against a pink background.	Cendawan Jelaga
89		 A green leaf specimen with a prominent dark purple spot near the base, set against a pink background.	Cendawan Jelaga
90		 A green leaf specimen with a prominent dark purple spot near the base, set against a pink background.	Cendawan Jelaga

91		<i>Downy Mildew</i>
92		<i>Downy Mildew</i>
93		<i>Downy Mildew</i>
94		<i>Downy Mildew</i>
95		<i>Downy Mildew</i>

96			<i>Downy Mildew</i>
97			<i>Downy Mildew</i>
98			<i>Downy Mildew</i>
99			<i>Downy Mildew</i>
100			<i>Downy Mildew</i>

101			<i>Downy Mildew</i>
102			<i>Downy Mildew</i>
103			<i>Downy Mildew</i>
104			<i>Downy Mildew</i>
105			<i>Downy Mildew</i>

106			<i>Downy Mildew</i>
107			<i>Downy Mildew</i>
108			<i>Downy Mildew</i>
109			<i>Downy Mildew</i>
110			<i>Downy Mildew</i>

111			<i>Downy Mildew</i>
112			<i>Downy Mildew</i>
113			<i>Downy Mildew</i>
114			<i>Downy Mildew</i>
115			<i>Downy Mildew</i>

116			<i>Downy Mildew</i>
117			<i>Downy Mildew</i>
118			<i>Downy Mildew</i>
119			<i>Downy Mildew</i>
120			<i>Downy Mildew</i>

LAMPIRAN B DATA UJI GAMBAR DAUN JERUK

No	Gambar Daun	Jenis Penyakit
1		CVPD
2		Cendawan Jelaga
3		<i>Downy Mildew</i>
4		Daun Sehat
5		Cendawan Jelaga

6		Cendawan Jelaga
7		Cendawan Jelaga
8		Cendawan Jelaga
9		Cendawan Jelaga
10		Cendawan Jelaga

11		Cendawan Jelaga
12		Cendawan Jelaga
13		Cendawan Jelaga
14		Cendawan Jelaga
15		<i>Downy Mildew</i>

16			<i>Downy Mildew</i>
17			<i>Downy Mildew</i>
18			Daun Sehat
19			Daun Sehat
20			Daun Sehat

21			Daun Sehat
22			Daun Sehat
23			Daun Sehat
24			Daun Sehat
25			Daun Sehat

26		Daun Sehat
27		Daun Sehat
28		Daun Sehat
29		Daun Sehat
30		Daun Sehat

31			Daun Sehat
32			Daun Sehat
33			Daun Sehat
34			Daun Sehat
35			Cendawan Jelaga

36			<i>Downy Mildew</i>
37			<i>Downy Mildew</i>
38			<i>Downy Mildew</i>
39			<i>Downy Mildew</i>
40			CVPD

41			CVPD
42			CVPD
43			CVPD
44			CVPD
45			CVPD

46		 A photograph of a single leaf showing significant yellowing, particularly along the veins and towards the edges, against a red background.	CVPD
47		 A photograph of a single leaf showing significant yellowing, particularly along the veins and towards the edges, against a red background.	CVPD
48		 A photograph of a single leaf showing significant yellowing, particularly along the veins and towards the edges, against a red background.	CVPD
49		 A photograph of a single leaf showing significant yellowing, particularly along the veins and towards the edges, against a red background.	CVPD
50		 A photograph of a single leaf showing dark spots and discoloration, characteristic of Downy Mildew, against a red background.	<i>Downy Mildew</i>

51		<i>Downy Mildew</i>
52		Cendawan Jelaga
53		Cendawan Jelaga
54		Cendawan Jelaga
55		<i>Downy Mildew</i>

56			<i>Downy Mildew</i>
57			<i>Downy Mildew</i>
58			Daun Sehat
59			<i>Downy Mildew</i>
60			<i>Downy Mildew</i>