

**RANCANG BANGUN APLIKASI BERBASIS ANDROID UNTUK DETEKSI
PENYAKIT HUANGLONGBING PADA TANAMAN JERUK DENGAN
CONVOLUTIONAL NEURAL NETWORK**

TUGAS AKHIR



RADITYA ARIYANTA PRADANA

NIM : 311910020

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MA CHUNG
MALANG
2023**

LEMBAR PENGESAHAN
TUGAS AKHIR

**RANCANG BANGUN APLIKASI BERBASIS ANDROID UNTUK
DETEKSI PENYAKIT HUANGLONGBING PADA TANAMAN JERUK
DENGAN CONVOLUTIONAL NEURAL NETWORK**

Oleh:


RADITYA ARIYANTA PRADANA
NIM. 311910020

dari:


PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS dan TEKNOLOGI
UNIVERSITAS MA CHUNG

Telah dinyatakan lulus dalam melaksanakan Tugas Akhir sebagai syarat kelulusan
dan berhak mendapatkan gelar Sarjana Komputer (S.Kom.)


Dosen Pembimbing I,


Dr. Kestrija Rega Prilianti, M.Si.
NIP. 20120035

Dosen Pembimbing II,


Hendry Setiawan, ST., M.Kom.
NIP. 20100006

Dekan Fakultas Sains dan Teknologi,


Dr. Kestrija Rega Prilianti, M.Si.
NIP. 20120035

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Rancang Bangun Aplikasi Berbasis Android Untuk Deteksi Penyakit Huanglongbing Pada Tanaman Jeruk Dengan Convolutional Neural Network” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupu merujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Malang 20 Juli 2023



Raditya Ariyanta Pradana

NIM. 311910020

RANCANG BANGUN APLIKASI BERBASIS ANDROID UNTUK DETEKSI PENYAKIT HUANGLONGBING PADA TANAMAN JERUK DENGAN CONVOLUTIONAL NEURAL NETWORK

Raditya Ariyanta Pradana, Kestrilia Rega Prilianti, Hendry Setiawan

Universitas Ma Chung

Abstrak

Jeruk merupakan buah yang kaya akan vitamin C yang bermanfaat bagi kesehatan tubuh. Salah satu permasalahan yang dihadapi dalam memenuhi kebutuhan buah jeruk adalah penyakit yang menyerang pohon jeruk yaitu HuangLongBing atau HLB. Deteksi penyakit HLB masih sulit dilakukan jika tidak didampingi oleh para ahli. Pembuatan model klasifikasi dengan algoritma *Convolutional Neural Network* untuk mengenali gambar dengan kondisi berbeda. Model CNN dibangun dengan arsitektur AlexNet, VGG16, dan SqueezeNet dan metode Optimasi Adam dan Adamax. Model yang berhasil dibuat terdiri dari 6 model yaitu AlexNet_Adam, AlexNet_Adamax, VGG16_Adam, VGG16_Adamax, SqueezeNet_Adam, dan SqueezeNet_Adamax. Pengujian dilakukan dengan melihat akurasi dan *loss* pada saat *training*, *validation*, dan *test*. Model dengan arsitektur VGG16 dan metode optimasi Adam ditentukan sebagai model terbaik dengan hasil akurasi yang paling baik pada training dengan akurasi sebesar 96%, validation dengan akurasi 85,7%, dan test dengan akurasi 92,8%. Model dengan arsitektur AlexNet memiliki ukuran file yang paling besar dengan ukuran 500 MB. Arsitektur VGG16 menghasilkan model dengan ukuran file 200 MB. Ukuran file paling kecil dihasilkan oleh arsitektur SqueezeNet dengan ukuran 2 MB. Model dengan arsitektur AlexNet tidak dapat melakukan prediksi karena ukuran file yang sangat besar sehingga tidak dapat mengetahui waktu yang dibutuhkan aplikasi untuk melakukan prediksi. Model dengan arsitektur VGG16 membutuhkan waktu sekitar 9 detik untuk melakukan prediksi. Sedangkan model dengan arsitektur SqueezeNet membutuhkan waktu kurang dari satu detik untuk melakukan prediksi.

Kata Kunci: AlexNet, *Convolutional Neural Network*, HuangLongBing, SqueezeNet, VGG16

DESIGN AND DEVELOPMENT OF AN ANDROID-BASED APPLICATION FOR HUANGLONGBING DISEASE DETECTION IN CITRUS PLANTS USING CONVOLUTIONAL NEURAL NETWORK

Raditya Ariyanta Pradana, Kestrilia Rega Prilianti, Hendry Setiawan

Ma Chung University

Abstract

Oranges are fruits rich in vitamin C, which is beneficial for overall health. One of the challenges faced in meeting the demand for oranges is the HuangLong-Bing (HLB) disease that affects orange trees. Detecting HLB disease is still difficult without the assistance of experts. A classification model was developed using the Convolutional Neural Network algorithm to recognize images with different conditions. The CNN model was built using the architectures of AlexNet, VGG16, and SqueezeNet, along with the Adam and Adamax optimization methods. The successful creation of the models resulted in six models: AlexNet_Adam, AlexNet_Adamax, VGG16_Adam, VGG16_Adamax, SqueezeNet_Adam, and SqueezeNet_Adamax. Testing was conducted by observing the accuracy and loss during training, validation, and testing. The VGG16 architecture with the Adam optimization method was determined as the best model, achieving the highest accuracy of 96% during training, 85.7% during validation, and 92.8% during testing. The AlexNet architecture had the largest file size, reaching 500 MB. The VGG16 architecture produced a model with a file size of 200 MB. The smallest file size was obtained with the SqueezeNet architecture, which was only 2 MB. The AlexNet model could not make predictions due to its extremely large file size, making it impossible to determine the time required for the application to make predictions. The VGG16 model took approximately 9 seconds to make predictions, while the SqueezeNet model required less than one second for prediction.

Keywords: AlexNet, *Convolutional Neural Network*, HuangLongBing, SqueezeNet, VGG16

KATA PENGANTAR

Puji dan syukur panjatkan kepada Tuhan yang Maha Esa atas segala anugerah dan petunjuk-Nya, penulis dapat menyelesaikan Tugas Akhir “Rancang Bangun Aplikasi Berbasis Android Untuk Deteksi Penyakit Huanglongbing Pada Tanaman Jeruk Dengan *Convolutional Neural Network*” dengan sebaik mungkin dan tepat pada waktunya. Penulis menyadari sepenuhnya bahwa dalam proses penyusunan Tugas Akhir ini tidak terlepas dari bimbingan, dukungan, serta semangat dari berbagai pihak yang telah berjasa terlibat dalam penyusunan Tugas Akhir ini. Oleh karena itu, dalam kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT Tuhan Yang Maha Esa yang telah memberkahi segala upaya penulis sehingga dapat menyelesaikan penelitian Tugas Akhir ini dengan sebaik mungkin.
2. Orang tua yang tidak pernah berhenti mendoakan dan memberikan dukungan kepada penulis.
3. Ibu Dr. Kestrilia Rega Prilianti, S.Si., M.Si., selaku Dekan Fakultas Sains dan Teknologi Universitas Ma Chung dan dosen pembimbing 1.
4. Bapak Hendry Setiawan, ST., M.Kom., selaku Kepala Program Studi Teknik Informatika Universitas Ma Chung dan Dosen Pembimbing 2.
5. Teman-teman penulis beserta seluruh pihak yang telah membantu memberi dukungan, menghibur, dan memberi semangat dalam penyusunan dan penyelesaian Tugas Akhir ini, yang tidak dapat penulis sebutkan satu per satu.

Laporan Tugas Akhir ini disusun berdasarkan hasil Tugas Akhir dengan judul “Rancang Bangun Aplikasi Berbasis Android Untuk Deteksi Penyakit Huanglongbing Pada Tanaman Jeruk Dengan *Convolutional Neural Network*”. Dalam penyusunannya, penulis telah berupaya sebaik dan semaksimal mungkin untuk memberikan hasil yang maksimal pula. Namun penulis menyadari sepenuhnya bahwa masih banyak kekurangan dan ketidak-sempurnaan di dalamnya. Oleh karena itu, demi pengembangan Tugas Akhir ini agar lebih baik, segala kritik dan saran yang bersifat membangun sangat penulis harapkan. Semoga

penelitian Tugas Akhir ini dapat berguna dan bermanfaat bagi pihak-pihak yang membutuhkan.

Malang, 20 Juli 2023

Raditya Ariyanta Pradana
NIM. 311910020

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN KEASLIAN TUGAS AKHIR	ii
Abstrak	iii
Abstract	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
DAFTAR PERSAMAAN	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Identifikasi Masalah	3
1.3 Batasan Masalah	3
1.4 Rumusan Masalah	3
1.5 Tujuan Penelitian	4
1.6 Manfaat Penelitian	4
1.7 Luaran Penelitian	4
1.8 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	6
2.1 Tanaman Jeruk	6
2.2 HuangLongBing (HLB)	6
2.3 Pigmen Tanaman	8
2.3.1 Klorofil	9
2.3.2 Karotenoid	9
2.3.3 Antosianin	10
2.4 Colorimeter	10
2.5 Lux meter	11
2.6 Python	12
2.7 Grey Level Co-occurrence Matrix (GLCM)	12
2.8 Tensorflow	13

2.9 Convolutional Neural Network	13
2.8.1 Feature Extraction Layer	14
2.8.2 Fully Connected Layer	16
2.10 Arsitektur VGG16	16
2.11 Arsitektur AlexNet	17
2.12 Arsitektur SqueezeNet	18
2.13 Optimizer Adam	19
2.14 Optimizer Adamax	20
2.15 SparseCategoricalCrossEntropy	20
2.16 Confusion Matrix	21
2.17 Java	22
2.18 Android Studio	22
2.19 Google Colaboratory	22
2.20 Penelitian Terdahulu	23
BAB III ANALISIS DAN PERANCANGAN SISTEM	25
3.1 Alur Penelitian	25
3.2 Analisis Kebutuhan	26
3.2.1 Kebutuhan Pengguna	26
3.2.2 Kebutuhan Peneliti	26
3.3 Studi Pustaka	27
3.4 Pengumpulan Data	28
3.5 Profiling	30
3.6 Desain dan Perancangan Sistem	31
3.6.1 Pembuatan Dataset	31
3.6.2 Preprocessing	31
3.6.3 Pelatihan Model CNN	32
3.6.4 Pengujian Model Menggunakan Data Test	33
3.7 Pengembangan Aplikasi	33
3.8 Pengujian Aplikasi	37
BAB IV HASIL DAN PEMBAHASAN	40
4.1 Dataset	40
4.2 Profiling	40

4.3 Implementasi CNN	44
4.4 Augmentasi	44
4.5 Kalibrasi	46
4.6 Pengujian Kinerja Model	47
4.6.1 AlexNet_Adam	47
4.6.2 AlexNet_Adamax	49
4.6.3 VGG16_Adam	51
4.6.4 VGG16_Adamax	53
4.6.5 SqueezeNet_Adam	55
4.6.6 SqueezeNet_Adamax	57
4.6.7 Ringkasan Pengujian Model	59
4.7 Hasil Aplikasi	61
BAB V SIMPULAN DAN SARAN	64
5.1 Simpulan	64
5.2 Saran	65
DAFTAR PUSTAKA	66
LAMPIRAN	69

DAFTAR GAMBAR

Gambar 2.1	Tanaman Jeruk Sehat	6
Gambar 2.2	Daun Tanaman Jeruk Terjangkit HLB	7
Gambar 2.3	Buah Jeruk Terjangkit HLB	8
Gambar 2.4	Grafik Absorbansi Pigmen Tanaman	10
Gambar 2.5	Colorimeter	11
Gambar 2.6	<i>Lux Meter</i>	12
Gambar 2.7	Arsitektur CNN	14
Gambar 2.8	Operasi Konvolusi dengan Kernel	15
Gambar 2.9	Max dan Average Pooling	15
Gambar 2.10	<i>Fully Connected Layer</i>	16
Gambar 2.11	Arsitektur VGG16	17
Gambar 2.12	Arsitektur AlexNet	17
Gambar 2.13	Arsitektur SqueezeNet	18
Gambar 2.14	Confusion Matrix	21
Gambar 3.1	Alur Penelitian	25
Gambar 3.2	Alur Pengumpulan Data	30
Gambar 3.3	Alur Desain dan Perancangan Sistem	31
Gambar 3.4	Halaman Utama	34
Gambar 3.5	Halaman Foto	35
Gambar 3.6	Halaman Riwayat	35
Gambar 3.7	Halaman Bantuan	36
Gambar 3.8	Halaman Tentang (about)	37
Gambar 4.1	Grafik Rata-rata RGB Data Colorimeter	41
Gambar 4.2	Grafik Standar Deviasi RGB Data Colorimeter	41
Gambar 4.3	Grafik Hasil GLCM <i>Dissimilarity</i>	42
Gambar 4.4	Daun dengan Kondisi HLB dan Sehat	43
Gambar 4.5	Grafik Hasil GLCM <i>Homogeneity</i>	43
Gambar 4.6	Confusion Matrix Data Test dengan Model Augmentasi	45
Gambar 4.7	Sebelum dan Sesudah Kalibrasi pada Daun HLB	46

Gambar 4.8	Grafik Indikator Kinerja AlexNet_Adam (a) Akurasi dan (b) Loss	48
Gambar 4.9	Confusion Matrix Data Test AlexNet_Adam	49
Gambar 4.10	Grafik Indikator Kinerja AlexNet_Adamax (a) Akurasi dan (b) Loss	50
Gambar 4.11	Confusion Matrix Data Test AlexNet_Adamax	51
Gambar 4.12	Grafik Indikator Kinerja VGG16_Adam (a) Akurasi dan (b) Loss	52
Gambar 4.13	Confusion Matrix Data Test VGG16_Adam	53
Gambar 4.14	Grafik Indikator Kinerja VGG16_Adamax (a) Akurasi dan (b) Loss	54
Gambar 4.15	Confusion Matrix Data Test VGG16_Adamax	55
Gambar 4.16	Grafik Indikator Kinerja SqueezeNet_Adam (a) Akurasi dan (b) Loss	56
Gambar 4.17	Confusion Matrix Data Test SqueezeNet_Adam	57
Gambar 4.18	Grafik Indikator Kinerja SqueezeNet_Adamax (a) Akurasi dan (b) Loss	58
Gambar 4.19	Confusion Matrix Data Test SqueezeNet_Adamax	59
Gambar 4.20	Tampilan Baru Halaman Prediksi	61

DAFTAR TABEL

Tabel 3.1	Ringkasan Penelitian Terdahulu	28
Tabel 3.2	Contoh Data pada Tiga Kondisi Tanaman Jeruk	29
Tabel 3.3	Confusion Matrix untuk Pengujian Model	33
Tabel 3.4	Tabel Evaluasi UI Aplikasi	38
Tabel 3.5	Tabel Evaluasi Kinerja Model CNN	39
Tabel 4.1	Jumlah Data Train, Test, dan Valid	40
Tabel 4.2	Perbandingan Nilai Akurasi dan Loss Model Dengan Augmentasi dan Tanpa Augmentasi	45
Tabel 4.3	Perbandingan Nilai Akurasi dan Loss Model Dengan Kalibrasi dan Tanpa Kalibrasi	47
Tabel 4.4	Ringkasan Akurasi Model CNN	60
Tabel 4.5	Ringkasan Loss Model CNN	60
Tabel 4.6	Tabel Evaluasi Kinerja Aplikasi	62
Tabel 4.7	Hasil Evaluasi Model	63

DAFTAR PERSAMAAN

(2-1)	Parameter Adam yang Dioptimasi	19
(2-2)	Persamaan Bias Adam	19
(2-3)	Bias-Corrected Dari Moving Average Kuadrat Gradien	19
(2-4)	Persamaan Moving Avarage Adam	19
(2-5)	Persamaan Moving Average Kuadrat Adam	19
(2-6)	Persamaan Moving Average Adamax	20
(2-7)	Moving Average dari Gradien Mutlak	20
(2-8)	Parameter Adamax yang Dioptimasi	20
(2-9)	Persamaan Loss SparseCategoricalCrossEntropy	21

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pangan merupakan bahan makanan yang dikonsumsi oleh manusia untuk memenuhi kebutuhan nutrisi dan energi dalam tubuh. Pangan bisa berasal dari berbagai sumber seperti tumbuhan dan hewan yang dapat diolah atau diproses menjadi berbagai jenis makanan dan minuman. Ketersediaan pangan yang cukup dan berkualitas merupakan salah satu aspek paling penting dalam memenuhi kebutuhan dasar manusia dan menjaga kesehatan (Sulaeman, 2014). Hal ini selaras dengan konsep swasembada pangan yang dikemukakan oleh Kementerian Pertanian Republik Indonesia (2021), dimana swasembada pangan merupakan kondisi pangan yang diperlukan oleh seluruh masyarakat, baik dari segi kualitas dan kuantitas. Oleh karena itu, kebutuhan pangan manusia akan pemenuhan nutrisi dalam tubuh tentunya bermacam-macam, salah satunya dapat diperoleh melalui vitamin.

Vitamin merupakan senyawa organik esensial yang dibutuhkan oleh tubuh dalam jumlah kecil untuk menjaga kesehatan dan fungsi normal. Vitamin penting bagi tubuh manusia karena berperan sebagai proses biologis, seperti metabolisme energi, pertumbuhan dan perkembangan, fungsi sistem saraf, dan sistem kekebalan tubuh (Linus Pauling Institute, 2016). tanpa adanya asupan vitamin yang cukup, maka tubuh manusia bisa menyebabkan berbagai masalah kesehatan dan dapat mempengaruhi fungsi normal tubuh. Setiap vitamin tentunya memiliki fungsi dan perannya masing-masing di dalam tubuh. adapun salah satu vitamin yang memiliki peran cukup besar bagi tubuh adalah vitamin C. hal tersebut karena tanpa adanya vitamin C, maka tubuh manusia tidak bisa menyebabkan berbagai masalah kesehatan pada tubuh (Linus Pauling Institute, 2023). vitamin C ini dapat diperoleh manusia melalui buah jeruk, dimana jeruk merupakan salah satu sumber terbaik vitamin C. Jeruk mengandung vitamin C dalam jumlah yang signifikan, yang bermanfaat bagi kesehatan tubuh.

Salah satu permasalahan yang dihadapi dalam memenuhi kebutuhan buah jeruk adalah penyakit pada pohon jeruk, salah satu penyakit serius yang menyerang

pohon jeruk adalah HuangLongBing atau HLB. Penyakit HLB atau bisa juga dikenal sebagai *Greening Disease* disebabkan oleh bakteri *Candidatus Liberibacter* spp. yang menyebar melalui serangga vektor seperti serangga kutu loncat dan serangga ulat buah. Bakteri tersebut hidup dan berkembang pada tanaman jeruk bagian jaringan floem yang mengakibatkan floem mengalami degenerasi sehingga terjadi penghambatan dalam penyerapan nutrisi tanaman. Gejala HLB yang bisa dilihat adalah adanya perubahan warna daun jeruk menjadi warna kuning dan bergaris-garis. Selain pada daun, gejala penyakit HLB juga bisa dilihat pada pucuk tanaman yang mati, buah yang kecil dan asam, serta bintik-bintik kuning pada daging buah. Pada tanaman jeruk muda yang sudah terinfeksi penyakit HLB menyebabkan ukuran daun lebih kecil, dan perkembangan kuncup daun lebih lambat. Sedangkan pada tanaman jeruk dewasa yang sudah terinfeksi penyakit HLB menyebabkan ukuran daun menjadi lebih kecil, kaku, tebal dan mengerasnya tulang daun. Penyakit HLB yang sangat sulit untuk di deteksi bagi para petani jika tidak didampingi oleh pakar penyakit tanaman, membuat penyebaran penyakit menjadi sangat sulit dikendalikan.

Deteksi penyakit HLB dengan cepat dan mudah dibutuhkan bantuan teknologi agar deteksi penyakit HLB secepat mungkin dapat mengurangi penyebaran penyakit HLB. Adapun salah satu cara yang bisa digunakan untuk deteksi penyakit HLB adalah dengan menggunakan model CNN. Model CNN ini dibentuk dengan melihat warna pada citra daun jeruk dan menghitung kandungan pigmen yang terkandung pada daun tanaman jeruk. Kandungan pigmen yang di dapat melalui model CNN akan menentukan apakah Tanaman jeruk terserang oleh penyakit HLB atau tidak. Beberapa penelitian terdahulu terkait dengan deteksi penyakit pada tanaman jeruk menggunakan CNN yang menjadi landasan penggunaan CNN adalah, Penelitian yang dilakukan oleh Natarajan dkk., (2020) berfokus pada meninjau efisiensi arsitektur deep learning untuk deteksi hama dan penyakit pada tanaman tomat. Data yang digunakan untuk mengembangkan model terdiri dari 1090 gambar daun tomat waktu nyata yang terinfeksi early blight, leaf curl, septoria leafspot, dan bacterial spot. Lalu Militante dkk., (2019) melakukan penelitian mengembangkan model *deep learning* untuk melakukan identifikasi penyakit tanaman melalui gambar daun sederhana dari tanaman yang sehat atau

sakit. Pelatihan model dilakukan menggunakan database terbuka yang terdiri dari 87.848 foto. Arsitektur model paling sukses, yaitu jaringan saraf konvolusi VGG, mencapai tingkat keberhasilan 99,53%.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah disampaikan, dapat diperoleh identifikasi masalah bahwa deteksi penyakit HLB masih sulit dilakukan jika tidak didampingi oleh para ahli, sehingga penyebaran penyakit HLB menjadi sulit dideteksi dan dikendalikan. Hal ini tentu menjadi masalah serius dalam industri perkebunan jeruk karena akan mempengaruhi jumlah jeruk yang berhasil dipanen dan memiliki nilai jual.

1.3 Batasan Masalah

Adapun beberapa batasan masalah dalam pengerjaan proyek ini antara lain sebagai berikut:

1. Menggunakan alat *Colorimeter* sebagai alat validasi nilai sebenarnya dari warna daun.
2. Bahasa pemrograman yang digunakan untuk pemrograman *Deep Learning* CNN adalah Python.
3. Bahasa pemrograman yang digunakan untuk pemrograman mobile adalah java.
4. Pembuatan dataset dan citra gambar hanya dilakukan dengan menggunakan kamera *smartphone*.
5. Jenis jeruk yang digunakan untuk membuat dataset adalah jeruk keprok.
6. Deteksi penyakit HLB dilakukan dengan citra daun tanaman jeruk.

1.4 Rumusan Masalah

Berdasarkan identifikasi masalah yang telah disampaikan, adapun masalah yang dapat dirumuskan dari proyek yang telah dikerjakan yaitu Bagaimana membangun aplikasi untuk melakukan deteksi penyakit HLB Secara *Realtime*?

1.5 Tujuan Penelitian

Adapun tujuan dari pengerjaan proyek ini berdasarkan masalah yang telah dirumuskan antara lain sebagai berikut membuat model CNN untuk deteksi penyakit HLB pada tanaman jeruk dan membangun aplikasi berbasis android dengan mengimplementasikan model CNN pada aplikasi.

1.6 Manfaat Penelitian

Adapun beberapa manfaat dari pengerjaan proyek ini antara lain yaitu:

1. Bagi masyarakat, dapat menggunakan aplikasi yang dapat mendeteksi penyakit HLB dengan melihat pigmen pada daun tanaman jeruk.
2. Bagi pembaca, dapat digunakan sebagai bahan referensi mengenai deteksi penyakit HLB dengan Convolutional Neural Network untuk belajar maupun kajian penelitian yang akan dilakukan.
3. Bagi Penulis, dapat menambah dan memperluas wawasan mengenai rancang bangun aplikasi android untuk deteksi penyakit HLB dengan Convolutional Neural Network.

1.7 Luaran Penelitian

Luaran dari penelitian ini adalah sebuah aplikasi berbasis android untuk melakukan deteksi penyakit HLB pada tanaman jeruk dengan menggunakan metode Convolutional Neural Network dan publikasi ilmiah.

1.8 Sistematika Penulisan

Sistematika dalam penulisan proposal Tugas Akhir ini akan dibagi menjadi lima bab seperti berikut.

Bab I Pendahuluan

Pada bab pendahuluan ini berisi latar belakang, identifikasi masalah, batasan masalah, tujuan penelitian, manfaat penelitian, luaran tugas akhir dan sistematika penulisan.

Bab II Tinjauan Pustaka

Bab tinjauan pustaka ini berisi uraian sistematis terkait dengan literatur yang digunakan dalam proses penyusunan Tugas Akhir sehingga diperoleh landasan teori terkait CNN, Android, dan penyakit HLB pada tanaman Jeruk.

Bab III Metodologi Penelitian

Bab ini menjelaskan tahapan pengerjaan serta analisis perancangan awal sistem yang akan dibuat. Tahapan ini sendiri terdiri atas identifikasi masalah, studi pustaka, pengumpulan data, *Profiling*, desain sistem, dan pengujian.

Bab IV Hasil dan Pembahasan

Bab ini menjelaskan tahapan pengerjaan serta analisis perancangan awal sistem yang akan dibuat. Tahapan ini sendiri terdiri atas identifikasi masalah, studi pustaka, pengumpulan data, desain sistem dan pengujian.

Bab V Kesimpulan dan Saran

Bab ini berisi simpulan dari hasil penelitian Tugas Akhir yang telah dilakukan beserta saran yang mungkin dapat dilakukan untuk memperbaiki sistem aplikasi untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Tanaman Jeruk

Tanaman jeruk adalah tanaman buah yang berasal dari genus citrus dalam keluarga Rutaceae. Jeruk mempunyai buah yang kaya akan vitamin C dan memiliki rasa yang asam-manis. Tanaman jeruk dapat tumbuh di daerah tropis dan subtropis, salah satunya di Indonesia. Menurut Yassin (2018), jeruk merupakan salah satu jenis tanaman buah yang penting di Indonesia dan merupakan sumber pendapatan bagi banyak petani karena memiliki nilai ekonomis yang tinggi. Tanaman jeruk dapat ditanaman di dataran rendah maupun dataran tinggi dengan ketinggian antara 100-1500 meter diatas permukaan laut (mdpl).



Gambar 2.1 Tanaman Jeruk Sehat

2.2 HuangLongBing (HLB)

HuangLongBing (HLB), atau bisa juga dikenal dengan naman *citrus vein phloem degeneration* (CVPD), merupakan penyakit yang menyerang tanaman jeruk

sejak awal 1960-an. Penyakit ini menyebabkan menurunnya produktivitas, kualitas bahkan menyebabkan kematian pada tanaman jeruk di berbagai negara di dunia. Penyebab dari penyakit HLB adalah bakteri *Candidatus Liberibacter*. Di Indonesia, penyakit HLB ditemukan pertama kali pada tanaman jeruk pada tahun 1964. Karakteristik tanaman jeruk yang terserang HLB adalah klorosis yang menyerupai defisiensi nitrogen, seng, mangan, dan besi. (Nurhadi, 2014). HLB menyebar melalui serangga vektor seperti kutu loncat yang menginfeksi tanaman saat menghisap getah tanaman yang terinfeksi. Bakteri tersebut akan masuk ke dalam jaringan floem atau jaringan pengangkut nutrisi dalam tanaman.

Gejala HLB bisa dilihat pada daun tanaman jeruk yang mengalami perubahan warna menjadi kuning secara tidak merata atau tidak berpola yang bisa dilihat pada gambar 2.2. Buah jeruk yang terkena penyakit HLB memiliki bentuk yang tidak normal, ukuran yang lebih kecil dari ukuran normal, dan memiliki rasa yang pahit atau tidak enak yang bisa dilihat pada gambar 2.3. Tanaman yang terinfeksi HLB umumnya tidak dapat dipulihkan dan akhirnya tanaman jeruk akan mati jika infeksi sudah sangat parah. Penanganan dan pengendalian HLB bisa dilakukan dengan penggunaan benih yang bebas dari infeksi, pengendalian hama serangga yang berpotensi untuk menyebarkan HLB dan penghapusan tanaman yang telah terinfeksi untuk mencegah penyebaran yang lebih lanjut.



Gambar 2.2 Daun Tanaman Jeruk Terjangkit HLB



Gambar 2.3 Buah Jeruk Terjangkit HLB

2.3 Pigmen Tanaman

Pigmen pada tanaman merupakan zat yang dapat menyerap dan memantulkan cahaya. Pada daun tanaman, pigmen bereaksi terhadap cahaya melalui sistem konjugasi atau rangkaian ikatan tunggal dan rangkap yang tersusun secara selang-seling. Sistem konjugasi memiliki kemampuan untuk menyerap dan memantulkan cahaya pada panjang gelombang tertentu. Hal ini membuat tanaman bisa melakukan proses fotosintesis dengan menyerap energi matahari dan menampilkan warna tertentu pada daun tanaman. Selain itu pada proses fotosintesis tanaman, pigmen berperan sangat penting untuk menyerap energi matahari yang akan digunakan untuk membuat makanan.

Sebuah tanaman tersusun dari beberapa jenis pigmen, tetapi hanya akan ada satu jenis pigmen yang mendominasi. Hal ini dapat terlihat pada beberapa tanaman yang akan memiliki warna daun berbeda, baik itu di satu tanaman maupun tanaman lain. Beberapa jenis pigmen pada tanaman dikelompokkan menjadi empat kelompok, yaitu klorofil, karotenoid (karoten, xantofil), flavonoid (chalcones, antosianin, flavon, flavonol) dan betalain (betaxanthin, betacyanin) (Młodzińska, 2009).

2.3.1 Klorofil

Menurut National Geographic Society (2019), klorofil atau zat hijau daun merupakan pigmen yang dimiliki oleh daun tanaman dan menjadi zat yang berperan penting dalam melakukan fotosintesis tanaman. Pada proses fotosintesis, klorofil berperan sebagai penyerap cahaya matahari untuk menjadi energi. Dalam tumbuhan, klorofil memiliki dua bentuk, yaitu klorofil-a yang terdapat pada semua organisme autotroph dan klorofil-b yang terdapat pada alga hijau serta tumbuhan darat.

Klorofil memantulkan cahaya hijau, serta menyerap cahaya merah dan biru. Molekul klorofil memantulkan cahaya hijau dengan panjang gelombang 450-650 nm. Selain memantulkan cahaya, klorofil juga menyerap cahaya merah dengan panjang gelombang 600 -700 nm dan cahaya biru dengan panjang gelombang 400-500 nm. Oleh karena adanya pantulan cahaya itulah yang membuat daun tanaman terlihat memiliki warna hijau. (Eichhorn Bilodeau dkk., 2019). Grafik spektrum absorbansi pada klorofil dapat dilihat pada gambar 2.1.

2.3.2 Karotenoid

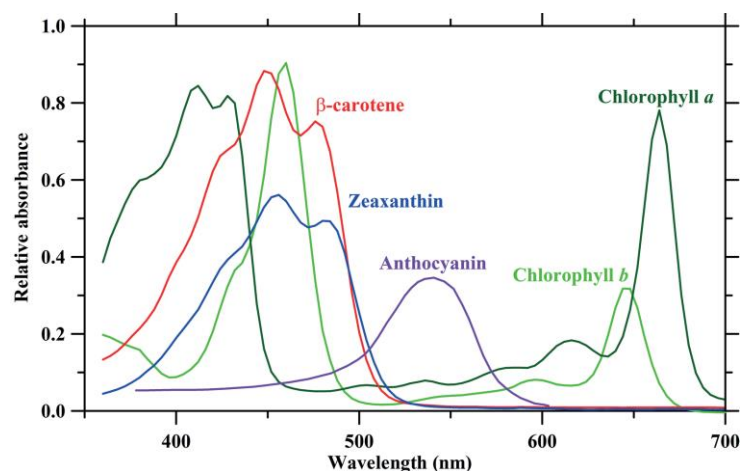
Karotenoid merupakan pigmen yang banyak ditemukan pada kromoplas tumbuhan dan alga serta sejumlah bakteri baik fotosintetik ataupun tidak dan beberapa fungi non-fotosintetik (Hirschberg dkk., 1997). Menurut Mithra (2011), menjelaskan karotenoid dikenal dengan manfaat sebagai antioksidan, berperan dalam proses fotosintesis sebagai pengantar elektron serta dalam beberapa karotenoid dapat diubah menjadi vitamin esensial.

Sebagai pigmen, karotenoid mampu menyerap cahaya biru dan memantulkan cahaya dengan warna-warna yang memiliki Panjang gelombang besar seperti merah sampai kuning kehijauan. Karotenoid menyerap cahaya biru pada dengan panjang 400-500 nm serta memantulkan cahaya kuning-merah dengan panjang gelombang 500-760 nm. Oleh karena adanya pantulan cahaya itulah yang membuat tampak warna kuning sampa kemerahan pada daun tanaman (Eichhorn Bilodeau dkk., 2019). Grafik spektrum absorbansi pada karotenoid dapat dilihat pada gambar 2.1.

2.3.3 Antosianin

Antosianin merupakan anggota dari kelas senyawa flavonoid yang larut di dalam air. Pigmen ini akan diproduksi oleh tanaman ketika tanaman berada dalam kondisi yang ekstrim dan menyebabkan tanaman mengalami stress, seperti suhu ekstrim serta tanaman memasuki masa penuaan. Senyawa ini memiliki kemampuan untuk menampilkan warna merah, ungu dan biru pada buah, bunga, dan daun tanaman. Antosianin sudah banyak di aplikasikan ke banyak pewarna alami pada berbagai produk pewarna makanan (Gould dkk., 2009).

Antosianin mampu menyerap cahaya biru-hijau serta memantulkan cahaya merah. Antosianin menyerap cahaya biru-hijau dengan panjang gelombang 450-550 nm. Antosianin juga memantulkan cahaya merah dengan Panjang gelombang 700 nm (Eichhorn Bilodeau dkk., 2019). Grafik spektrum absorbansi pada antosianin dapat dilihat pada gambar 2.1.



Gambar 2.4 Grafik Absorbansi Pigmen Tanaman

(Eichhorn Bilodeau dkk., 2019)

2.4 Colorimeter

Colorimeter merupakan alat yang digunakan untuk mengukur warna. Hasil output yang ditampilkan oleh Colorimeter dapat digunakan untuk membedakan warna. Colorimeter mendapatkan nilai warna dengan detektor untuk menentukan

konsentrasi atau kepekatan warna dari larutan zat warna dengan analisis intensitas cahaya. Cahaya yang dihasilkan oleh Colorimeter akan menembus sampel, sehingga berkas cahaya akan ditangkap oleh sensor photodiode. Sensor photodiode memiliki fungsi mengonversi cahaya menjadi tegangan *output*. Nilai *output* yang dihasilkan oleh colorimeter berupa ruang warna dengan format Lab (Putra dan Yulkifli, 2019).



Gambar 2.5 Colorimeter

2.5 Lux meter

Lux meter merupakan alat yang digunakan untuk mengukur tingkat kecerahan atau intensitas cahaya dalam suatu tempat. Nilai yang ditunjukkan oleh lux meter didapat dari sebuah sensor yang peka dan linier terhadap cahaya. Semakin kecil nilai intensitas cahaya yang ditunjukkan oleh lux meter maka semakin jauh jarak sumber cahaya, sebaliknya jika semakin besar nilai intensitas cahaya maka semakin dekat jarak sumber cahaya (Wijaya dan Sutrimo, 2020)



Gambar 2.6 *Lux Meter*

2.6 Python

Python merupakan bahasa pemrograman yang dibuat dan dikembangkan oleh Guido Van Rossum dan rilis secara publik pada tahun 1991. Bahasa pemrograman ini merupakan salah satu yang dapat melakukan intruksi multi guna secara langsung atau interpretatif, berorientasi objek dan semantic dinamis untuk memberikan tingkat keterbacaan syntax. Python memberikan fasilitas pendukung seperti modul, paket, dan library yang tersedia secara gratis.

2.7 *Grey Level Co-occurrence Matrix (GLCM)*

Gray-Level Co-Occurrence Matrix (GLCM) merupakan metode statistik untuk memeriksa tekstur yang mempertimbangkan hubungan spasial piksel dalam suatu gambar. Fungsi GLCM mengkarakterisasi tekstur gambar dengan menghitung seberapa sering pasangan piksel dengan nilai tertentu dan dalam hubungan spasial tertentu muncul dalam gambar, membuat GLCM, dan kemudian mengekstrak ukuran statistik dari matriks (MATLAB, n.d.). Berikut ini adalah beberapa fitur tekstur GLCM:

1. *Contrast*

Mengukur frekuensi spasial dari citra dan perbedaan momen pada GLCM. Perbedaan yang dimaksudkan merupakan perbedaan tinggi dan rendah nya pixel. Contrast akan bernilai mendekati 0 jika pixel tetangganya mempunyai nilai yang sama.

2. *Dissimilarity*

Mengukur ketidakmiripan tekstur, akan memiliki nilai yang besar bila acak dan sebaliknya akan bernilai kecil bila seragam.

3. *Homogeneity*

Nilai ini sensitif terhadap nilai disekitar diagonal utama. Bernilai tinggi apabila semua pixel mempunyai nilai yang sama.

4. *Energy*

Energi akan mengukur tentang keseragaman. Energi akan bernilai tinggi jika nilai pixel mirip satu sama lain sebaliknya akan bernilai kecil yang berarti nilai dari GLCM normalisasi adalah heterogen. Nilai maksimum dari energi adalah 1 yang berarti distribusi pixel dalam kondisi konstan atau bentuk nya yang tidak acak.

5. *Correlation*

Mengukur linearitas dari sejumlah pasangan pixel (pairs).

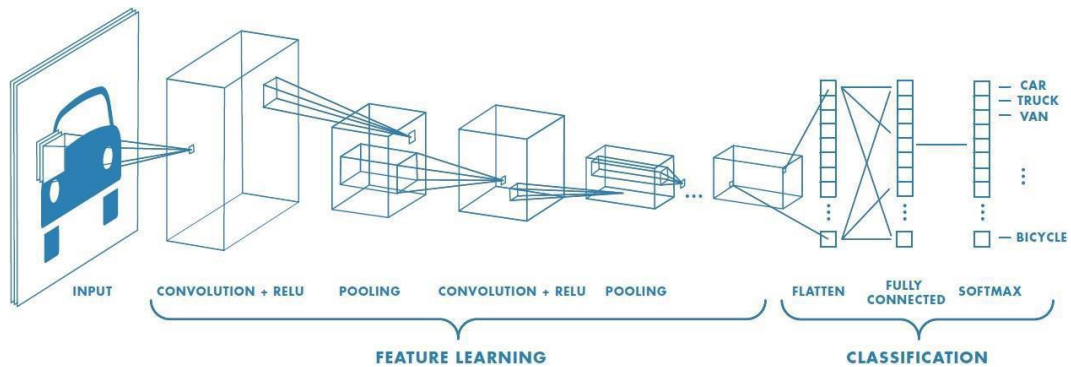
2.8 Tensorflow

Tensorflow merupakan *framework open-source* yang dikembangkan oleh google pada tahun 2015. *Framework* ini digunakan untuk membuat dan menjalankan model *machine learning* dan *deep learning*. TensorFlow memiliki fitur yang sangat kuat, seperti kemampuan untuk bekerja dengan data yang sangat besar, mendukung parallel computing, dan memberikan dukungan yang luas untuk berbagai jenis model machine learning dan deep learning, termasuk convolutional neural networks (CNN), recurrent neural networks (RNN), dan banyak lagi (Tensorflow, n.d.).

2.9 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu algoritma deep learning yang mana algoritma ini berkembang dari *Multilayer Perceptron*. CNN pada umumnya digunakan untuk pembelajaran terhadap berbagai macam citra dua dimensi (Saha, 2018). Dalam proses pembuatan model CNN diperlukan pre-processing untuk melakukan proses training. Model CNN yang melalui proses training yang cukup akan menghasilkan karakteristik yang bisa dipelajari. Peran dari CNN adalah untuk mereduksi citra ke bentuk yang sudah diproses tanpa

menghilangkan fitur-fitur penting, dimana fitur tersebut akan digunakan untuk melakukan proses prediksi (Saha, 2018).



Gambar 2.7 Arsitektur CNN

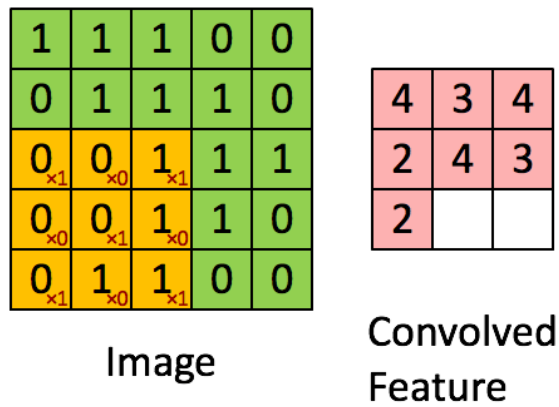
(Saha, 2018)

2.8.1 Feature Extraction Layer

Feature Extraction Layer merupakan layer yang berfungsi untuk melakukan proses encoding pada sebuah citra menjadi angka-angka yang mempresentasikan citra. Dalam *Feature Extraction Layer* terbagi menjadi dua bagian, yaitu *Convolutional Layer* dan *Pooling Layer*.

a. Convolutional Layer

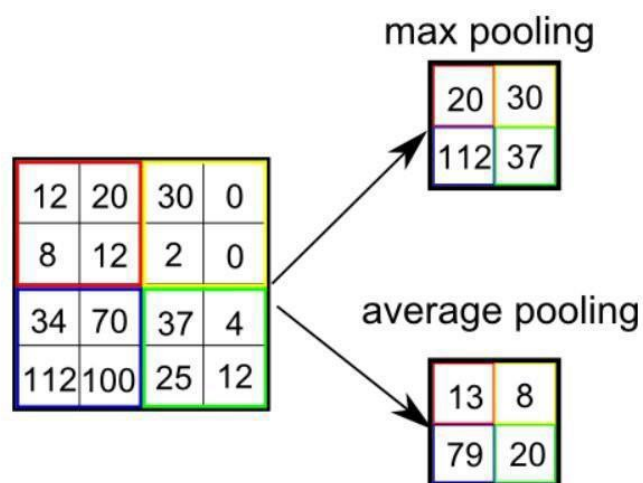
Convolutional layer atau layer konvolusi merupakan operasi konvolusi pada *output* dari *layer* sebelumnya. Pada prosesnya layer konvolusi diaplikasikan pada sebuah kernel dalam citra pada seluruh posisi yang memungkinkan untuk dilakukan proses konvolusi. Hal tersebut bisa dilihat pada gambar 2.8. Dalam prosesnya kernel akan bergerak dari sudut kiri atas sampai kanan bawah pada citra untuk melakukan ekstraksi. Hasil yang didapat dari proses konvolusi adalah *Convolved Feature* seperti kotak di kanan. Tujuan dari proses ekstraksi adalah untuk mendapatkan fitur dari setiap citra yang dimasukkan. Hasil konvolusi akan membentuk sebuah transformasi linear dari input sesuai informasi spasial pada data. *Weight* atau bobot dalam layer adalah hasil dari kernel konvolusi yang nantinya akan digunakan untuk melatih model.



Gambar 2.8 Operasi Konvolusi dengan Kernel
(Saha, 2018)

b. Pooling Layer

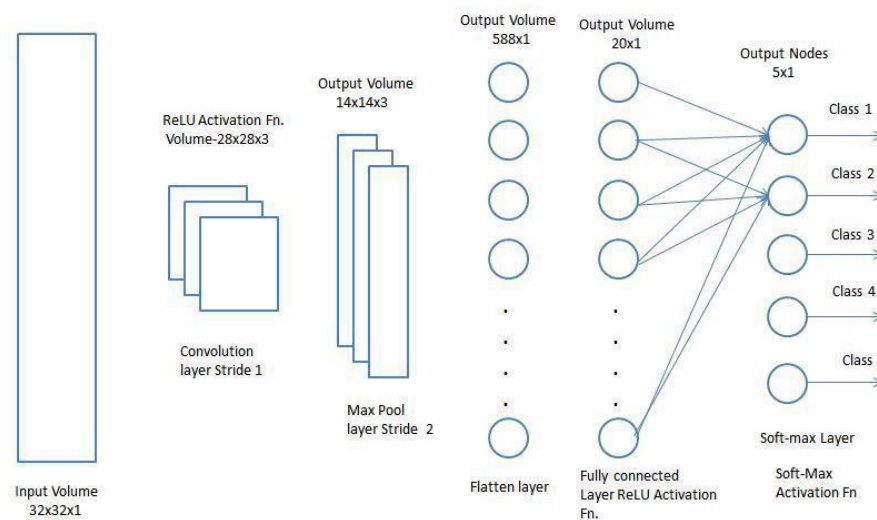
Pooling layer merupakan *layer* yang berfungsi untuk mengurangi ukuran spasial *Convolved Feature*. Pengurangan ukuran ini bertujuan untuk mengurangi daya komputasi yang digunakan saat data diproses dengan cara melakukan reduksi dimensi. Pooling layer terdiri dari dua jenis, yaitu *Max Pooling* dan *Average Pooling*. *Max Pooling* akan mengembalikan nilai maksimum dari gambar yang dicakup oleh kernel, sedangkan *Average Pooling* melakukan pengurangan dimensi dengan tujuan *noise suppressing mechanism*.



Gambar 2.9 Max dan Average Pooling
(Saha, 2018)

2.8.2 Fully Connected Layer

Fully connected layer merupakan *layer* yang berfungsi untuk mengubah data sehingga dapat digunakan untuk klasifikasi. Data yang dihasilkan oleh *feature extraction layer* berupa *feature map* yang berbentuk array multi dimensi, Oleh karena itu *feature map* tersebut harus di-reshape terlebih dahulu, sehingga menjadi sebuah *vector* yang dapat digunakan sebagai input dalam *Fully connected layer*.

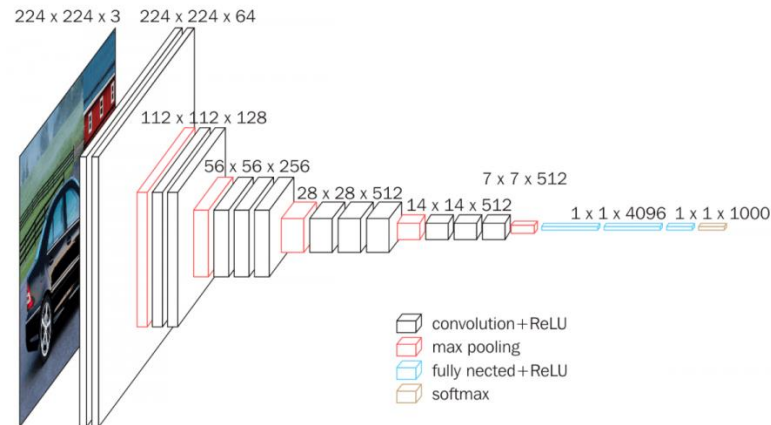


Gambar 2.10 *Fully Connected Layer*

(Saha, 2018)

2.10 Arsitektur VGG16

VGG16 adalah arsitektur CNN yang mampu bekerja sangat baik dalam pengenalan gambar. Arsitektur ini diusulkan oleh Karen Simonyan dan Andrew Zisserman dari Visual Geometry Group Lab of Oxford University pada tahun 2014. Arsitektur ini menggunakan *convolution layers* dengan filter 3x3 dan *stride* 1, selain itu pada lapisan *padding* dan *maxpool* dengan filter 2x2 stride 2. Angka 16 pada nama arsitektur ini mengacu pada arsitektur ini memiliki 16 lapisan yang memiliki bobot. Arsitektur VGG16 memiliki dua *fully connected layers*, diikuti dengan softmax untuk output (Boesch, 2021).

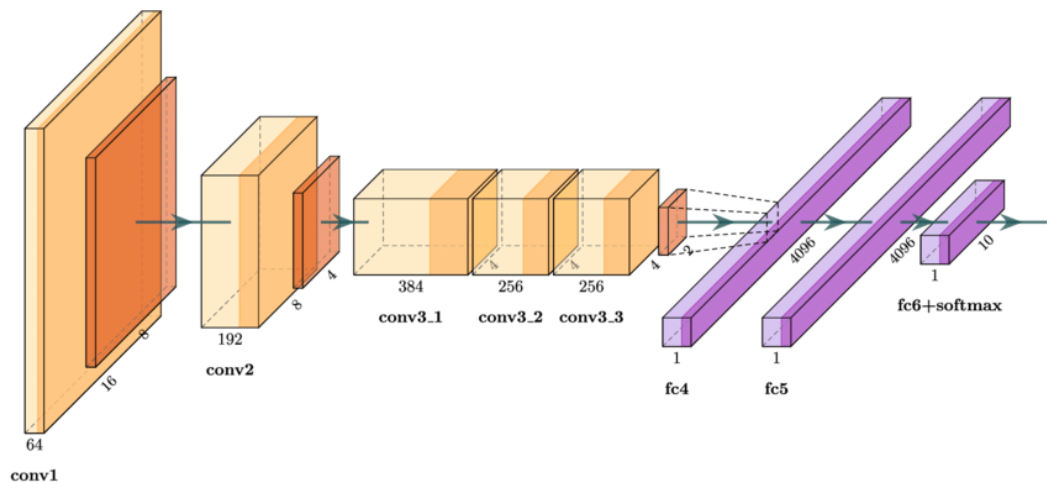


Gambar 2.11 Arsitektur VGG16

(Boesch, 2021)

2.11 Arsitektur AlexNet

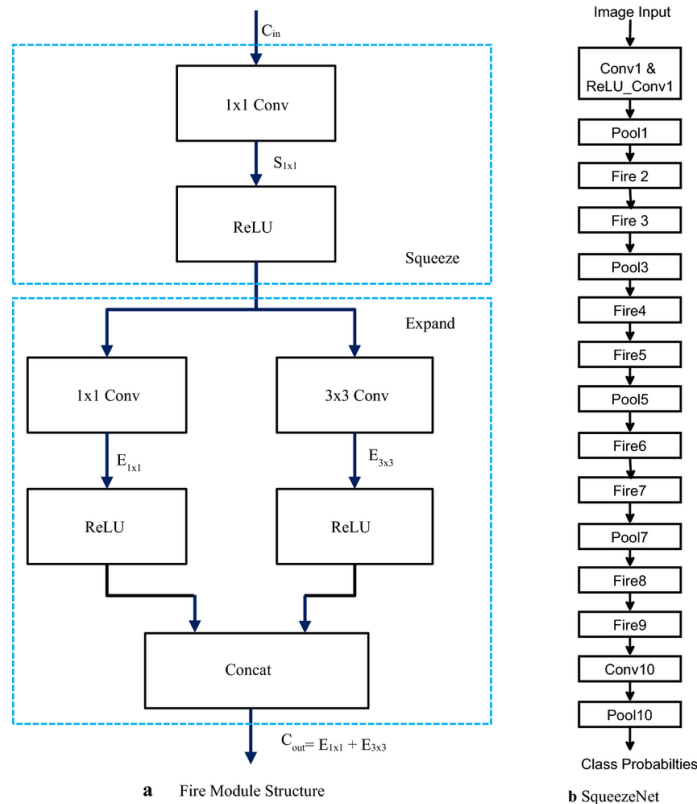
AlexNet merupakan arsitektur CNN yang dirancang oleh Alex Krizhevsky bekerja sama dengan Ilya Sutskever dan Geoffrey Hinton. Arsitektur ini merupakan arsitektur CNN pertama yang memenangkan Tantangan Pengenalan Visual Skala Besar ImageNet pada tahun 2012. AlexNet terdiri dari 5 *convolutional layers*, 3 *max-pooling layers*, 2 *normalization layers*, 2 *fully connected layers*, 1 *softmax layer* (Great Learning Team, 2020).



Gambar 2.12 Arsitektur AlexNet

2.12 Arsitektur SqueezeNet

SqueezeNet merupakan arsitektur *Convolutional Neural Network* yang menerapkan rancangan yang berbeda yaitu dengan mengurangi jumlah parameter, terutama dengan menggunakan module *fire* yang *squeeze* parameter menggunakan *convolution layer 1x1*. Arsitektur ini pertama kali diusulkan oleh para peneliti di DeepScale, UC Berkeley, dan Universitas Stanford pada tahun 2016. Pengurangan jumlah parameter dengan menggunakan modul *fire* menghasilkan model dengan ukuran *file* yang kecil. Arsitektur CNN yang lebih kecil seperti SqueezeNet menawarkan beberapa keuntungan, termasuk membutuhkan lebih sedikit komunikasi antar server selama pelatihan terdistribusi, lebih sedikit bandwidth untuk mengeksport model baru dari cloud, dan lebih layak untuk digunakan pada perangkat keras dengan memori terbatas (Tsang, 2019).



Gambar 2.13 Arsitektur SqueezeNet

2.13 Optimizer Adam

Optimizer Adam merupakan algoritma optimasi yang sering digunakan pada pemrograman deep learning. Adam adalah algoritma yang populer di bidang deep learning karena mencapai hasil yang baik dengan cepat. Adam adalah turunan dari metode optimasi SGD yang didasari oleh estimasi adaptif momen orde pertama dan kedua. Optimasi Adam menggabungkan RMSprop dan AdaGrad. Teknik optimasi ini mempertahankan satu *learning rate* untuk semua pembaruan bobot dan *learning rate* tidak berubah sama sekali (Brownlee, 2017). Untuk persamaan optimizer adam dapat dilihat pada persamaan berikut.

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \quad (2-1)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2-2)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2-3)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2-4)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2-5)$$

Dimana w_t adalah parameter yang dioptimasi pada iterasi ke- t , η adalah laju pembelajaran (learning rate), \hat{m}_t adalah perhitungan bias-corrected (dikoreksi kesalahan) dari moving average gradien, \hat{v}_t adalah perhitungan bias-corrected dari moving average kuadrat gradien, dan ϵ adalah nilai kecil yang digunakan untuk mencegah pembagian dengan nol. Dengan perhitungan m_t merupakan perhitungan moving average dari gradien g pada iterasi ke- t dan v_t adalah perhitungan moving average dari kuadrat gradien g pada iterasi ke- t . Dengan β_1 adalah parameter yang mengendalikan eksponensial moving average dari gradien dan β_2 adalah parameter yang mengendalikan eksponensial moving average dari kuadrat gradien.

2.14 Optimizer Adamax

Adamax merupakan metode optimasi first-order gradient-based yang berarti varian dari Adam berdasarkan norma tak terhingga. Optimasi ini merupakan perpanjangan dari gradien keturunan versi Adam yang mengeneralisirkan pendekatan koma menuju norma tak terbatas (Max) dan dapat menghasilkan pengoptimalan yang lebih efektif pada beberapa masalah (Brownlee, 2021). Untuk Persamaan optimizer Adamax dapat dilihat sebagai berikut.

$$m_t = \beta_1^\infty m_{t-1} + (1 - \beta_1^\infty) |g_t|^\infty \quad (2-6)$$

$$v_t = \max(\beta_2 \cdot v_{t-1}, |g_t|) \quad (2-7)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{1 - \beta_1^t} \frac{m_t}{v_t + \epsilon} \quad (2-8)$$

Dimana θ_t adalah parameter yang dioptimasi pada iterasi ke-t, η adalah laju pembelajaran (learning rate), m_t adalah perhitungan moving average untuk gradien g pada iterasi ke-t, v_t adalah moving average dari gradien mutlak g pada iterasi ke-t, dan ϵ adalah nilai kecil yang digunakan untuk mencegah pembagian dengan nol. Dengan β_1 adalah parameter yang mengendalikan eksponensial moving average dari gradien dan β_2 adalah parameter yang mengendalikan eksponensial moving average dari kuadrat gradien.

2.15 SparseCategoricalCrossEntropy

SparseCategoricalCrossentropy merupakan fungsi loss yang digunakan dalam algoritma machine learning untuk melatih model klasifikasi. Ini biasanya digunakan ketika label output direpresentasikan dalam format matriks jarang, di mana hanya satu nilai di setiap baris yang bukan nol. Fungsi ini digunakan dalam kasus di mana kelas saling eksklusif, artinya setiap input hanya dapat dimiliki oleh satu kelas. Fungsi menghitung loss cross-entropy antara label sebenarnya dan label yang diprediksi. Label sebenarnya dikodekan bilangan bulat, sedangkan label yang diprediksi adalah probabilitas yang ditetapkan untuk setiap kelas. Fungsi ini dapat

didefinisikan di TensorFlow sebagai `tf.keras.losses.SparseCategoricalCrossentropy` (Rahman, 2023).

$$J(w) = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (2-9)$$

Dimana w merupakan parameter model seperti bobot *neural network*, y_i merupakan label yang sebenarnya, dan \hat{y}_i merupakan label yang diprediksi.

2.16 Confusion Matrix

Confusion Matrix merupakan tabel yang meringkas kinerja algoritma klasifikasi. Ini digunakan untuk mengevaluasi keakuratan model pembelajaran mesin dengan membandingkan nilai prediksi dengan nilai sebenarnya. Matriks menampilkan jumlah *true positives* (TP), *true negatives* (TN), *false positives* (FP), dan *false negatives* (FN) (Narkhede, 2018). Gambar 2.11 merupakan contoh dari *confusion matrix* untuk ukuran 2x2.

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Gambar 2.14 Confusion Matrix

True positives (TP) adalah kasus di mana model memprediksi kelas positif dengan benar. False positives (FP) adalah kasus di mana model memprediksi kelas positif, tetapi kelas sebenarnya negatif. False negatives (FN) adalah kasus di mana

model memprediksi kelas negatif, tetapi kelas sebenarnya adalah positif. True negatives (TN) adalah kasus di mana model memprediksi kelas negatif dengan benar.

Confusion Matrix berfungsi untuk mengevaluasi kinerja model klasifikasi, terutama ketika kumpulan data tidak seimbang. Akurasi saja tidak cukup untuk melihat kinerja model karena akurasi saja bisa menyesatkan jika jumlah pengamatan di kelas yang berbeda sangat bervariasi (Narkhede, 2018).

2.17 Java

Java merupakan bahasa pemrograman yang *object-oriented* dan populer untuk mengembangkan aplikasi desktop, web, dan mobile. Java dikembangkan pada awal tahun 1990-an oleh James Gosling dan timnya di Sun Microsystems (sekarang bagian dari Oracle Corporation). Java dirancang untuk mudah dipelajari, dipahami, dan digunakan oleh para pengembang, serta memiliki keunggulan dalam portabilitas dan keamanan (Oracle, n.d.).

2.18 Android Studio

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) yang dibuat khusus untuk pengembangan aplikasi berbasis android. Android Studio dikembangkan oleh Google dan dirilis pertama kali pada tahun 2013. Android Studio adalah salah satu aplikasi yang sangat populer digunakan oleh para pengembang aplikasi berbasis android di seluruh dunia. Android Studio menyediakan berbagai macam fitur yang dapat membantu pengembangan aplikasi android seperti Layout Editor, Gradle Build System, Code Editor, Debugger, dan Emulator (Wenderlich, n.d.).

2.19 Google Colaboratory

Google Colaboratory adalah coding environment yang disediakan oleh Google dengan fasilitas berupa notebook. Notebook yang disediakan oleh google colab dapat melakukan berbagai macam proses pemrograman dalam bahasa

Python. Saat membuat sebuah notebook maka akan otomatis tersimpan kedalam tersimpan kedalam google drive. Google colabory juga bisa memberikan fasilitas untuk berbagi notebook agar dapat bekerja bersama rekan untuk mengedit ataupun mengomentari notebook.

2.20 Penelitian Terdahulu

Beberapa penelitian terdahulu terkait dengan deteksi penyakit pada tanaman jeruk yang menjadi referensi dalam penelitian yang dilakukan.

Militante dkk., (2019) melakukan penelitian mengembangkan model *deep learning* untuk melakukan identifikasi penyakit tanaman melalui gambar daun sederhana dari tanaman yang sehat atau sakit. Pelatihan model dilakukan menggunakan database terbuka yang terdiri dari 87.848 foto yang diambil baik dalam kondisi laboratorium maupun kondisi nyata di lapangan pertanian. Data tersebut terdiri dari 25 spesies tanaman dalam 58 kelas kombinasi (tanaman, penyakit), termasuk beberapa tanaman yang sehat. Arsitektur model paling sukses, yaitu jaringan saraf konvolusi VGG, mencapai tingkat keberhasilan 99,53%.

Penelitian yang dilakukan oleh Natarajan dkk., (2020) berfokus pada meninjau efisiensi arsitektur *deep learning* untuk deteksi hama dan penyakit pada tanaman tomat. Data yang digunakan untuk mengembangkan model terdiri dari 1090 gambar daun tomat waktu nyata yang terinfeksi *early blight*, *leaf curl*, *septoria leafspot*, dan *bacterial spot*. Berbagai arsitektur deep learning digunakan untuk mengidentifikasi hama dan penyakit menggunakan detektor deep: Faster R-CNN, R-FCN, dan SSD, yang digabungkan dengan VGG Net dan ResNet, AlexNet, SqueezeNet. Hasil yang di dapat arsitektur deep learning Faster R-CNN yang digabungkan dengan ResNet memberikan kinerja lebih baik.

Ferentinos, (2018) melakukan penelitian dengan tujuan untuk mendeteksi dan mengenali 32 variasi tanaman dan penyakit-penyakit tanaman yang berbeda menggunakan CNN. Model yang telah dilatih dapat digunakan untuk menguji gambar waktu nyata untuk mendeteksi dan mengenali penyakit-penyakit tanaman. Arsitektur CNN lainnya juga dapat menggunakan laju pembelajaran dan optimizer yang berbeda untuk menguji kinerja dan akurasi model. Dengan akurasi yang

dicapai sebesar 96,5%, model yang diusulkan dapat membantu petani untuk mendeteksi dan mengenali penyakit-penyakit tanaman seperti *Alteria solani* pada tanaman kentang dan tomat, *Diplocarpon earlianum* pada tanaman strawberry, Cucumber mosaic virus (CMV) pada tanaman labu, dan seterusnya.

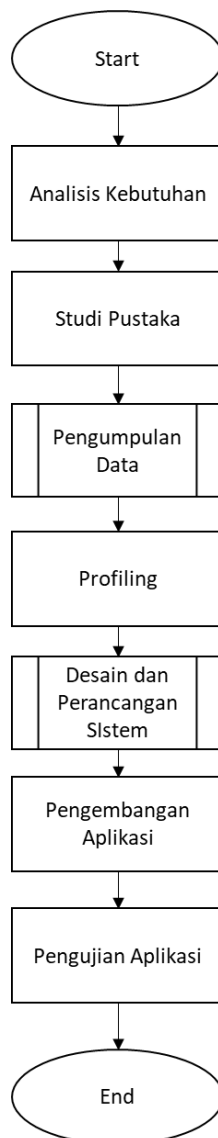
Pada penelitian yang dikerjakan pada topik ini juga memuat tentang deteksi penyakit HLB pada tanaman jeruk dengan menggunakan citra daun tanaman jeruk. Melalui beberapa referensi yang telah dijelaskan, arsitektur yang digunakan untuk topik ini adalah AlexNet, VGG16, dan SqueezeNet. Ketiga arsitektur tersebut dipilih karena memiliki akurasi yang cukup baik dalam melakukan deteksi penyakit pada tanaman menggunakan model *deep learning*. Khusus untuk arsitektur SqueezeNet, dengan tambahan pertimbangan yaitu memiliki ukuran file yang lebih kecil dan membutuhkan waktu yang lebih cepat untuk melakukan prediksi.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Alur Penelitian

Pengerjaan proyek Tugas Akhir ini bertujuan untuk merancang dan membangun aplikasi deteksi penyakit HLB dengan Convolutional Neural Network. Alur penelitian pada proyek ini dapat dilihat pada gambar 3.1.



Gambar 3.1 Alur Penelitian

Penelitian dimulai dengan melakukan memahami lebih dalam permasalahan yang sedang dibahas di penelitian ini. Selanjutnya dilakukan studi pustaka untuk mempelajari jurnal dan artikel dari penelitian yang berkaitan dengan permasalahan dan metode yang sedang dibahas pada penelitian ini. Pengumpulan data adalah langkah selanjutnya, pada tahap ini akan dibahas bagaimana data dikumpulkan dengan perangkat dan metode yang digunakan. Langkah selanjutnya adalah melakukan perancangan model CNN yang akan dikembangkan untuk aplikasi deteksi penyakit HLB.

3.2 Analisis Kebutuhan

Perancangan sistem untuk aplikasi android yang akan dikerjakan memerlukan analisis agar pada pengerjaan proyek dapat dilakukan dengan baik dan benar. Analisis yang dilakukan untuk pengembangan aplikasi android meliputi perangkat dan kebutuhan dari sisi pengguna dan peneliti.

3.2.1 Kebutuhan Pengguna

Kebutuhan Pengguna pada penelitian ini adalah pengguna yang ingin melakukan deteksi penyakit HLB pada tanaman jeruk dengan cepat dan mudah menggunakan smartphone. Pengguna dapat menggunakan aplikasi dengan menekan tombol ambil gambar yang terdapat pada aplikasi lalu dilakukan pengambilan gambar daun tanaman jeruk yang ingin diidentifikasi terkena penyakit HLB atau tidak. Setelah melakukan pengambilan gambar maka aplikasi akan melakukan proses pengenalan kadar pigmen daun tanaman jeruk untuk mendeteksi penyakit HLB.

3.2.2 Kebutuhan Peneliti

Berikut merupakan perangkat-perangkat yang akan digunakan untuk membantu kegiatan penelitian yang dilakukan.

1. Perangkat Keras
 - a. Samsung Galaxy A53

- i. Exynos 1280 (5 nm)
 - ii. Mali-G68 GPU
 - iii. 8GB RAM/256 GB Internal Memory
 - iv. Android 12, One UI 5.1
- b. Laptop Acer Nitro 5
 - i. Processor: AMD Ryzen 7 3750
 - ii. RAM: 8GB DDR4-2400
- c. Digital Lux Meter AR823
- d. Colorimeter
 - i. Model: WR-10
 - ii. Color Space: CIELAB
 - iii. Sensor: Photodiode array
- e. Colorchecker
- 2. Perangkat Lunak
 - a. Python 3
 - b. Java
 - c. Android Studio
 - d. Google Colaboratory

3.3 Studi Pustaka

Pada tahap ini dilakukan studi pustaka mengenai penelitian-penelitian terdahulu serta referensi-referensi yang akan menjadi landasan dalam menyelesaikan permasalahan pada penelitian ini. Penelitian yang menjadi rujukan adalah penelitian yang berkaitan dengan deteksi penyakit tanaman dengan menggunakan metode CNN. Terdapat tiga rujukan utama dalam mengembangkan aplikasi pada penelitian ini yaitu tentang deteksi penyakit pada tanaman menggunakan Deep Learning dengan algoritma CNN.

Tabel 3.1 Ringkasan Penelitian Terdahulu




No	Topik	Pengetahuan	Temuan
1	Deteksi Daun Tanaman dan Pengenalan Penyakit dengan Deep Learning	Pemanfaatan Algoritma Deep Learning yaitu Convolutional Neural Network untuk Mengidentifikasi penyakit tanaman melalui gambar daun dari tanaman sehat atau tidak. Model dikembangkan dengan menggunakan arsitektur VGG. Model memiliki tingkat keberhasilan deteksi sebesar 99,53%.	Pemanfaatan Algoritma CNN untuk melakukan identifikasi terhadap penyakit tanaman (Militante et al., 2019).
2	Deteksi Penyakit pada Tanaman Tomat Menggunakan Teknik Deep Learning	Membandingkan beberapa arsitektur seperti VGG Net, ResNet, AlexNet, dan SqueezeNet untuk melakukan deteksi penyakit pada tanaman tomat dan digabungkan dengan detektor deep seperti R-CNN, R-FCN, dan SSD menggunakan 1090 gambar daun tomat.	Penggunaan detektor deep dan Arsitektur CNN untuk deteksi penyakit pada tanaman tomat (Natarajan et al., 2020).
3	Model Deep Learning untuk Deteksi Penyakit Tanaman	Mendeteksi dan mengenali 32 penyakit pada tanaman yang berbeda dengan menggunakan metode CNN yang dilatih dengan optimizer yang berbeda. Hasil akurasi yang dihasilkan oleh model sebesar 96,5%.	Penggunaan optimizer yang berbeda untuk menguji kinerja dan akurasi model (Ferentinos, 2018).

3.4 Pengumpulan Data

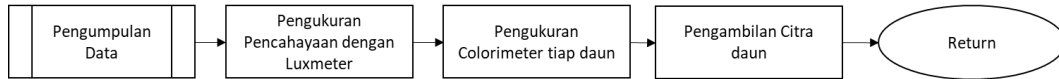
Pada tahap pengumpulan data yang akan digunakan untuk mengembangkan model CNN. Data yang akan digunakan adalah gambar atau citra daun tanaman jeruk yang dibagi menjadi 3 yaitu: daun dari tanaman jeruk yang sehat, daun dari tanaman jeruk yang terkena penyakit HLB, dan daun tanaman jeruk yang tidak sehat namun bukan terkena HLB. Pemilihan daun yang akan diproses dalam

pembuatan dataset dibantu oleh pakar yang ahli dalam mengenali penyakit tanaman jeruk dengan melihat warna pada daun. Contoh data yang berhasil diambil dapat dilihat pada tabel 3.2.

Tabel 3.2 Contoh Data pada Tiga Kondisi Tanaman Jeruk

Citra Daun	Kondisi Tanaman Jeruk
	Sehat
	Tidak Sehat (Bukan HLB)
	HLB

Pengambilan citra dataset dilakukan dengan menggunakan kamera smartphone yang spesifikasinya ada subbab kebutuhan peneliti. Pengambilan gambar dilakukan langsung di perkebunan jeruk dengan menggunakan lux meter, dan colorimeter. Tahapan Prosesnya dapat dilihat pada gambar 3.2.



Gambar 3.2 Alur Pengumpulan Data

Berdasarkan gambar 3.2 Daun dari tanaman jeruk mulai dari yang sehat, tidak sehat tapi bukan HLB, dan yang terkena HLB dikumpulkan. Hal yang dilakukan pertama adalah mengukur kecerahan cahaya menggunakan luxmeter dengan tujuan untuk mengetahui data di ambil pada tingkat kecerahan berapa dan mengetahui data di ambil pada kondisi pencahayaan apa. Kondisi pencahayaan yang berbeda-beda menghasilkan pantulan warna yang berbeda sehingga dibutuhkan koreksi pencahayaan. Setelah dilakukan pengukuran cahaya dilanjutkan dengan setiap daun yang akan diproses adalah memberikan label dan diteruskan pada dilakukan pengukuran warna dengan menggunakan colorimeter untuk mengetahui warna asli yang dipantulkan oleh data daun tanaman jeruk. Hasil yang diberikan oleh colorimeter berupa ruang warna dengan format Lab.

3.5 *Profiling*

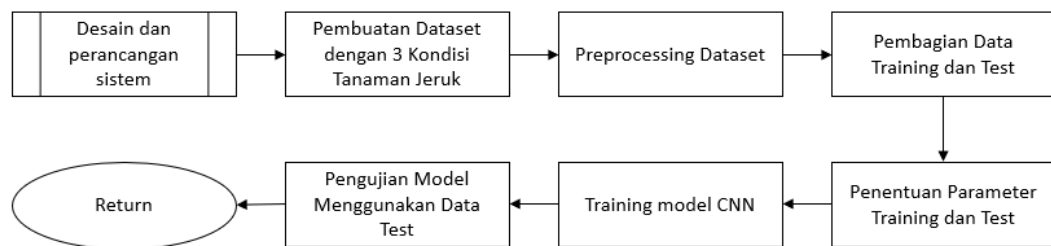
Pada proses ini dilakukan pembuatan profil pada setiap kategori daun dengan menggunakan data colorimeter yang diambil bersamaan dengan pengumpulan data citra daun tanaman jeruk. *Profiling* bertujuan untuk mendapatkan perbedaan dari ketiga kelas data citra daun. Data *colorimeter* yang dimiliki akan diubah dari Lab ke format RGB lalu dihitung rata-rata dan standar deviasinya untuk mendapatkan perbedaan dari ketiga kelas data citra daun.

Selain dengan menggunakan warna dari data *colorimeter*, *Profiling* yang dilakukan adalah mendapatkan nilai tekstur pada citra daun setiap kelasnya menggunakan metode *Grey Level Co-occurrence Matrix* atau GLCM. Citra daun

tanaman jeruk yang sudah diambil akan dimasukan ke dalam metode GLCM untuk mendapatkan beberapa nilai texture. Setelah Nilai didapatkan maka selanjutnya adalah melakukan perbandingan pada setiap kondisi tanaman jeruk.

3.6 Desain dan Perancangan Sistem

Pengembangan proyek ini terdiri dari sebuah model yang dapat melakukan deteksi penyakit HLB pada tanaman jeruk. Model dikembangkan dengan menggunakan python sampai mendapatkan model yang sesuai untuk melakukan deteksi penyakit HLB pada tanaman jeruk. Untuk membuat sebuah model tersebut maka dilakukan beberapa tahap yang dapat dilihat pada gambar 3.3



Gambar 3.3 Alur Desain dan Perancangan Sistem

3.6.1 Pembuatan Dataset

Pada tahap ini dilakukan pembuatan dataset dari hasil pengumpulan data citra daun tanaman jeruk. Data akan dikelompokkan menjadi tiga yaitu sehat, tidak sehat tapi bukan HLB, dan HLB. Setelah itu dilakukan juga augmentasi data dengan melakukan *flip* dan *random rotation*. Tujuan dilakukannya augmentasi adalah untuk memperbanyak jumlah dan variasi data pada dataset.

3.6.2 Preprocessing

Tahap ini dilakukan setelah mengumpulkan data untuk memproses data sebelum masuk kedalam tahap pelatihan model CNN. Tahap ini dilakukan beberapa proses yaitu mengumpulkan gambar sesuai dengan kondisi tanaman jeruk, dilakukan juga resize untuk menyamakan ukuran gambar yang masuk ke dalam

proses pelatihan model CNN. Ukuran yang ditetapkan adalah 456×656 . Dengan tujuan adalah untuk mempercepat proses pelatihan sehingga model tidak menghabiskan banyak waktu dalam memproses satu gambar. Langkah selanjutnya data yang telah melalui proses resize akan dimasukkan kedalam proses koreksi pencahayaan untuk menyamakan seluruh kondisi pencahayaan pada data yang telah dikumpulkan. Pada proses koreksi cahaya metode yang digunakan adalah menggunakan model yang telah dikembangkan pada penelitian sebelumnya yaitu koreksi cahaya menggunakan model CNN (Pradana, 2023). Sebagai acuannya Tabel 3.3 adalah tingkat pencahayaan yang telah ditentukan.

Tabel 3.3 Acuan Tingkat Pencahayaan

Tingkat Pencahayaan	<i>Lux</i>		
	Minimal	Average	Maximal
Sangat Gelap	0	7	10
Gelap	30	45	70
Normal	500	550	700
Terang	40000	65000	70000
Sangat Terang	90000	100000	100000

3.6.3 Pelatihan Model CNN

Setelah melakukan tahap *preprocessing*, maka tahap selanjutnya adalah melakukan proses pelatihan model CNN. Pada Pelatihan model dilakukan dengan data citra daun jeruk yang telah di proses pada tahap sebelumnya. Masing-masing data memiliki kondisi tanaman jeruk. Tiap kondisi dari tanaman jeruk ini akan di ekstraksi cirinya untuk dipelajari oleh model. Beberapa arsitektur yang digunakan untuk membuat model yaitu AlexNet, VGG16, dan SqueezeNet. Terdapat tiga *node output* dari model yang berhasil dibuat yaitu sehat, tidak sehat tapi bukan HLB, dan HLB. Susunan dari setiap arsitektur dapat dilihat pada gambar 2.8, 2.9, dan 2.10. Metode optimasi yang digunakan pada penelitian ini adalah Adam dan Adamax. Sedangkan fungsi *loss* yang digunakan adalah SparseCategoricalCrossentropy karena label yang dimiliki lebih dari dua label.

3.6.4 Pengujian Model Menggunakan Data Test

Setelah model berhasil dibuat maka tahap selanjutnya adalah melakukan evaluasi model dengan melakukan pengujian terhadap model. Pengujian dilakukan dengan melakukan deteksi menggunakan data testing yang tidak digunakan dalam proses pelatihan model. Setelah itu akan menghasilkan akurasi, precision, dan recall dari model dalam melakukan proses deteksi penyakit seperti *confusion matrix* yang dapat dilihat pada tabel 3.4. Selain mengevaluasi hasil dari ketiga indikator tersebut, evaluasi juga dilakukan dengan melihat apakah model terjadi *overfitting* saat dilakukan tahap pengujian.

Tabel 3.4 Confusion Matrix untuk Pengujian Model

		Predicted		
		Bukan HLB	HLB	Sehat
True Class	Bukan HLB			
	HLB			
	Sehat			

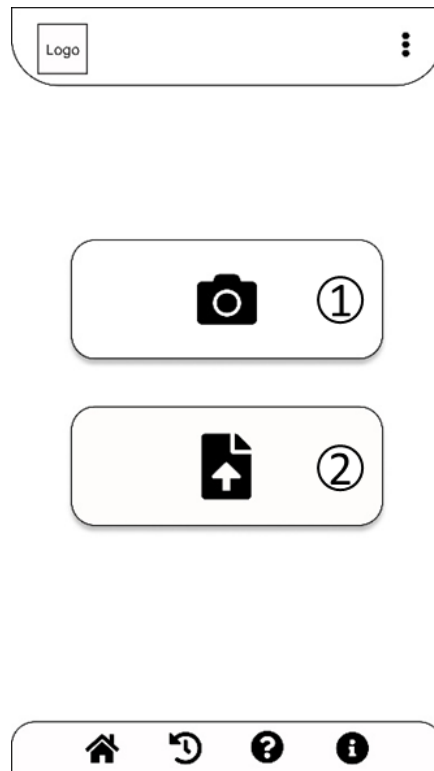
3.7 Pengembangan Aplikasi

Pada tahap pengembangan aplikasi, model yang telah berhasil dibuat akan di implementasikan ke dalam aplikasi. Model CNN akan bekerja untuk melakukan identifikasi apakah daun tanaman jeruk yang difoto akan mengeluarkan keluaran identifikasi ke dalam tanaman jeruk yang sehat, tidak sehat tapi bukan HLB, atau terjangkit penyakit HLB. Untuk pengembangan aplikasi android akan digunakan bahasa pemrograman java.

3.7.1 Mockup Aplikasi

Tahap ini menjelaskan seperti apa rancangan mockup yang dikerjakan oleh peneliti dan kegunaan dari setiap tampilan yang terdapat pada aplikasi. tampilan halaman utama akan muncul seperti pada gambar 3.8. Pada halaman tersebut

terdapat dua tombol utama yaitu tombol untuk ambil foto atau nomor (1) pada gambar 3.4 dan ambil gambar dari galeri atau nomor (2) pada gambar 3.8.



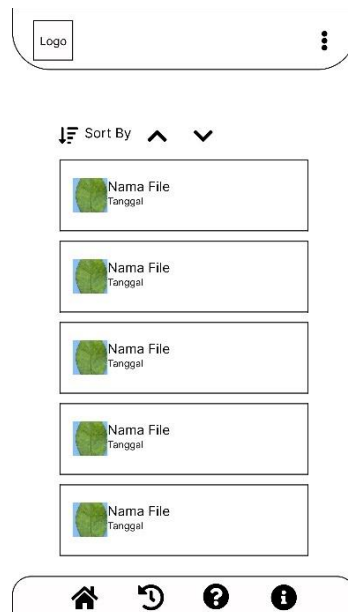
Gambar 3.4 Halaman Utama

Jika pengguna menekan tombol ambil gambar maka pengguna akan dialihkan ke kamera. Setelah mengambil gambar, pengguna dapat melihat hasilnya berupa hasil prediksi dan seberapa besar tingkat keyakinan prediksi dari aplikasi atau nilai confidence. Nilai confidence ditunjukkan untuk memberikan pertimbangan kepada pengguna apakah akan mempercayai hasil prediksi dari aplikasi atau tidak. Untuk tampilan pada halaman prediksi yaitu gambar 3.5.



Gambar 3.5 Halaman Foto

Selain terdapat halaman utama, ada pula halaman lainnya yaitu halaman riwayat, bantuan, dan tentang (*about*). Halaman riwayat memiliki fungsi untuk melihat hasil prediksi kondisi tanaman jeruk. Ilustrasi mockup dapat dilihat pada gambar 3.6.



Gambar 3.6 Halaman Riwayat

Halaman bantuan berisikan informasi tentang cara menggunakan aplikasi mulai dari pengambilan gambar hingga hasil prediksi kondisi tanaman jeruk yang dapat dilihat seperti pada gambar 3.7.



Gambar 3.7 Halaman Bantuan

Sedangkan halaman tentang (about) berisikan informasi singkat mengenai aplikasi yang dapat dilihat pada gambar 3.8.



Gambar 3.8 Halaman Tentang (about)

3.8 Pengujian Aplikasi

Tahap ini dilakukan untuk melihat seberapa baik kinerja aplikasi dalam melakukan prediksi kondisi tanaman jeruk. Pengujian dilakukan menjadi dua bagian yaitu pengujian terhadap *User Interface* dan kinerja model.

3.7.2 Evaluasi *User Interface* Aplikasi

Pengujian terhadap aplikasi juga dilakukan seperti mengukur seberapa baik tombol berfungsi, seberapa cepat halaman berpindah ketika tombol ditekan, apakah kamera berfungsi dengan baik, dan apakah gambar berhasil disimpan. Keseluruhan evaluasi dirangkum dan ditampilkan pada tabel 3.4.

Tabel 3.5 Tabel Evaluasi UI Aplikasi

No	Keterangan	Fitur	Berfungsi	Delay
1	Home Screen	Take Photo Button		
		Upload Photo Button		
		History Page Button		
		Guide Page Button		
		Credit Page Button		
		More Action Button		
2	Take Photo Page	Take Photo Button		
		Home Page Button		
		History Page Button		
		Guide Page Button		
		Credit Page Button		
3	Prediction Result Screen	Delete Image Button		
4	History Screen	Sort By Button		
		Home Page Button		
		History Page Button		
		Guide Page Button		
		Credit Page Button		
		More Action Button		
5	Guide Screen	Next Button		
		Previous Button		
		Home Page Button		
		History Page Button		
		Guide Page Button		
		Credit Page Button		
6	About Screen	More Action Button		
		Home Page Button		
		Photo Page Button		
		History Page Button		
		Guide Page Button		

3.7.3 Evaluasi Kinerja Model

Evaluasi kinerja model dilakukan untuk melihat seberapa besar ukuran file atau size model, akurasi, dan seberapa lama waktu yang diperlukan model untuk melakukan kalibrasi satu gambar dengan menggunakan aplikasi yang telah dibuat. Evaluasi kinerja model setelah di aplikasikan pada aplikasi android dirangkum pada tabel 3.6.

Tabel 3.6 Tabel Evaluasi Kinerja Model CNN

Model	Ukuran File	Waktu
AlexNet_Adam		
AlexNet_Adamax		
VGG16_Adam		
VGG16_Adamax		
SqueezeNet_Adam		
SqueezeNet_Adamax		

BAB IV

HASIL DAN PEMBAHASAN

4.1 Dataset

Seperti yang dijelaskan pada bab sebelumnya bahwa dataset berisi 3 kelas yaitu Sehat, HLB, dan Bukan HLB. Pengambilan nilai colorimeter juga dilakukan. Setiap daun akan diambil 3 titik dengan menggunakan bantuan alat colorimeter. Pengambilan colorimeter ini juga berguna untuk mendapatkan ciri-ciri dari setiap kondisi dari tanaman jeruk dengan melakukan *Profiling*. Secara keseluruhan pembagian citra daun jeruk pada dataset HLB, Sehat, dan Bukan HLB untuk *training*, *validation*, dan *testing* dapat dilihat pada tabel 4.1.

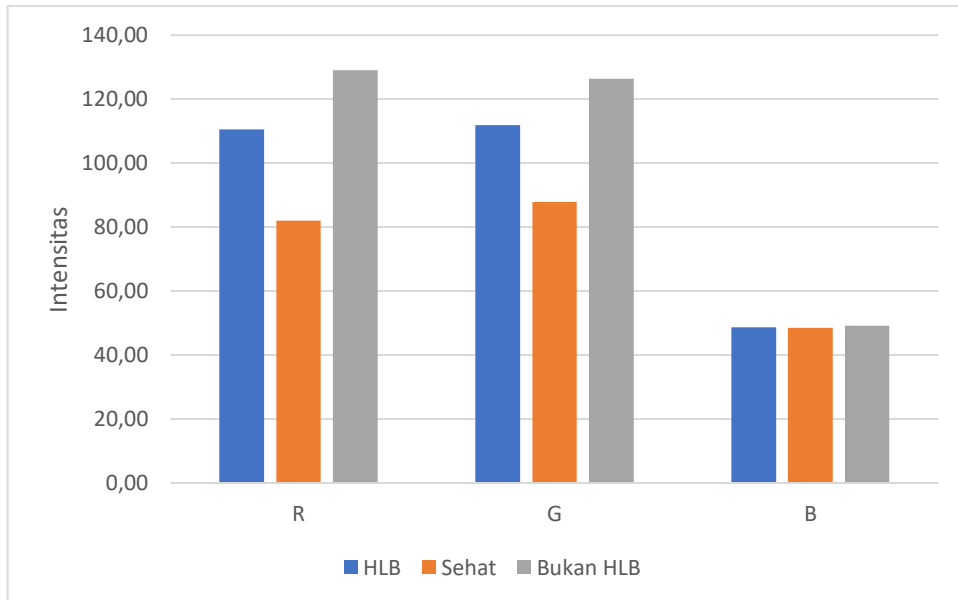
Tabel 4.1 Jumlah Data Train, Test, dan Valid

	Train	Test	Valid	Total
HLB	61	8	8	77
Sehat	30	4	4	38
Bukan HLB	15	2	2	19

Jumlah dataset tersebut dihitung berdasarkan pembagian dataset yaitu 80% untuk dataset *training*, 10% untuk *validation*, dan 10% untuk dataset test. Pembagian tersebut dilakukan berdasarkan jumlah data yang terbatas, sehingga membutuhkan lebih banyak data train untuk mendapatkan akurasi yang baik.

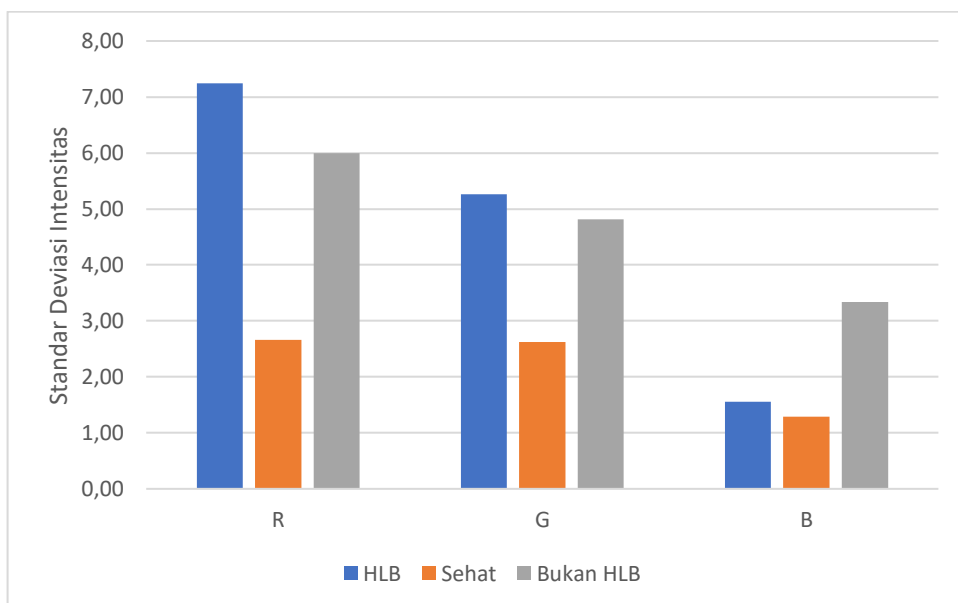
4.2 Profiling

Hasil dari *Profiling* ini digunakan untuk mengetahui perbedaan dari setiap kondisi dengan menggunakan perolehan data dari alat *colorimeter*. Berikut adalah hasil dari *Profiling* dengan data *colorimeter*.



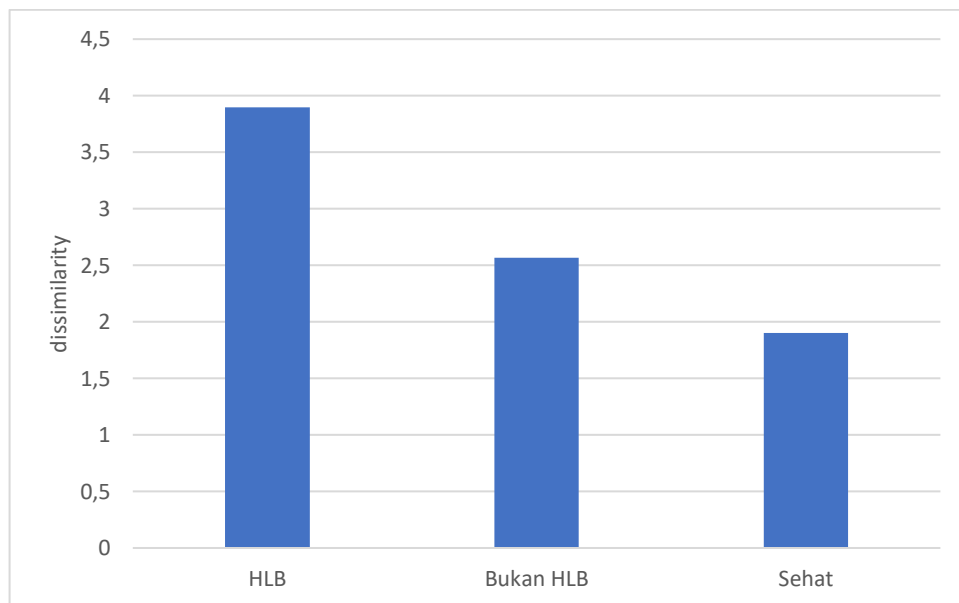
Gambar 4.1 Grafik Rata-rata RGB Data Colorimeter

Berdasarkan gambar 4.1, Hasil rata-rata dari nilai RGB pada setiap kondisi daun tanaman jeruk menunjukkan bahwa nilai red dan green yang tampak menunjukkan perbedaan. Nilai red dan green pada daun tanaman jeruk dalam kondisi sehat memiliki nilai yang lebih rendah dibandingkan dengan HLB dan bukan HLB. Untuk kondisi bukan HLB memiliki nilai red dan green yang paling tinggi.



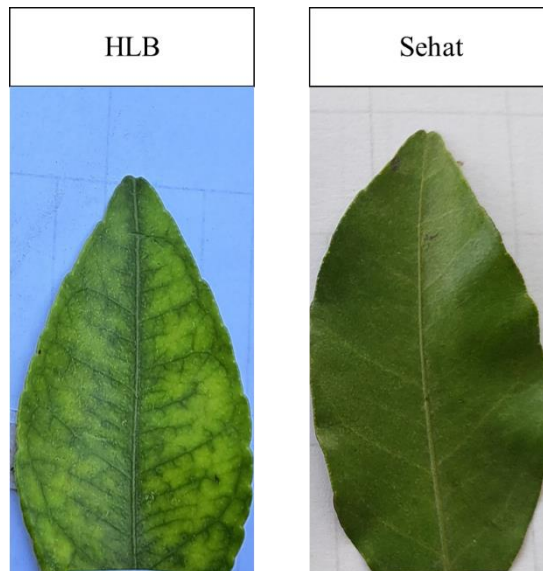
Gambar 4.2 Grafik Standar Deviasi RGB Data Colorimeter

Pada gambar 4.2 menjelaskan persebaran nilai RGB pada setiap kondisi daun tanaman jeruk. Untuk warna red dan green pada kondisi sehat memiliki persebaran nilai yang paling rendah dibandingkan dengan kondisi HLB dan bukan HLB. Nilai red pada kondisi HLB memiliki nilai red dan green yang paling tinggi. Hal tersebut membuktikan bahwa warna pada daun yang terjangkit penyakit HLB memiliki persebaran warna daun yang acak dan tidak merata. Selain menggunakan data dari colorimeter proses *Profiling* juga dilakukan dengan metode Gray-Level Co-Occurrence Matrix atau GLCM. Berikut hasil dikeluarkan oleh GLCM untuk setiap kondisi pada daun tanaman jeruk.



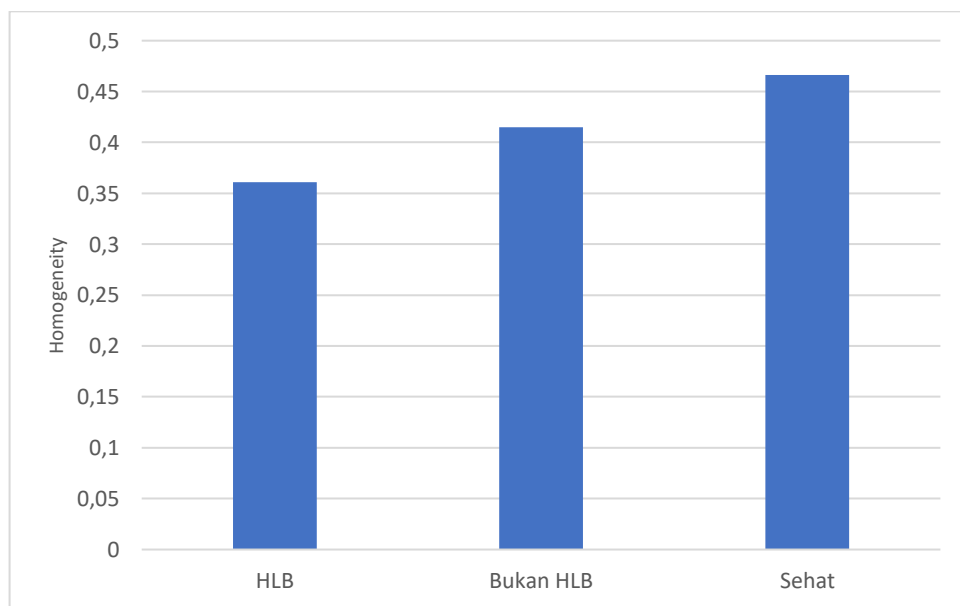
Gambar 4.3 Grafik Hasil GLCM *Dissimilarity*

Berdasarkan gambar 4.3, Hasil GLCM berfokus pada dissimilarity dan homogeneity. Hasil dissimilarity menunjukkan bahwa daun yang terjangkit HLB memiliki nilai yang paling besar karena memiliki pola yang acak. Sedangkan pada kondisi sehat nilai dissimilarity memiliki nilai yang paling rendah karena daun memiliki pola yang sama. Untuk melihat perbedaannya dapat dilihat pada gambar 4.4.



Gambar 4.4 Daun dengan Kondisi HLB dan Sehat

Pada gambar 4.4 dapat dilihat bahwa daun dengan kondisi HLB memiliki warna pada daun yang tidak rata atau memiliki bercak-bercak kuning pada daun. Hal itu menyebabkan nilai *dissimilarity* menjadi tinggi. Sedangkan pada kondisi sehat daun memiliki warna yang sama atau rata dengan warna hijau sehingga memiliki nilai *dissimilarity* yang lebih rendah. Selanjutnya GLCM untuk hasil *homogeneity* dapat dilihat pada gambar 4.5.



Gambar 4.5 Grafik Hasil GLCM *Homogeneity*

Pada homogeneity dapat dilihat pada grafik pada kondisi sehat nilai homogeneity memiliki nilai paling tinggi. Hal tersebut disebabkan karena sebagian besar pixel mempunyai nilai yang sama / seragam. Sedangkan pada kondisi HLB memiliki nilai homogeneity paling rendah karena pixel yang bernilai tidak seragam.

4.3 Implementasi CNN

Implementasi pada CNN menggunakan *library* Keras sebagai sarana membangun arsitektur, melakukan *preprocessing* data, *training*, dan *testing*. Penulisan kode dilakukan dengan menggunakan *google colab* untuk mempercepat proses training dengan menggunakan GPU *google colab*. Implementasinya terdiri dari beberapa arsitektur yaitu AlexNet, VGG16, dan SqueezeNet. Sebagai contoh dalam kode untuk membangun arsitektur AlexNet dan SqueezeNet dapat dilihat pada lampiran nomor 1 dan 2. Untuk arsitektur VGG16 menggunakan *library* keras dari tensorflow untuk membangun arsitektur.

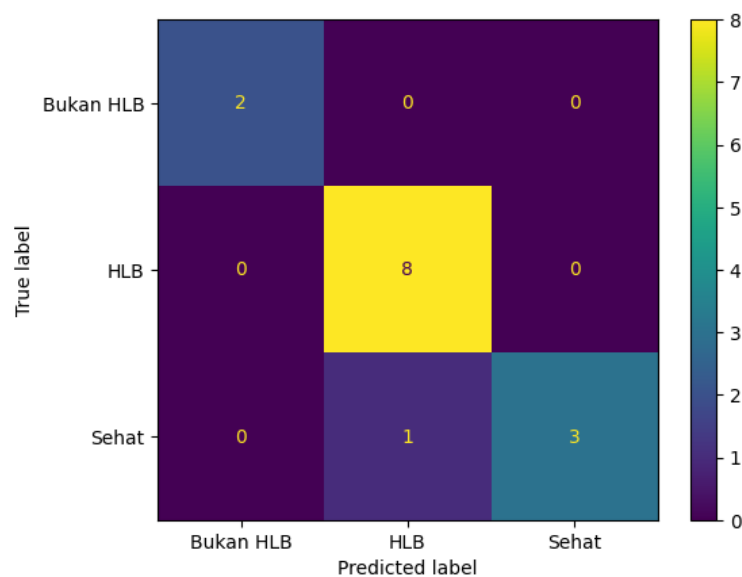
4.4 Augmentasi

Pada perancangan sistem 3.6.2, augmentasi digunakan untuk memperbanyak variasi dari dataset. Pada pengaplikasiannya dibutuhkan pengujian terlebih dahulu apakah augmentasi bisa memperbaiki kinerja model atau tidak. Pengujian dilakukan dengan menggunakan salah satu arsitektur dan teknik optimasi yaitu arsitektur VGG16 dan teknik optimasi Adam dengan jumlah epoch sebanyak 30 epoch. Untuk melihat perbandingan kinerja model dengan menggunakan augmentasi dan tanpa augmentasi dapat dilihat pada tabel 4.2.

Tabel 4.2 Perbandingan Nilai Akurasi dan Loss Model Dengan Augmentasi dan Tanpa Augmentasi

	Tanpa Augmentasi	Augmentasi
Akurasi Training	96.2%	89.6%
Loss Training	0.1822	0.337
Akurasi Validation	85.7%	85.7%
Loss Validation	0.2440	0.3923
Akurasi Test	92.8%	92.8%
Loss Test	0.2103	0.2834

Berdasarkan tabel 4.2, dapat dilihat pada blok yang diberi warna abu-abu merupakan kinerja model yang lebih baik atau sama dengan model lainnya. Model yang dilatih dengan menggunakan data tanpa augmentasi memiliki kinerja yang lebih baik dimana pada loss training, validation, dan test lebih kecil dibandingkan dengan model yang dilatih dengan data augmentasi. Untuk akurasi training model tanpa augmentasi memiliki hasil yang lebih baik sedangkan akurasi validation dan test memiliki hasil yang sama dengan model dengan augmentasi. Dengan hasil pengujian diatas maka diputuskan bahwa tidak akan menggunakan augmentasi data pada pelatihan model. Untuk hasil pengujian dengan menggunakan data test dapat dilihat pada gambar 4.6



Gambar 4.6 Confusion Matrix Data Test dengan Model Augmentasi

4.5 Kalibrasi

Pada perancangan sistem 3.6.2, kalibrasi digunakan untuk mengatasi perbedaan pencahayaan pada data citra daun tanaman jeruk. Sebelum mengimplementasikan pada model yang akan digunakan pada aplikasi android maka perlu dilakukan pengujian apakah kalibrasi memang diperlukan dan meningkatkan kinerja model pada deteksi penyakit HLB menggunakan CNN. Contoh dari hasil proses kalibrasi dapat dilihat pada gambar 4.7.



Gambar 4.7 Sebelum dan Sesudah Kalibrasi pada Daun HLB

Pada gambar 4.7 dapat dilihat sebelum kalibrasi memiliki pencahayaan yang lebih terang sehingga setelah melalui proses kalibrasi Cahaya diturunkan menjadi gambar sesudah kalibrasi. Namun pada hasil, gambar yang dihasilkan oleh model kalibrasi mengubah warna pada daun khususnya daun dengan kondisi HLB dan bukan HLB. Daun yang awalnya memiliki warna kuning di beberapa bagian berubah menjadi hijau. Hal ini disebabkan oleh model kalibrasi hanya dilatih dengan daun yang tidak memiliki pola pada daunnya atau hijau polos. Citra diambil pada tingkat kecerahan 70-90 lux atau pada kondisi gelap berdasarkan pada penelitian (Prasetyo, 2022), sehingga kalibrasi cukup berpengaruh pada dataset yang dimiliki. Untuk melakukan perbandingan, dilakukan pelatihan model menggunakan data yang telah dikalibrasi dan data yang tanpa melewati kalibrasi. Untuk hasil pengujian model tanpa kalibrasi dan tanpa dapat dilihat pada tabel 4.3.

Tabel 4.3 Perbandingan Nilai Akurasi dan Loss Model Dengan Kalibrasi dan Tanpa Kalibrasi

	Tanpa Kalibrasi	Kalibrasi
Akurasi Training	96.2%	84.2%
Loss Training	0.1822	0.4910
Akurasi Validation	85.7%	71.4%
Loss Validation	0.2440	0.7640
Akurasi Test	92.8%	78.57%
Loss Test	0.2103	0.6164

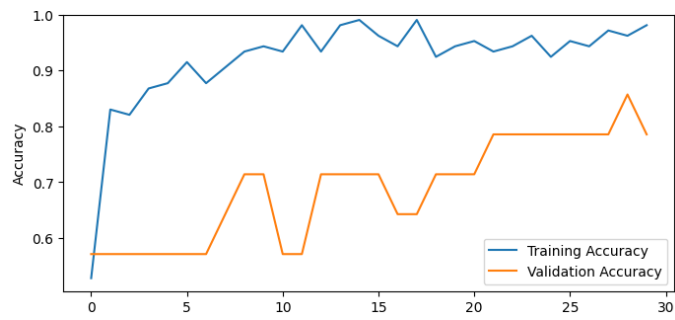
Berdasarkan tabel 4.3, semua hasil yang menunjukkan bahwa model yang dilatih dengan data tanpa melewati proses kalibrasi pencapaian memiliki kinerja yang lebih baik dibandingkan dengan model yang dilatih dengan data melewati proses kalibrasi pencapaian. Akurasi dari model yang dilatih dengan data tanpa kalibrasi memiliki nilai yang lebih tinggi baik pada training, validation, ataupun test. Begitu juga pada loss training, validation, dan test model tanpa kalibrasi memiliki nilai yang lebih rendah dibandingkan dengan model yang dilatih dengan data kalibrasi. Dengan hasil yang ditunjukkan, diputuskan model tidak akan melalui proses kalibrasi.

4.6 Pengujian Kinerja Model

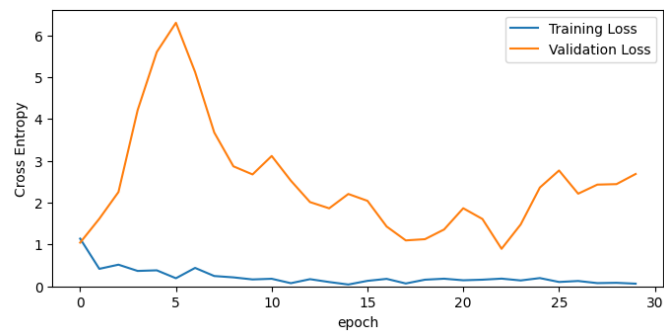
Model untuk melakukan deteksi penyakit HLB terdiri dari enam macam model dengan parameter optimasi adam dan adamax yang menggunakan tiga arsitektur yang berbeda-beda. Pengujian kinerja model dilakukan dengan menggunakan data *test*. Hasil dari pengujian menggunakan data *test* adalah berupa akurasi dan confusion matrix.

4.6.1 AlexNet_Adam

Model AlexNet_Adam dibuat dengan arsitektur AlexNet dan menggunakan teknik optimasi Adam dengan jumlah epoch yang dimiliki sebanyak 30. Pada akhir proses training, nilai akurasi dan loss pada training dan test bisa dilihat pada gambar 4.8.



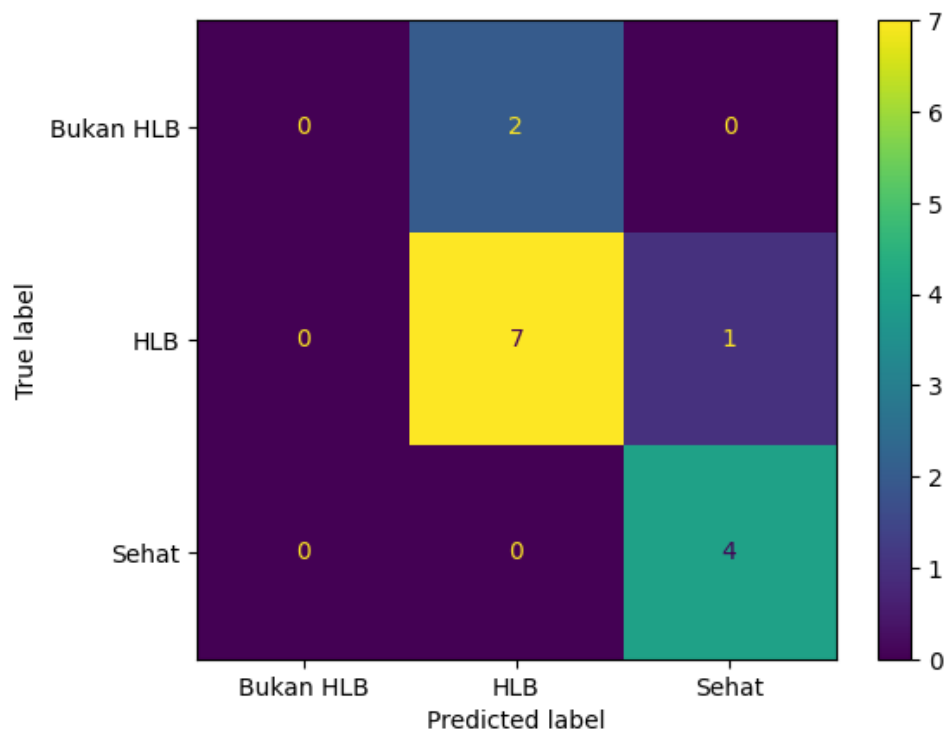
(a)



(b)

Gambar 4.8 Grafik Indikator Kinerja AlexNet_Adam (a) Akurasi dan (b) Loss

Berdasarkan Gambar 4.8, bisa dilihat bahwa nilai akurasi *training* dan *validation* mengalami peningkatan dan nilai loss mengalami penurunan. Pada akhir proses train hasil yang di dapat adalah akurasi train sebesar 98% dan akurasi validasi sebesar 78,5%. Selain itu nilai loss yang di hasilkan untuk training sebesar 0.0644 dan nilai loss untuk validasi sebesar 2.6855. Model dengan arsitektur AlexNet dan teknik optimasi Adam memiliki tingkat akurasi training yang tinggi namun validasi masih belum bisa mencapai akurasi dan loss yang baik seperti training. Untuk pengujian menggunakan data test mendapatkan akurasi sebesar 78.5% dengan confusion matrix yang bisa dilihat pada gambar 4.9.

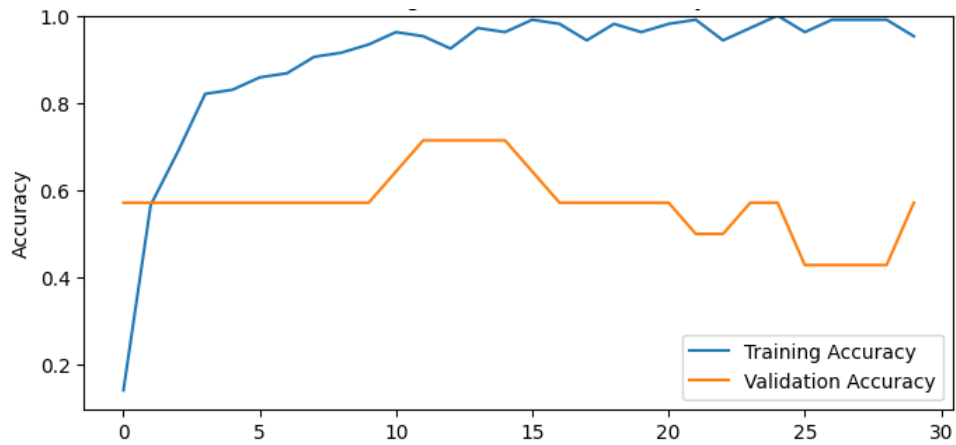


Gambar 4.9 Confusion Matrix Data Test AlexNet_Adam

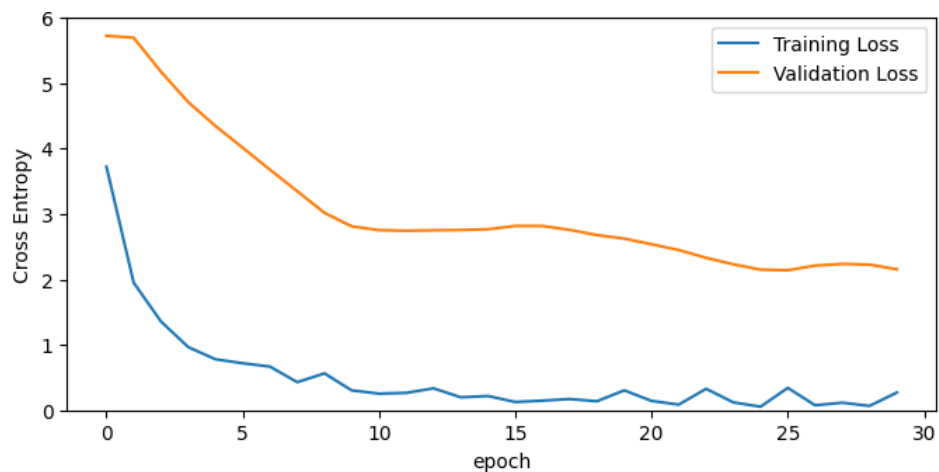
Berdasarkan gambar 4.9, dapat dilihat bahwa model mampu memprediksi benar untuk seluruh data test citra daun tanaman jeruk yang sehat. Namun untuk citra daun tanaman jeruk pada kondisi HLB mampu diprediksi benar sebanyak 7 citra dan 1 citra diprediksi sehat. Model masih belum bisa memprediksi untuk citra daun tanaman jeruk dengan kondisi bukan HLB, karena semua citra diprediksi HLB.

4.6.2 AlexNet_Adamax

Model AlexNet_Adamax dibuat dengan menggunakan arsitektur AlexNet dan metode optimasi Adamax dengan jumlah epoch sejumlah 30. Pada akhir proses training, nilai akurasi dan loss pada training bisa dilihat pada gambar 4.10.



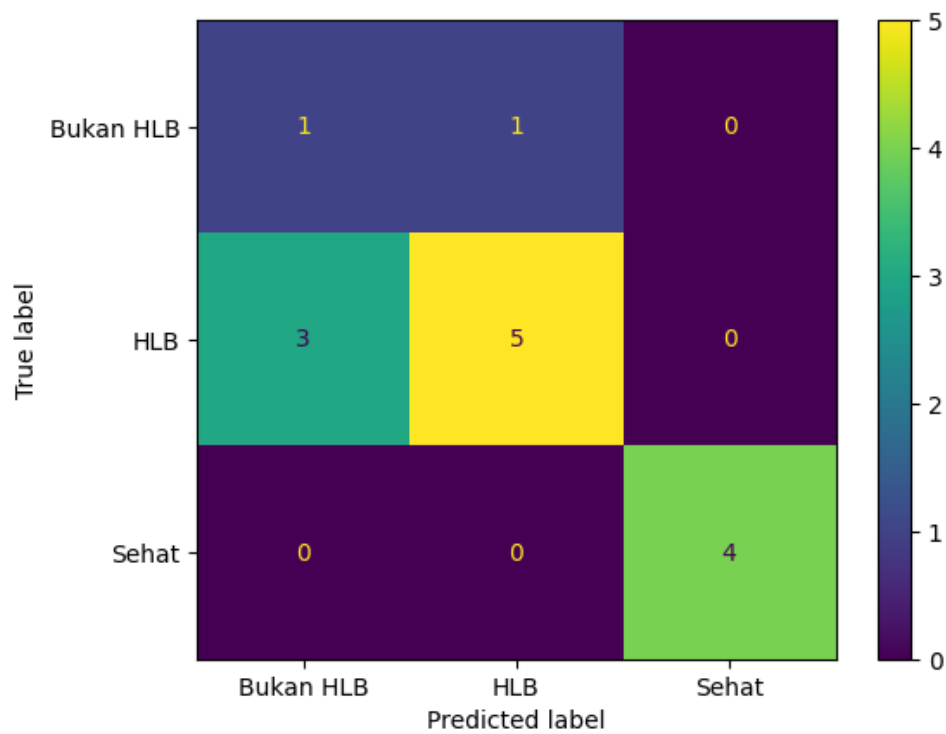
(a)



(b)

Gambar 4.10 Grafik Indikator Kinerja AlexNet_Adamax (a) Akurasi dan (b) Loss

Berdasarkan gambar 4.10 nilai akurasi untuk train mengalami peningkatan yang cukup tinggi sampai 95.2% namun untuk akurasi validasi tidak mendapatkan peningkatan yang sama dengan train dan hanya mencapai 57.1%. Nilai loss pada training mengalami penurunan yang cukup signifikan dengan mencapai 0.27 sedangkan untuk validasi penurunan loss terjadi namun tidak terlalu signifikan dan hanya mencapai 2.155. Model dengan arsitektur AlexNet dan metode optimasi Adamax memiliki akurasi yang tinggi pada training namun akurasi cukup rendah pada validasi. Pengujian model menggunakan data test mendapatkan akurasi sebesar 71.4% dengan *confusion matrix* yang bisa dilihat pada gambar 4.11.

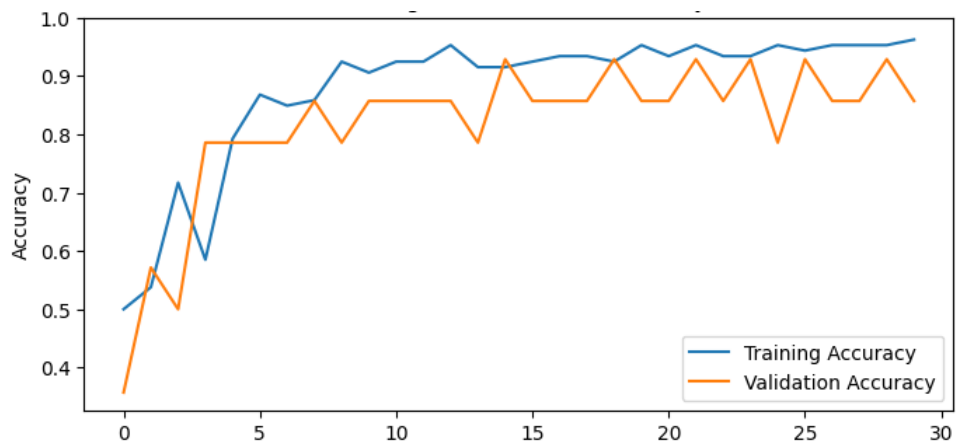


Gambar 4.11 Confusion Matrix Data Test AlexNet_Adamax

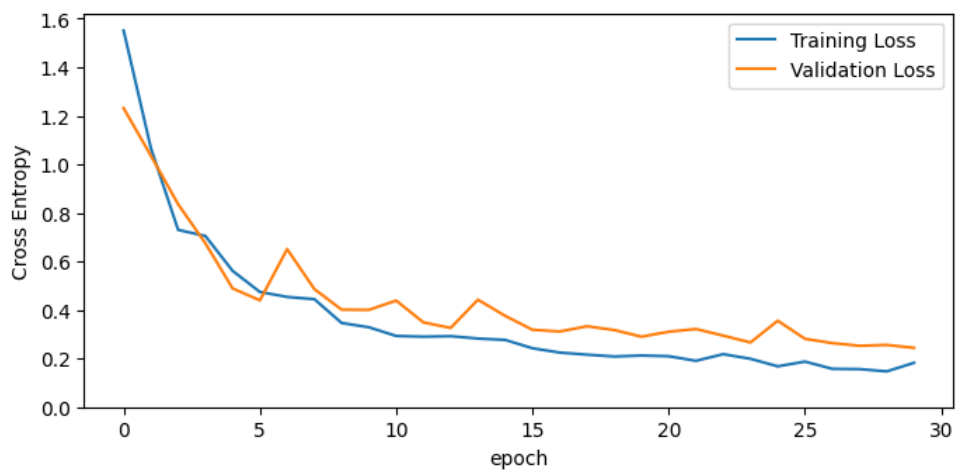
Pada gambar 4.11 dapat dilihat bahwa model AlexNet dengan metode optimasi Adamax mampu memprediksi seluruh data citra daun tanaman jeruk dengan kondisi sehat. Untuk data citra daun tanaman jeruk dengan kondisi HLB, dari 8 citra model berhasil memprediksi benar sebanyak 5 citra. 3 citra gagal diprediksi HLB dan model memprediksi bahwa daun termasuk bukan HLB. Citra daun dengan kondisi bukan HLB berhasil diprediksi benar sebanyak 1 citra dan tidak berhasil diprediksi benar sebanyak 1 citra.

4.6.3 VGG16_Adam

Model VGG16_Adam merupakan model yang dibangun dengan menggunakan arsitektur VGG16 dan teknik optimasi Adam. Jumlah epoch yang digunakan untuk membangun model ini adalah sebanyak 30 epoch. Pada akhir proses training, nilai akurasi dan loss pada training dan test bisa dilihat pada gambar 4.12.



(a)

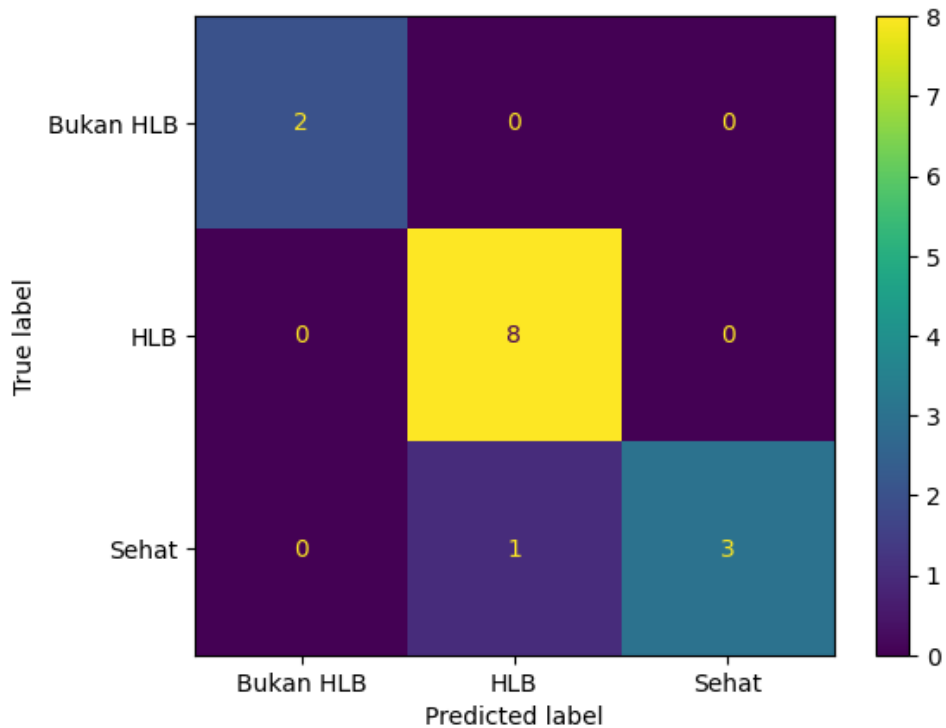


(b)

Gambar 4.12 Grafik Indikator Kinerja VGG16_Adam (a) Akurasi dan (b) Loss

Gambar 4.12 menjelaskan bahwa nilai akurasi dari training dan validation mengalami peningkatan yang cukup baik secara bersamaan. Akurasi training pada akhir proses mendapatkan nilai 96% dan akurasi validation sebesar 85.7%. Pada proses training nilai loss train dan valid semakin menurun hingga nilai akhirnya 0.182 untuk training dan 0.244 untuk validation. Model VGG16 dengan teknik optimasi Adam memiliki kinerja yang baik dengan akurasi training dan validation yang tinggi. Selain itu nilai loss yang ditunjukkan juga tidak memiliki perbedaan yang terlalu jauh sehingga model bisa dikatakan tidak dalam kondisi overfit. Pengujian model dengan menggunakan data test mendapatkan akurasi sebesar

92.8% dan loss sebesar 0.21. Confusion matrix untuk pengujian model bisa dilihat pada gambar 4.13.

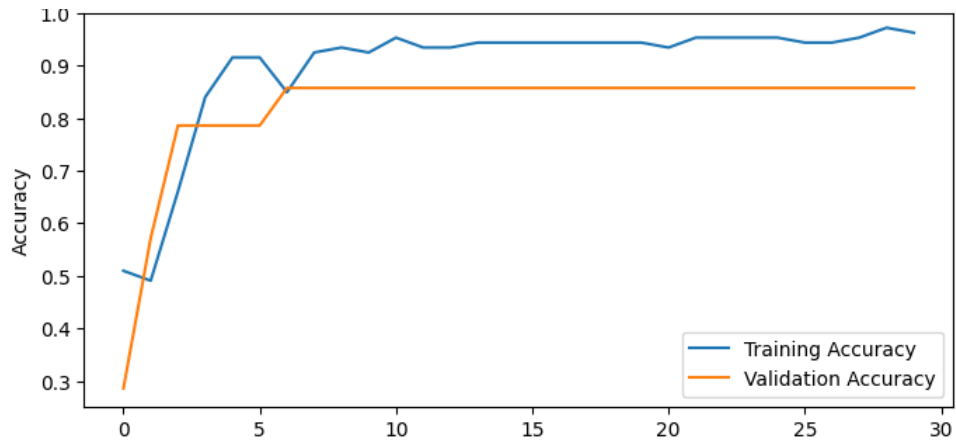


Gambar 4.13 Confusion Matrix Data Test VGG16_Adam

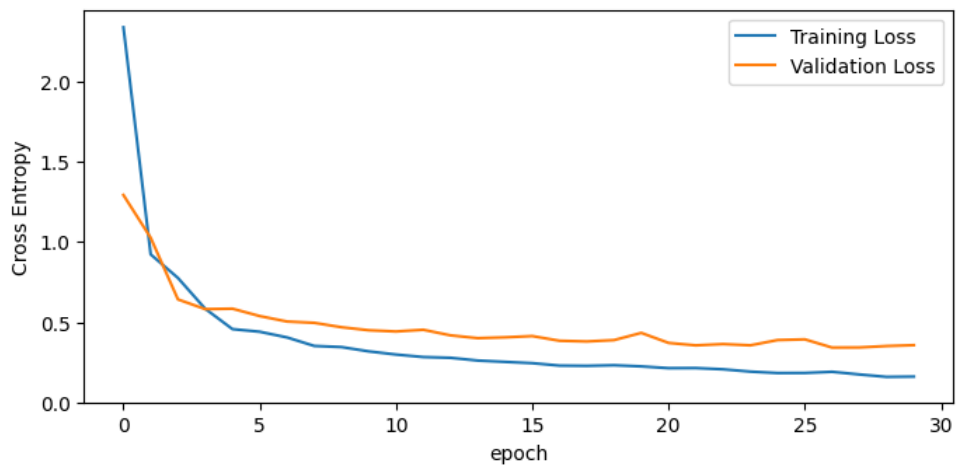
Gambar 4.13 menunjukkan bahwa model dengan arsitektur VGG16 dan teknik optimasi Adam mampu bekerja dengan sangat baik. Hasil prediksi yang ditunjukkan model mampu memprediksi benar seluruhnya pada citra daun tanaman jeruk dengan kondisi HLB dan Bukan HLB. Untuk Kondisi Sehat dari 4 citra daun tanaman jeruk hanya 1 yang gagal diprediksi sehat. Model dengan arsitektur VGG16 dan teknik optimasi Adam yang berhasil dibuat memiliki ukuran file sebesar 476MB.

4.6.4 VGG16_Adamax

Model VGG16_Adamax adalah model yang dibentuk dengan menggunakan arsitektur VGG15 dan teknik optimasi Adamax. Model ini dibuat dengan jumlah epoch sebanyak 30 epoch. Nilai akurasi dan loss yang dihasilkan oleh model pada saat proses training dapat dilihat pada gambar 4.14.



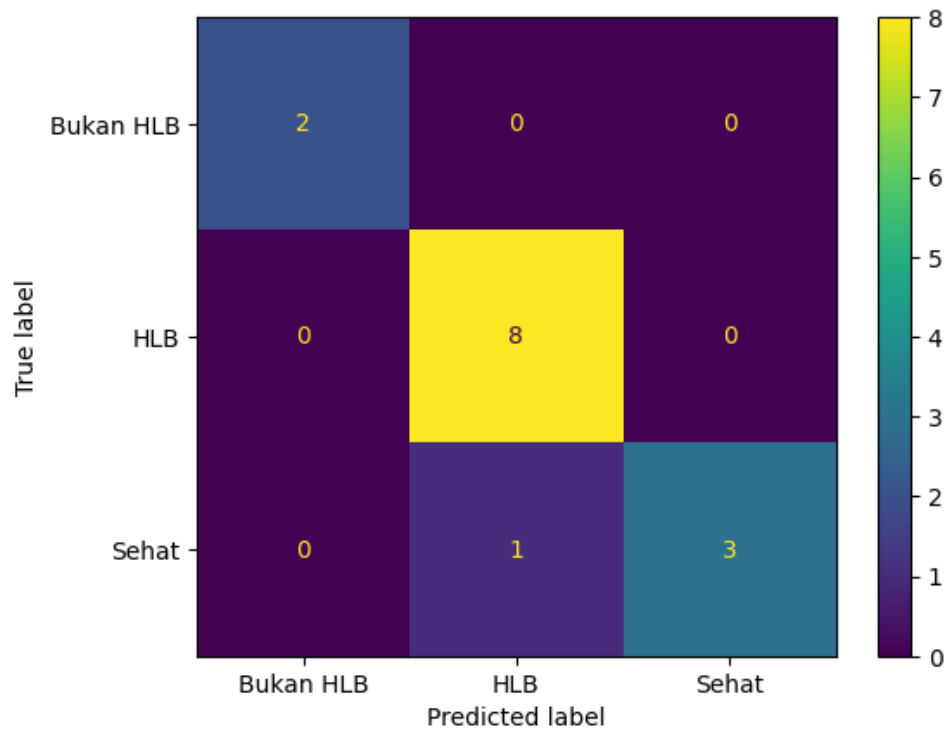
(a)



(b)

Gambar 4.14 Grafik Indikator Kinerja VGG16_Adamax (a) Akurasi dan (b) Loss

Dari gambar grafik 4.14, dapat dilihat bahwa model mampu mencapai akurasi yang baik pada akhir proses training yaitu sebesar 96.2% dan akurasi validasi sebesar 85.7%. Nilai loss yang ditunjukkan pada grafik juga mendapatkan hasil yang baik dengan nilai akhir yaitu 0.16 untuk train loss dan 0.356 untuk validation loss. Hasil yang diperoleh dari proses training menunjukkan bahwa model memiliki akurasi yang baik dan loss yang rendah. Nilai loss yang tidak begitu jauh juga menunjukkan bahwa tidak terjadi overfit sehingga model mampu memprediksi data diluar data train. Pengujian menggunakan data test pada model menghasilkan akurasi sebesar 92.8% dan loss sebesar 0.288. Hasil prediksi dapat dilihat pada confusion matrix pada gambar 4.15.

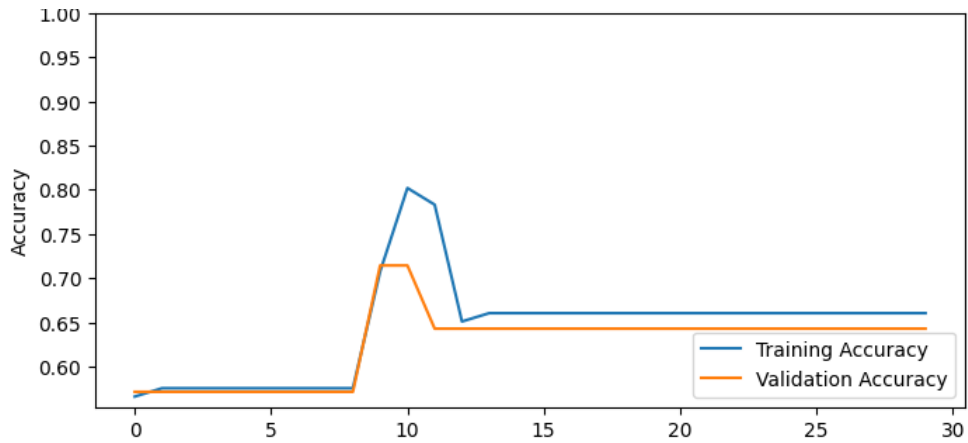


Gambar 4.15 Confusion Matrix Data Test VGG16_Adamax

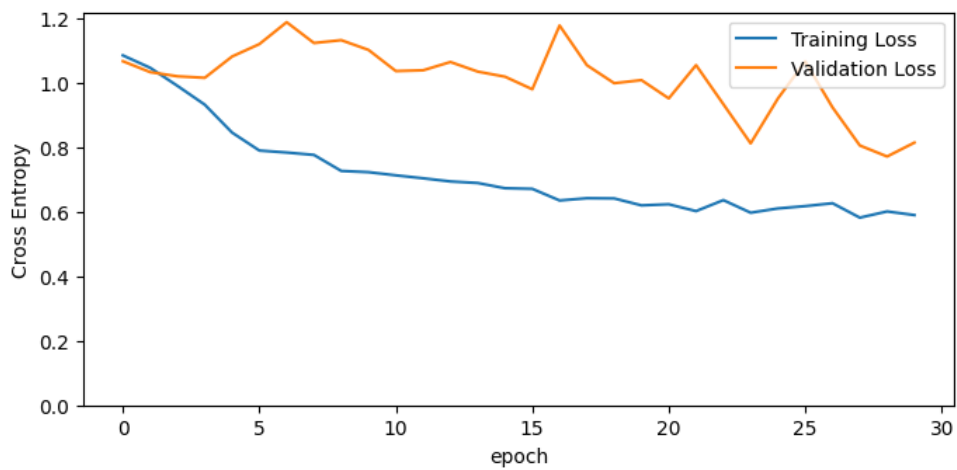
Dari gambar 4.15, dapat dilihat bahwa model mampu bekerja dengan baik karena dapat memprediksi benar seluruh data test citra daun tanaman pada kondisi bukan HLB dan HLB. Pada data test citra daun tanaman jeruk pada kondisi sehat berhasil diprediksi sehat sebanyak 3 citra dan gagal 1 citra karena diprediksi kedalam HLB. Model yang berhasil disimpan memiliki ukuran file sebesar 476MB.

4.6.5 SqueezeNet_Adam

Model SqueezeNet_Adam dibentuk dengan arsitektur SqueezeNet dan teknik optimasi Adam dengan jumlah epoch sebanyak 30 epoch. Akurasi dan loss dari hasil training model dapat dilihat pada gambar 4.16.



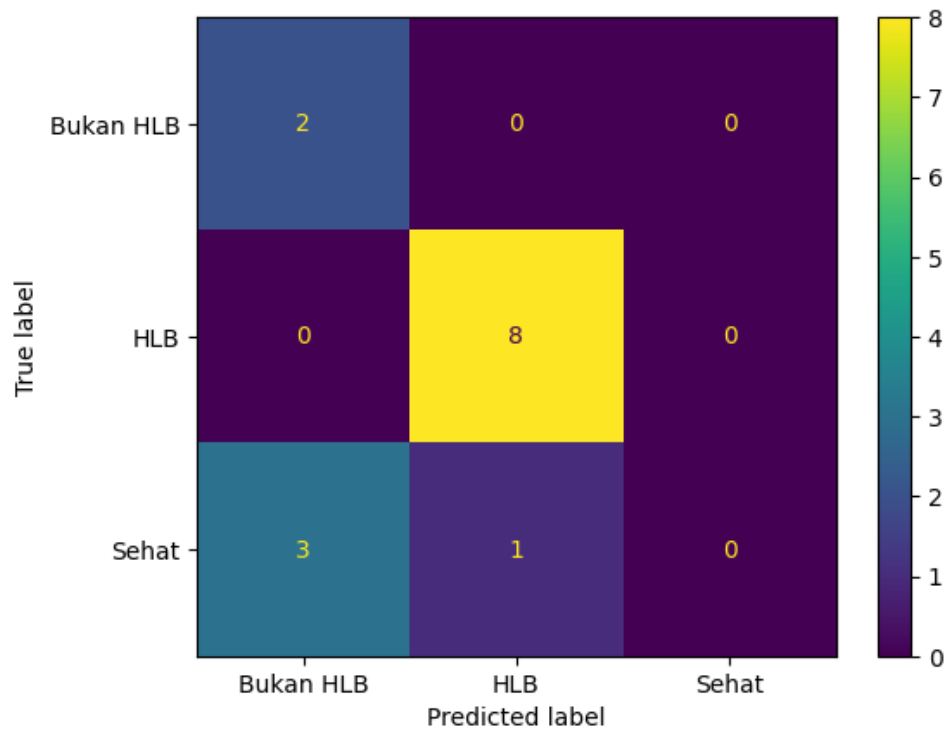
(a)



(b)

Gambar 4.16 Grafik Indikator Kinerja SqueezeNet_Adam (a) Akurasi dan (b) Loss

Berdasarkan gambar 4.16, dapat dilihat grafik akurasi mengalami peningkatan walau tidak signifikan. Hasil akhir dari akurasi training sebesar 66% dan akurasi validation sebesar 64.2%. Nilai loss yang dihasilkan model adalah 0.59 untuk train loss dan 0.81 untuk validation loss. Model ini selanjutnya diuji dengan menggunakan data test dan mendapatkan akurasi sebesar 71.4% dengan loss sebesar 0.775. Hasil prediksi untuk setiap kondisi daun dapat dilihat pada gambar confusion matrix 4.17.

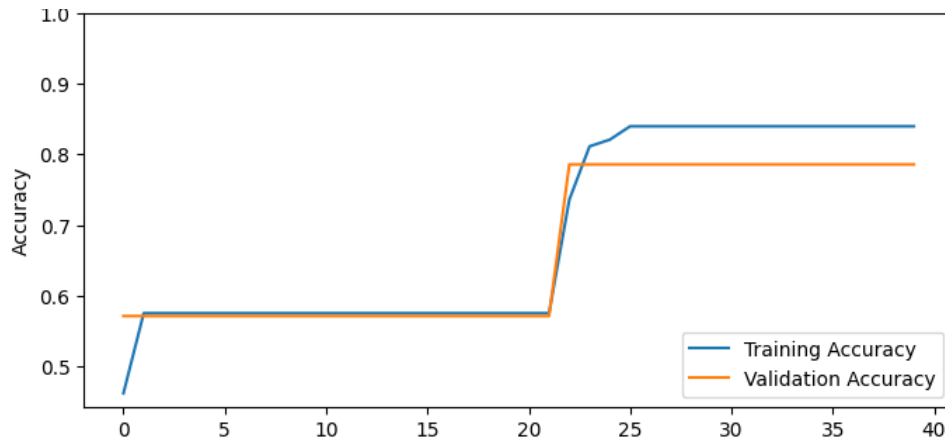


Gambar 4.17 Confusion Matrix Data Test SqueezeNet_Adam

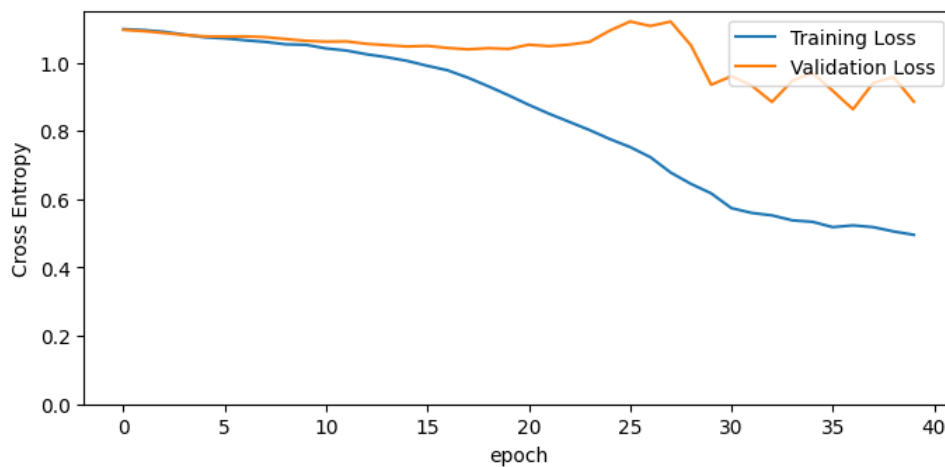
Berdasarkan gambar 4.17, confusion matrix yang ditunjukkan model dapat memprediksi seluruh data test citra daun tanaman jeruk pada kondisi bukan HLB dan HLB. Namun pada kondisi sehat model masih belum bisa memprediksi dengan benar. Citra daun tanaman jeruk dalam kondisi sehat diprediksi Bukan HLB sebanyak 3 citra dan HLB sebanyak 1 citra. Ukuran model SqueezeNet_Adam yang telah berhasil dibuat adalah sebesar 8.65MB.

4.6.6 SqueezeNet_Adamax

Model SqueezeNet_Adamax merupakan model yang dibentuk dengan arsitektur SqueezeNet dan teknik optimasi Adamax dengan jumlah epoch sebanyak 30 Epoch. Hasil akurasi dan loss pada proses melakukan training dapat dilihat pada gambar 4.18.



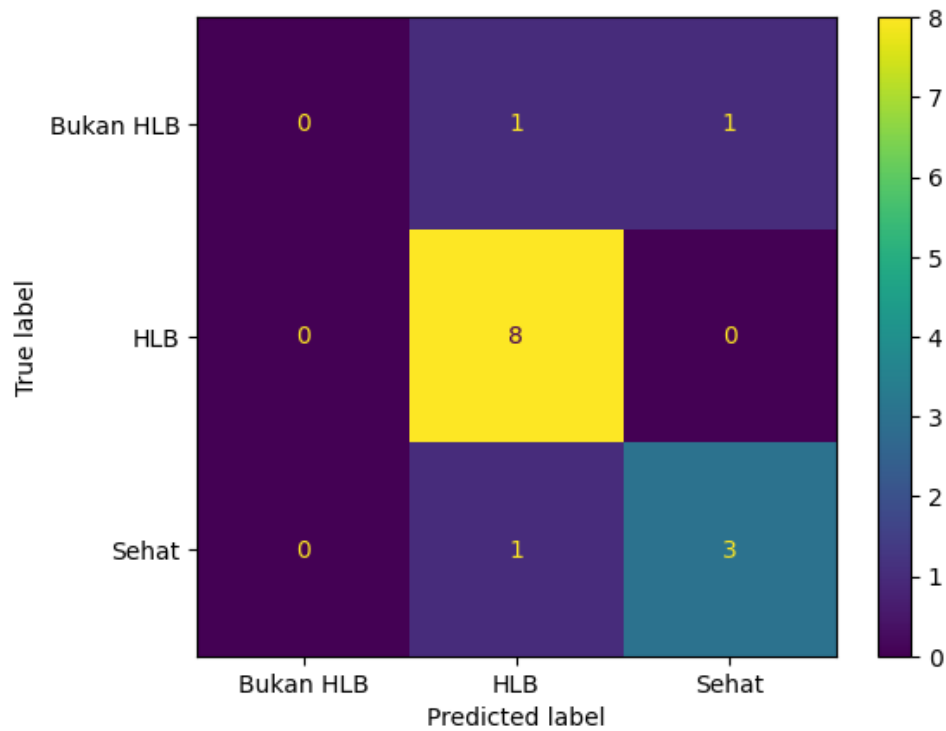
(a)



(b)

Gambar 4.18 Grafik Indikator Kinerja SqueezeNet_Adamax (a) Akurasi dan (b) Loss

Berdasarkan gambar 4.18. dapat dilihat model pada proses training mengalami peningkatan akurasi. Hasil akhir akurasi yang didapatkan adalah 83.9% untuk akurasi training dan 78.5% untuk akurasi validation. Nilai loss yang dihasilkan oleh model pada akhir proses training menunjukkan 0.495 untuk training loss dan 0.885 untuk validation loss. Dari hasil itu, model yang dibentuk dengan arsitektur SqueezeNet_Adamax memiliki akurasi yang cukup baik namun memiliki nilai loss yang masih tinggi.



Gambar 4.19 Confusion Matrix Data Test SqueezeNet_Adamax

Gambar 4.19 menjelaskan bahwa model SqueezeNet_Adamax berhasil melakukan prediksi pada data test citra daun tanaman jeruk dengan kondisi HLB. Untuk citra kondisi sehat model berhasil memprediksi 3 citra dengan benar dan gagal dalam memprediksi 1 citra. Model ini masih belum bisa memprediksi seluruh citra dengan benar pada kondisi daun bukan HLB. Model yang berhasil dibentuk memiliki ukuran file sebesar 8.65MB

4.6.7 Ringkasan Pengujian Model

Dari penjelasan diatas dapat diringkaskan beberapa hasil yang di dapat dari pengujian terhadap model yaitu akurasi dimana terdapat *Training Accuracy*, *Validation Accuracy*, dan *Test Accuracy*. Selain akurasi terdapat juga hasil loss dimana terdapat *Training Loss*, *Validation Loss*, dan *Test Loss* yang bisa dilihat pada tabel

Tabel 4.4 Ringkasan Akurasi Model CNN

	Training	Validation	Test
	Accuracy	Accuracy	Accuracy
AlexNet_Adam	98,0%	78,5%	78,5%
AlexNet_Adamax	95,2%	57,1%	71,4%
VGG16_Adam	96,0%	85,7%	92,8%
VGG16_Adamax	96,2%	85,7%	92,8%
SqueezeNet_Adam	66,0%	64,2%	71,4%
SqueezeNet_Adamax	83,9%	78,5%	78,5%

Berdasarkan tabel 4.4, akurasi model terbaik pada saat melakukan prediksi dengan data *test* adalah model VGG16_Adam dan VGG16_Adamax dengan akurasi 92,8%. Diikuti dengan model AlexNet_Adam dan SqueezeNet_Adamax dengan akurasi 78,5%. Dan model dengan akurasi terendah adalah model AlexNet_Adamax dan SqueezeNet_Adam dengan akurasi sebesar 71,4%.

Tabel 4.5 Ringkasan Loss Model CNN

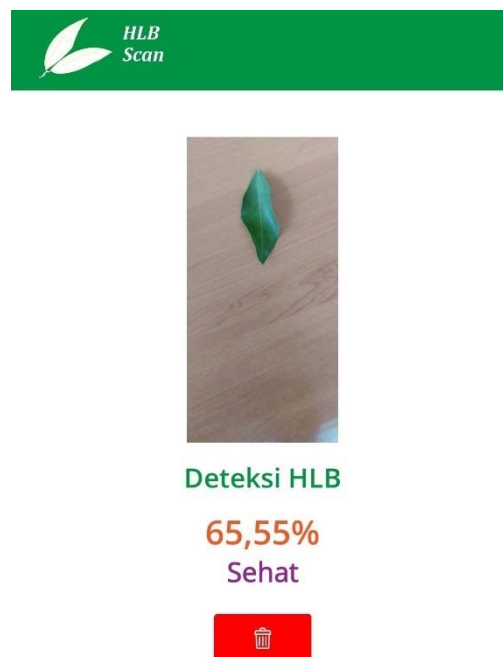
	Training	Validation	Test
	Loss	Loss	Loss
AlexNet_Adam	0,064	2,685	1,745
AlexNet_Adamax	0,270	2,155	1,907
VGG16_Adam	0,182	0,244	0,210
VGG16_Adamax	0,160	0,356	0,288
SqueezeNet_Adam	0,590	0,815	0,775
SqueezeNet_Adamax	0,495	0,885	0,870

Berdasarkan tabel 4.5, pada saat model melakukan prediksi dengan menggunakan data test, model dengan arsitektur VGG16 memiliki nilai loss yang paling rendah dengan nilai loss 0,2103 untuk model VGG16_Adam dan 0,2883 untuk model VGG16_Adamax. Diikuti dengan model yang dibangun dengan arsitektur SqueezeNet memiliki nilai loss sebesar 0,7754 untuk model

SqueezeNet_Adam dan 0,8708 untuk model SqueezeNet_Adamax. Terakhir model dengan arsitektur AlexNet memiliki nilai loss yang paling tinggi dengan nilai loss sebesar 1,7453 untuk AlexNet_Adam dan 1,9074 untuk AlexNet_Adamax.

4.7 Hasil Aplikasi

Hasil akhir aplikasi kurang lebih memiliki tampilan yang hampir sama dengan *mockup* aplikasi yang tertera pada bab 3. Berikut adalah tampilan yang dirubah sehingga dapat digunakan dengan mudah dan memberikan kenyamanan untuk pengguna.



Gambar 4.20 Tampilan Baru Halaman Prediksi

Pada gambar 4.20 tampilan baru yang ditambahkan fitur delete untuk menghapus gambar yang telah di deteksi baik dari kamera ataupun upload dari galeri. Fitur hapus ini memberikan kemudahan bagi pengguna terutama apabila terjadi kesalahan dalam melakukan pengambilan foto.

Tabel 4.6 Tabel Evaluasi Kinerja Aplikasi

No	Keterangan	Fitur	Berfungsi	Delay
1	Home Screen	Take Photo Button	✓	1 (s)
		Upload Photo Button	✓	0,5 (s)
		History Page Button	✓	0,5 (s)
		Guide Page Button	✓	0,5 (s)
		Credit Page Button	✓	0,5 (s)
		More Action Button	✓	0,5 (s)
2	Take Photo Page	Take Photo Button	✓	0,5 (s)
		Home Page Button	✓	0,5 (s)
		History Page Button	✓	0,5 (s)
		Guide Page Button	✓	0,5 (s)
		Credit Page Button	✓	0,5 (s)
3	Prediction Result Screen	Delete Image Button	✓	0,5 (s)
4	History Screen	Sort By Button	✓	0,5 (s)
		Home Page Button	✓	0,5 (s)
		History Page Button	✓	0,5 (s)
		Guide Page Button	✓	0,5 (s)
		Credit Page Button	✓	0,5 (s)
		More Action Button	✓	0,5 (s)
5	Guide Screen	Next Button	✓	0,5 (s)
		Previous Button	✓	0,5 (s)
		Home Page Button	✓	0,5 (s)
		History Page Button	✓	0,5 (s)
		Guide Page Button	✓	0,5 (s)
		Credit Page Button	✓	0,5 (s)
		More Action Button	✓	0,5 (s)
6	About Screen	Home Page Button	✓	0,5 (s)
		Photo Page Button	✓	0,5 (s)
		History Page Button	✓	0,5 (s)
		Guide Page Button	✓	0,5 (s)

Model yang diterapkan pada aplikasi merupakan semua model yang sudah dibuat dan dilakukan pengujian. Sebelum diterapkan pada aplikasi model harus melewati proses konversi. Konversi dilakukan karena android tidak menerima file dengan format hdf5 sehingga dibutuhkan konversi ke format yang dapat diterima oleh android yaitu tensorflowlite (tflite). Pada penerapannya model akan diuji membutuhkan waktu berapa lama untuk melakukan prediksi. Pengujian terhadap kecepatan prediksi penting untuk kenyamanan pengguna. Waktu prediksi yang lama akan mengganggu pengguna dalam menggunakan aplikasi. Evaluasi untuk model yang digunakan oleh android dirangkum pada tabel 4.7.

Tabel 4.7 Hasil Evaluasi Model

Model	Ukuran File	Waktu
AlexNet_Adam	500 MB	✖
AlexNet_Adamax	500 MB	✖
VGG16_Adam	200 MB	9,14 (s)
VGG16_Adamax	200 MB	9,61 (s)
SqueezeNet_Adam	2 MB	0,84 (s)
SqueezeNet_Adamax	2 MB	0,53 (s)

Kedua model yang dibangun dengan arsitektur AlexNet tidak dapat berjalan pada aplikasi karena memiliki ukuran file yang sangat besar yaitu sebesar 500 MB. Pada android studio hanya menerima file tensorflowlite maksimal dengan ukuran 200 MB. Untuk Model dengan arsitektur VGG16 berhasil berjalan pada aplikasi android dengan waktu prediksi 9,14 detik untuk model dengan optimasi Adam dan 9,61 detik untuk model dengan optimasi Adamax. Model dengan arsitektur SqueezeNet memiliki ukuran file yang sangat jauh jika dibandingkan dengan arsitektur lain. Dimana ukuran file model dengan SqueezeNet hanya 2 MB. Selain ukuran file yang jauh lebih kecil, waktu prediksi yang dibutuhkan oleh model juga sangat singkat yaitu 0,84 detik untuk model dengan optimasi Adam dan 0,53 detik untuk model dengan optimasi Adamax.

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Penelitian tugas akhir yang bertujuan membuat model CNN untuk deteksi penyakit HLB pada tanaman jeruk dan membangun aplikasi berbasis android dengan mengimplementasikan model CNN pada aplikasi sudah berjalan dengan baik. Berdasarkan penelitian yang telah dilakukan dapat disimpulkan yaitu pembuatan model CNN untuk deteksi HLB berhasil dilakukan dengan jumlah model yang terbagi menjadi 6 macam yaitu: AlexNet_Adam, AlexNet_Adamax, VGG16_Adam, VGG16_Adamax, SqueezeNet_Adam, dan SqueezeNet_Adamax. Model VGG16_Adam ditentukan sebagai model terbaik dengan hasil akurasi yang paling baik pada training dengan akurasi sebesar 96%, validation dengan akurasi 85,7%, dan test dengan akurasi 92,8%.

Perubahan terjadi pada pembuatan model yaitu tidak melakukan augmentasi data dan kalibrasi atau koreksi pencahayaan. Augmentasi tidak dilakukan karena tidak dapat meningkatkan akurasi model dan memiliki nilai loss yang lebih tinggi dari model yang dilatih tanpa data augmentasi. Kalibrasi tidak dilakukan pada membuat model karena akurasi yang lebih rendah dibandingkan dengan model yang dilatih dengan data yang tidak melalui proses kalibrasi. Nilai loss yang dihasilkan oleh model dengan kalibrasi juga lebih tinggi dibandingkan dengan model yang dibangun tanpa kalibrasi.

Pengukuran kinerja aplikasi dilakukan dengan mengukur besar ukuran file dan lamanya waktu yang dibutuhkan model untuk melakukan prediksi. Model dengan arsitektur AlexNet memiliki ukuran file yang paling besar dengan ukuran 500 MB. Arsitektur VGG16 menghasilkan model dengan ukuran file 200 MB. Ukuran file paling kecil dihasilkan oleh arsitektur SqueezeNet dengan ukuran 2 MB. Model dengan arsitektur AlexNet tidak dapat melakukan prediksi karena ukuran file yang sangat besar sehingga tidak dapat mengetahui waktu yang dibutuhkan aplikasi untuk melakukan prediksi. Model dengan arsitektur VGG16 membutuhkan waktu sekitar 9 detik untuk melakukan prediksi. Sedangkan model dengan arsitektur SqueezeNet membutuhkan waktu kurang dari satu detik untuk

melakukan prediksi. Dari pengujian tersebut, Model SqueezeNet memiliki kinerja paling cepat dalam melakukan prediksi dan memiliki ukuran file yang paling kecil.

5.2 Saran

Adapun saran-saran dan masukan yang dapat digunakan sebagai referensi untuk penelitian selanjutnya terutama dengan topik yang sama yaitu deteksi penyakit HLB pada tanaman jeruk.

1. Model kemungkinan akan bekerja lebih baik dengan menambahkan datanya, terutama data citra daun dengan kondisi bukan HLB dan Sehat untuk mengimbangi jumlah data yang ada pada kondisi HLB.
2. Membuat model dengan kombinasi augmentasi dan jumlah epoch yang beragam. Hal ini bertujuan untuk melihat apakah jumlah epoch bisa meningkatkan kinerja model dengan augmentasi.
3. Arsitektur SqueezeNet sebaiknya dimodifikasi dengan menambahkan atau mengurangi layer yang ada. Hal ini bertujuan untuk meningkatkan akurasi model dalam melakukan prediksi.

DAFTAR PUSTAKA

- Boesch, G. (2021, October 6). *VGG Very Deep Convolutional Networks (VGGNet)* – *What you need to know*. Viso.Ai. <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/>
- Brownlee, J. (2017, July 3). *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning*. Machinelearningmastery. <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>
- Brownlee, J. (2021, June 7). *Gradient Descent Optimization With AdaMax From Scratch*. Machinelearningmastery. <https://machinelearningmastery.com/gradient-descent-optimization-with-adamax-from-scratch/>
- Eichhorn Bilodeau, S., Wu, B. Sen, Rufyikiri, A. S., MacPherson, S., & Lefsrud, M. (2019). An update on plant photobiology and implications for cannabis production. In *Frontiers in Plant Science* (Vol. 10). Frontiers Media S.A. <https://doi.org/10.3389/fpls.2019.00296>
- Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311–318. <https://doi.org/10.1016/j.compag.2018.01.009>
- Gould, K., Davies, K., & Winefield, C. (2009). *Anthocyanins* (C. Winefield, K. Davies, & K. Gould, Eds.). Springer New York. <https://doi.org/10.1007/978-0-387-77335-3>
- Great Learning Team. (2020, June 24). *AlexNet: The First CNN to win Image Net / What is AlexNet?* <https://www.facebook.com/GreatLearningOfficial/>. <https://www.mygreatlearning.com/blog/alexnet-the-first-cnn-to-win-image-net/>
- Hirschberg, J., Cohen, M., Harker, M., Lotan, T., Mann, V., & Pecker, I. (1997). Molecular genetics of the carotenoid biosynthesis pathway in plants and algae. In *Chem* (Vol. 69, Issue 10). IUPAC.

- Kementerian Pertanian Republik Indonesia. (2021). *Swasembada Pangan Suatu Keharusan*. Kementerian Pertanian Republik Indonesia. <https://www.pertanian.go.id/home/?show=news&act=view&id=2496>
- Linus Pauling Institute. (2016, October 5). *Vitamins*. Linus Pauling Institute. <https://lpi.oregonstate.edu/mic/vitamins>
- Linus Pauling Institute. (2023, January 3). *Vitamin C*. Linus Pauling Institute. <https://lpi.oregonstate.edu/mic/vitamins/vitamin-C>
- MATLAB. (n.d.). *Texture Analysis Using the Gray-Level Co-Occurrence Matrix (GLCM)- MATLAB & Simulink*. MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink. Retrieved June 27, 2023, from <https://www.mathworks.com/help/images/texture-analysis-using-the-gray-level-co-occurrence-matrix-glcm.html>
- Militante, S. V, Gerardo, B. D., & Dionision, N. V. (2019). *Plant Leaf Detection and Disease Recognition using Deep Learning*.
- Mithra, S. (2011). *What are Carotenoids?* Wisegeek. <https://www.thehealthboard.com/what-are-carotenoids.htm>
- Młodzińska, E. (2009). *Survey Of Plant Pigments: Molecular And Environmental Determinants Of Plant Colors*. 7–16.
- Narkhede, S. (2018, May 9). *Understanding Confusion Matrix*. Towardsdatascience. <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>
- Natarajan, V. A., Macha Babitha, M., & Sunil Kumar, M. (2020). Detection of disease in tomato plant using Deep Learning Techniques. In *International Journal of Modern Agriculture* (Vol. 9, Issue 4). <https://www.researchgate.net/publication/349860175>
- Nurhadi. (2014). *PENYAKIT HUANGLONGBING TANAMAN JERUK (Candidatus Liberibacter asiaticus): ANCAMAN DAN STRATEGI PENGENDALIAN*.
- Oracle. (n.d.). *The Java Tutorials*. Retrieved March 31, 2023, from <https://docs.oracle.com/javase/tutorial/>

- Pradana, R. A. (2023). *PENGEMBANGAN MODEL CONVOLUTIONAL NEURAL NETWORK UNTUK PERBAIKAN WARNA CITRA PADA PROSES PENGUKURAN PIGMEN SECARA NON-DESTRUKTIF*.
- Prasetyo, M. C. D. (2022). *Rancang Bangun Aplikasi Berbasis Android Untuk Kalibrasi Citra Tanaman Secara Automatis Dengan Convolutional Neural Network*. Universitas Ma Chung.
- Putra, H., & Yulkifli, D. (2019). Studi Awal Rancang Bangun Colorimeter Menggunakan Sensor Opt101 Berbasis Sistem Android Dengan Display Smartphone. In *Jurusan Fisika FMIPA Univ. Riau Pekanbaru p-ISSN* (Vol. 16, Issue 2).
- Rahman, M. (2023, March 7). *What You Need to Know about Sparse Categorical Cross Entropy*.
- Saha, S. (2018). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Towards Data Science.
- Society, N. G. (2019, September 13). *Chlorophyll*. National Geographic Society. <https://education.nationalgeographic.org/resource/chlorophyll>
- Sulaeman, A. (2014). *Buku Pegangan Sosial Ekonomi Pertanian*. Penebar Swadaya.
- Tensorflow. (n.d.). *Tensorflow*. Retrieved March 31, 2023, from <https://www.tensorflow.org/>
- Tsang, S.-H. (2019, April 22). *Review: SqueezeNet (Image Classification)*. Towards Data Science. <https://towardsdatascience.com/review-squeezenet-image-classification-e7414825581a>
- Wenderlich, R. (n.d.). *What is Android Studio?*
- Wijaya, N. H., & Sutrimo, S. (2020). Lux Meter as A Measuring Instrument for Operating Lamp Light Intensity Based on Arduino Uno R3. *Jurnal Ecotipe (Electronic, Control, Telecommunication, Information, and Power Engineering)*, 8(1), 1–8. <https://doi.org/10.33019/jurnalecotipe.v8i1.1927>
- Yassin, M. R. (2018). *IDENTIFIKASI KARAKTERISTIK LAHAN PERKEBUNAN JERUK PAMELO DI KECAMATAN MA'RANG KABUPATEN PANGKEP* Miss Rahma Yassin. www.agroplantaonline.com

LAMPIRAN

Lampiran nomor 1:

Kode Arsitektur AlexNet

```
model = Sequential()
model.add(Convolution2D(96, 11, strides = (4, 4),
padding = 'valid', input_shape=(656, 456, 3), activation
= 'relu'))
model.add(MaxPooling2D(pool_size = (2, 2), strides = (2,
2), padding = 'valid'))
model.add(BatchNormalization())
model.add(Convolution2D(256, 11, strides = (1, 1),
padding='valid', activation = 'relu'))
model.add(MaxPooling2D(pool_size = (2, 2), strides = (2,
2), padding='valid'))
model.add(BatchNormalization())
model.add(Convolution2D(384, 3, strides = (1, 1),
padding='valid', activation = 'relu'))
model.add(BatchNormalization())
model.add(Convolution2D(384, 3, strides = (1, 1),
padding='valid', activation = 'relu'))
model.add(BatchNormalization())
model.add(Convolution2D(256, 3, strides=(1,1),
padding='valid', activation = 'relu'))
model.add(MaxPooling2D(pool_size = (2, 2), strides = (2,
2), padding = 'valid'))
model.add(BatchNormalization())
model.add(Flatten())
# Full Connection Step
model.add(Dense(units = 4096, activation = 'relu'))
model.add(Dropout(0.4))
model.add(BatchNormalization())
model.add(Dense(units = 4096, activation = 'relu'))
```

```
model.add(Dropout(0.4))
model.add(BatchNormalization())
model.add(Dense(units = 1000, activation = 'relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(Dense(units = 3, activation = 'softmax'))
```

Lampiran nomor 2:

Kode Arsitektur SqueezeNet

```
# Definisi blok Fire
def fire_module(x, squeeze_channels,
expand1x1_channels, expand3x3_channels):
    squeeze = Convolution2D(squeeze_channels, (1, 1),
activation='relu')(x)
    expand1x1 = Convolution2D(expand1x1_channels, (1,
1), activation='relu')(squeeze)
    expand3x3 = Convolution2D(expand3x3_channels, (3,
3), padding='same', activation='relu')(squeeze)
    return Concatenate()([expand1x1, expand3x3])

# Definisi arsitektur SqueezeNet
def SqueezeNet(input_shape, num_classes):
    inputs = Input(shape=input_shape)
    x = Convolution2D(96, (7, 7), strides=(2, 2),
activation='relu', padding='same')(inputs)
    x = MaxPooling2D(pool_size=(3, 3), strides=(2, 2),
padding='same')(x)
    x = fire_module(x, 16, 64, 64)
    x = fire_module(x, 16, 64, 64)
    x = fire_module(x, 32, 128, 128)
    x = MaxPooling2D(pool_size=(3, 3), strides=(2, 2),
padding='same')(x)
    x = fire_module(x, 32, 128, 128)
    x = fire_module(x, 48, 192, 192)
    x = fire_module(x, 48, 192, 192)
    x = fire_module(x, 64, 256, 256)
    x = MaxPooling2D(pool_size=(3, 3), strides=(2, 2),
padding='same')(x)
    x = fire_module(x, 64, 256, 256)
```

```

x = Dropout(0.5)(x)
x = Convolution2D(num_classes, (1, 1),
activation='relu', padding='same')(x)
x = GlobalAveragePooling2D()(x)
outputs = Dense(num_classes,
activation='softmax')(x)

model = Model(inputs, outputs)
return model

```