

**KLASIFIKASI PENYAKIT DAUN PADA TANAMAN KOPI
DENGAN PENERAPAN METODE *FASTER REGION
CONVOLUTIONAL NEURAL NETWORK (FASTER R-CNN)***

SKRIPSI

IAN ARIESSA SITORUS

181402093



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

2024

KLASIFIKASI PENYAKIT DAUN PADA TANAMAN DENGAN
PENERAPAN METODE *FASTER REGION CONVOLUTIONAL*
NEURAL NETWORK (FASTER R-CNN)

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh
ijazah Sarjana Teknologi Informasi

IAN ARIESSA SITORUS

181402093



PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

2024

PERSETUJUAN

Judul : KLASIFIKASI PENYAKIT DAUN PADA TANAMAN
KOPI DENGAN PENERAPAN METODE *FASTER
REGION CONVOLUTIONAL NEURAL NETWORK
(FASTER R-CNN)*

Kategori : SKRIPSI

Nama Mahasiswa : IAN ARIESSA SITORUS

Nomor Induk Mahasiswa : 181402093

Program Studi : TEKNOLOGI INFORMASI

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

Medan, 10 Januari 2024

Komisi Pembimbing:

Pembimbing 2,

Pembimbing 1,

Fahrurrozi Lubis B.IT., M.Sc.IT

Seniman S.Kom., M.Kom.

NIP. 198610122018052001

NIP. 198705252014041001

Diketahui/disetujui oleh

Program Studi S-1 Teknologi Informasi

Ketua,

Dedy Arisandi S.T., M.Kom.

NIP. 197908312009121002

PERNYATAAN

KLASIFIKASI PENYAKIT DAUN PADA TANAMAN KOPI
DENGAN PENERAPAN METODE *FASTER REGION*
CONVOLUTIONAL
NEURAL NETWORK (FASTER R-CNN)

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 10 Januari 2024

Ian Ariessa Sitorus

181402093

ABSTRAK

Kopi merupakan suatu komoditas agrikultur bernilai tinggi dengan sumbangan devisa cukup besar bagi Negara Indonesia. Dengan lahan budidaya kopi mencapai 1,24 juta hektare, Indonesia termasuk salah satu penghasil kopi terbesar di dunia. Potensi besar tersebut belum terserap dengan baik diakibatkan efektivitas pencegahan dan pengendalian penyakit belum dikelola dengan baik. Penyakit tersebut dapat diindikasikan dari perubahan bentuk dan warna pada daun. Namun, faktor penglihatan, pengalaman, serta kompleksitas lahan yang luas dan jumlah tanaman kopi yang besar menjadi kendala yang dialami oleh petani. Penelitian dilakukan guna menemukan solusi alternatif dengan pengembangan sistem berbasis *computer vision* dengan penerapan *Faster R-CNN* dalam melakukan klasifikasi 3 jenis penyakit pada daun, yaitu Karat Daun (*Leaf Rust*), Hawar Daun (*Leaf Blight*) dan Penggorok Daun (*Leaf Miner*). Penelitian ini menggunakan *dataset* yang terdiri dari 3.600 citra, terbagi menjadi 2.880 *data training*, 360 *data validation*, dan 360 *data testing*. Hasil pengujian terhadap sistem yang menerapkan metode *Faster R-CNN* mampu mencapai nilai akurasi sebesar 95%.

Kata kunci : Kopi, Penyakit Daun pada Tanaman Kopi, *Computer Vision*, *Faster Region Convolutional Neural Network (Faster R-CNN)*.

**CLASSIFICATION OF LEAF DISEASES IN COFFEE PLANTS USING THE
FASTER REGION CONVOLUTIONAL NEURAL NETWORK
(FASTER R-CNN) METHOD**

ABSTRACT

Coffee is a high-value agricultural commodity with a significant contribution to foreign exchange for the Indonesian state. With a cultivation area of 1.24 million hectares, Indonesia is among the largest coffee producers in the world. Despite this substantial potential, it has not been fully realized due to the ineffective management of disease prevention and control. Diseases can be indicated by changes in the shape and color of the leaves. However, factors such as vision, experience, as well as the vast complexity of the land and the large number of coffee plants pose challenges for farmers. Research is conducted to find an alternative solution by developing a computer vision-based system using the Faster R-CNN application to classify three types of leaf diseases: Leaf Rust, Leaf Blight, and Leaf Miner. This study employs a dataset comprising 3,600 images, divided into 2,880 training data, 360 validation data, and 360 testing data. Testing results of the system implementing the Faster R-CNN method achieve an accuracy value of 95%.

Keywords: *Coffee, Leaf Diseases in Coffee Plants, Computer Vision, Faster Region Convolutional Neural Network (Faster R-CNN).*

UCAPAN TERIMA KASIH

Segala puji dan syukur penulis panjatkan kepada Yesus Kristus atas berkat dan penyertaan-Nya, penulis dapat menyelesaikan penyusunan skripsi sebagai salah satu syarat untuk menyandang gelar Sarjana Komputer di Program Studi S-1 Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

Dengan rasa sukacita dan penuh syukur, penulis mempersembahkan skripsi ini kepada orang tua terkasih, Ayahanda A. Sitorus dan M. Purba, S.Si. Kasih sayang sepanjang masa, doa, nasihat, serta keteguhan hati dengan prinsip “*Anakkonki do Hamoraon di Ahu*”, selalu memberikan yang terbaik tanpa rasa lelah demi masa depan anak-anaknya. Terima kasih atas segalanya.

Penulis juga ingin menyampaikan ucapan terima kasih kepada semua pihak yang telah memberikan doa, dukungan, serta bantuan selama proses penulisan skripsi ini. Dengan hati yang bersukacita, penulis ucapkan terima kasih sebesar-besarnya kepada :

1. Bapak Prof. Dr. Muryanto Amin, S.Sos., M.Si., selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia, B.Sc., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Bapak Dedy Arisandi, S.T., M.Kom., selaku Ketua Program Studi S-1 Teknologi Informasi Universitas Sumatera Utara.
4. Bapak Ivan Jaya, S.Si., M.Kom., selaku Sekretaris Program Studi S-1 Teknologi Informasi Universitas Sumatera Utara.
5. Bapak Seniman, S.Kom., M.Kom., selaku Dosen Pembimbing I dan Bapak Fahrurrozi Lubis, B.IT., M.Sc.IT., selaku Dosen Pembimbing II. Terima kasih atas segala bimbingan, saran, motivasi, dan nasihat yang telah diberikan kepada penulis dalam penelitian dan penyusunan penulisan skripsi ini. Semoga Tuhan Yang Maha Esa selalu memberkati.
6. Bapak Dr. Muhammad Anggia Muchtar, S.T., MM.IT., selaku Dosen Pembimbing I dan Ibu Fanindia Purnamasari, S.TI., M.IT., selaku Dosen

Pembanding II. Terima kasih penulis ucapkan kepada Bapak dan Ibu karena sudah memberikan saran, kritik yang membangun, nasihat, dan motivasi kepada penulis dalam penelitian dan penyusunan penulisan skripsi ini. Semoga Tuhan Yang Maha Esa selalu memberkati.

7. Bapak dan Ibu Dosen Program Studi S-1 Teknologi Informasi yang telah memberikan ilmu-ilmu yang berharga kepada penulis selama masa perkuliahan.
8. Bang Fadly Idris, S.E., selaku staff akademik yang banyak memberikan bantuan, nasihat, serta motivasi kepada penulis selama masa penyusunan tugas akhir. Semoga Tuhan Yang Maha Esa selalu memberkati.
9. Brain Brilliant Sitorus, Ardian Steven Sitorus dan Risto Panjaitan, saudara kandung penulis yang memberikan doa terbaik untuk penulis dalam menapaki setiap proses.
10. Keluarga besar penulis, terkhusus kepada Oppung boru Br. Panjaitan dan Oppung boru Br. Sirait yang selalu memberikan doa dan dukungan dalam setiap proses yang dilalui oleh penulis.
11. Samuel Agusta Emri Surbakti, Cici Sianturi, Angga Subaru Hutagaol, Chindy Siagian, Yohannes Bedi Ginting, Fernando Abelta Kaban, Ricky Martin Abdi Negara Purba, dan Monang Limbong yang telah menjadi saudara penulis dalam melewati suka dan duka selama merantau di Kota Medan. Memberikan banyak bantuan, motivasi, dan nasihat kepada penulis dalam menjalani perkuliahan hingga penyusunan tugas akhir. Semoga Tuhan Yesus selalu menyertai dan memberkati setiap langkah dan rencana.
12. CatchMeIfYouCan Team (Albert, Darius J. Simamora, Andrian Sebayang, Janrian Simbolon, dan Ammar R. Hasibuan), yang telah menemani penulis selama masa perkuliahan sampai tahap penyusunan tugas akhir.
13. Yoman Despi (Joyfull Banjarnahor, Boy Charto Sihombing, Christanta A. Ginting, Alfaro Tambunan, Yehezkiel Simanjuntak, Felix G. Hutahaeen, Helmut Pakpahan, Theo Ifanka Sebayang), yang telah menemani penulis dalam masa perkuliahan sampai tahap penyusunan tugas akhir,
14. Anak AMPARA 409, terima kasih telah memberikan warna kehidupan kepada penulis selama merantau di Kota Medan.
15. Teman-teman Teknologi Informasi 2018, yang selalu bersama dengan penulis dalam menjalani masa perkuliahan.

16. Teman-teman Alumni SMA N 1 Silima Pungga-Pungga Angkatan 2018, yang telah memberikan doa dan dukungan.

Medan, 10 Januari 2024

Penulis

DAFTAR ISI

PERSETUJUAN	i
PERNYATAAN	ii
ABSTRAK	iii
UCAPAN TERIMA KASIH	v
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian	3
1.4. Batasan Masalah	3
1.5. Manfaat Penelitian	4
1.6. Metodologi Penelitian	4
1.7. Sistematika Penulisan	5
BAB 2 LANDASAN TEORI	7
2.1. Tanaman Kopi	7
2.2. Penyakit Tanaman Kopi	7
2.2.1. Penyakit karat daun (<i>leaf rust</i>)	7
2.2.2. Penyakit penggorok daun (<i>leaf miner</i>)	8
2.2.3. Penyakit hawar daun (<i>leaf blight</i>)	8
2.3. Pengolahan Citra Digital	9
2.4. Citra Digital	9
2.5. <i>Convolutional Neural Network (CNN)</i>	11
2.6. <i>Regional Convolutional Neural Network (R-CNN)</i>	12
2.7. <i>Faster Regional Convolutional Neural Network (Faster R-CNN)</i>	14
2.8. <i>Pytorch</i>	15
2.9. <i>Python</i>	15
2.10. <i>Android</i>	16
2.11. <i>Android Studio</i>	16
2.12. <i>Confussion Matrix</i>	16

2.12.1 Accuracy	17
2.12.2 Precision	17
2.12.3 Recall	18
2.12.4 F1-Score	18
2.13. Penelitian Terdahulu	18
BAB 3	21
ANALISIS DAN PERANCANGAN SISTEM	21
3.1. Data Penelitian	21
3.2. Analisis Sistem	21
3.2.1 Image Acquisition	24
3.2.2 Image Pre-Processing	24
3.2.3 Feature Extraction	26
3.2.4 Image Classifier	26
3.2.5 Learned Model	31
3.2.6 Proses Training	31
3.2.7 Testing	32
3.2.8 Output	32
3.3. Perancangan Antarmuka Sistem	32
3.3.1 Rancangan Tampilan Splash Screen	32
3.3.2 Rancangan Tampilan Menu Check	33
3.3.3 Rancangan Tampilan Halaman Menu Disease	33
3.3.4 Rancangan Tampilan Menu About	34
BAB 4	35
IMPLEMENTASI DAN PENGUJIAN SISTEM	35
4.1. Implementasi Sistem	35
4.1.1 Perangkat (Hardware) dan Perangkat Keras Lunak (Software)	35
4.1.2 Implementasi Data	35
4.2. Implementasi Perancangan Antarmuka	37
4.2.1 Tampilan Splashscreen	37
4.2.2 Tampilan Menu Check	37
4.2.4 Tampilan Menu About	39
4.3. Prosedur Penggunaan Aplikasi	39
4.4. Pelatihan Sistem	40
4.5. Pengujian Sistem	42

BAB 5	49
KESIMPULAN DAN SARAN.....	49
5.1. Kesimpulan.....	49
5.2. Saran.....	49
DAFTAR PUSTAKA	51

DAFTAR TABEL

Tabel 2. 1 Confussion Matrix	17
Tabel 2. 2 Penelitian Terdahulu	18
Tabel 2. 3 Penelitian Terdahulu Lanjutan	19
Tabel 2. 4 Penelitian Terdahulu Lanjutan	20
Tabel 3. 1 Kategori Dataset	21
Tabel 4. 1 Hasil Pelatihan Sistem.....	41
Tabel 4. 2 Hasil Pelatihan Sistem Lanjutan	42
Tabel 4. 3 Hasil Pengujian Sistem	43
Tabel 4. 4 Hasil Pengujian Sistem Lanjutan	44
Tabel 4. 5 Hasil Pengujian Sistem Lanjutan	45
Tabel 4. 6 Hasil Confussion Matrix	46
Tabel 4. 7 Nilai TP, FP, dan FN	46
Tabel 4. 8 Nilai Precision, Recall, dan F1-Score	47

DAFTAR GAMBAR

Gambar 2. 1 Karat Daun (Leaf Rust) (Sumber : Kaggle.com).....	8
Gambar 2. 2 Penggorok Daun (Leaf Miner) (Sumber : Kaggle.com).....	8
Gambar 2. 3 Hawar Daun (Leaf Blight) (Sumber : Kaggle.com)	9
Gambar 2. 4 Koordinat Pengolahan Citra (Karpathy, 2017).....	10
Gambar 2. 5 Citra Warna (Sumber : majalahguru.net)	10
Gambar 2. 6 Citra Keabuan (Grayscale) (Sumber : majalahguru.net)	11
Gambar 2. 7 Citra Biner (Sumber : majalahguru.net)	11
Gambar 2. 8 Arsitektur Umum CNN (Karpathy, 2017).....	11
Gambar 2. 9 Arsitektur Umum R-CNN (Sumber : blog.paperspace.com)	12
Gambar 2. 10 Selective Search (Sumber : geeksforgeeks.org)	13
Gambar 2. 11 Teknik <i>ROI Pooling</i> pada <i>Fast R-CNN</i>	13
Gambar 2. 12 Arsitektur Umum Faster R-CNN (Gavrilescu, 2018).....	14
Gambar 3. 1 Arsitektur Umum.....	23
Gambar 3. 2 Resizing Citra	24
Gambar 3. 3 Perhitungan proses Resizing (Nguyen & et al, 2018)	25
Gambar 3. 4 Nilai matrix setelah proses resize (Nguyen & et al, 2018).....	25
Gambar 3. 5 Pseudocode Augmentasi Data	26
Gambar 3. 6 Pseudocode Proses Klasifikasi	27
Gambar 3. 7 Mekanisme Feature Map	28
Gambar 3. 8 Region Proposal Network (RPN)	28
Gambar 3. 9 Target ROI.....	29
Gambar 3. 10 Proses ROI Pooling (Karpathy, 2017).....	30
Gambar 3. 11 Pemetaan Data Pooling (Karpathy, 2017)	30
Gambar 3. 12 ROI Pooling 3 x 3 (Karpathy, 2017)	30
Gambar 3. 13 Rancangan Tampilan Splash Screen.....	33
Gambar 3. 14 Rancangan Tampilan Halaman Menu Check	33
Gambar 3. 15 Rancangan Tampilan Halaman Menu Disease	34
Gambar 3. 16 Rancangan Tampilan Halaman Menu About	34
Gambar 4. 1 Data Karat Daun (Leaf Rust).....	36
Gambar 4. 2 Data Hawar Daun (Leaf Blight)	36
Gambar 4. 3 Data Penggorok Daun (Leaf Miner).....	36
Gambar 4. 4 Tampilan Splashscreen	37
Gambar 4. 5 Tampilan Menu Check	38
Gambar 4. 6 Tampilan Menu Disease	38
Gambar 4. 7 Tampilan Menu About.....	39

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Kopi merupakan salah satu primadona komoditas agrikultur karena bernilai tinggi yang mampu menyumbang devisa cukup besar bagi pendapatan Negara Indonesia. Sebagai salah satu negara penghasil kopi terbesar, Negara Indonesia memiliki total lahan budidaya mencapai 1,24 juta ha. Potensi besar tersebut belum dimanfaatkan dengan baik, diakibatkan kurangnya tingkat produktivitas tanaman kopi di Indonesia. (Nasution, B.B, 2018).

Menurut hasil penelitian yang dilakukan oleh (CIPS), rendahnya tingkat produktivitas budidaya kopi di Indonesia disebabkan oleh dua faktor utama, yaitu rata-rata usia tanaman kopi yang sudah mencapai masa tua, dan kurangnya implementasi upaya peremajaan dan perawatan tanaman dengan benar. Kondisi ini menjadikan tanaman kopi lebih rentan terhadap penyakit yang pada akhirnya mengganggu produktivitasnya.

Tanda-tanda penyakit pada tanaman kopi dapat diindikasikan melalui perubahan bentuk fisik, warna, dan bekas yang ditimbulkan pada daun. Namun, diperlukan pengalaman dan ketelitian untuk dapat mengenali dan membedakan jenis penyakit tersebut guna penanganan yang tepat sasaran sehingga pengendalian penyebaran penyakit dapat dilakukan dengan baik (Putri, Ramadhani, & Yunanto, 2021). Pada penelitian ini, jenis penyakit yang akan diklasifikasikan adalah karat daun (*leaf rust*) disebabkan oleh *Hemilia Vastarix*, serangga penambang daun *Leucoptera Coffela* vektor penyakit *leaf miner*, dan *Phoma Costaricensis* penyebab penyakit hawar daun (*leaf blight*). Jenis penyakit ini mampu memberikan dampak yang sangat buruk karena dapat mengakibatkan daun berguguran sehingga proses pembentukan makanan pada fotosintesis akan terganggu (Sutarman, 2017).

Pencegahan dan pengendalian penyebaran penyakit dapat dilakukan dengan berbagai metode, salah satunya dengan penggunaan obat-obatan pertanian berbahan kimia anorganik. Namun, penggunaan obat-obatan pertanian dapat menimbulkan efek

buruk berkelanjutan apabila tidak sesuai dosis dan penggunaannya terhadap suatu penyakit (Purbosari et al, 2021). Oleh karena itu, dibutuhkan sebuah solusi alternatif yang dapat membantu petani dalam mengenali jenis penyakit yang menjangkit dengan akurat dan dalam waktu singkat sehingga penggunaan obat-obatan pertanian berbahan kimia buatan dapat diterapkan dengan tepat sasaran.

Solusi alternatif untuk mengetahui jenis penyakit daun dapat dilakukan dengan pendekatan disiplin ilmu teknologi informasi yang sudah diterapkan dalam berbagai bidang, yaitu teknologi *computer vision* dan *machine learning*. Saat ini, teknologi tersebut berkembang dengan sangat pesat. Telah dilakukan beberapa penelitian mengenai identifikasi jenis penyakit pada daun dengan pemanfaatan teknologi *computer vision*. Penelitian yang dilakukan oleh (Ma, et al., 2018) untuk mengklasifikasikan jenis penyakit pada buah mentimun menggunakan *Convolutional Neural Network (CNN)* dengan *dataset* 1.184 citra dapat menghasilkan nilai akurasi sebesar 94%. (Esgario, Krohling, & Ventura, 2020) telah melakukan penelitian untuk mengklasifikasikan tingkat kerusakan disebabkan oleh patogen pada beberapa tanaman dengan penerapan *CNN*. Penelitian ini juga bertujuan untuk menguji kinerja arsitektur *AlexNet*, *GoogleNet*, *ResNet50* dan *VGG16* dengan hasil menunjukkan bahwa arsitektur *ResNet50* memiliki performa terbaik mencapai nilai akurasi 97,07%. Berikutnya, (Irfansyaj, 2021) melakukan penelitian dengan penerapan arsitektur *CNN AlexNet* untuk mengklasifikasikan 3 kategori kondisi daun, yaitu sehat, *rust*, dan *red spider mite* dengan nilai akurasi yang mencapai 81%. (I., Golpur; ParianJA, R.A; Chayjan,; 2021) telah melakukan penelitian mengenai identifikasi kultivar padi dengan menggunakan citra digital menggunakan algoritma . Adapun akurasi yang capai oleh penelitian ini mencapai 96,66%. Penelitian yang dilakukan oleh (Sabrina & Al Maki, 2022) dalam mengklasifikasikan 1.560 citra terdiri dari 6 kategori menggunakan algoritma *CNN* dengan hasil akurasi mencapai 94%.

Berdasarkan uraian pada latar belakang, penulis mengajukan judul **“KLASIFIKASI PENYAKIT PADA DAUN TANAMAN KOPI DENGAN PENERAPAN METODE *FASTER REGION CONVOLUTIONAL NEURAL NETWORK (FASTER R-CNN)*”**.

1.2. Rumusan Masalah

Produktivitas pada tanaman kopi dipengaruhi oleh beberapa faktor, salah satu faktor utama adalah kesehatan daun. Daun memiliki fungsi sebagai tempat terjadinya proses pembentukan makanan (fotosintesis). Peran penting daun pada proses pembentukan makanan sangat memengaruhi proses pertumbuhan dan perkembangan tanaman kopi. Penggunaan obat-obatan pertanian berbahan kimia anorganik dilakukan sebagai upaya dalam pencegahan dan pengendalian berbagai penyakit pada tanaman. Penggunaan yang tidak tepat sasaran dapat memengaruhi proses produktivitas karena menimbulkan efek buruk berkelanjutan. Faktor indra penglihatan dan pengalaman berperan penting dalam proses identifikasi suatu penyakit sehingga dapat dilakukan penanganan yang tepat. Maka, dibutuhkan sebuah teknologi yang dapat membantu petani dalam menangani masalah tersebut sehingga penanganan yang cepat dan tepat sasaran dapat dilakukan sebagai upaya pencegahan dan pengendalian jenis penyakit yang menyerang daun tanaman kopi.

1.3. Tujuan Penelitian

Tujuan penelitian ini dilakukan adalah untuk mengembangkan sebuah pendekatan yang dapat mendeteksi dan mengklasifikasikan jenis penyakit yang menyerang daun tanaman kopi dengan penerapan algoritma *Faster R-CNN*.

1.4. Batasan Masalah

Pada penelitian ini, penulis menetapkan beberapa batasan masalah yang bertujuan sebagai fokus ruang lingkup dari permasalahan penelitian, yaitu :

- 1) Penelitian ini menggunakan citra daun tanaman kopi yang bersumber dari *platform open data Kaggle*.
- 2) Objek pada penelitian ini adalah warna pada bekas perubahan fisik akibat dari penyakit yang menyerang.
- 3) Sistem mampu mendeteksi lebih dari 1 jenis penyakit pada 1 citra daun dan mengklasifikasikan.
- 4) Untuk hasil maksimal, citra yang digunakan adalah selebar daun tanaman kopi.
- 5) Data yang digunakan berekstensi .jpg/.jpeg.

- 6) Jenis penyakit yang menjadi cakupan penelitian ini adalah : Karat Daun (*Leaf Rust*), Daun Penggorok Daun (*Leaf Miner*), Hawar Daun (*Leaf Blight*).

1.5. Manfaat Penelitian

Beberapa manfaat yang diperoleh dari penelitian ini adalah :

- 1) Menjadi salah satu solusi alternatif dalam membantu petani untuk mengenali jenis penyakit yang menyerang daun kopi agar petani dapat melakukan pencegahan dan pengendalian yang tepat sasaran sehingga memberikan dampak yang positif terhadap produktivitas tanaman kopi.
- 2) Menjadi salah satu sumber referensi dan bahan pembelajaran untuk penelitian lebih lanjut dalam pengembangan algoritma *Faster R-CNN* dan materi penerapan disiplin ilmu Teknologi Informasi pada sektor pertanian.

1.6. Metodologi Penelitian

Pada penelitian ini, penulis menyusun tahapan-tahapan yang akan dilakukan sebagai acuan pada penelitian. Adapun tahapan-tahapan tersebut adalah :

1. Studi Literatur dan Pengumpulan Data

Pada tahapan awal, penulis melakukan pembelajaran dan pemahaman mengenai literatur yang berkaitan dengan *image processing*, algoritma *Faster Region Convolutional Neural Network (Faster R-CNN)*, serta karya ilmiah yang memuat materi mengenai jenis-jenis penyakit pada daun tanaman kopi yang diperoleh dari berbagai jurnal, artikel, buku, dan skripsi. Penulis juga melakukan pengumpulan data untuk mempelajari ciri dari penyakit tersebut. Data citra daun kopi berasal dari *platform open source* Kaggle yang merupakan *dataset public*.

2. Analisis Permasalahan

Tahapan ini dilakukan setelah studi literatur telah selesai. Analisis dan materi mengenai studi literatur yang telah dipersiapkan akan menjadi acuan pemahaman mengenai materi *computer vision*, *Faster R-CNN*, dan informasi mengenai penyakit daun pada tanaman kopi yang akan diterapkan dalam penelitian.

3. Perancangan Sistem

Pada tahapan ini, penulis melakukan perancangan arsitektur umum, menentukan pembagian *dataset* menjadi *data_training*, *data_validation*, dan *data_testing* dengan rasio 80:10:10, serta perancangan desain antarmuka sistem.

4. Implementasi

Pada tahapan ini, dilakukan implementasi dari analisis sesuai rancangan yang telah disusun pada tahapan sebelumnya untuk pembangunan sistem.

5. Pengujian Sistem

Pada tahap ini, sistem yang telah dikembangkan akan diuji untuk mengevaluasi kinerja serta menghitung tingkat akurasi dari penerapan metode *Faster Region Convolutional Neural Network* dalam proses klasifikasi jenis penyakit pada daun kopi. Hasil klasifikasi akan dianalisis menggunakan *Confusion Matrix* untuk menghitung nilai *Accuracy*, *Precision*, *Recall*, dan *F1-Score*.

6. Dokumentasi dan Penyusunan Laporan

Tahapan akhir dilakukan penyusunan dokumentasi dan laporan yang menjabarkan alur, proses, dan hasil dari keseluruhan penelitian.

1.7. Sistematika Penulisan

Sistematika pada skripsi ini terdiri dari lima bab yang masing-masing bab dijelaskan secara singkat sebagai berikut :

BAB 1. PENDAHULUAN

Bab pendahuluan disusun dari latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB 2. LANDASAN TEORI

Bab landasan teori disusun dari berbagai teori-teori yang berkaitan dengan penyakit pada daun kopi, pengolahan citra digital, algoritma *Faster Region Convolutional Neural Network (Faster R-CNN)* serta analisis penelitian terdahulu yang berkaitan dengan penelitian ini.

BAB 3. ANALISIS DAN PERANCANGAN SISTEM

Bab analisis dan perancangan sistem disusun dari penjabaran berbagai tahapan yang telah dilakukan dalam perancangan sistem yang berkaitan dengan arsitektur umum metode *Faster Region Convolutional Neural Network (Faster R-CNN)*, tahapan *pre-processing*, proses *data training*, dan perancangan desain antarmuka (*user interface*).

BAB 4. IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab implementasi dan pengujian sistem menguraikan implementasi hasil analisis, serta perancangan sistem yang telah dilakukan. Sistem yang telah dibangun akan diuji kinerja dalam menyelesaikan masalah, sehingga diperoleh nilai akurasi yang dapat dicapai oleh sistem.

BAB 5. KESIMPULAN DAN SARAN

Pada bab ini, penulis menyajikan kesimpulan dari keseluruhan tahapan penelitian serta saran yang diajukan untuk diimplementasikan pada penelitian pengembangan berikutnya.

BAB 2

LANDASAN TEORI

2.1. Tanaman Kopi

Kopi (*Coffe. Sp*) adalah tanaman berupa pohon yang termasuk ke dalam *famili Rubiaceae*, bergenus *Coffeaa*. Tanaman kopi tumbuh secara tegak hingga mencapai tinggi 12 meter, bercabang, serta memiliki daun dengan bentuk bulat telur yang tumbuh pada percabangan dan ranting. Tanaman kopi memiliki varian sekitar 70 spesies yang tersebar di seluruh dunia, tetapi spesies Kopi Robusta (*Coffe Canephora var Robusta*) dan Kopi Arabika (*Coffe Arabica*) menjadi varian kopi yang paling banyak dibudidayakan. (Nasution, 2018).

2.2. Penyakit Tanaman Kopi

Pada penelitian ini, penulis menentukan tiga jenis penyakit daun pada tanaman kopi. Adapun jenis-jenis penyakit tersebut, yaitu : penyakit karat daun (*leaf rust*), penyakit daun gugur (*leaf miner*), dan hawar daun (*leaf blight*).

2.2.1. Penyakit karat daun (*leaf rust*)

Penyakit karat daun (*leaf rust*) merupakan kondisi pada daun tanaman kopi yang diinfeksi oleh patogen berupa jamur dari ordo *Pucciniales*, yang disebut *Hemileia Vastatrix*. Patogen ini menginfeksi tanaman kopi melalui stomata pada daun. Gejala awal yang ditimbulkan adalah munculnya bercak kuning pada permukaan atas daun yang selanjutnya dengan perubahan warna menjadi cokelat dengan kondisi bagian bawah daun terdapat spora berwarna jingga. Akibatnya, jaringan klorofil yang berfungsi dalam proses fotosintesis rusak dan menimbulkan dampak buruk dalam proses pertumbuhan dan perkembangan tanaman kopi. Penanganan yang kurang tepat dapat mengakibatkan keguguran daun sehingga tingkat produktivitas tanaman kopi menurun. (Siska, 2018).



Gambar 2. 1 Karat Daun (*Leaf Rust*) (Sumber : Kaggle.com)

2.2.2. Penyakit penggorok daun (*leaf miner*)

Penyakit Penggorok Daun (*leaf miner*) merupakan penyakit yang disebabkan oleh serangan serangga vektor penambang daun (*Leucoptera Caffeine*) yang dapat ditemukan pada area perkebunan yang rimbun. Selain menjadi vektor, serangga tersebut juga memakan jaringan *parenkim palisade* yang mengakibatkan kematian sel pada jaringan daun. Hama ini pertama kali ditemukan pada lahan perkebunan kopi di Brazil. (Filho, 2006)



Gambar 2. 2 Penggorok Daun (*Leaf Miner*) (Sumber : Kaggle.com)

2.2.3. Penyakit hawar daun (*leaf blight*)

Hawar daun (*leaf blight*) adalah suatu penyakit pada daun tanaman kopi yang diakibatkan oleh patogen jamur yang dikenal dengan nama *Phoma Costaricensis*. Perubahan yang terjadi sebagai dampak dari penyakit jenis ini adalah dahan menjadi kering, daun berguguran, serta terjadinya kerontokan tunas buah dan kelopak bunga. Indikasi penyakit ini adalah perubahan fisik permukaan daun dengan perubahan warna menjadi coklat gelap dengan pola bertekstur. Penyakit ini dapat ditemukan di lingkungan perkebunan berhawa dingin dan berangin (Soesanto, 2020).



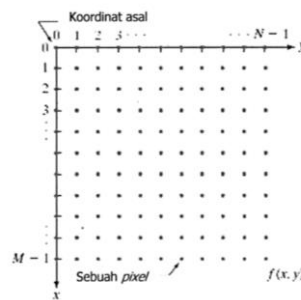
Gambar 2. 3 Hawar Daun (*Leaf Blight*) (Sumber : Kaggle.com)

2.3. Pengolahan Citra Digital

Pengolahan citra digital (*Digital Image Processing*) merupakan suatu disiplin ilmu yang berfokus pada teknik-teknik pengolahan citra, mencakup gambar diam (foto) dan gambar bergerak (video), dan dilakukan secara digital menggunakan komputer (Sutoyo & Mulyanto, 2009). Secara matematis, citra merupakan sebuah fungsi kontinu yang dapat mendeskripsikan intensitas cahaya pada bidang dua dimensi. Pada penerapannya, citra akan direpresentasikan menjadi bentuk numerik dengan nilai diskrit sehingga dapat diproses oleh perhitungan komputasi pada *computer digital* atau disebut digitalisasi citra.

2.4. Citra Digital

Citra digital adalah hasil representasi nilai-nilai yang menggambarkan bilangan nyata dan kompleks melalui susunan bit tertentu. Sebuah gambar dapat direpresentasikan sebagai fungsi $f(x, y)$ dengan M kolom dan N baris, di mana x dan y akan mewakili koordinat spasial, dan nilai f pada koordinat (x, y) yang dikenal sebagai tingkat intensitas atau keabuan citra pada titik tersebut. Citra dapat disebut dengan citra digital apabila nilai x , y , dan f adalah diskrit.



Gambar 2. 4 Koordinat Pengolahan Citra (*Karpathy, 2017*)

Citra digital dapat merepresentasikan sebuah matriks yang disusun oleh M kolom dan N baris dengan perpotongan antar kolom dan baris disebut dengan *pixel*. *Pixel* adalah elemen terkecil dari sebuah citra dan memiliki 2 parameter, yaitu koordinat intensitas dan warna. Nilai yang terdapat pada koordinat (x, y) adalah $f(x, y)$ yang menyatakan intensitas atau warna dari *pixel* pada titik tersebut. Beberapa jenis citra digital yang sering digunakan adalah citra biner, citra *grayscale* dan citra warna (Sutoyo & Mulyanto, 2009).

2.4.1 Citra Berwarna (Color Image)

Citra berwarna merupakan citra yang terdiri dari tiga komponen utama pada warna yaitu R (*Red*), G (*Green*), dan B (*Blue*). Gambar yang berwarna memiliki memori 24 bit dan setiap warna berisi 8 bit, memungkinkan sekitar 16 juta variasi warna untuk ditampilkan.



Gambar 2. 5 Citra Warna (Sumber : majalahguru.net)

2.4.2 Citra Keabuan (Grayscale Image)

Citra ke abuan (*Grayscale*) merupakan citra digital yang di setiap *pixel* hanya terdiri dari satu nilai saja, yaitu nilai abu-abu. Kehalusan gradasi warna pada citra bergantung pada besar jumlah bit warna yang dihasilkan.



Gambar 2. 6 Citra Ke Abuan (*Grayscale*) (Sumber : majalahguru.net)

2.4.3 Citra Biner (*Binary Image*)

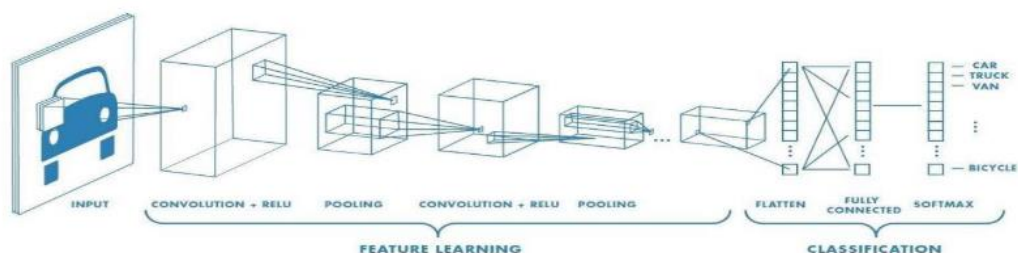
Citra biner merupakan citra yang terdiri dari beberapa *pixel* dengan intensitas yang sangat terbatas. Setiap *pixel* hanya memiliki dua nilai, yaitu hitam dan putih. Memori penyimpanan yang dibutuhkan pada citra biner hanya sebesar 1 bit.



Gambar 2. 7 Citra Biner (Sumber : majalahguru.net)

2.5. *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) merupakan suatu teknik dalam *machine learning* bertipe *feed-forward* (tidak berulang) untuk mengolah data dua dimensi, analisis gambar visual, dan mengenali objek pada gambar berdasarkan beberapa parameter. *CNN* dirancang untuk mengurangi jumlah beberapa parameter dengan cara melakukan penyederhanaan arsitektur jaringan (*neuron network*) dalam melakukan pengenalan objek (Karpathy, 2017). Arsitektur *CNN* terdiri dari 2 bagian, yaitu *Feature Extraction Layer* dan *Fully-Connected Layer (Multilayer Perception)*.

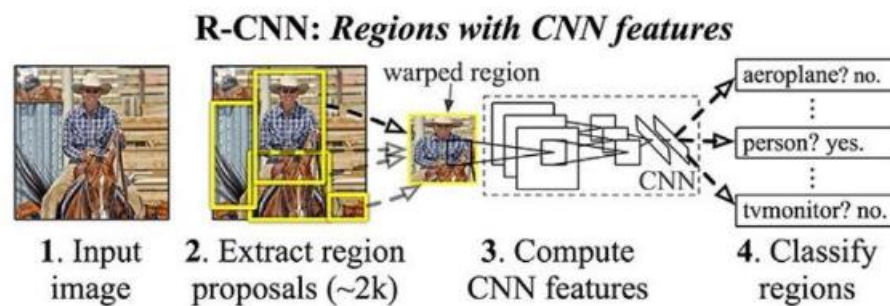


Gambar 2. 8 Arsitektur Umum *CNN* (Karpathy, 2017)

2.6. Regional Convolutional Neural Network (R-CNN)

Regional Convolutional Neural Network (R-CNN) merupakan algoritma dalam pemrosesan citra pada tahap *image processing* yang memanfaatkan metode *selective search*. Pada tahapan ini, citra gambar akan dikelompokkan menjadi 2000 *proposal regional* berdasarkan parameter warna pada citra gambar.

Dengan alur proses yang kompleks, proses algoritma *R-CNN* tergolong lambat dikarenakan proses pelatihan pada *R-CNN* mengolah 2000 *proposal regional* pada setiap citra gambar. Hal demikian membuat *R-CNN* kurang efisien dan efektif untuk diimplementasikan dalam sistem *real-time* dikarenakan sistem membutuhkan waktu berkisar 50 detik untuk pemrosesan satu gambar (Gavrilescu, 2018).



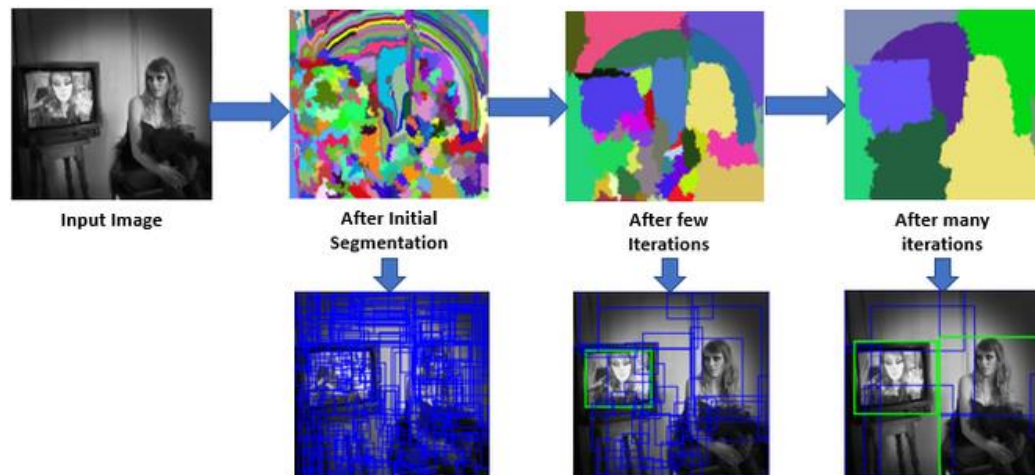
Gambar 2. 9 Arsitektur Umum *R-CNN* (Sumber : blog.paperspace.com)

Dalam algoritma *selective search*, dua wilayah yang memiliki kemiripan akan digabungkan secara bersamaan. Hal ini bertujuan untuk mengurangi jumlah wilayah yang perlu diuji lebih lanjut. Proses penggabungan ini dapat membantu meningkatkan efisiensi dan mempercepat proses pencarian objek dalam gambar. Proses tersebut dapat direpresentasikan ke dalam persamaan (2.1) :

$$S(a,b) = S_{\text{texture}}(a,b) + S_{\text{size}}(a,b) \quad (2.1)$$

Keterangan : $S_{\text{texture}}(a,b)$ = Mengukur kesamaan visual

$S_{\text{size}}(a,b)$ = Menggabungkan wilayah terkecil secara bersamaan untuk menghindari area yang lain.

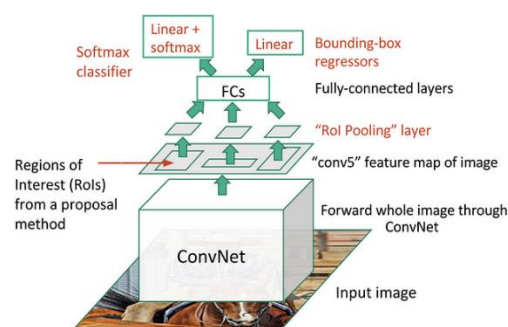


Gambar 2. 10 *Selective Search* (Sumber : geeksforgeeks.org)

2.7. Fast Region Convolutional Neural Network

Fast Region Convolutional Neural Network (Fast R-CNN) merupakan metode lanjutan dari *R-CNN*. *Fast R-CNN* dikembangkan untuk mengatasi masalah waktu pada proses *training* dan *testing*. *Fast R-CNN* menggunakan metode *selective search* untuk menghasilkan beberapa *proposal region* dalam melakukan identifikasi kemungkinan lokasi objek pada citra gambar. Dengan penerapan *ROI (Region of Interest) Pooling*, yang bertujuan untuk mengekstrak fitur pada bagian citra gambar yang sesuai dengan *proposal region* sehingga dapat mengurangi jumlah komputasi yang dibutuhkan.

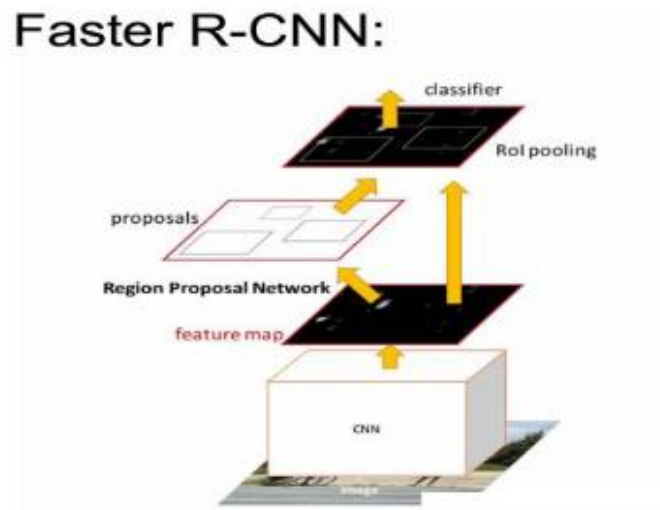
Feature yang telah diekstrak akan dikirimkan ke *Convolutional Neural Network (CNN)* untuk proses klasifikasi. Jaringan ini akan memprediksi probabilitas setiap kelas objek untuk setiap *proposal region*. *Fast R-CNN* dapat melakukan prediksi *refinement* atau perbaikan pada setiap *bounding box* yang diusulkan oleh *selective search* dengan tujuan untuk meningkatkan akurasi lokasi objek.



Gambar 2. 11 Teknik *ROI Pooling* pada *Fast R-CNN*
(Sumber : towardsdatascience.com)

2.7. *Faster Regional Convolutional Neural Network (Faster R-CNN)*

Faster Regional Convolutional Neural Network (Faster R-CNN) merupakan salah satu metode *deep learning* yang digunakan untuk mengenali suatu objek pada citra (Alamsyah & Pratama, 2019). *Faster R-CNN* mampu mendeteksi objek dengan sangat baik dengan teknik melakukan penelusuran ciri-ciri yang dimiliki oleh objek pada citra yang merujuk kepada atribut-atribut yang dapat digunakan untuk melakukan identifikasi objek tersebut. Terdapat tiga fungsi pada metode *Faster R-CNN*, yaitu menghasilkan dan mengklasifikasikan *feature map*, menemukan regresi, dan menambah lapisan *convolutional* oleh *regressor*.



Gambar 2. 12 Arsitektur Umum *Faster R-CNN* (Gavrilescu, 2018)

Faster R-CNN memiliki beberapa kelebihan utama, yaitu :

1. Menghasilkan performa yang lebih cepat dan akurat.
2. Proposal *region* tidak sebesar pada *R-CNN* sehingga mempercepat dan mempermudah proses kerja.
3. Dapat diterapkan pada proses klasifikasi berbasis *real-time*.

Metode *Faster R-CNN* memiliki dua tahapan utama, yakni proses ekstraksi dan pelatihan menggunakan *Region Proposal Network (RPN)*, serta klasifikasi fitur yang dihasilkan. Proses pelatihan terdiri dari empat tahap, yaitu: melatih jaringan *R-CNN*,

melatih jaringan *Faster R-CNN* dengan *RPN*, melatih ulang *RPN* menggunakan *Faster R-CNN*, dan melatih ulang *Faster R-CNN* dengan *RPN* yang telah diperbaharui.

Berikut merupakan penjabaran arsitektur umum dari *Faster R-CNN* :

- 1) *Convolutional Layer*, mempelajari citra gambar yang diunggah, mengekstraksi ciri khas dari citra gambar yang diunggah, kemudian akan membentuk *feature map* dari objek.
- 2) *Feature Map*, merupakan sebuah *map* yang berisikan representasi *vector* dari citra gambar yang telah diekstraksi oleh *Convolutional Layer*.
- 3) *Region Proposal Network (RPN)* adalah sebuah model yang berfungsi dalam pengelolaan *feature map*. *RPN* akan menghasilkan 2 *Convolutional Layer*, di mana *layer* yang pertama berfungsi untuk memprediksi letak sebuah objek pada citra gambar, sedangkan *layer* berikutnya akan memberikan *bounding box* di sekitar objek tersebut.
- 4) *ROI Pooling*, merupakan *Layer* yang berperan untuk melakukan ekstraksi informasi yang telah diproses oleh *RPN* dan selanjutnya diklasifikasikan oleh *classification layer*.
- 5) *Classification Layer*, merupakan *Layer* yang berperan dalam mengelompokkan, memberikan label, dan memberikan *bounding box* pada suatu objek pada citra gambar yang diproses oleh *RPN*.

2.8. Pytorch

Pytorch merupakan sebuah *framework* atau *library* pada bahasa pemrograman *python* yang digunakan dalam pengembangan *deep learning*. Pada proses pelatihan, *pytorch* memiliki kelebihan fleksibilitas komputasi grafis (*dynamic computation graph*) sehingga dapat melakukan perubahan struktur grafik komputasi selama *runtime*. Hal ini memudahkan pengembang melakukan *debugging* dan bereksplorasi pengembangan sistem ke depannya (Mishra, 2019).

2.9. Python

Python merupakan sebuah bahasa pemrograman yang *interpretatif*, semantik, dan dinamis dengan teori perancangan yang berfokus pada tingkat pembacaan kode untuk memudahkan dalam membaca *source code*. Dengan bahasa pemrograman *Python*,

source code dapat terlihat sederhana sehingga pengembang dapat membangun sistem terbaru karena *Python* memiliki *library* yang lengkap (Perkovic, 2015).

2.10. Android

Android adalah sebuah sistem operasi berbasis *Linux* untuk *device mobile* yang bersifat *open source*. *Android* menyediakan *Android SDK (Software Development Kit)* yang memberikan kemungkinan kepada pihak *developer* (pengembang) untuk merancang sebuah aplikasi dengan *library*, *tools*, dan *framework* yang telah disediakan. Dalam proses pengembangan, pihak pengembang dapat merancang dan membangun aplikasi menggunakan *Android Studio*.

2.11. Android Studio

Android Studio merupakan sebuah sistem perangkat lunak yang memiliki berbagai fitur dalam mengembangkan sebuah aplikasi berbasis *android*. Beberapa fitur yang dapat digunakan untuk membantu pengembang dalam membangun aplikasi berbasis *android* adalah sebagai berikut :

- Memiliki *framework* (kerangka kerja).
- *Support* bahasa pemrograman C++ dan NDK.
- Memiliki fitur alat pengujian yang ekstensif.
- Memiliki integrasi GitHub.
- *Fitur Lint* yang dapat membantu meningkatkan kinerja, kegunaan, dan mengatasi masalah seperti masalah kompatibilitas versi dalam pengembangan aplikasi android.

2.12. Confussion Matrix

Confussion Matrix adalah suatu metode yang digunakan dalam menilai kinerja pada sistem dalam melakukan proses klasifikasi. Metode ini menghasilkan *output* yang dapat dikategorikan menjadi dua *class* atau lebih (*multiclass*). Perbedaan antara nilai prediksi dan nilai aktual pada *Confussion Matrix* menghasilkan empat kombinasi.

Tabel 2. 1 *Confussion Matrix*

Aktual	Prediksi	
	Positive	Negative
	Positive	Negative
	TP	FN
	FP	TN

Berdasarkan tabel 2.1, *Confussion Matrix* merupakan hasil dari proses klasifikasi yang dikategorikan menjadi empat, yaitu :

- 1) TP (*True Positive*), merupakan data yang bernilai positif dan menghasilkan prediksi benar.
- 2) FN (*False Negative*), merupakan data yang bernilai positif, tetapi menghasilkan prediksi salah.
- 3) FP (*False Positive*), merupakan data yang bernilai negatif, tetapi menghasilkan prediksi benar.
- 4) TN (*True Negative*), merupakan data yang bernilai negatif dan menghasilkan prediksi salah.

Confussion Matrix dapat melakukan perhitungan *Performance Matrix* sebagai berikut :

2.12.1 Accuracy

Accuracy merupakan sebuah metode perhitungan yang digunakan dalam mengukur tingkat akurasi sebuah model dalam proses klasifikasi. *Accuracy* dapat menghitung tingkat jarak antara nilai prediksi dan nilai aktual, sehingga semakin tinggi tingkat benar sebuah data diklasifikasi oleh sebuah sistem maka nilai akurasi yang diperoleh juga tinggi. Persamaan yang digunakan untuk menentukan nilai *Accuracy* tertulis pada Persamaan 2.2.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \dots \dots \dots (2.2)$$

2.12.2 Precision

Precision merupakan suatu perbandingan antara total data relevan yang diperoleh dengan total semua data, baik yang relevan maupun tidak. Persamaan yang digunakan untuk menentukan nilai *Precision* tertulis di Persamaan 2.3.

$$Precission = \frac{TP + TN}{TP + FP} \times 100\% \dots \dots \dots (2.3)$$

2.12.3 Recall

Recall merupakan suatu metode perhitungan yang digunakan untuk menentukan perbandingan jumlah data yang sesuai berdasarkan hasil pengolahan data oleh sistem. Persamaan yang digunakan untuk menentukan nilai *Recall* terdapat pada Persamaan 2.4.

$$Recall = \frac{TP}{TP + FP} \times 100\% \dots \dots \dots (2.4)$$

2.12.4 F1-Score

F1-Score merupakan sebuah metode perhitungan yang digunakan untuk menentukan perbandingan dari nilai rata-rata *Precision* dan *Recall*. Persamaan yang digunakan untuk menentukan nilai *F1-Score* terdapat pada Persamaan 2.5.

$$F1 - Score = 2 \times \frac{Recall \times Precission}{Recall + Precision} \dots \dots \dots (2.5)$$

2.13. Penelitian Terdahulu

Berikut adalah beberapa penelitian terdahulu yang dijadikan penulis sebagai materi referensi. Penentuan referensi ini berdasarkan relevansi dan pengembangan lebih lanjut dalam menguji implementasi *computer vision* dalam melakukan identifikasi dan klasifikasi sebuah objek dengan penerapan metode yang relevan dengan algoritma *Faster R-CNN*.

Tabel 2. 2 Penelitian Terdahulu

No.	Peneliti	Judul	Keterangan
1	(I., Golpur; ParianJA, R.A; Chayjan,; 2021)	“ <i>Identification and Classification of Bulk Paddy, Brown, and White Rice Cultivars with Colour Features Extraction Using Image Analysis and Network</i> ”	Penelitian ini menunjukkan bahwa metode <i>Faster R-CNN</i> efektif dalam melakukan deteksi dan identifikasi penyakit pada tanaman padi, seperti penyakit blas-96%, bercak coklat-95%, dan hawar daun-94%.

Tabel 2. 3 Penelitian Terdahulu Lanjutan

No	Peneliti	Judul	Keterangan
2	(Esgario, Krohling, & Ventura, 2020)	<i>“Deep Learning for Classification and Severity Estimation of Coffee Leaf Biotic Stress”</i>	Pengujian kinerja algoritma <i>CNN</i> dengan struktur <i>AlexNet</i> , <i>GoogleNet</i> , <i>ResNet 50</i> , dan <i>VGG16</i> untuk klasifikasi 5 jenis penyakit dengan <i>dataset Biotic Stress</i> dan <i>Severity</i> yang terdiri dari 1685 citra, serta <i>dataset Symptom</i> yang terdiri dari 2722 citra. Hasil penelitian menunjukkan bahwa <i>ResNet50</i> mencapai kinerja tertinggi pada <i>dataset Symptom</i> dengan tingkat akurasi mencapai 97,07%. Sebaliknya, performa terendah <i>ResNet50</i> terlihat pada data <i>Severity</i> yang menunjukkan tingkat ketidakstabilan dengan nilai akurasi sebesar 78,57%.
3	(Montalbo & Hernandez, 2020)	<i>“Classifying Barako Coffe Leaf Diseases Using Deep Convolutional Models”</i>	Penelitian ini memanfaatkan <i>Convolutional Neural Network (CNN)</i> dengan struktur <i>ResNetV2-152</i> , <i>VGG16</i> , dan <i>Xception</i> untuk mengklasifikasi 4 jenis penyakit pada 4667 citra data. Tingkat akurasi tertinggi tercatat sebesar 97,20% dengan <i>VGG16</i> , sementara tingkat akurasi terendah adalah 90,83% pada arsitektur <i>ResNetV2-152</i> .
4	(Sabrina & Al Maki, 2022)	<i>“Klasifikasi Penyakit pada Tanaman Kopi Robusta Berdasarkan Citra Daun Menggunakan Convolutional Neural Network”</i>	Penelitian ini menerapkan <i>Convolutional Neural Network (CNN)</i> dengan struktur <i>Efficient-B0</i> untuk mengklasifikasi 1560 citra daun ke dalam 6 kelas yang berbeda. Akurasi hasil penelitian mencapai 94%.

Tabel 2. 4 Penelitian Terdahulu Lanjutan

No	Peneliti	Judul	Keterangan
5	(Ma, et al., 2018)	<i>“A Recognition Method for Cucumber Disease Using Leaf Symptom Images Based On Deep Convolutional Neural Network”</i>	Penelitian ini melakukan klasifikasi penyakit pada <i>dataset</i> daun timun berjumlah 1.184 data citra . Metode yang digunakan adalah algoritma <i>CNN</i> dengan arsitektur <i>AlexNet</i> dengan memperoleh tingkat akurasi mencapai 94%




BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1. Data Penelitian

Data yang digunakan sebagai bahan penelitian ini adalah *dataset* penyakit pada daun kopi bersumber dari *platform open source* Kaggle. *Dataset* terdiri dari 3 kelas, yaitu Karat Daun (*Leaf Rust*), Pengorok Daun (*Leaf Miner*), dan Hawar Daun (*Leaf Blight*). *Dataset* terdiri dari 3.600 data foto dengan 1.200 data citra setiap *class* dan dibagi menjadi *data_testing*, *data_validation*, dan *data_testing*. *Data_validation* digunakan untuk mencegah terjadinya *overfitting*. Pembagian data ini dilakukan dengan presentase *data_testing* sebesar 80%, *data_validation* sebesar 10%, dan *data_testing* sebesar 10%.

Tabel 3. 1 Kategori Dataset

No	Gambar	Keterangan
1		Hawar Daun (<i>Leaf Blight</i>)
2		Penggorok Daun (<i>Leaf Miner</i>)
3.		Karat Daun (<i>Leaf Rust</i>)

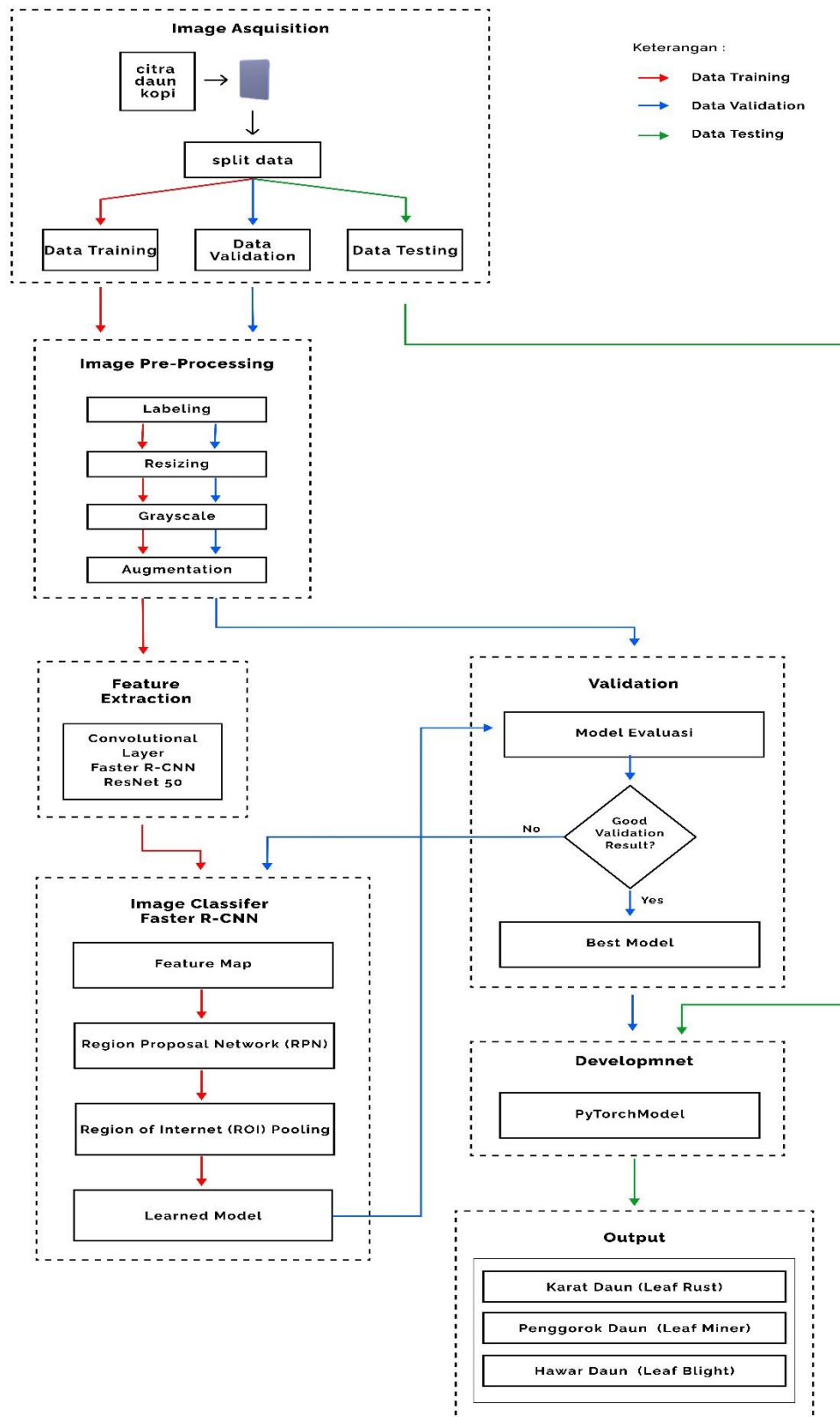
3.2. Analisis Sistem

Penelitian dilakukan dengan beberapa tahapan. Adapun tahapan pertama ialah pengumpulan data (*asquisition*) citra yang digunakan sebagai *data_training* dan *data_validation*. Jenis citra yang dikumpulkan berdasarkan kategori yang akan

digunakan, yaitu Karat Daun (*Leaf Rust*), Pengorok Daun (*Leaf Miner*), dan Hawar Daun (*Leaf Blight*).

Tahapan kedua adalah *image pre-processing*. Pada tahapan ini, citra akan diberikan *labelling*, *resizing*, dan *augmentation*. Proses *labelling* bertujuan untuk memberikan label atau tanda pada setiap citra yang akan diklasifikasikan di proses pelatihan data. Proses *resizing* bertujuan untuk mengubah ukuran setiap citra gambar menjadi 300 x 300 *pixel* agar semua citra gambar yang akan diunggah ke sistem memiliki format ukuran yang sama. Proses *augmentation* bertujuan untuk membuat beberapa variasi dari citra gambar asli dengan perubahan-perubahan seperti memutar data citra pada interval tertentu. Perubahan tersebut bisa dilakukan menjadi *flip* secara *vertical* dan *horizontal* dengan *rotate 90%*. Hal ini bertujuan untuk peningkatan kuantitas *dataset* sehingga dapat mengurangi *overfitting* (model yang terlalu mirip dengan *data_training*) dan membantu model dalam memprediksi *data_test* selain *dataset*. Setelah data citra melalui proses tahapan *pre-processing*, data citra akan dibagi menjadi *data_training* dan *data_testing*.

Tahapan selanjutnya adalah tahapan klasifikasi pada data citra. Penelitian ini menggunakan ekstraksi fitur yang disediakan oleh *Faster Regional Convolutional Neural Network* dengan bantuan *pre-trained* model *ResNet50*. Setelah proses tahapan ini selesai, maka akan diperoleh *output* berupa *class* yang diklasifikasikan.



Gambar 3. 1 Arsitektur Umum

Berikut merupakan penjelasan dari arsitektur umum :

3.2.1 Image Acquisition

Image Acquisition atau akusisi citra merupakan tahapan proses penangkapan citra analog sehingga diperoleh citra digital. Pada tahapan ini, penulis melakukan pengumpulan *dataset* yang akan digunakan dalam pengembangan sistem. *Dataset* yang digunakan berekstensi .jpeg/.jpg yang dikelompokkan (*split data*) menjadi tiga bagian, yaitu 80% data latih (*data_training*), 10% data validasi (*data_validation*), dan 10% data uji (*data_test*).

3.2.2 Image Pre-Processing

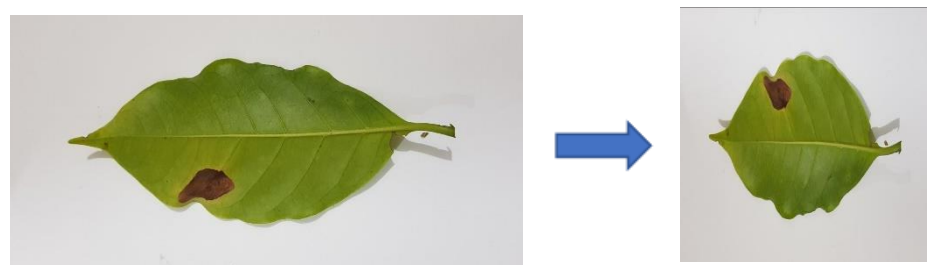
Tahapan ini merupakan serangkaian tahapan-tahapan yang dibutuhkan dalam menyiapkan data citra agar data yang akan digunakan lebih bervariasi dan efisien untuk melakukan pelatihan pada tahapan berikutnya. Adapun proses-proses yang terjadi pada tahapan *image pre-processing* adalah sebagai berikut :

1) Labeling

Labeling dilakukan untuk memberikan label pada setiap citra gambar yang akan digunakan untuk proses *training*.

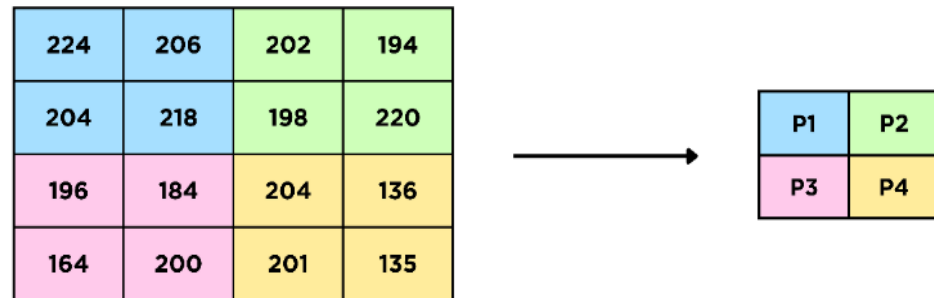
2) Resizing

Resizing merupakan proses untuk mengubah ukuran suatu gambar yang bertujuan agar seluruh gambar yang akan diunggah ke sistem memiliki ukuran yang sama. Semakin banyak jumlah *pixel*, maka pemrosesan sistem akan semakin lama. Berdasarkan hal tersebut, *resizing* dibutuhkan agar ukuran piksel lebih sedikit dan semua gambar memiliki ukuran *pixel* yang sama. (Gavrilescu, 2018)



Gambar 3. 2 *Resizing* Citra

Pada tahapan ini, citra gambar akan dilakukan *resizing* atau mengubah ukuran *pixel* citra. Citra gambar akan diubah menjadi ukuran 300 x 300 *pixel*. Hal ini dilakukan guna pada saat proses *training* menjadi lebih efektif dan efisien.



Gambar 3. 3 Perhitungan proses *Resizing* (Nguyen & et al, 2018)

Gambar 3.2 merupakan perhitungan *resizing* suatu citra dari ukuran 4 x 4 *pixel* menjadi 2 x 2 *pixel*. Proses ini melibatkan perhitungan nilai rata-rata dari 4 *pixel* yang berdekatan sebagai *pixel* baru dalam citra yang lebih kecil yaitu 2 x 2 *pixel*. Adapun detail dari perhitungan tersebut dapat dilihat pada perhitungan berikut :

1. $(224 + 206 + 204 + 218) : 4 = 213$
2. $(202 + 194 + 198 + 220) : 4 = 203$
3. $(196 + 184 + 164 + 200) : 4 = 186$
4. $(204 + 136 + 207 + 135) : 4 = 169$

Nilai *pixel* baru yang dihasilkan oleh perhitungan *resizing* dengan *matrix* di atas terdapat pada gambar 3.6 berikut.

213	203
186	169

Gambar 3. 4 Nilai matrix setelah proses *resize* (Nguyen & et al, 2018)

3) *Augmentation*

Augmentation merupakan tahapan yang berfungsi untuk memodifikasi citra dasar guna menghasilkan *dataset* baru yang memiliki posisi dan bentuk yang berbeda dari citra dasar. Proses yang terjadi pada tahapan ini ialah

penambahan jumlah dan variasi *dataset* yang digunakan pada penelitian ini. Proses *Augmentation* citra yang digunakan oleh penulis adalah *horizontal flip*, *vertical flip*, dan *90° rotate*. Hasil dari tahapan *Augmentation* akan menambah jumlah *dataset* awal yang berjumlah 3600 menjadi 10.800 citra gambar.

```

PROCEDURE save_transformed_images(data, folder, name, index)
  CALL data.transpose(pil.FLIP_LEFT_RIGHT).save(folder + '/' + name + '/'
+ STR(index) + '.JPG')
  index += 1
  CALL data.transpose(pil.ROTATE_90).save(folder + '/' + name + '/' +
STR(index) + '.JPG')
  index += 1
  CALL
data.transpose(pil.ROTATE_90).transpose(pil.FLIP_TOP_BOTTOM).save(fo
lder + '/' + name + '/' + STR(index) + '.JPG')
  index += 1
END PROCEDURE

```

Gambar 3. 5 *Pseudocode* Augmentasi Data

3.2.3 Feature Extraction

Feature Extraction merupakan tahapan dalam pengambilan *feature* atau ciri dari suatu bentuk objek yang berfungsi dalam pengenalan gambar pada *dataset*. Pada tahapan ini, *convolutional layer* digunakan untuk melakukan ekstraksi fitur-fitur dan memeriksa bagian-bagian penting yang terdapat pada citra gambar. Citra gambar yang diunggah selanjutnya diproses ke dalam beberapa *convolutional layer* menggunakan filter ekstraksi dengan perhitungan *dot product* antar gambar dan *filter*. Selanjutnya dilakukan *convolve* (menggeser) *filter* pada gambar untuk membentuk *activation map*. Hasilnya, *layer* akan menyusun dan membentuk *feature map* dari objek yang terdapat pada citra.

3.2.4 Image Classifier

Image Classifier merupakan tahapan di mana mesin akan dilatih untuk mengenali objek dan melakukan klasifikasi citra gambar ke dalam kelas jenis penyakit. Proses klasifikasi dilakukan dengan metode *Convolutional Neural Network* berbasis algoritma *Faster R-*

CNN dengan *platform tools open source* yaitu *google collab* dengan menggunakan *framework Pytorch*.

```
PROCEDURE train_and_summarize_model(train_data,
validation_data, batch_size, epochs)
    model = image_classifier.create(train_data,
validation_data=validation_data, batch_size=batch_size,
epochs=epochs)
    CALL model.summary()
END PROCEDURE
```

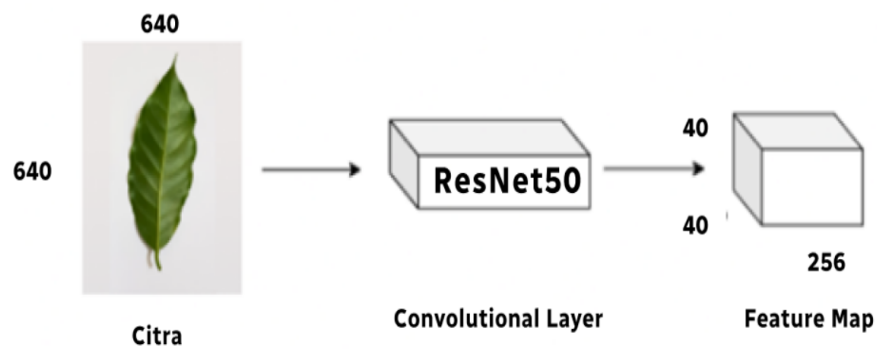
Gambar 3. 6 *Pseudocode* Proses Klasifikasi

Pseudocode pada gambar 3.6 merupakan representasi model pada proses *image classification* dengan dua *argument*, yaitu *data_training* dan *data_validation* yang digunakan untuk melatih data dan melakukan evaluasi pada model. *Batch size* berfungsi menentukan ukuran *dataset* yang digunakan dalam pelatihan model. Jumlah *epochs* berfungsi menentukan berapa kali model akan melakukan pelatihan pada *dataset*. *Model summary* merupakan suatu metode untuk menampilkan arsitektur dari model yang telah dirancang. Melalui *data_train* dan *data_validation*, model dapat memahami korelasi antar citra gambar dan label yang sudah ditetapkan, sehingga model dapat memprediksi label yang tepat untuk citra gambar yang tidak ada di *dataset*.

Berikut merupakan tahapan-tahapan proses dari *Faster R-CNN*.

3.2.4.1 *Feature Map*

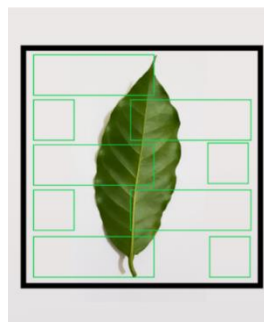
Feature Map merupakan *map* yang dihasilkan oleh *convolutional layer*, merepresentasikan informasi mengenai *vector* dan *image*. *Convolutional layer* akan menghasilkan dua *feature map*, di mana *feature map* pertama akan diproses oleh *RPN (Region Proposal Networ)* untuk mendapatkan *region proposal*, sementara *feature map* kedua akan dikirimkan langsung ke *polling layer*. Citra gambar direpresentasikan dengan *height*, *width*, dan *depth* yang akan diolah pada lapisan tengah menggunakan *Convolutional Neural Network (CNN)* yang telah dilatih sebelumnya sehingga dapat ditentukan *convolutional feature map*. Ukuran dimensi citra gambar pada *feature map* awalnya yaitu 640 x 640 x 3, setelah dilatih menggunakan *ResNet50*. Representasi dari proses *feature map* dapat dilihat pada gambar 3.7 berikut.



Gambar 3. 7 Mekanisme *Feature Map*

3.2.4.2 Region Proposal Network (RPN)

Region Proposal Network berfungsi memproses *feature map* yang telah diolah pada *convolutional layer*, kemudian melakukan prediksi objek pada citra gambar dengan menandai objek tersebut ke dalam *bounding box*. *RPN* tersusun dari dua *convolutional layer* yang memiliki peran berbeda, yaitu *layer* pertama berfungsi mendeteksi lokasi objek dan *layer* kedua berfungsi memprediksi *bounding box*. Nilai 40 x 40 pada *feature map* disebut dengan *anchors*. Penentuan rasio dan ukuran secara spesifik diperlukan untuk setiap *anchors*. Adapun rasio dari *anchors* tersebut ialah 1:1, 1:2, 2:1, dan 128, 256%, 512% untuk ukuran tiga citra. Seluruh *anchors* memiliki ukuran $3 \times 3 = 9$ kotak yang berarti $40 \times 40 \times 9 = 14.400$ kotak pada satu citra. Hasil yang terlalu besar maka dipilih sebanyak 256 kotak sebagai *mini batch*. Proses *RPN* dapat dilihat pada gambar 3.8.



Gambar 3. 8 *Region Proposal Network (RPN)*

Pada gambar 3.8, kotak berwarna hitam merupakan *anchors* yang menilai sebagai objek citra daun kopi yang akan dideteksi dan kotak berwarna hijau merupakan kumpulan *anchors* yang menilai bukan objek penyakit pada daun tanaman kopi yang dideteksi. Dalam proses *training*, semua *anchors* akan

menentukan suatu area kotak berisi objek dan kemudian melakukan pengelompokan ke dalam dua jenis kategori yang berbeda, yaitu $(IoU) > 0,7$ dan $(IoU) < 0,3$. *Anchor*s yang mempunyai nilai kebenaran objek *intersection over union* (IoU) $> 0,7$ akan dikategorikan sebagai objek dan apabila $(IoU) < 0,3$ akan dinyatakan bukan objek.

3.2.4.3 Region of Interest Polling

Region of Interest (ROI) adalah suatu area tertentu pada citra yang diprediksi memiliki tingkat kemungkinan tertinggi terdapat objek yang dideteksi. Fungsi *layer* ini berguna untuk menyesuaikan ukuran *feature map* dan *region proposal* yang telah dihasilkan oleh *RPN (Region Proposal Network)* dan selanjutnya diproses kembali pada *classification layer*. Sebelum menjalani proses klasifikasi, objek harus melewati proses *ROI Polling* yang memperhitungkan koordinat dan ukuran asli dari *ROI*.



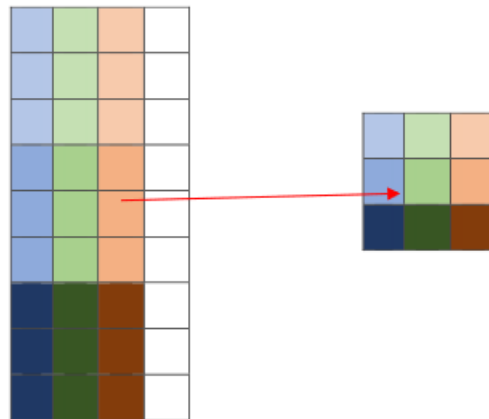
Gambar 3. 9 Target ROI

Gambar 3.9 merupakan objek yang ditargetkan oleh *ROI* dengan ukuran 150 x 300 yang kemudian akan dibagi dengan 32 sebagai faktor skala.

Lebar : $150/32 = 4,6$ (dibulatkan menjadi 4)

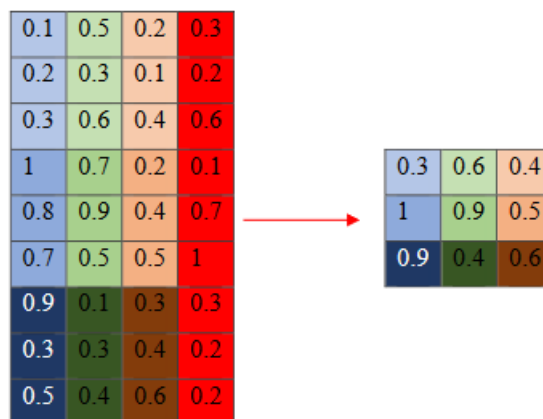
Tinggi : $300/32 = 9,3$ (dibulatkan menjadi 9)

Tahapan selanjutnya mengubah *ROI* berukuran 9 x 4 x 512 melalui *fully connected layer* yang memiliki ukuran tetap 3 x 3 x 512.



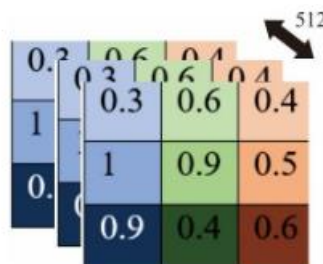
Gambar 3. 10 Proses *ROI Pooling* (Karpathy, 2017)

Setelah proses *ROI Pooling*, data digabungkan menjadi *matrix* berukuran $3 \times 3 \times 512$. Hal ini dilakukan karena faktor kuantitas, maka baris paling kanan akan dihilangkan seluruhnya. Proses penggabungan data dapat dilihat pada gambar 3.11.



Gambar 3. 11 Pemetaan *Data Pooling* (Karpathy, 2017)

Gambar 3.11 merupakan representasi dari hasil penggabungan keseluruhan *data pooling*. Penggabungan pada *data pooling* hanya mengambil nilai tertinggi dari *matrix*, maka ukuran awal yang beragam akan diubah menjadi satu ukuran yang bernilai sama pada semua matriks *ROI*.



Gambar 3. 12 *ROI Pooling 3 x 3* (Karpathy, 2017)

Gambar 3.12 merupakan representasi ratusan *matrix* $3 \times 3 \times 512$ dari hasil *ROI Pooling*. Masing-masing *matrix* akan dikirim melalui seluruh *neural network* untuk setiap model, di mana setiap model akan menghasilkan klasifikasi objek berupa klasifikasi jenis penyakit daun.

3.2.5 Learned Model

Learned Model merupakan model pembelajaran mesin yang telah melakukan proses pelatihan data. Model ini digunakan sistem untuk melakukan klasifikasi objek yang dideteksi ke dalam *class* yang ditentukan.

3.2.6 Proses Training

Proses pelatihan (*training*) dilakukan untuk mempelajari sistem pada citra yang diunggah agar menghasilkan *output* yang diinginkan melalui metode *machine learning*. *Output* yang dihasilkan akan digunakan dalam proses *testing*. Pada tahapan pelatihan ini, data *input* akan dibagi menjadi tiga bagian, yaitu *data_training*, *data_validation*, dan *data_testing*.

- *Data Training*

Data Training merupakan data yang digunakan untuk pelatihan model. Jumlah citra yang digunakan sebanyak 2.880 citra.

- *Data Validation*

Data Validation adalah data yang akan digunakan untuk melakukan evaluasi kinerja model pada proses pelatihan. Tujuan penggunaan data ini adalah untuk mencegah model tidak *overfitting* ketika proses pelatihan. Jumlah citra yang digunakan sebanyak 360 citra.

- *Data Testing*

Data Testing merupakan data yang akan digunakan untuk menguji dan evaluasi kinerja pada model. Hal ini diperlukan untuk mengetahui kemampuan model dalam melakukan klasifikasi pada citra yang tidak dikenali sebelumnya. *Data testing* menjadi data yang objektif dalam mengevaluasi kinerja model karena data ini tidak terlibat dalam proses pelatihan model. Jumlah citra yang digunakan sebanyak 360 citra.

3.2.7 Testing

Testing adalah tahapan yang dilakukan untuk menguji kinerja sistem dalam melakukan klasifikasi jenis penyakit daun pada tanaman kopi dengan menggunakan citra yang tidak dikenali sebelumnya sebagai *input*. Kinerja sistem dapat diketahui dengan cara menghitung nilai akurasi yang diperoleh sistem saat melakukan proses klasifikasi. Sistem akan melakukan klasifikasi berdasarkan model yang telah dibuat melalui serangkaian algoritma *Faster R-CNN*.

3.2.8 Output

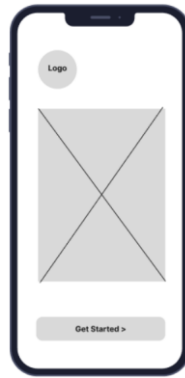
Output merupakan hasil akhir dari serangkaian proses yang terjadi pada sistem. Pada penelitian ini, *output* yang diharapkan adalah informasi mengenai jenis penyakit daun berdasarkan objek yang terdapat pada citra gambar daun tanaman kopi. *Output* pada penelitian ini dibagi menjadi 3 *class*, Karat Daun (*Leaf Rust*), Hawar Daun (*Leaf Blight*), dan Pengorok Daun (*Leaf Miner*).

3.3. Perancangan Antarmuka Sistem

Perancangan antarmuka sistem merangkum skema alur perancangan dan penyusunan elemen-elemen pendukung dalam pembuatan desain antarmuka (*user interface*) sistem yang bertujuan untuk menciptakan pengalaman pengguna yang efisien dan efektif. Pada penelitian ini, *interface* yang digunakan untuk pengguna adalah desain berbasis *mobile*. Desain ini digunakan dengan tujuan agar penggunaan sistem yang lebih praktis sehingga mempermudah pengguna (*user*) dalam pemanfaatan fungsi yang disediakan.

3.3.1 Rancangan Tampilan Splash Screen

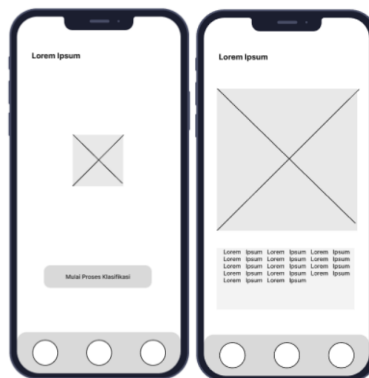
Rancangan tampilan awal (*splash screen*) pada aplikasi terdapat tombol mulai “*Get Started*”. Ketika pengguna menekan tombol “*Get Started*” maka tampilan akan dialihkan ke halaman utama. Pada halaman ini, terdapat 3 menu, yaitu menu “*Check*”, menu “*Disease*”, dan menu “*About*”. Setiap menu memiliki fungsi yang berbeda sesuai dengan penamaan tombol menu. Menu “*Check*” berfungsi untuk memulai proses unggah foto dari galeri dan melakukan proses klasifikasi, menu “*Disease*” berisi konten informasi mengenai jenis penyakit yang dapat diklasifikasikan pada aplikasi, dan menu “*About*” berisi konten mengenai pihak yang terlibat dalam penelitian.



Gambar 3. 13 Rancangan Tampilan *Splash Screen*

3.3.2 Rancangan Tampilan Menu Check

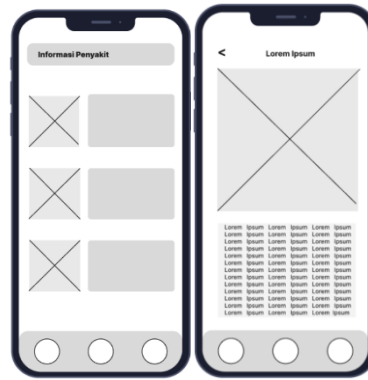
Rancangan desain antarmuka pada menu “*Check*” memuat proses klasifikasi yang mencakup tahapan-tahapan yang akan dilakukan oleh pengguna, di mana pengguna dapat mengunggah citra gambar dari galeri. Halaman ini dapat diakses oleh pengguna melalui tombol menu “*Check*” yang terdapat pada tampilan beranda. Pengguna dapat unggah foto dan menekan tombol “Mulai Proses Klasifikasi” untuk memulai proses. Setelah proses klasifikasi selesai, maka tampilan secara otomatis akan dialihkan ke tampilan hasil klasifikasi yang memuat informasi mengenai jenis penyakit dan *time prediction*.



Gambar 3. 14 Rancangan Tampilan Halaman Menu *Check*

3.3.3 Rancangan Tampilan Halaman Menu Disease

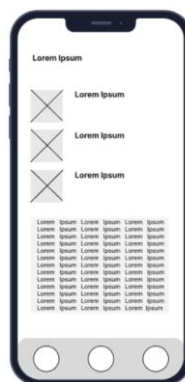
Rancangan desain antarmuka menu “*Disease*” memuat halaman yang berisikan informasi jenis-jenis penyakit pada daun tanaman kopi yang dapat diklasifikasikan oleh aplikasi. Rancangan halaman ini dapat dilihat pada gambar 3.15.



Gambar 3. 15 Rancangan Tampilan Halaman *Menu Disease*

3.3.4 Rancangan Tampilan Menu About

Rancangan desain antarmuka menu “About” memuat konten berisi informasi mengenai pihak yang terlibat dalam pengembangan sistem.



Gambar 3. 162 Rancangan Tampilan Halaman *Menu About*

BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem

Tahapan implementasi sistem pada penelitian pengembangan sistem yang mampu melakukan klasifikasi jenis penyakit daun tanaman kopi menggunakan algoritma *Faster Regional Convolutional Neural Network (Faster R-CNN)*, dibutuhkan perangkat lunak (*software*), perangkat keras (*hardware*), dan pendukung lainnya, yaitu :

4.1.1 Perangkat (Hardware) dan Perangkat Keras Lunak (Software)

Spesifikasi perangkat *hardware* dan *software* yang digunakan dalam pengembangan aplikasi ini adalah :

1. Laptop Acer Nitro AN515-57
2. *Processor* Intel(R) Core(TM) i5-11400H @ 2.70GHz (12 CPUs), ~2.7GHz
3. *RAM* 8 GB
4. *Storage SSD* 512 GB
5. Sistem Operasi *Windows 11 Home 64-bit*
6. *Android Studio*
7. *Google Collab Notebooks*
8. *Tensorflow*
9. *Node JS 18.18.0*
10. *React Native 0.71.11*

4.1.2 Implementasi Data

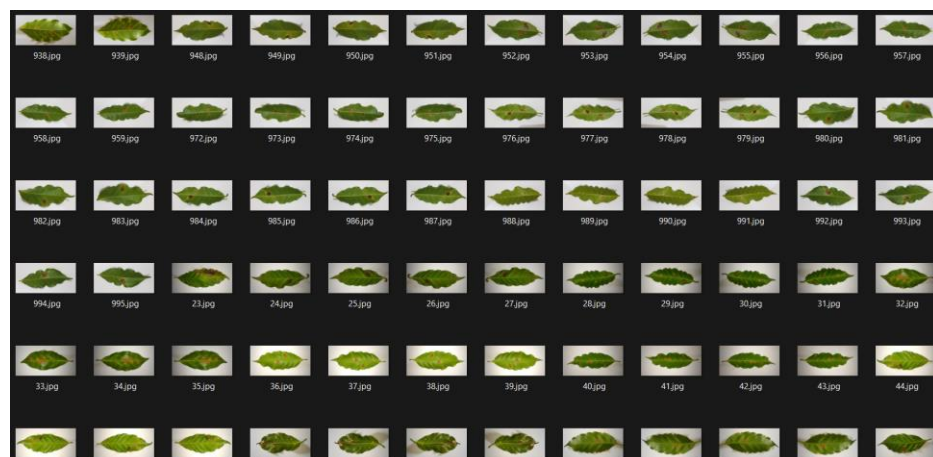
Dataset yang digunakan dalam penelitian bersumber dari *platform open dataset* yaitu Kaggle. Pengelompokan citra gambar daun pada *dataset* berdasarkan jenis penyakit daun yang menyerang tanaman kopi, yaitu penyakit Karat Daun (*Leaf Rust*), Penggorok Daun (*Leaf Miner*), dan penyakit Hawar Daun (*Leaf Blight*). Jumlah total citra gambar daun pada *dataset* berjumlah 3600 citra gambar dengan pembagian sebanyak 1.200 citra gambar untuk setiap kategori penyakit



Gambar 4. 1 Data Karat Daun (*Leaf Rust*)



Gambar 4. 2 Data Hawar Daun (*Leaf Blight*)



Gambar 4. 3 Data Penggorok Daun (*Leaf Miner*)

4.2. Implementasi Perancangan Antarmuka

4.2.1 Tampilan Splashscreen

Rancangan tampilan awal (*splash screen*) pada aplikasi terdapat logo aplikasi dan tombol mulai “*Get Started*”. Ketika pengguna menekan tombol “*Get Started*” maka tampilan akan dialihkan ke halaman utama. Pada halaman ini, terdapat 3 menu, yaitu menu “*Scan*”, menu “*Disease*”, dan menu “*About*”. Setiap menu memiliki fungsi yang berbeda sesuai dengan penamaan tombol menu. Menu “*Scan*” digunakan oleh pengguna untuk memulai proses *scanning* atau *upload photo from gallery*, menu “*Disease*” berisi konten informasi mengenai jenis penyakit yang dapat diklasifikasikan pada aplikasi, dan menu “*Tutorial*” berisi konten tata cara penggunaan aplikasi.

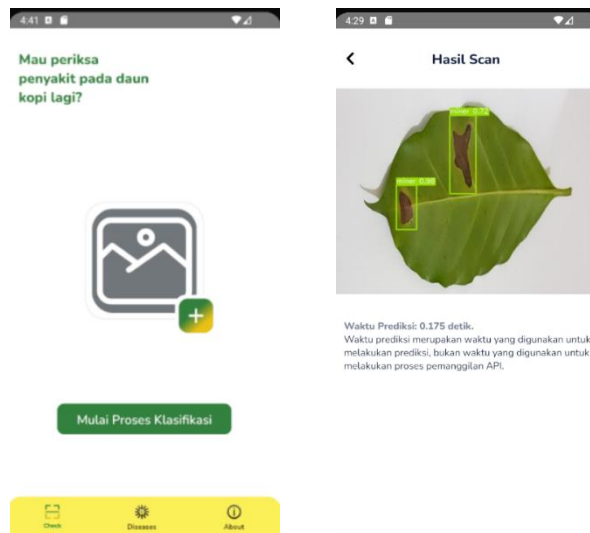


Gambar 4. 4 Tampilan *Splashscreen*

4.2.2 Tampilan Menu Check

Tampilan menu “*Check*” mencakup tahapan-tahapan yang dilakukan oleh pengguna untuk melakukan proses klasifikasi. Halaman ini dapat diakses oleh pengguna melalui tombol menu “*Check*” yang terdapat pada bagian bawah tampilan. Pada tampilan ini, terdapat sejumlah tombol yang memiliki fungsinya masing-masing. Tombol *icon* gambar yang digunakan untuk mengunggah citra gambar daun yang tersimpan di galeri, dan tombol “Mulai Proses Klasifikasi” dapat digunakan untuk memulai klasifikasi setelah muncul citra gambar yang akan diklasifikasikan pada kotak *review*. Setelah

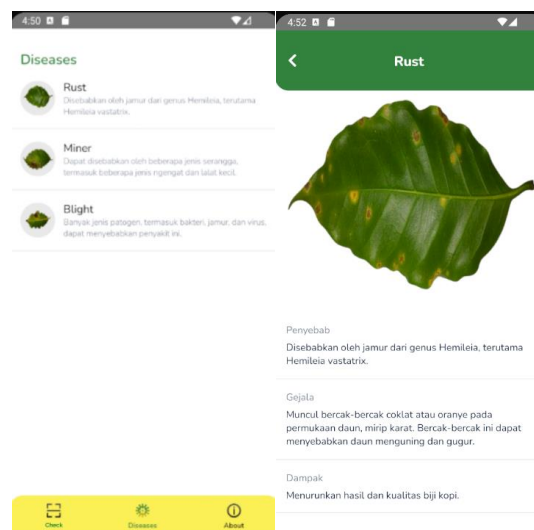
proses klasifikasi selesai, maka tampilan secara otomatis akan dialihkan ke tampilan hasil klasifikasi yang memuat informasi mengenai hasil klasifikasi dan *time predict*.



Gambar 4. 5 Tampilan Menu Check

4.2.3 Tampilan Menu Disease

Tampilan menu “Disease” memuat konten informasi mengenai jenis penyakit daun pada tanaman kopi yang dapat diklasifikasikan oleh aplikasi. Setiap jenis penyakit disertakan penyebab, gejala dan dampak yang ditimbulkan oleh penyakit tersebut.



Gambar 4. 6 Tampilan Menu Disease

4.2.4 Tampilan Menu About

Tampilan menu *About* berisi konten informasi pihak yang terlibat dalam penelitian ini. Pada tampilan ini akan tertera informasi mengenai penulis dan dosen pembimbing penulis.



Gambar 4. 7 Tampilan *Menu About*

4.3. Prosedur Penggunaan Aplikasi

Prosedur penggunaan aplikasi merupakan panduan umum bagi pengguna dalam pengoperasian aplikasi yang telah dibuat. Pengguna dapat mengakses aplikasi secara terbuka. Ketika pengguna menggunakan aplikasi, maka halaman *splash screen* akan tampil yang memuat informasi mengenai aplikasi. Pengguna dapat menekan tombol “*Get Started*” untuk mengeksplor aplikasi. Selanjutnya, tampilan halaman utama akan muncul dan terdapat 3 tombol utama dalam pengoperasian aplikasi, yaitu *Check*, *Disease*, dan *About*. Tombol *Check* digunakan apabila pengguna ingin melakukan proses klasifikasi.

Pada tampilan menu *Check*, pengguna dapat memilih menggunakan data citra gambar yang sudah tersimpan di galeri. Setelah memilih, pengguna dapat melihat tampilan *review* dari citra gambar dan menekan tombol “Mulai Proses Klasifikasi” yang terdapat di bawah untuk memulai proses. Hasil klasifikasi akan muncul dengan perubahan tampilan yang akan memuat informasi mengenai hasil klasifikasi. Informasi lengkap mengenai jenis penyakit tersebut dapat dilihat pada menu *Disease* yang telah

disediakan. Adapun informasi yang dimuat adalah konten mengenai penyebab, gejala, dan dampak yang disebabkan oleh penyakit tersebut.

4.4. Pelatihan Sistem

Pelatihan sistem dilakukan dengan menerapkan algoritma *Faster R-CNN* pada 8.640 data citra gambar daun kopi setelah melewati proses *augmentasi*. Jumlah data tersebut berasal dari 2.880 citra gambar daun tanaman kopi dengan presentase 80% data digunakan sebagai *data_training* kemudian dikali 3 kali proses *augmentasi*.

Pelatihan sistem menggunakan beberapa *layer*. *Layer* yang digunakan adalah *conv2d layer*, *maxpooling2d layer*, *batch normalization*, *drop out layer*, *flatten layer*, dan *dense layer*. *Conv2D layer* adalah *convolutional layer* dua dimensi yang digunakan untuk melakukan ekstraksi fitur dari citra gambar. *MaxPooling2D* adalah operasi *max-pooling* pada citra untuk membantu mengurangi dimensi data dan mempertahankan fitur-fitur penting dari citra. *Batch Normalization* digunakan untuk melakukan menormalisasi *batch* pada data agar membantu dalam pelatihan model yang lebih stabil dengan konvergensi yang lebih cepat. Kemudian, *flatten layer* digunakan sebelum *Layer Dense* untuk meratakan *tensor* multidimensi menjadi *tensor* satu dimensi. Terakhir yaitu *Dense Layer*, yaitu untuk menghubungkan setiap *neuron* yang ada di setiap lapisan.

Terdapat beberapa percobaan konfigurasi pelatihan yang telah dilakukan pada penelitian ini, di antaranya adalah percobaan dengan menggunakan *Batch Size* 16 dan *epoch* 24 dengan perolehan akurasi sebesar 88%. Kemudian percobaan selanjutnya dilakukan dengan meningkatkan ukuran *batch size* menjadi 30 dan *epoch* 15 dengan perolehan akurasi sebesar 90%. Kemudian, dilakukan Kembali percobaan dengan menaikkan ukuran *batch size* menjadi 64 dan *epoch* 32 memperoleh akurasi maksimal yaitu 95%.

Hal demikian didukung dengan penelitian yang dilakukan oleh (Suwinto & Kaunang, 2022) yang menyatakan bahwa ukuran *batch size* yang lebih besar walaupun dengan *epoch* yang lebih sedikit dapat menghasilkan *accuracy* yang lebih tinggi dibandingkan dengan ukuran *batch size* yang lebih kecil dan *epoch* yang lebih besar. Berdasarkan beberapa percobaan yang telah dilakukan maka penelitian ini menggunakan *batch size* sebesar 64 dan *epoch* 32.

Tabel 4. 1 Hasil Pelatihan Sistem

<i>Epoch</i>	<i>Loss</i>	<i>Accuracy</i>	<i>Val_Loss</i>	<i>Val_Accuracy</i>
1	0.4751	0.7966	0.3123	0.8695
2	0.1829	0.9282	0.2734	0.8980
3	0.0763	0.9725	0.4441	0.8814
4	0.0484	0.9830	0.4937	0.8621
5	0.0313	0.9903	0.4209	0.4209
6	0.0345	0.9875	0.3497	0.9219
7	0.0461	0.9845	0.3529	0.9200
8	0.0321	0.9886	0.4151	0.9053
9	0.0167	0.9938	0.3448	0.9210
10	0.0183	0.9935	0.3989	0.9191
11	0.0204	0.9929	0.4625	0.9044
12	0.0238	0.9915	0.3115	0.9329
13	0.0159	0.9938	0.4379	0.9182
14	0.0228	0.9917	0.4065	0.9283
15	0.0053	0.9986	0.3423	0.9375
16	0.0042	0.9995	0.3423	0.9375
17	0.0064	0.9989	0.3788	0.9329
18	0.0364	0.9856	0.3638	0.9274
19	0.0067	0.9982	0.4109	0.9283
20	0.0168	0.9944	0.7020	0.8732
21	0.0346	0.9893	0.3758	0.9246
22	0.0104	0.9960	0.4006	0.9329
23	0.0114	0.9967	0.3829	0.9375
24	0.0054	0.9980	0.3322	0.9384
25	0.0038	0.9985	0.4104	0.9320
26	0.0161	0.9945	0.6046	0.8980
27	0.0052	0.9983	0.3883	0.9366
28	0.0242	0.9937	0.4176	0.9228
29	0.0216	0.9936	0.4286	0.9265
30	0.0084	0.9983	0.4146	0.9292

Tabel 4. 2 Hasil Pelatihan Sistem Lanjutan

<i>Epoch</i>	<i>Loss</i>	<i>Accuracy</i>	<i>Val_Loss</i>	<i>Val_Accuracy</i>
31	0.0086	0.9995	0.3877	0.9366
32	0.0042	0.9984	0.4312	0.9311

Berdasarkan Tabel 4.1 dan 4.2, dapat diperhatikan bahwa sistem mengalami peningkatan performa dalam mengenali objek yang terdapat pada citra daun. Hal ini dapat dinilai dari nilai *loss* yang semakin berkurang. Nilai *Loss* merupakan nilai yang mengukur kinerja model dalam mendapati kesalahan dalam memprediksi label pada data *training* dengan tujuan untuk meminimalkan nilai *loss* selama pelatihan model. Peningkatan performa juga dapat diperhatikan pada nilai *Accuracy* yang berperan mengukur kinerja model dalam memprediksi dengan benar pada *data_training*. Nilai *Accuracy* dihitung sebagai presentasi prediksi benar terhadap total jumlah sampel pelatihan dan menggambarkan seberapa baik model dapat mengklasifikasikan *data_training*. Kemudian nilai *Val_Loss* yang merupakan nilai pengukuran pada *data_validation* yang membantu dalam memantau kinerja model dalam memproses citra gambar yang tidak dikenali selama proses *training*. Tujuannya adalah mencegah *overfitting* dan dapat melakukan generalisasi dengan baik pada data yang baru. *Val_Accuracy* berfungsi untuk memberikan gambaran kinerja model dalam memprediksi data citra yang belum dikenali sebelumnya.

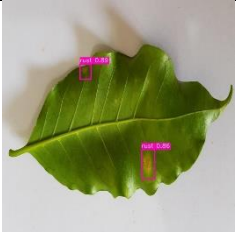
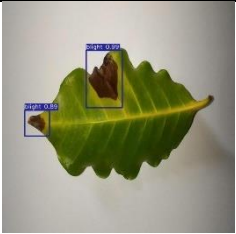


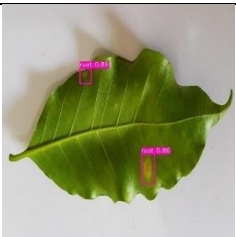
Pelatihan sistem ini menggunakan *optimizer adam* yang mempunyai *learning rate 0,001* menggunakan *initiazer uniform* dan juga *activation relu* untuk mendapatkan nilai terbaik dalam menentukan *Loss*, *Accuracy*, *Val_Loss*, dan *Val_Accuracy*. Pada tabel 4.1 menunjukkan bahwasanya hasil validasi akurasi paling tinggi diperoleh adalah 0.9375 yang terjadi pada *epoch* ke-16.

4.5. Pengujian Sistem


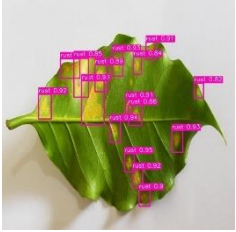

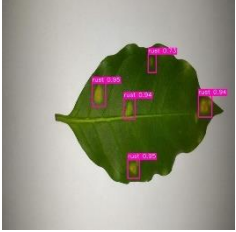
Dalam proses pengujian sistem, *data_testing* digunakan untuk mengevaluasi kinerja sistem. Pengujian dilakukan mulai dari tahapan *pre-processing* dan identifikasi objek dengan algoritma *Faster R-CNN* dengan *output* hasil klasifikasi dari objek yang dideteksi ke dalam kategori penyakit yang ditentukan.

Pengujian sistem dilakukan dengan menggunakan *data_testing* sebanyak 360 citra gambar yang merupakan presentasi 10% dari jumlah total *dataset* awal yang digunakan.

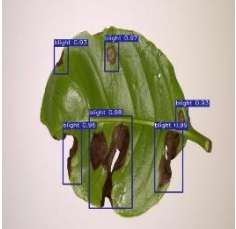
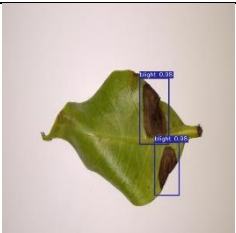

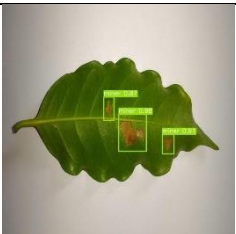
Tabel 4. 3 Hasil Pengujian Sistem

No	Citra	Ukuran Citra	Time Predict	Total Execution Time	Jumlah Objek	Manual	Hasil Sistem	Status
1		29,1 KB	0.159 s	11.09 s	2	Rust	Rust	Benar
2		27,3 KB	0.207 s	7.69 s	2	Blight	Blight	Benar
3		19,0 KB	0.257 s	7.56 s	2	Miner	Miner	Benar
4		28,7 KB	0.224 s	6.30 s	1	Miner	Miner	Benar
5		30,3 KB	0.216 s	18.49 s	2	Rust	Rust	Benar

Tabel 4. 4 Hasil Pengujian Sistem Lanjutan

No	Citra	Ukuran Citra	Time Predict	Total Execution Time	Jumlah Objek	Manual	Hasil Sistem	Status
6		24,0 KB	0.237 s	11.55 s	3	Blight	Blight	Benar
7		25 KB	0.333 s	8.67 s	4	Rust; Miner	Rust; Miner	Benar
8		33 KB	0.245 s	7.82	16	Rust	Rust	Benar
9		30 KB	0.355 s	7.58 s	1	Blight	Blight	Benar
10		24 KB	0.314 s	8.60 s	2	Rust; Miner	Rust; Miner	Benar
11		28 KB	0.256 s	10.93 s	5	Rust	Rust	Benar

Tabel 4. 5 Hasil Pengujian Sistem Lanjutan

No	Citra	Ukuran Citra	Time Predict	Total Execution Time	Jumlah Objek	Manual	Hasil Sistem	Status
12		37 KB	0.222 s	6.17 s	5	Blight	Blight	Benar
13		22 KB	0.220 s	9.96 s	2	Blight	Blight	Benar
14		29 KB	0.302 s	12.26 s	1	Miner	Miner	Benar
15		26 KB	0.225	10.12	3	Miner	Miner	Benar

Tabel 4.3, 4.4, 4.5 merupakan 15 hasil pengujian (*testing*) yang dilakukan untuk menguji kinerja aplikasi. Untuk pengujian lengkap terdapat pada lampiran. Dari 360 citra yang digunakan untuk melakukan pengujian terhadap kinerja aplikasi, terdapat kesalahan klasifikasi oleh sistem sebanyak 18 citra, yaitu 5 citra pada *class* Penggorok Daun (*Leaf Miner*), 7 citra pada *class* Hawar Daun (*Leaf Blight*), dan 6 citra pada *class* Karat Daun (*Leaf Rust*).

Model yang telah melalui proses pelatihan dan pengujian akan memasuki proses metode evaluasi *confussion matrix* untuk dapat melihat hasil kinerja sistem. Adapun hasil dari *confussion matrix* dapat dilihat pada Tabel 4.6.

Tabel 4. 6 Hasil *Confussion Matrix*

		Prediksi			
		<i>Leaf Miner</i>	<i>Leaf Blight</i>	<i>Leaf Rust</i>	Total
Aktual	<i>Leaf Miner</i>	115	5	0	120
	<i>Leaf Blight</i>	4	114	3	120
	<i>Leaf Rust</i>	0	6	116	120
	Total	119	125	119	360

Tabel 4.6 menjabarkan hasil dari uji sistem dalam melakukan klasifikasi *data testing* sebanyak 360 data yang dibagi menjadi 120 citra per *class*. Terdapat beberapa kesalahan prediksi sebanyak 18 citra saat melakukan identifikasi objek pada permukaan daun dikarenakan pada citra gambar tersebut terdapat beberapa faktor yang mempengaruhi, yaitu intensitas cahaya, bayangan dan objek yang memiliki ciri fisik yang mirip. Hal tersebut membuat terjadinya kesalahan dalam proses klasifikasi. Dari data tersebut, dapat ditentukan nilai *True Positive (TP)*, *False Positive*, dan *False Negative (FN)* pada tabel 4.7.

Tabel 4. 7 Nilai TP, FP, dan FN

	TP	FP	FN
<i>Leaf Miner</i>	115	5	4
<i>Leaf Blight</i>	113	7	10
<i>Leaf Rust</i>	114	6	4
Total	342	18	18

Pada tabel 4.7 terdapat 126 data citra yang bernilai *True Positive (TP)* yang merepresentasikan data yang bernilai positif dan diprediksi benar oleh sistem. Kemudian terdapat 14 data bernilai *False True (FP)* yang merepresentasikan data yang bernilai *negative*, namun sistem memprediksi sebagai data positif. Selanjutnya terdapat 18 data yang bernilai *False Negative (FN)* yang merepresentasikan bahwa sistem memprediksi data sebagai nilai yang lain. Dengan nilai-nilai tersebut, dapat dihitung nilai *Precision*, *Recall*, dan *F1-Score* untuk setiap datanya.

a. *Precision*

Rumus perhitungan *Precision* terdapat pada Persamaan 2.2

$$\text{Precision Penggorok Daun (Leaf Miner)} = \frac{115}{120} \times 100\% = 95,83\%$$

$$\text{Precision Hawar Daun (Leaf Blight)} = \frac{113}{120} \times 100\% = 94,71\%$$

$$\text{Precision Karat Daun (Leaf Rust)} = \frac{114}{120} \times 100\% = 95\%$$

b. *Recall*

Rumus perhitungan *Recall* dapat dilihat pada Persamaan 2.3

$$\text{Recall Penggorok Daun (Life Miner)} = \frac{115}{120} \times 100\% = 95,83\%$$

$$\text{Recall Hawar Daun (Leaf Blight)} = \frac{113}{120} \times 100\% = 94,16\%$$

$$\text{Recall Karat Daun (Leaf Rust)} = \frac{114}{120} \times 100\% = 95\%$$

c. *F1-Score*

Rumus perhitungan *F1-Score* dapat dilihat pada Persamaan 2.

$$\text{F1-Score Penggorok Daun (Leaf Minert)} = 2 \times \frac{(0,95 \times 0,95)}{(0,95 + 0,95)} \times 100\% = 95\%$$

$$\text{F1-Score Hawar Daun (Leaf Blight)} = 2 \times \frac{(0,94 \times 0,94)}{(0,94 + 0,94)} \times 100\% = 94\%$$

$$\text{F1-Score Karat Daun (Leaf Rust)} = 2 \times \frac{(0,95 \times 0,95)}{(0,95 + 0,95)} \times 100\% = 96\%$$

Dari perhitungan persamaan untuk mencari nilai *Precision*, *Recall*, dan *F1-Score* di dalam Tabel 4.5.

Tabel 4. 8 Nilai *Precision*, *Recall*, dan *F1-Score*

<i>Class</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Penggorok Daun (<i>Lead Miner</i>)	95,83%	95,83%	95%
Hawar Daun (<i>Leaf Blight</i>)	94%	94%	94%
Karat Daun (<i>Leaf Rust</i>)	95%	95%	95%

Berdasarkan pengujian kinerja sistem yang telah dilakukan, sistem mampu mengklasifikasikan 342 citra dengan benar dari keseluruhan *data test* berjumlah 360 citra yang terdiri dari 115 citra Penggorok Daun (*Leaf Miner*), 113 citra penyakit Hawar Daun (*Leaf Blight*), dan 114 citra penyakit Karat Daun (*Leaf Rust*). Berdasarkan uji

kinerja sistem, maka dapat dihitung nilai akurasi dengan rumus yang terdapat pada Persamaan 4.4.

$$Accuracy = \frac{N_{Benar}}{N} \times 100\%$$
$$Accuracy = \frac{342}{360} \times 100\%$$
$$= 95\%$$

Berdasarkan perhitungan nilai *Accuracy*, maka diperoleh nilai akurasi dari penelitian klasifikasi jenis penyakit daun pada tanaman kopi dengan penerapan metode *Faster R-CNN* ini mencapai sebesar 95%. Terdapat beberapa faktor yang menyebabkan terjadinya kekurangan pada sistem dalam melakukan identifikasi, yaitu adanya pengaruh faktor intensitas cahaya dan kemiripan ciri pada objek perubahan permukaan daun yang disebabkan oleh penyakit yang menyerang.

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang sudah dilakukan, maka dapat ditarik kesimpulan sebagai berikut.

1. Algoritma *Faster R-CNN* dapat mendeteksi dan mengklasifikasikan objek dari perubahan fisik yang disebabkan oleh jenis penyakit daun pada tanaman kopi.
2. Sistem dapat menghasilkan performa yang cukup baik dengan tingkat akurasi sebesar 95%.
3. Ukuran *batch size* dan *epoch* yang tepat dapat mengurangi nilai *loss* dan meningkatkan nilai akurasi sehingga berpengaruh terhadap keberhasilan sistem dalam melakukan klasifikasi.
4. Penerapan *augmentation dataset* memberikan dampak efektif dan efisien untuk meningkatkan nilai akurasi dari model dikarenakan jumlah citra gambar pada *dataset* menjadi lebih variatif.
5. Ukuran citra gambar, kualitas citra gambar dan jumlah objek mempengaruhi waktu yang dibutuhkan untuk melakukan prediksi.
6. Algoritma *Faster R-CNN* membutuhkan infrastruktur *hardware* yang tinggi untuk memberikan hasil yang lebih maksimal.
7. Kekurangan pada sistem dalam melakukan klasifikasi disebabkan oleh faktor tingkat resolusi citra, pengaruh intensitas cahaya, dan kemiripan ciri yang dimiliki oleh bekas timbul dari penyakit yang ada pada permukaan daun tanaman kopi.

5.2. Saran

Adapun beberapa saran dari penulis untuk pengembangan pada penelitian selanjutnya adalah sebagai berikut.

1. Sistem *non-realtime* dibangun untuk mencapai tingkat akurasi yang lebih baik dengan membutuhkan waktu yang lebih dari sistem *real-time* yang berfokus kepada kecepatan dalam memprediksi. Diharapkan untuk penelitian berikutnya,

dapat membangun sebuah sistem yang lebih baik dengan tingkat akurasi tinggi serta waktu yang dibutuhkan lebih singkat dengan alur kerja sistem lebih ringkas guna antisipasi pada perangkat yang dimiliki oleh petani.

2. *Dataset* yang digunakan pada penelitian ini berdasarkan penilaian personal yang tersedia di *platform open source* Kaggle. Dibutuhkan penelitian lebih lanjut yang berkolaborasi dengan pihak ahli dalam bidang penyakit pada tanaman kopi guna pelabelan jenis penyakit pada citra gambar yang lebih tepat. Hal ini dibutuhkan agar sistem dapat memberikan hasil yang lebih memuaskan serta tepat sasaran.
3. Diharapkan pada penelitian selanjutnya dapat membentuk sebuah model yang mampu mendeteksi dan mengklasifikasi berbagai jenis penyakit pada tanaman kopi sehingga mempermudah petani dalam pemantauan dan pengendalian penyebaran penyakit pada perkebunan budidaya tanaman kopi.
4. Diharapkan pada penelitian selanjutnya mampu mencari solusi untuk menangani kekurangan-kekurangan data agar metode *Faster R-CNN* mampu memberikan hasil yang jauh lebih baik lagi.

DAFTAR PUSTAKA

- Alamsyah, D., & Pratama, D. (2019). Deteksi Ujung Jari Menggunakan Faster R-CNN Dengan Arsitektur Inception V2 pada Citra Derau. *JuSiTik : Jurnal Sistem dan Teknologi Informasi Komunikasi*, 2(1). doi:10.32524/jusitik.v2i1.435
- Esgario, J. G., Krohling, R. A., & Ventura, J. (2020, February). Deep learning for classification and severity estimation of coffee leaf biotic stress. *Computers and Electronics in Agriculture*, 169, 105162. doi:10.1016/j.compag.2019.105162
- Filho, O. (2006). Coffe Leaf Miner Resistance. *Brazillian Journal of Plant Physiology*, 110-117. doi:10.1590/S1677-04202006000100009
- Gavrilescu, R. (2018). Faster R-CNN:an Approach to Real-Time Object Detection. Dalam *2018 International Conference and Exposition on Electrical And Power Engineering (EPE)* (hal. 165-168). Romania: IEEE. doi:10.1109/ICEPE.2018.8559776
- I., Golpur; ParianJA, R.A; Chayjan;. (2021). Identification and classification of bulk paddy, brown, and white rice cultivars with colour features extraction using image analysis and neural network. Retrieved from <https://doi.org/10.17221/238/2013-CJFS>
- Irfansyah, D., Mustikasari, M., & Suroso, A. (2021). Arsitektur Convolutional Neural Network (CNN) Alexnet untuk Klasifikasi Hama pada Citra Daun Tanman Kopi. 87-92.
- Irfansyaj, D. e. (2021). Arsitektur Convolutional Neural Network (CNN) AlexNet Untuk Klasifikasi Hama Pada Citra Daun Tanaman Kopi. *Jurnal Informatika : Jurnal Pengembangan IT (JPIT)*, 6, 87-89.
- Karpathy, A. (2017). Convolutional Neural Network for Visual Recognition. *Notes Accompany The Stanford CS Class CS231*. Diambil kembali dari <https://cs231n.github.io/convolutional-networks/>
- Krisnaindra. (t.thn.). *Klasifikasi dan Morfologi Tanaman Kopi dengan Pengertiannya*. Dipetik Agustus 2022, dari teoriemo.com: <https://www.teoriemo.com/2016/10/klasifikasi-dan-morfologi-tanaman-kopi.html>
- Ma, J., Du, K., Zheng, F., Zhang, L., Gong, Z., & Zun , Z. (2018). A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Computer and Electronics in Agriculture*, 154, 18-24. doi:doi.org/10.1016/j.compag.2018.08.048
- Mishra, P. (2019). *yTorch Recipes: A Problem-Solution Approach*. Apress. Diambil kembali dari <https://books.google.co.id/books?id=X5OFDwAAQBAJ>

- Montalbo, F., & Hernandez, A. (2020). An Optimized Classification Model for Coffea Liberica Disease using Deep Convolutional Neural Networks. *2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA)*, 213-218. doi:10.1109/CSPA48992.2020.9068683
- Montalbo, F., & Hernandez, A. (2020). Classifying barako coffee leaf diseases using deep convolutional models. *International Journal of Advances in Intelligent Informatics*, 6, 197-209.
- Nasution, B. (2018). Specialty Kopi Indonesia. Jakarta: Warta Ekspor. Dipetik Agustus 2021
- Nguyen, C., & et al. (2018). Towards Real-Time Smile Detection based on Faster Region Convolutional Neural Network. *1st International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*.
- Odhiambo, D. (2021). Coffe Leaf Disease. Retrieved from <http://www.kaggle.com/badasstechie/coffe-leaf-disease>
- Perkovic, L. (2015). *Introduction to Computing Using Python: An Application Development Focus*. Wiley. Diambil kembali dari <https://books.google.co.id/books?id=rE9FEAAQBAJ>
- Purba, J. (2022). *Purba, Jeremy Dosdo. Aplikasi Berbasis Android untuk Pengklasifikasian Kain Ulos Batak Menggunakan Algoritma Faster R-CNN*. Medan: Universitas Sumatera Utara.
- Purbosari, P., Sasongko, H., Salamah, Z., & Utami, N. (2021). Peningkatan Kesadaran Lingkungan dan Kesehatan Masyarakat Desa Somongari melalui Edukasi Dampak Pupuk dan Pestisida Anorganik. *Agrokreatif Jurnal Ilmiah Pengabdian Kepada Masyarakat*, 7, 2. doi:<https://doi.org/10.29244/agrokreatif.7.2.131-137>
- Putri, F. D., Ramadhani, K. N., & Yunanto, P. E. (2021). *Identifikasi Penyakit Pada Daun Tanaman Apel Menggunakan Local Binary Pattern (LBP) dan Color Histogram*. Bandung.
- Rahardjo, P. (2017). *Berkebun Kopi*. (A. Mu'min., Penyunt.) Jakarta: Penebar Swadaya. Dipetik Oktober 2022, dari <https://books.google.co.id/books?id=Qy0-DwAAQBAJ&lpg=PP1&hl=id&pg=PA20#v=onepage&q&f=false>
- Ren, S., He, K., Girchick, R., & Sun, J. (2017). Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137-1149. doi:10.1109/TPAMI.2016.2577031
- Sabrina, S., & Al Maki, W. (2022). Klasifikasi Penyakit Pada Tanaman Kopi Robusta Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *eProceedings of Engineering*, 9.

- Siska, R. (2018). Serangan Karat Daun Kopi (*Hemila vastatrix* B et Br) pada Tanaman Kopi Arabika di Perkebunan Rakyat Kabupaten Mandailing Ntal Sumatera Utara. *TALENTA Confrence Series : Agriculture & Natural Resources (ANR)*, 1(1). doi:10.32734/anr.v1i1.101
- Soesanto, L. (2020). *Kompendium Penyakit-Penyakit Halaman*. (J. Deviyanti, Penyunt.) Yogyakarta: Penerbit Andi. Diambil kembali dari <https://books.google.co.id/books?id=TgMGEAAAQBAJ>
- Sugiarti, L. (2019). *Identifikasi Hama dan Penyakit Pada Tanaman Kopi di Kebun Percobaan Fakultas Pertanian Universitas Winaya Mukti*.
- Sutarman. (2017). *Dasar-Dasar Ilmu Penyakit Tanaman*. Sidoarjo.
- Sutoyo, T., & Mulyanto, d. (2009). *Teori Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.