

**KLASIFIKASI PENYAKIT KARAT DAUN DAN TUNGAU MERAH
PADA TANAMAN KOPI MENGGUNAKAN METODE
CONVOLUTIONAL NEURAL NETWORK (CNN)
DENGAN ARSITEKTUR *SQUEEZENET***

(Skripsi)

Oleh

**APRILIA ANGGUN SARI RAHMAWATI
NPM 2117051038**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK

KLASIFIKASI PENYAKIT KARAT DAUN DAN TUNGAU MERAH PADA TANAMAN KOPI MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN ARSITEKTUR SQUEEZENET

Oleh

APRILIA ANGGUN SARI RAHMAWATI

Penyakit pada daun kopi seperti karat daun dan serangan tungau merah, merupakan salah satu penyebab penurunan produktivitas kopi di Indonesia. Dampak yang ditimbulkan tidak hanya pada berkurangnya jumlah panen, tetapi juga menurunkan kualitas biji kopi. Oleh karena itu, dibutuhkan metode klasifikasi penyakit yang mampu mengklasifikasikan kondisi daun secara akurat guna mendukung pengelolaan tanaman yang lebih efektif. Penelitian ini mengimplementasikan model klasifikasi otomatis menggunakan *Convolutional Neural Network* (CNN) dengan memanfaatkan arsitektur *SqueezeNet* yang dikenal ringan serta efisien dalam penggunaan sumber daya komputasi. Dataset yang digunakan berjumlah 1.000 citra daun kopi, terbagi menjadi tiga kelas, yaitu daun sehat, daun terinfeksi karat, dan daun terserang tungau merah. Seluruh citra diproses melalui tahapan *preprocessing* dan augmentasi untuk meningkatkan keragaman data latih pada proses pelatihan. Model dilatih dengan pengaturan *hyperparameter* tertentu, seperti jumlah *epoch*, *batch size*, serta *learning rate*. Evaluasi dilakukan dengan data uji terpisah, menghasilkan akurasi klasifikasi sebesar 93,68%. Hasil ini menunjukkan bahwa *SqueezeNet* mampu mengenali kategori daun kopi dengan baik. Dengan keunggulannya yang ringan, model ini berpotensi diterapkan pada perangkat dengan keterbatasan komputasi, sehingga mendukung pemantauan kondisi tanaman kopi secara praktis dan berkelanjutan.

Kata Kunci: *SqueezeNet*, *Deep Learning*, Klasifikasi Penyakit Daun Kopi, *Convolutional Neural Network* (CNN).

ABSTRACT

CLASSIFICATION OF LEAF RUST AND RED SPIDER MITE DISEASES IN COFFEE PLANTS USING THE CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD WITH THE SQUEEZENET ARCHITECTURE

By

APRILIA ANGGUN SARI RAHMAWATI

Coffee leaf diseases, such as leaf rust and red spider mite infestation, are among the causes of declining coffee productivity in Indonesia. The impact is not only reduced harvests but also reduced coffee bean quality. Therefore, a disease classification method capable of accurately classifying leaf conditions is needed to support more effective crop management. This study implemented an automatic classification model using a Convolutional Neural Network (CNN) utilizing the SqueezeNet architecture, known for its lightweight and efficient use of computing resources. The dataset used consisted of 1,000 coffee leaf images, divided into three classes: healthy leaves, rust infected leaves, and red spider mite infested leaves. All images underwent preprocessing and augmentation stages to increase the diversity of the training data. The model was trained with specific hyperparameter settings, such as the number of epochs, batch size, and learning rate. Evaluation was conducted on separate test data, resulting in a classification accuracy of 93.68%. These results indicate that SqueezeNet is capable of recognizing coffee leaf categories well. With its lightweight advantages, this model has the potential to be applied to devices with computing limitations, thus supporting practical and continuous monitoring of coffee plant conditions.

Keywords: SqueezeNet, Deep Learning, Classification Coffee Leaf Disease, Convolutional Neural Network (CNN).

**KLASIFIKASI PENYAKIT KARAT DAUN DAN TUNGAU MERAH
PADA TANAMAN KOPI MENGGUNAKAN METODE
CONVOLUTIONAL NEURAL NETWORK (CNN)
DENGAN ARSITEKTUR *SQUEEZENET***

Oleh

APRILIA ANGGUN SARI RAHMAWATI

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

Judul Skripsi

**KLASIFIKASI PENYAKIT KARAT
DAUN DAN TUNGAU MERAH PADA
TANAMAN KOPI MENGGUNAKAN
METODE CONVOLUTIONAL NEURAL
NETWORK (CNN) DENGAN
ARSITEKTUR SQUEEZENET**

Nama Mahasiswa

Aprilia Anggun Sari Rahmawati

Nomor Pokok Mahasiswa

2117051038

Program Studi

S1 Ilmu Komputer

Fakultas

Matematika dan Ilmu Pengetahuan Alam



1. Komisi Pembimbing

Favorisen R. Lumbanraja, Ph.D.

NIP. 198301102008121002

Yunda Heningtyas, M. Kom.

NIP. 198901082019032014

2. Mengetahui

Ketua Jurusan Ilmu Komputer

Ketua Program Studi S1 Ilmu
Komputer

Dwi Sakethi, S.Si., M.Kom.

NIP. 196806111998021001

Tristiyanto, M.I.S., Ph.D.

NIP. 198104142005011001

MENGESAHKAN

1. Tim Penguji

Ketua

Favorisen R. Lumbanraja, Ph.D.

Sekretaris

Yunda Heningtyas, M.Kom.

Penguji Bukan

Pembimbing

Dr. rer. nat. Akmal Junaidi, S.Si., M.Sc.

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Heri Satria, S.Si., M.Si.

NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: 21 Agustus 2025

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Aprilia Anggun Sari Rahmawati

NPM : 2117051038

Dengan ini menyatakan bahwa skripsi yang berjudul **“Klasifikasi Penyakit Karat Daun dan Tungau Merah Pada Tanaman Kopi Menggunakan Metode *Convolutional Neural Network* (CNN) Dengan Arsitektur *SqueezeNet*”** merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertulis pada skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari ditemukan bukti bahwa skripsi saya merupakan hasil penjiplakan atau hasil karya orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 15 September 2025
Yang menyatakan



Aprilia Anggun Sari Rahmawati
NPM. 2117051038

RIWAYAT HIDUP



Penulis bernama Aprilia Anggun Sari Rahmawati, lahir di Desa Kota Raman pada tanggal 4 April 2003 sebagai anak keempat dari Ibu Suhartin dan Bapak Suyarto. Penulis menyelesaikan pendidikan formal Sekolah Dasar di SD Negeri 2 Kota Raman dan selesai pada Tahun 2015. Dilanjutkan ke jenjang Sekolah Menengah Pertama di MTs Negeri 2 Lampung Timur yang diselesaikan pada Tahun 2018, dan lulus dari pendidikan Sekolah Menengah Atas di SMA Negeri 1 Purbolinggo pada Tahun 2021.

Pada Tahun 2021 penulis terdaftar sebagai mahasiswa jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SNMPTN. Kegiatan-kegiatan yang dilakukan penulis selama menjadi mahasiswa antara lain.

1. Menjadi anggota Bidang Internal Himpunan Mahasiswa Jurusan Ilmu Komputer pada Tahun 2022.
2. Menjadi Asisten Dosen Jurusan Ilmu Komputer pada Mata Kuliah logika pada Tahun 2022.
3. Menjadi Sekretaris Bidang Internal Himpunan Mahasiswa Jurusan Ilmu Komputer pada Tahun 2023.
4. Menjadi Asisten Dosen Jurusan Ilmu Komputer pada Mata Kuliah Rekayasa Perangkat Lunak pada Tahun 2023.
5. Melaksanakan Kerja Praktik di Balai Desa Kota Raman pada Tahun 2024.
6. Melaksanakan Kuliah Kerja Nyata di Desa Braja Fajar, Kecamatan Way Jepara, Kabupaten Lampung Timur pada periode II Tahun 2024.

MOTTO

”Semua orang harus berusaha untuk mengikuti apa yang benar dan bukan apa yang ditetapkan.”

(Aristoteles)

“Rahasia perubahan adalah memusatkan semua energimu bukan untuk melawan yang lama, tetapi untuk membangun yang baru.”

(Socrates)

“Katanya hatiku telah lama terbelah, bagai cangkang kosong terpisah. Ragaku ada di sini, tapi hatiku bersamamu. Bukan maaf yang ku minta, tapi peluk yang ku lupa.”

(Nadin Amizah)

PERSEMBAHAN

Puji dan syukur tercurahkan kepada Allah Subhanahu Wa Ta'ala karena atas limpahan Rahmat, dan Karunia-Nya saya dapat menyelesaikan skripsi ini. Shalawat serta salam selalu tercurahkan kepada Nabi Muhammad Shallallahu Alaihi Wasallam.

Kupersembahkan karya ini kepada:

Kedua Orang Tuaku Tercinta

Yang senantiasa memberikan kasih sayang, pengorbanan, perjuangan, perhatian, dan sujud lama yang ibu lakukan untuk merayu Allah Subhanahu Wa Ta'ala agar melancarkan setiap langkah baikku. Terimakasih juga kepada bapak yang telah memberikan pembelajaran hidup yang sangat luar biasa.

Keluarga Besar Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung

Yang senantiasa memberikan dukungan dan semangat.

SANWACANA

Puji syukur atas kehadiran Allah *Subhanahu Wa Ta'ala*, Karena limpahan nikmat, rahmat, dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul **“Klasifikasi Penyakit Karat Daun dan Tungau Merah Pada Tanaman Kopi Menggunakan Metode *Convolutional Neural Network* (CNN) dengan arsitektur *SqueezeNet*”** dengan baik.

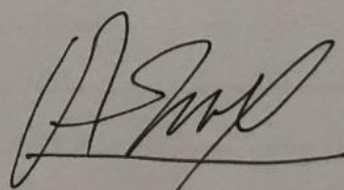
Selesaiannya skripsi ini merupakan hasil dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Untuk itu, pada kesempatan kali ini penulis menyampaikan terimakasih kepada.

1. Ibu yang selalu memberikan doa serta dukungan moral dan material selama perkuliahan.
2. Koko, Mba Prima, Mas Candra yang selalu memberikan dukungan, kasih sayang, dan semangat sehingga penulis dapat menyelesaikan skripsi ini.
3. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., IPM., ASEAN Eng. Selaku Rektor Universitas Lampung.
4. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. Selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
5. Bapak Dwi Sakheti, S.Si., M.Kom. Selaku Ketua Jurusan Ilmu Komputer.
6. Ibu Yunda Heningtyas, M.Kom. Selaku Sekretaris Jurusan Ilmu Komputer sekaligus dosen pembimbing kedua yang senantiasa membantu serta membimbing selama pengerjaan skripsi.
7. Bapak Tristiyanto, S.Kom, M.I.S., Ph.D. Selaku Ketua Program Studi S1 Ilmu Komputer FMIPA Universitas Lampung.
8. Ibu Dewi Asiah Shofiana, S.Komp., M.Kom. Selaku dosen Pembimbing Akademik yang senantiasa memberikan arahan selama masa studi.

9. Bapak Favorisen Rosyking Lumbanraja, S.Kom., M.Si., Ph.D. selaku dosen Pembimbing Utama yang senantiasa membimbing, memberikan arahan, saran, dan masukan selama pengerjaan skripsi.
10. Bapak Dr. rer. nat. Akmal Junaidi, S.Si., M.Sc. selaku dosen Pembahas yang telah memberikan kritik dan saran yang bermanfaat dalam perbaikan skripsi.
11. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu serta pembelajaran selama penulis menempuh pendidikan di Jurusan Ilmu Komputer Universitas Lampung.
12. Staff, dan Karyawan Jurusan Ilmu Komputer yang telah memberi bantuan untuk kelancaran selama penulis menempuh pendidikan di Jurusan Ilmu Komputer Universitas Lampung.
13. Teman seperjuangan semasa kuliah Ika Rahma Alia, Siska Hermayanti, Hobi Ngoding, dan Sahabat Jannah, yang selalu memberikan bantuan serta semangat kepada penulis.
14. Teman-Teman Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom) yang telah memberikan pengalaman berharga.
15. Keluarga besar Ilmu Komputer 2021 yang telah berjuang bersama selama perkuliahan.
16. Almamater Tercinta.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan dan belum sempurna. Meski demikian, penulis berharap karya ini dapat memberikan manfaat bagi seluruh sivitas akademika Universitas Lampung secara umum, serta mahasiswa Ilmu Komputer pada khususnya.

Bandar Lampung, 15 September 2025



Aprilia Anggun Sari Rahmawati
NPM. 2117051038

DAFTAR ISI

	Halaman
DAFTAR GAMBAR	vi
DAFTAR TABEL	vii
DAFTAR KODE PROGRAM.....	viii
I. PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	4
II. TINJAUAN PUSTAKA	5
2.1. Penelitian Terdahulu	5
2.2. Tanaman Kopi	10
2.3. Penyakit Karat Daun.....	12
2.4. Penyakit Tungau Merah.....	13
2.5. <i>Machine Learning</i>	14
2.6. <i>Deep Learning</i>	16
2.7. <i>Convolutional Neural Network</i> (CNN).....	16
2.8. <i>Python</i>	17
2.9. <i>Preprocessing</i>	18
2.10. Augmentasi Data	22
2.11. <i>SqueezeNet</i>	23
2.12. <i>Library</i>	27
2.13. <i>Hyperparameter</i>	30
2.14. <i>Data Train</i>	34
2.15. <i>Data Validation</i>	35

2.16. <i>Data Test</i>	35
2.17. <i>Confusion Matrix</i>	35
2.18. <i>Overfitting</i>	38
2.19. <i>Underfitting</i>	38
2.20. <i>Roboflow</i>	39
III. METODOLOGI PENELITIAN	40
3.1. Tempat dan Waktu	40
3.2. Data dan Alat	40
3.3. Metodologi	46
IV. HASIL DAN PEMBAHASAN	59
4.1. <i>Preprocessing</i>	59
4.2. <i>Splitting Data</i>	68
4.3. Augmentasi Data	70
4.4. <i>Modeling</i>	72
4.5. <i>Testing Model</i>	75
4.6. Evaluasi Model	78
V. SIMPULAN DAN SARAN	81
5.1. Simpulan	81
5.2. Saran	81
DAFTAR PUSTAKA	83

DAFTAR GAMBAR

Gambar	Halaman
1. Tanaman Kopi.....	10
2. Penyakit Karat Daun.	12
3. Bercak Kuning.	13
4. Bercak Cokelat.	13
5. Penyakit Tungau Merah.	14
6. Struktur <i>Convolutional Neural Network</i> (CNN).....	17
7. Arsitektur <i>SqueezeNet</i>	24
8. Ilustrasi Proses Pada <i>Pooling Layer</i>	25
9. Ilustrasi <i>Neuron</i> Sebelum Terdapat <i>DropOut</i>	33
10. Ilustrasi <i>Neuron</i> Setelah Terdapat <i>DropOut</i>	33
11. Ilustrasi <i>Overfitting</i>	38
12. Ilustrasi <i>Underfitting</i>	39
13. Tahapan Penelitian Klasifikasi Penyakit Daun Kopi.	47
14. Notifikasi Duplikasi Gambar pada <i>Roboflow</i>	60
15. Grafik Hasil <i>Training Model</i>	75
16. Gambar Yang Diprediksi Salah Oleh Model.	76
17. <i>Confusion Matrix</i>	80

DAFTAR TABEL

Tabel	Halaman
1. Penelitian Terdahulu	5
2. <i>Confusion Matrix</i>	36
3. Alur Kegiatan Penelitian Klasifikasi Penyakit Daun Kopi	41
4. Jumlah Dan Persentase Data Masing-Masing Kelas.....	42
5. Sampel Data Masing-Masing Kelas.....	47
6. Lapisan Model dengan Arsitektur <i>SqueezeNet</i>	54
7. Perbandingan data sebelum dan sesudah <i>cleaning</i>	61
8. Contoh Anotasi Masing-Masing Kelas	62
9. Hasil <i>Cropping</i>	64
10. Hasil Segmentasi.....	66
11. Hasil <i>Padding</i>	67
12. Contoh Gambar Sebelum dan Sesudah <i>resize</i>	68
13. Persebaran Data Per Kelas Masing-Masing Subset	69
14. Kebutuhan Penambahan Data Masing-Masing Kelas.....	70
15. Contoh Hasil Penerapan Teknik Augmentasi	71
16. Penyesuaian Layer pada Bagian <i>Classifier</i>	73
17. <i>Hyperparameter</i> Yang Digunakan.....	74
18. Analisis <i>Misclassification</i>	77
19. <i>Classification Report</i>	79

DAFTAR KODE PROGRAM

Kode Progam	Halaman
1. Pengecekan Duplikasi Gambar.	60
2. Memindahkan <i>Polygon</i> Objek Utama Daun Kopi	65
3. <i>Splitting</i> Data <i>Train</i> dan Data Sementara.....	69
4. <i>Splitting</i> Data <i>Validation</i> dan Data <i>Test</i>	69
5. <i>Load Model Squeezenet</i> Versi 1.1.....	72

I. PENDAHULUAN

1.1. Latar Belakang

Kopi merupakan salah satu komoditas penting di Indonesia yang memiliki peran strategis dalam mendukung perekonomian masyarakat. Namun seiring berjalanya waktu, produksi kopi di Indonesia mengalami penurunan. Saat ini, produktivitas rata-rata kopi hanya mencapai sekitar 600 kg/ha, padahal potensi produksinya bisa mencapai 1000 kg/ha (Widiyani & Hartono, 2021). Penyebab terjadinya penurunan produktivitas kopi salah satunya adalah adanya penyakit yang menyerang daun kopi sehingga dapat mengganggu proses fotosintesis dan menurunkan kualitas kopi tersebut. Contoh penyakit daun yang dapat menyerang tanaman kopi adalah karat daun (*leaf rust*). Penyakit karat daun dicirikan dengan bercak kuning yang berubah jingga dan bertepung di bawah daun, menyebabkan daun menguning dan gugur. Penyebaran melalui spora jamur yang tumbuh dalam 1-3 jam dibantu air, suhu hangat, dan angin (Caron & Markusen, 2023). Selain karat daun (*leaf rust*) yang disebabkan jamur, daun kopi juga rentan terhadap serangan tungau merah (*Red Spider Mite*). Serangan tungau merah ditandai dengan *nekrosis* (jaringan mati) di sekitar tulang daun yang menyebar ke seluruh permukaan. Pada tingkat serangan tinggi, terjadi defoliasi (kerontokan daun) hingga kematian tanaman (Santoso & Astuti, 2019).

Perkembangan teknologi *deep learning* telah mengalami kemajuan pesat dan memberikan dampak signifikan di berbagai bidang, termasuk pertanian. *Convolutional Neural Network* (CNN) merupakan salah satu algoritma *deep learning* yang memiliki performa baik dalam analisis citra.

CNN telah menjadi metode yang baik untuk mengklasifikasikan gambar. CNN terinspirasi dari cara kerja otak manusia dalam memproses informasi visual, sehingga mampu mengenali pola-pola yang rumit dalam gambar dengan sangat baik (Nurdin *et al.*, 2024). *SqueezeNet* adalah arsitektur CNN yang dirancang dengan ukuran model lebih kecil dibanding *AlexNet* atau *VGG*, namun tetap efisien dalam pengolahan citra. *SqueezeNet* menawarkan arsitektur efisien dengan parameter minimal namun tetap memberikan akurasi tinggi dalam klasifikasi gambar, membuatnya ideal untuk perangkat dengan sumber daya terbatas (Fauzan *et al.*, 2023). *SqueezeNet* diharapkan dapat digunakan untuk mengklasifikasikan penyakit daun kopi secara akurat.

Identifikasi penyakit tanaman termasuk daun kopi telah banyak dilakukan. Windiawan & Suharso (2019) menggunakan arsitektur *VGG16* untuk identifikasi penyakit *Rust* dan *Spider Mite* pada daun kopi menggunakan 360 gambar. Tingkat akurasi *VGG16* mencapai 89% untuk data *train* dan 95% untuk data *validation*. Supriyanto *et al.* (2023) juga mengklasifikasikan penyakit daun Kopi Robusta. Dataset primer yang digunakan sebanyak 400 citra yang diekstraksi menggunakan GLCM menunjukkan akurasi sebesar 97,50% dengan *kernel Radial Basis Function (RBF) Gaussian*. Penelitian lain yang dilakukan oleh Theerthagiri *et al.* (2024) berhasil menunjukkan keunggulan *SqueezeNet* dalam mengklasifikasikan penyakit daun jagung. Pada penelitian tersebut *SqueezeNet* berhasil mencapai akurasi tertinggi sebesar 97% dibandingkan dengan arsitektur lain seperti *VGG16*, *ResNet34*, dan *ResNet50*.

Klasifikasi penyakit daun kopi sangat krusial untuk menjaga kesehatan tanaman dan hasil panen. Penyakit seperti karat daun dan serangan tungau merah dapat mengurangi fotosintesis, mengakibatkan kerugian ekonomi yang signifikan. Dengan metode klasifikasi yang efektif, petani dapat mengambil tindakan pencegahan, melindungi daun yang sehat dari kerusakan lebih lanjut dan mendukung keberlanjutan produksi kopi.

Arsitektur *SqueezeNet* diharapkan dapat memberikan performa yang lebih baik dalam mengklasifikasikan penyakit karat daun dan tungau merah pada tanaman kopi, dengan efisiensi memori yang lebih baik tanpa mengorbankan akurasi.

1.2. Rumusan Masalah

Rumusan masalah penelitian ini terletak pada efektivitas *Convolutional Neural Network* (CNN) dengan arsitektur *SqueezeNet* dalam mengklasifikasikan daun kopi yang terinfeksi penyakit karat daun, tungau merah dan yang sehat.

1.3. Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut.

- 1.3.1. Penelitian ini menggunakan dataset dari *Kaggle* yang berjumlah 1000 gambar dengan 3 kelas yaitu *healthy*, *rust*, dan *red spider mite*.
- 1.3.2. Penelitian ini dibatasi pada pembuatan model tanpa adanya pengembangan sistem lebih lanjut.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah.

- 1.4.1. Mengimplementasikan metode *Convolutional Neural Network* (CNN) dengan arsitektur *SqueezeNet* untuk mengklasifikasikan daun sehat, daun yang terinfeksi penyakit karat daun, dan daun yang terinfeksi penyakit tungau merah pada tanaman kopi.
- 1.4.2. Mengetahui performa metode CNN dengan arsitektur *SqueezeNet* dalam mengklasifikasikan daun sehat, penyakit karat daun, dan tungau merah pada tanaman kopi.

1.5. Manfaat Penelitian

Adapun manfaat dari penelitian ini yaitu.

- 1.5.1. Meningkatkan pemahaman pembaca tentang klasifikasi daun sehat, penyakit karat daun, dan tungau merah pada tanaman kopi menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur *SqueezeNet*.
- 1.5.2. Memberikan kontribusi pada pengembangan sistem klasifikasi penyakit tanaman berbasis kecerdasan buatan.

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian terdahulu merupakan tinjauan sistematis terhadap studi atau publikasi ilmiah yang relevan dengan topik penelitian ini. Penelitian terdahulu yang berkaitan disajikan pada Tabel 1.

Tabel 1. Penelitian Terdahulu

No	Penelitian	Data	Metode	Hasil
1.	Identifikasi Penyakit pada Daun Kopi Menggunakan Metode <i>Deep Learning</i> VGG16 (Windiawan & Suharso, 2019)	<i>Kaggle</i> : 1539 gambar, 3 kategori yaitu gambar gambar sehat, penyakit <i>red spider mite</i> , dan penyakit <i>rust</i> .	<i>Preprocessing</i> • <i>Data Cleaning</i> • Augmentasi Data <i>Splitting Data</i> • 90% <i>training</i>, 10% <i>validation</i> • 80% <i>training</i>, 20% <i>validation</i> • 70% <i>training</i>, 30% <i>validation</i> • 60% <i>training</i>, 40% <i>validation</i> <i>Modelling</i> • VGG16 Evaluasi • Metrik akurasi	Data 90:10: • Akurasi data <i>train</i>: 83% • Akurasi data <i>validation</i>: 89% Data 80:20 • Akurasi data <i>train</i>: 87% • Akurasi data <i>validation</i>: 94% Data 70:30 • Akurasi data <i>train</i>: 88% • Akurasi data <i>validation</i>: 92% Data 60:40 • Akurasi data <i>train</i>: 89% • Akurasi data <i>validation</i>: 95%

Tabel 1. (lanjutan)

No	Penelitian	Data	Metode	Hasil
2.	Klasifikasi Penyakit Daun Kopi Robusta Menggunakan Metode SVM dengan Ekstraksi Ciri GLCM (Supriyanto <i>et al.</i> , 2023)	400 data citra, diambil dari kebun Desa Plaosan, Cluwak, Pati, Jawa Timur	<i>Preprocessing</i> • Peningkatan nilai intensitas (kontras) • Segmentasi <i>K-Means Clustering</i> • Ekstraksi ciri <i>Gray Level Co-occurrence Matrix (GLCM)</i> <i>Modelling</i> • <i>Support Vector Machine (SVM)</i> dengan <i>Kernel Linear</i>, <i>Kernel Polinomial</i>, dan <i>Kernel Radial Basis Function (RBF) Gaussian</i> Evaluasi • <i>Confusion Matrix</i>	<i>Kernel Linear</i> • Akurasi = 70,00% • Presisi = 66,6% • <i>Recall</i> = 80,00% • <i>F1-Score</i> = 72,73% <i>Kernel Polinomial</i> • Akurasi = 86,25% • Presisi = 96,77% • <i>Recall</i> = 75,00% • <i>F1-Score</i> = 84,51% <i>Kernel Radial Basis Function (RBF) Gaussian</i> • Akurasi = 97,50% • Presisi = 95,24% • <i>Recall</i> = 100,00% • <i>F1-Score</i> = 97,56%
3.	<i>Deep SqueezeNet learning model for diagnosis and prediction of maize leaf diseases</i> (Theerthagiri <i>et al.</i> , 2024)	<i>Kaggle:</i> 4188 gambar dengan 4 kelas yaitu sehat, common rust, grey leaf spot, dan blight	<i>Preprocessing</i> • Augmentasi Data • <i>Splitting data</i> • <i>Resizing</i> • SMOTE <i>Modeling</i> • CNN Arsitektur • <i>SqueezeNet</i> • VGG16 • ResNet34 • ResNet50	<i>SqueezeNet</i> • Accuracy = 97% • Precision - <i>Blight</i> = 0,95 - <i>Common Rust</i> = 0,97 - <i>Gray Leaf Spot</i> = 0,96 - <i>Healthy</i> = 1,00 • Recall - <i>Blight</i> = 0,95 - <i>Common Rust</i> = 0,98 - <i>Gray Leaf Spot</i> = 0,93 - <i>Healthy</i> = 1,00 • <i>F1-Score</i> - <i>Blight</i> = 0,95 - <i>Common Rust</i> = 0,98 - <i>Gray Leaf Spot</i> = 0,94 - <i>Healthy</i> = 1,00

Tabel 1. (lanjutan)

No	Penelitian	Data	Metode	Hasil
			Evaluasi	VGG16
			• <i>Confusion Matrix</i>	• Accuracy = 92% • Precision - <i>Blight</i> = 0,90 - <i>Common Rust</i> = 0,98 - <i>Gray Leaf Spot</i> = 0,69 - <i>Healthy</i> = 1,00 • Recall - <i>Blight</i> = 0,84 - <i>Common Rust</i> = 0,97 - <i>Gray Leaf Spot</i> = 0,69 - <i>Healthy</i> = 1,00 • Recall - <i>Blight</i> = 0,84 - <i>Common Rust</i> = 0,97 - <i>Gray Leaf Spot</i> = 0,81 - <i>Healthy</i> = 1,00 • F1-Score - <i>Blight</i> = 0,87 - <i>Common Rust</i> = 0,98 - <i>Gray Leaf Spot</i> = 0,75 - <i>Healthy</i> = 1,00
				ResNet34
				• Accuracy = 95% • Precision - <i>Blight</i> = 0,94 - <i>Common Rust</i> = 0,97 - <i>Gray Leaf Spot</i> = 0,84 - <i>Healthy</i> = 1,00 • Recall - <i>Blight</i> = 0,90 - <i>Common Rust</i> = 0,98 - <i>Gray Leaf Spot</i> = 0,91 - <i>Healthy</i> = 1,00 • F1-Score - <i>Blight</i> = 0,92 - <i>Common Rust</i> = 0,98 - <i>Gray Leaf Spot</i> = 0,88 - <i>Healthy</i> = 1,00
				ResNet50
				• Accuracy = 94% • Precision - <i>Blight</i> = 0,89 - <i>Common Rust</i> = 0,98 - <i>Gray Leaf Spot</i> = 0,85 - <i>Healthy</i> = 1,00 • Recall - <i>Blight</i> = 0,93 - <i>Common Rust</i> = 0,97 - <i>Gray Leaf Spot</i> = 0,81 - <i>Healthy</i> = 1,00

Tabel 1. (lanjutan)

No	Penelitian	Data	Metode	Hasil
				• <i>F1-Score</i> - <i>Blight</i> = 0,91 - <i>Common Rust</i> = 0,98 - <i>Gray Leaf Spot</i> = 0,83 - <i>Healthy</i> = 1,00

Windiawan & Suharso (2019) menggunakan dataset yang berasal dari *Kaggle*. Dataset awal terdiri dari 1.539 gambar yang terbagi dalam tiga kategori, yaitu daun kopi sehat, daun kopi yang terinfeksi penyakit *Red Spider Mite*, dan daun kopi yang terinfeksi penyakit *Rust*. Setelah melalui proses *data cleaning* untuk menghilangkan gambar yang kurang jelas dan menyeimbangkan jumlah data pada setiap kategori, dataset yang digunakan menjadi 360 gambar, dengan 120 gambar untuk setiap kategori. Tahapan *preprocessing* meliputi *data preparation*, *data cleaning*, *splitting data* menjadi data *train* dan data *validation* dengan beberapa perbandingan yaitu 90:10, 80:20, 70:30, 60:40, serta *data augmentation* untuk meningkatkan variasi data. Pemodelan dilakukan menggunakan arsitektur *deep learning* VGG16 dengan 20 epoch dan ukuran *input layer* 150×150 . Evaluasi model dilakukan dengan metrik akurasi (*accuracy*) untuk menemukan kombinasi model terbaik. Penelitian ini memperoleh akurasi terbesar pada *split ratio* 60:40 dengan akurasi data *train* sebesar 89% dan akurasi data *validation* sebesar 95%.

Supriyanto *et al.* (2023) mengklasifikasikan penyakit daun Kopi Robusta yaitu bercak daun (*Cercospora coffeicola*) dan karat daun (*Hemileia vastatrix*). Dataset yang digunakan berupa citra daun Kopi Robusta yang diperoleh dari perkebunan di Desa Plaosan, Kecamatan Cluwak, Kabupaten Pati, Jawa Tengah. Dataset ini terdiri dari 320 citra sebagai data latih dan 80 citra sebagai data uji. Penelitian ini melibatkan beberapa tahapan seperti *preprocessing* yaitu dengan cara peregangan kontras untuk meningkatkan kualitas citra agar lebih mudah dianalisis, segmentasi yang dilakukan dengan metode *K-Means Clustering* yaitu membagi citra

menjadi tiga *cluster* untuk memisahkan area yang terpengaruh penyakit dan area yang sehat, ekstraksi fitur menggunakan *Gray Level Co-occurrence Matrix* (GLCM) untuk mendapatkan fitur tekstur *Angular Second Moment* (ASM/Energi), kontras, korelasi, *Inverse Difference Moment* (IDM/Homogenitas) dan entropi. Fitur yang diekstraksi dihitung berdasarkan sudut 0° , 45° , 90° , dan 135° dengan jarak antar piksel 1 hingga 3 piksel. Setelah tahap *preprocessing* selanjutnya adalah tahap *modeling* dimana pada tahap ini dilakukan pengklasifikasian menggunakan metode *Support Vector Machine* (SVM) dengan tiga *kernel* yaitu linear, polinomial, dan *Radial Basis Function* (RBF) *Gaussian*. Evaluasi kinerja dilakukan menggunakan *confusion matrix* dengan metric akurasi, presisi, recall, dan *F1-Score*. Pada penelitian ini performa terbaik diraih oleh *kernel Radial Basis Function* (RBF) *Gaussian* dengan akurasi sebesar 97,50%, presisi 95,24%, *recall* 100,00%, dan *F1-Score* 97,56%.

Theerthagiri *et al.* (2024) memprediksi serta mengklasifikasikan penyakit daun jagung dengan menggunakan dataset yang diambil dari repositori *Kaggle*, terdiri dari total 4.188 gambar yang terbagi ke dalam empat kelas yaitu *common rust*, *grey leaf spot*, *blight*, dan sehat. Dalam penelitian ini, metode *preprocessing* dilakukan dengan menggunakan teknik SMOTE untuk menyeimbangkan jumlah gambar antar kelas, serta augmentasi gambar yang mencakup rotasi dan pemotongan guna memperkaya dataset yang ada. Metode *modeling* yang digunakan meliputi beberapa arsitektur *neural network* seperti *SqueezeNet*, VGG16, *ResNet34*, dan *ResNet50*. Hasil dari penelitian ini menunjukkan bahwa *SqueezeNet* mencapai akurasi tertinggi sebesar 97% dalam mengklasifikasikan penyakit daun jagung, mengungguli model-model lainnya dengan peningkatan akurasi berkisar antara 2% hingga 5%. Selain itu, *SqueezeNet* juga berhasil mengurangi *Mean Square Error* (MSE) antara 4% hingga 11% dibandingkan dengan arsitektur VGG16, *ResNet34*, dan *ResNet59*. Hasil ini menyoroti potensi penggunaan *SqueezeNet* untuk meningkatkan produktivitas pertanian dan manajemen penyakit tanaman.

2.2. Tanaman Kopi

Indonesia memiliki luas 1.266.848 hektar kebun kopi yang terdiri dari Perkebunan Besar Negara, Perkebunan Besar Swasta, dan Perkebunan Rakyat (Badan Pusat Statistik, 2023). Menurut *International Coffee Organization* (2025), produksi kopi dunia mencapai jutaan karung dan Indonesia berada pada urutan ketiga setelah Brazil dan India negara yang mempengaruhi industri kopi global. Sehingga menjaga tanaman kopi dari serangan penyakit merupakan hal yang sangat penting untuk terus meningkatkan produktivitas kopi. Tanaman kopi dapat dilihat pada Gambar 1.



Gambar 1. Tanaman Kopi (Dinas Pertanian dan Pangan Kabupaten Bandung, 2018).

Menurut Anggraini (2024), ada tiga jenis kopi yang dibudidayakan yaitu

2.2.1. Kopi Arabika

Kopi Arabika merupakan salah satu varietas kopi unggulan di Indonesia setelah Kopi Robusta, dikenal luas karena mutu dan cita rasanya yang tinggi sehingga banyak diminati hingga ke berbagai negara. Jenis kopi ini memiliki karakteristik rasa yang lebih kompleks, kaya, dan bervariasi dibandingkan dengan kopi Robusta maupun Liberika. Keasaman pada kopi Arabika cenderung lebih

dominan, memberikan sensasi rasa yang lebih segar dan unik, sementara kandungan kafeinnya lebih rendah, yaitu berkisar antara 0,9 – 1,8%, sehingga lebih ringan dan tidak terlalu menimbulkan efek rangsangan berlebih bagi penikmatnya (Anggraini, 2024).

2.2.2. Kopi Robusta

Kopi Robusta adalah varietas kopi yang paling luas dibudidayakan di Indonesia karena dikenal memiliki daya adaptasi yang baik di berbagai kondisi lingkungan. Tanaman ini relatif lebih mudah dalam hal perawatan dan memiliki ketahanan yang lebih kuat terhadap serangan hama serta penyakit bila dibandingkan dengan kopi Arabika, sehingga lebih efisien bagi para petani. Dari segi kandungan, kopi Robusta memiliki kadar kafein paling tinggi di antara dua varietas kopi lainnya, yakni sekitar 1,6 – 2,5% yang menjadikannya lebih pahit dan memberikan efek rangsangan yang lebih kuat. Selain itu, tingkat keasaman (pH) kopi Robusta juga lebih tinggi, yaitu sekitar 4,85 – 5,15, sehingga mempengaruhi rasa dan sensasi asam yang dihasilkan (Anggraini, 2024).

2.2.3. Kopi Liberika

Kopi Liberika dikenal sebagai varietas kopi yang memiliki ukuran buah paling besar dibandingkan varietas lainnya, dengan panjang biji berkisar antara 0,83 – 1,10cm dan lebar sekitar 0,61cm. meski berukuran besar, rendemen biji kopi Liberika tergolong rendah, rata-rata hanya sekitar 9,03%, sehingga jumlah biji yang dihasilkan dari proses pengolahan relatif sedikit. Kopi ini sering disebut sebagai kopi nangka karena mempunyai aroma khas menyerupai buah nangka yang cukup unik dan mudah dikenali. Dari segi kandungan, kafein pada kopi Liberika cenderung rendah, yaitu hanya sekitar 1,1 – 1,3%, menjadikannya relatif aman dikonsumsi

oleh individu yang sensitif terhadap kafein. Selain itu, kopi Liberika memiliki potensi ekonomi yang menjanjikan berkat cita rasanya yang khas, tidak sepahit kopi Robusta, menghadirkan perpaduan aroma nangka dengan sedikit rasa asam mirip Arabika, dan sentuhan coklat yang memberi karakter unik tersendiri (Anggraini, 2024).

2.3. Penyakit Karat Daun

Salah satu penyakit yang kerap menyerang tanaman kopi adalah karat daun (*rust*). Karat daun disebabkan oleh *Hemileia vastatrix*. *Hemileia vastatrix* merupakan jamur parasit obligat yang artinya jamur ini tidak dapat bertahan hidup di luar jaringan inangnya yang bersifat biotik. Oleh karena itu, penyebaran penyakit karat daun ini hanya dapat terjadi pada tanaman inang di sekitarnya. Dampak dari penyakit ini dapat menyebabkan kerugian produksi kopi mencapai 20-70% (Mutmainah *et al.*, 2023). Penyakit karat daun dapat dilihat pada Gambar 2.



Gambar 2. Penyakit Karat Daun (Kaggle, 2020).

Gejala penyakit karat daun awalnya daun dipenuhi bercak klorosis dengan ukuran bervariasi yang berubah warna dari kuning muda menjadi kuning tua. Pada bagian bawah daun, tampak tepung berwarna oranye yang merupakan *uredospora* jamur. Dalam perkembangan selanjutnya, bercak coklat mulai muncul dan saling bergabung, membesar, dan menyebabkan daun mengalami kekeringan (*nekrosis*). Tahap akhir dari serangan

penyakit ini adalah gugurnya daun, yang menandakan kerusakan yang signifikan pada tanaman (Harni *et al.*, 2015).



Gambar 3. Bercak Kuning (Kaggle, 2020).



Gambar 4. Bercak Cokelat (Kaggle, 2020).

2.4. Penyakit Tungau Merah

Tungau merah disebabkan oleh hama tungau *tetranychus sp.* Hama ini menyerang daun tanaman, baik daun muda maupun relatif tua, dengan cara menghisap cairan pada jaringan epidermis. Akibatnya, muncul bercak-bercak putih yang semakin bertambah, yang berangsur mengubah warna daun menjadi kuning. Serangan tungau juga menyebabkan daun berlekuk tidak teratur dan pada tingkat serangan berat dapat mengakibatkan kerontokan daun (Santoso *et al.*, 2014). Penyakit tungau merah dapat dilihat pada Gambar 5.



Gambar 5. Penyakit Tungau Merah (Kaggle, 2020).

2.5. *Machine Learning*

Machine Learning atau pembelajaran mesin merupakan bagian *Artificial Intelligence* (AI) yang memungkinkan sistem komputer belajar dan menyelesaikan masalah secara mandiri. Teknologi ini dirancang untuk mengotomatisasi tugas-tugas yang biasanya dilakukan manusia, dengan kemampuan belajar dan beradaptasi tanpa perlu pemrograman eksplisit. Sistem *Machine Learning* beroperasi berdasarkan analisis data historis untuk mengidentifikasi pola dan mengambil keputusan dengan intervensi minimal dari manusia. Kapasitas pembelajaran sistem ini bersifat dinamis dan terus berkembang seiring berjalannya waktu (Wijoyo *et al.*, 2024).

Ada tiga jenis utama *Machine Learning* seperti yang dijelaskan pada penelitian yang dilakukan oleh (Roihan *et al.*, 2020) yaitu.

2.5.1. *Supervised Learning*

Dalam *Supervised Learning*, salah satu teknik yang diterapkan adalah metode klasifikasi, yaitu proses di mana model dilatih menggunakan data yang telah diberi label secara lengkap. Setiap sampel data dalam kumpulan pelatihan memiliki label atau kategori yang diketahui, yang digunakan untuk membimbing model dalam mempelajari pola dan karakteristik yang membedakan setiap kelas. Tujuan dari pendekatan ini adalah agar model mampu memprediksi label dari data baru yang belum pernah dilihat sebelumnya, dengan mengidentifikasi pola serupa

berdasarkan pelatihan sebelumnya. Teknik ini sangat efektif ketika tersedia data yang cukup dan teranotasi dengan baik untuk setiap kelas yang ingin dikenali (Roihan *et al.*, 2020).

2.5.2. *Unsupervised Learning*

Unsupervised Learning sering kali dikenal dengan istilah *clustering* karena pendekatan ini tidak memerlukan label atau anotasi pada kumpulan data yang digunakan dalam pelatihan. Berbeda dengan *supervised learning*, metode ini bekerja tanpa mengetahui kategori atau kelas dari data yang dianalisis. Tujuan utama dari *unsupervised learning* adalah untuk menemukan pola tersembunyi, struktur internal, atau kemiripan antar data secara otomatis. Model mencoba mengelompokkan data ke dalam beberapa grup atau klaster berdasarkan karakteristik atau fitur yang serupa, tanpa panduan eksplisit mengenai label atau kelas. Oleh karena itu, hasil dari proses ini bukan berupa identifikasi terhadap kelas-kelas tertentu yang telah ditentukan sebelumnya, melainkan berupa pembentukan klaster yang merepresentasikan struktur alami dari data tersebut (Roihan *et al.*, 2020).

2.5.3. *Reinforcement Learning*

Reinforcement Learning adalah metode *machine learning* di mana suatu model berinteraksi secara langsung dengan lingkungan yang bersifat dinamis dan terus berubah. Dalam proses ini, model tidak diberikan instruksi atau label eksplisit mengenai tindakan mana yang benar atau salah. Sebaliknya, model belajar melalui pengalaman dengan mencoba berbagai aksi dan mengamati konsekuensinya. Model akan menerima umpan balik dalam bentuk *reward* atau *punishment* berdasarkan tindakan yang dilakukan, yang kemudian digunakan untuk memperbaiki strategi

pengambilan keputusan di masa mendatang. Meskipun tidak ada pemberitahuan langsung dari komputer mengenai apakah tujuan akhirnya telah tercapai, model secara bertahap belajar untuk memilih tindakan yang memaksimalkan total *reward* jangka panjang. Pendekatan ini sangat cocok untuk permasalahan yang melibatkan pengambilan keputusan secara bertahap dalam kondisi yang kompleks dan tidak sepenuhnya diketahui sebelumnya (Roihan *et al.*, 2020).

2.6. *Deep Learning*

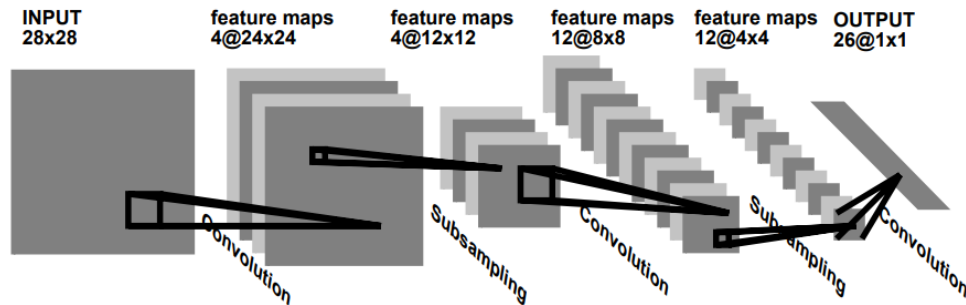
Deep Learning adalah proses pembelajaran yang menggunakan algoritma yang berfungsi seperti otak manusia untuk memprediksi peluang dan peristiwa serta mengenali objek. Pengolahan gambar digital adalah bidang dimana *deep learning* dapat diterapkan. Sistem pengolahan gambar bertujuan untuk membantu manusia mengenali atau mengklasifikasikan objek secara cepat dan akurat. *Deep learning* juga memiliki kemampuan untuk memproses sejumlah besar data sekaligus (Wijayanto *et al.*, 2024).

Deep Learning memberikan hasil *computer vision* terbaik, sehingga baru-baru ini menjadi fokus pengembangan *Machine Learning*. *Deep Learning* adalah teknik pembelajaran otomatis yang digunakan mesin untuk memahami dan mengklasifikasikan benda. Pengenalan objek dan citra (*face recognition*, *object detection*, *image recognition*, *medical image classification* dan lain sebagainya) adalah fokus utama dari *Deep Learning* (Husna *et al.*, 2022).

2.7. *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) memiliki mekanisme kerja yang mirip dengan *Artificial Neural Network (ANN)* yang meniru sistem kerja *neuron* manusia. Namun, pada CNN setiap *neuron* direpresentasikan dalam bentuk dua dimensi sehingga prosesnya lebih efisien. Dalam CNN,

operasi linier dilakukan melalui operasi konvolusi, di mana bobotnya terdiri dari empat dimensi, bukan satu dimensi yang merupakan kumpulan *kernel* konvolusi (Rasidi *et al.*, 2022). Struktur *Convolutional Neural Network* (CNN) dapat dilihat pada Gambar 6.



Gambar 6. Struktur *Convolutional Neural Network* (CNN)
(Yann & Yoshua, 1995).

Cara kerja *Convolutional Neural Network* (CNN) mirip dengan *Neural Network Standart*, namun perbedaannya terletak pada penggunaan *kernel* dua dimensi atau dimensi tinggi di setiap unit pada lapisan CNN yang akan menjalani proses dengan media *input*, CNN memanfaatkan *kernel* yang ada di dalamnya. Selain itu, untuk mengurangi jumlah variabel agar proses pembelajaran menjadi lebih sederhana CNN menggunakan beberapa parameter. CNN juga dilatih untuk mengenali fitur dari objek agar dapat melakukan prediksi (Jinan *et al.*, 2022).

2.8. Python

Python adalah bahasa pemrograman *open-source* yang bersifat tingkat tinggi dan diinterpretasikan sehingga mempermudah pengembangan program berorientasi objek melalui pendekatan yang sederhana. Bahasa ini sangat populer di kalangan ilmuwan data karena kemampuannya yang kuat dalam menangani perhitungan matematika, statistik, serta fungsi-fungsi ilmiah (Raihan & Yulianto, 2023).

Python merupakan salah satu bahasa pemrograman yang paling diminati oleh industri secara global. Sekitar 30% industri membutuhkan *programmer* yang menguasai bahasa pemrograman *python*. Kepopuleran *python* di seluruh dunia didorong oleh aturan sintaksnya yang mudah dipahami sehingga menjadikannya pilihan yang ideal bagi pemula yang baru memulai perjalanan di dunia *pemrograman* (Hermawan *et al.*, 2024).

2.9. *Preprocessing*

Preprocessing adalah tahap awal dalam pengolahan data yang bertujuan untuk mempersiapkan data mentah sebelum dianalisis lebih lanjut. Proses ini mencakup berbagai langkah penting seperti membersihkan data dari kesalahan, duplikasi, atau gangguan (*noise*), mengatasi data yang hilang, serta mengubah format data agar sesuai dengan kebutuhan analisis atau model yang digunakan. *Preprocessing* juga dapat melibatkan normalisasi, standarisasi, *encoding* dan kategorikal, hingga transformasi data agar lebih mudah diolah dan menghasilkan hasil analisis yang lebih akurat (Ariyani *et al.*, 2022). Adapun tahapan-tahapan *preprocessing* adalah sebagai berikut.

2.9.1. *Cleaning*

Cleaning merupakan proses penting untuk memastikan kualitas dataset sebelum digunakan dalam pelatihan model atau analisis lanjutan. Proses ini mencakup identifikasi dan penanganan berbagai masalah seperti duplikasi gambar, gambar yang rusak atau tidak dapat dibuka, serta ketidaksesuaian format atau ukuran gambar. Tujuan utama dari *cleaning* adalah untuk memastikan bahwa dataset yang digunakan bersih, akurat, dan representatif sehingga dapat meningkatkan performa dan keandalan model yang dibangun (Telkom University, 2025).

2.9.2. Anotasi Gambar

Anotasi gambar (*image annotation*) adalah proses penting dalam bidang *image retrieval* dan *computer vision* yang bertujuan untuk memberikan informasi tekstual atau label pada sebuah gambar agar isi visual dari gambar tersebut dapat dipahami dan diolah oleh sistem komputer. Label yang diberikan dapat berupa nama objek, aktivitas, lokasi, atribut visual, atau hubungan antar objek dalam gambar. Proses anotasi bisa dilakukan secara manual oleh manusia, semi-otomatis, atau sepenuhnya otomatis menggunakan algoritma kecerdasan buatan. Anotasi dilakukan berdasarkan konten gambar seperti objek misalnya daun, posisi objek dengan menggunakan *bounding box* atau *segmentation mask*, serta konteks gambar misalnya rusak dan warna dominan. Hasil dari anotasi sangat berguna untuk berbagai aplikasi seperti pelabelan data untuk pelatihan model *machine learning*, pengelompokan dan pengorganisasian gambar, dan peningkatan akurasi sistem pengenalan visual. Dengan anotasi yang tepat dan akurat. Dengan anotasi yang tepat dan akurat, sistem komputer dapat memahami konteks dan isi visual gambar secara lebih mendalam sehingga mampu mengambil keputusan dan melakukan analisis yang lebih cerdas dan efisien (Anwari *et al.*, 2018).

2.9.3. Cropping

Cropping image atau pemotongan citra merupakan teknik yang digunakan untuk menentukan spesifik bagian dari gambar yang mengandung objek utama yang ingin dianalisis atau diolah lebih lanjut. Proses *cropping* dilakukan pada tahap *preprocessing* yang secara umum bertujuan untuk meningkatkan kualitas gambar. Hal ini meliputi penghilangan *noise*, penajaman tepi objek, serta pengurangan efek blur atau keburaman. *Noise* sendiri adalah

informasi yang terekam dalam citra namun tidak dibutuhkan karena dapat mengganggu analisis, dan bisa saja berada di baik dalam maupun luar objek yang diteliti. Untuk *noise* yang terletak di luar area objek, proses *cropping* menjadi salah satu cara efektif untuk menghilangkannya. Dengan memangkas citra berdasarkan koordinat tertentu, ukuran gambar menjadi lebih kecil dan fokus hanya pada area yang relevan. Hasil dari proses ini adalah potongan gambar atau objek dengan ukuran tertentu yang telah dipisahkan dari bagian gambar lain yang tidak diperlukan. Teknik *cropping* ini sangat penting dalam mempersiapkan citra agar lebih bersih dan siap digunakan pada tahap pengolahan selanjutnya, seperti klasifikasi, segmentasi, atau deteksi objek (Yuhandri, 2019).

2.9.4. Segmentasi

Segmentasi citra adalah salah satu proses penting dalam pengolahan citra digital yang berfungsi untuk memisahkan area objek dari area latar belakang dalam sebuah gambar. Tujuan utama dari segmentasi ini adalah untuk mengisolasi bagian-bagian tertentu dari citra agar lebih mudah dianalisis dan diidentifikasi pada tahap selanjutnya. Proses ini sangat penting dalam berbagai aplikasi yang melibatkan pemahaman visual seperti pengenalan pola, klasifikasi objek, deteksi tepi, dan analisis bentuk. Dengan memisahkan wilayah yang relevan, segmentasi membantu sistem komputer fokus pada informasi penting dalam citra, terutama saat objek yang diamati memiliki kemiripan warna atau bentuk dengan latar belakangnya (Kumaseh *et al.*, 2013).

Segmentasi citra merupakan proses yang dapat diterapkan baik pada gambar berwarna maupun pada gambar dalam skala keabuan. Teknik ini bertujuan untuk membagi citra menjadi beberapa bagian

atau wilayah yang memiliki karakteristik serupa, sehingga objek yang terdapat dalam gambar dapat dipisahkan secara lebih jelas dari latar belakang atau objek lainnya. Proses segmentasi ini bertujuan untuk menyederhanakan representasi citra agar lebih mudah dianalisis baik secara visual maupun komputasional. Dengan memisahkan objek dari latar belakang, batas objek menjadi lebih terdefinisi sehingga memungkinkan deteksi fitur dan pengambilan nilai piksel objek secara lebih akurat. Hal ini sangat penting dalam berbagai aplikasi seperti pengenalan objek, klasifikasi, atau deteksi bentuk, di mana kejelasan batas objek berperan besar dalam keberhasilan analisis citra (Rachmat *et al.*, 2022).

2.9.5. *Padding*

Padding adalah sebuah parameter dalam operasi konvolusi yang berfungsi untuk menambahkan sejumlah piksel dengan nilai nol di sekitar tepi citra *input*. tujuan utama dari pemberian *padding* ini adalah untuk mengatur dimensi *output* dari *convolutional layer* agar tidak jauh berbeda dari dimensi *input*, atau bahkan tetap sama. Dengan menambahkan *padding*, hilangnya informasi di tepi gambar akibat proses konvolusi dapat diminimalkan, karena piksel di bagian pinggir tetap terlibat dalam perhitungan filter. Selain itu, *padding* memungkinkan jaringan konvolusional memiliki arsitektur yang lebih dalam karena ukuran fitur map tidak cepat mengecil seiring bertambahnya lapisan. Hal ini penting untuk memungkinkan ekstraksi lebih banyak fitur secara bertahap dan mendalam, yang pada akhirnya meningkatkan kemampuan jaringan dalam mengenali pola-pola kompleks dalam citra (Magdalena *et al.*, 2021).

2.9.6. Resizing

Resizing merupakan proses penyesuaian ulang dimensi citra, baik dengan cara memperkecil (*downscaling*) maupun memperbesar (*upscaling*) resolusinya agar sesuai dengan kebutuhan sistem atau algoritma pengolahan citra yang digunakan. Proses ini bertujuan untuk mengatur resolusi gambar ke ukuran tertentu yang diinginkan, sehingga gambar menjadi lebih efisien untuk diproses. Jika citra memiliki resolusi tinggi maka ukuran data yang besar dapat membebani memori dan memperlambat kinerja aplikasi serta meningkatkan waktu pemrosesan secara signifikan. Sebaliknya, jika resolusi gambar terlalu rendah maka informasi visual yang dibutuhkan bisa jadi tidak cukup akurat untuk dianalisis, yang dapat mengurangi efektivitas sistem terutama dalam konteks klasifikasi, deteksi objek, atau segmentasi. Oleh karena itu, proses *resizing* harus dilakukan dengan pertimbangan yang tepat agar diperoleh keseimbangan antara efisiensi komputasi dan kualitas informasi visual yang tetap memadai untuk kebutuhan analisis lebih lanjut (Sari *et al.*, 2023).

2.10. Augmentasi Data

Augmentasi data adalah teknik penting dalam pemrosesan dan analisis data, khususnya dalam konteks *machine learning* dan kecerdasan buatan. Metode augmentasi data meningkatkan kinerja deteksi dengan memasukkan variasi tambahan ke dalam dataset pelatihan melalui transformasi seperti rotasi, penskalaan, dan penambahan *noise* (Toyib *et al.*, 2024).

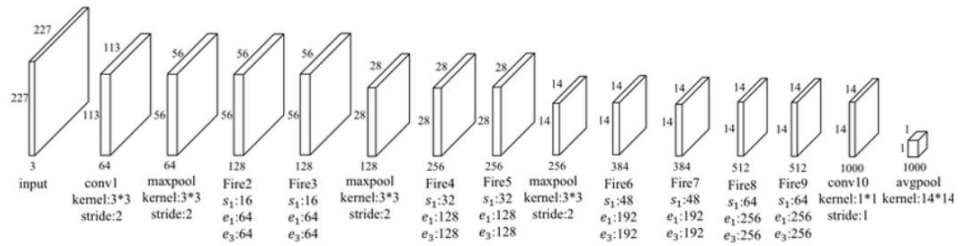
Augmentasi data merupakan metode untuk mengatasi masalah *overfitting* dengan cara memperbanyak data, terutama pada dataset yang terbatas. Teknik ini menerapkan berbagai metode untuk memperkaya atribut dan

memperbesar ukuran dataset yang digunakan dalam pelatihan (Auliaddina & Arifin, 2024).

2.11. *SqueezeNet*

SqueezeNet merupakan salah satu arsitektur model *Convolutional Neural Network* (CNN). Arsitektur ini terkenal karena kemampuannya menghasilkan model yang lebih ringan tanpa mengorbankan performa signifikan. *SqueezeNet* dikembangkan dengan tujuan untuk mengurangi jumlah parameter dalam model yang pada gilirannya mengurangi kompleksitas dan ukuran keseluruhan model. Hal ini sangat penting untuk penerapan model pada perangkat dengan sumber daya terbatas, seperti perangkat *mobile* atau *Internet of Things* (IoT) yang memiliki keterbatasan dalam hal komputasi dan memori (Armiady & Muslem, 2023).

Arsitektur *SqueezeNet* dirancang dengan fokus pada penerapan di perangkat-perangkat tertanam, seperti yang digunakan dalam mobil. Berbeda dengan arsitektur *AlexNet* yang dikenal besar dan kompleks, *SqueezeNet* menawarkan pendekatan yang jauh lebih ringkas dengan ukuran yang bahkan 50 kali lebih kecil. Dalam pengembangannya, *SqueezeNet* memperhatikan batasan perangkat keras yang mungkin memiliki keterbatasan dalam kapasitas memori dan daya komputasi. Untuk mencapai tujuan ini, *SqueezeNet* mengusulkan beberapa pendekatan penting, termasuk strategi untuk mengurangi jumlah parameter yang diperlukan oleh arsitektur, serta pengenalan modul *fire* yang menjadi elemen kunci dalam struktur arsitektur ini. Modul *fire* dirancang khusus untuk mengoptimalkan keseimbangan antara pengurangan parameter dan pemeliharaan kualitas fitur yang relevan dalam proses ekstraksi fitur (Harahap *et al.*, 2024). Arsitektur *SqueezeNet* dapat dilihat pada Gambar 7.



Gambar 7. Arsitektur *SqueezeNet* (Deng & Wu, 2019).

Adapun lapisan-lapisan yang ada dalam model tersebut adalah sebagai berikut

2.11.1. *Convolutional Layer*

Dalam CNN *convolutional layer* merupakan lapisan awal yang bertugas menerima *input* berupa citra. Lapisan ini melakukan operasi konvolusi, yaitu proses penerapan sejumlah filter (*kernel*) di area-area kecil dari citra untuk mengekstraksi fitur penting seperti tepi, tekstur, atau pola tertentu. Setiap filter bergerak melintasi gambar dengan langkah tertentu (*stride*) dan menghasilkan peta fitur (*feature map*) baru yang merepresentasikan informasi lokal dari gambar. Ukuran filter ditentukan oleh panjang dan lebar dalam satuan piksel, sementara kedalamannya disesuaikan dengan jumlah *channel* pada citra masukan, misalnya 3 *channel* untuk gambar berwarna (RGB). Proses ini memungkinkan jaringan untuk mengenali berbagai pola visual secara bertahap di lapisan-lapisan berikutnya (Ahadin *et al.*, 2024)

2.11.2. *Pooling Layer*

Setelah melewati *convolutional layer*, data citra akan diproses oleh lapisan *pooling* yang berfungsi untuk mengurangi dimensi data dan menerima *output* dari lapisan sebelumnya. Lapisan ini menggunakan filter dengan ukuran dan langkah (*stride*) tertentu

untuk menyaring informasi dari *feature map*. Jarak pergerakan filter saat menyapu area pada *feature map* ditentukan oleh nilai *stride*. Misalnya pada metode *Max Pooling* dengan ukuran 2×2 dan *stride* 2, nilai tertinggi dari setiap blok 2×2 akan dipilih dalam setiap langkah pergeseran. Sementara itu, *Average Pooling* mengambil rata-rata nilai dalam area tersebut. Proses ini membantu mengurangi ukuran representasi data dan jumlah parameter yang dibutuhkan di lapisan berikutnya, menghasilkan *output* citra dengan ukuran yang lebih kecil dibandingkan *input* aslinya (Ahadin *et al.*, 2024). Ilustrasi proses pada *pooling layer* dapat dilihat pada Gambar 8.



Gambar 8. Ilustrasi Proses Pada *Pooling Layer*
(Ahadin *et al.*, 2024).

2.11.3. Fire Modules

Fire module merupakan komponen utama dalam arsitektur *SqueezeNet* yang dirancang untuk mengurangi jumlah parameter model tanpa mengorbankan akurasi secara signifikan. *Fire module* terdiri dari dua bagian utama, yaitu *squeeze layer* dan *expand layer*. *Squeeze layer* berfungsi untuk mengurangi jumlah saluran pada *input* dengan menggunakan sejumlah filter tertentu, yang memungkinkan ekstraksi fitur dengan jumlah parameter yang lebih sedikit serta efisiensi komputasi yang lebih tinggi. *Output* dari *squeeze layer* akan diteruskan ke *expand layer* yang bertugas memperluas kembali jumlah saluran (Anjanadevi *et al.*, 2023).

2.11.4. *Global Average Pooling*

Salah satu kelebihan *Global Average Pooling* (GAP) dibandingkan dengan lapisan *fully connected* adalah sifatnya lebih menyatu dengan struktur konvolusional karena GAP secara langsung menghubungkan peta fitur dengan kelas-kelas yang diklasifikasikan. Dengan cara ini, setiap peta fitur dapat ditafsirkan sebagai representasi tingkat keyakinan terhadap suatu kategori (*confidence map*). Keuntungan lainnya adalah bahwa GAP tidak memiliki bobot parameter yang harus dilatih, sehingga kemungkinan terjadinya *overfitting* di tahap ini bisa diminimalkan. Secara keseluruhan, GAP dapat dipandang sebagai bentuk regulasi struktural yang secara eksplisit mendorong peta fitur berperan sebagai indikator keyakinan terhadap masing-masing kategori (Lin *et al.*, 2014).

2.11.5. *Kernel*

Kernel atau filter pada *Convolutional Layer* adalah sebuah matriks yang memiliki ukuran tertentu, dimana panjang, lebar, dan kedalamannya disesuaikan dengan dimensi dari data input yang diproses. Matriks ini memiliki peran yang sangat krusial dalam operasi konvolusi karena bertugas mengekstraksi berbagai informasi penting dari citra. Cara kerjanya adalah dengan memperhatikan nilai-nilai piksel yang ada saat ini beserta nilai piksel-piksel di sekitarnya. Dengan menggunakan *kernel* ini, CNN mampu mendeteksi pola-pola khas seperti tepi, sudut, atau tekstur dalam citra, kemudian menerjemahkan informasi tersebut ke dalam bentuk *feature map* yang digunakan oleh lapisan-lapisan berikutnya untuk memahami gambar secara lebih mendalam (Sari, 2020).

2.11.6. *Stride*

Stride merupakan parameter penting dalam operasi konvolusi yang menentukan jarak pergeseran *kernel* ketika melewati citra *input*. Jika *stride* bernilai 1, *kernel* akan berpindah satu piksel demi satu piksel baik ke arah horizontal maupun vertikal, sehingga area citra yang diproses menjadi lebih banyak detail informasi yang dihasilkan semakin tinggi. Sebaliknya, jika *stride* bernilai 2, *kernel* bergeser dua piksel pada setiap langkah, sehingga ukuran *output feature map* menjadi lebih kecil karena sebagian informasi citra dilewati. Dengan demikian, semakin kecil nilai *stride* yang digunakan, semakin kaya detail representasi fitur yang diperoleh, tetapi hal ini juga meningkatkan jumlah operasi komputasi yang harus dilakukan sehingga memerlukan waktu pemrosesan yang lebih lama (Magdalena *et al.*, 2021).

2.12. *Library*

Dalam upaya menghemat waktu dan efisiensi kerja, para ilmuwan data maupun pengembang umumnya memanfaatkan berbagai kerangka kerja (*framework*) dan pustaka (*library*) yang disediakan oleh bahasa pemrograman *Python*. *Python* dikenal memiliki ekosistem pustaka yang sangat kaya dan mengandung berbagai kebutuhan dalam pengolahan serta analisis data. Dengan memanfaatkan pustaka-pustaka tersebut, permasalahan yang kompleks dapat diselesaikan dalam waktu yang relatif singkat, sehingga proses pengembangan dan analisis menjadi lebih efektif dan terstruktur (Wijoyo *et al.*, 2024).

2.12.1. *PyTorch*

PyTorch adalah *library python* yang digunakan untuk komputasi *deep learning*, dikenal karena fleksibilitas dan kemudahannya

dalam membangun model menggunakan sintaks *python*. Sejak rilis, *PyTorch* banyak digunakan oleh komunitas riset dan kini menjadi salah satu alat utama dalam pengembangan *deep learning*. *Library* ini menggunakan struktur data utama bernama *Tensor*, yaitu *array* multidimensi mirip *NumPy* yang memungkinkan eksekusi operasi matematika secara efisien, terutama dengan dukungan perangkat keras modern. *PyTorch* juga mendukung pelatihan terdistribusi dan pemrosesan paralel untuk efisiensi pemuatan data. Saat dipadukan dengan *torchvision*, *PyTorch* sangat efektif untuk pengolahan data visual seperti gambar dan video, terutama dalam penerapan *Convolutional Neural Network* (CNN), berkat dukungan terhadap arsitektur CNN populer (Hendri *et al.*, 2021).

2.12.2. *Scikit_learn*

Scikit_learn merupakan *library python* yang dirancang untuk mempermudah pengembangan *machine learning* melalui API yang intuitif hasil kontribusi dari berbagai pengembang internasional. *Library* ini banyak digunakan di kalangan industri maupun akademisi dan telah dioptimalkan karena dibangun di atas *library NumPy* dan *SciPy*, sehingga mendukung efisiensi dalam perhitungan numerik (Fahmi, 2023).

2.12.3. *PIL*

Python Imaging Library (PIL) adalah pustaka *Python* yang dirancang khusus untuk memproses dan memanipulasi file gambar. *Library* ini menyediakan berbagai fungsi yang memungkinkan pengguna untuk membuka, membaca, dan menyimpan gambar dalam berbagai format seperti JPEG, PNG, BMP, dan lainnya. Selain itu, *PIL* juga mendukung berbagai operasi pengolahan gambar dasar seperti mengubah ukuran (*resize*), memutar gambar

(*rotate*), memotong (*crop*), menyesuaikan warna atau kecerahan, serta melakukan konversi antar format file gambar (Asaad *et al.*, 2023).

2.12.4. *Matplotlib*

Salah satu *library python* yang paling banyak digunakan untuk membuat diagram analisis data adalah *matplotlib*. Proses ini dikenal sebagai visualisasi data. Dengan menambahkan *library matplotlib*, pengguna dapat menampilkan gambar dan grafik yang sangat membantu dalam analisis matematika untuk memahami berbagai informasi secara lebih jelas melalui tampilan visual (Asaad *et al.*, 2023).

2.12.5. *Seaborn*

Seaborn merupakan *library* yang dikembangkan di atas *matplotlib* dan dirancang khusus untuk visualisasi data statistik. *Library* ini menawarkan fitur dan tampilan yang lebih menarik dan informatif. *Seaborn* memungkinkan pembuatan berbagai jenis grafik seperti *heatmap*, diagram batang, dan jenis grafik lainya yang membantu memberikan pemahaman yang lebih dalam terhadap data (Hermanto *et al.*, 2022).

2.12.6. *Numpy*

NumPy merupakan *library* dalam *Python* yang memungkinkan berbagai operasi manipulasi data dilakukan dengan mudah. Interaksi antara *array NumPy* dan *Python* sangat menyerupai penggunaan variabel standar dalam *Python*. Selain itu, *NumPy* juga menyediakan fungsi-fungsi lanjutan untuk keperluan seperti aljabar

linier, *transformasi Fourier*, dan operasi pada matriks (Putra *et al.*, 2023).

2.13. *Hyperparameter*

Hyperparameter merupakan parameter yang nilainya tidak ditentukan dari data melainkan ditetapkan secara langsung oleh pengguna. Secara singkat *hyperparameter* adalah nilai-nilai yang perlu ditentukan terlebih dahulu sebelum proses pelatihan model dimulai. *Hyperparameter* berfungsi untuk mengatur bagaimana model belajar serta membantu dalam mengoptimalkan performa model *machine learning*. Pengaturannya perlu dilakukan secara cermat, karena kesalahan dalam penentuan *hyperparameter* dapat berdampak negatif terhadap kinerja model, bahkan bisa menyebabkan *overfitting* atau *underfitting* (Telkom University, 2025).

2.13.1. *Batch Size*

Batch Size merupakan istilah dalam *machine learning* yang merujuk pada jumlah sampel pelatihan yang digunakan dalam satu iterasi. *Batch size* juga termasuk salah satu *hyperparameter* krusial yang perlu disesuaikan dalam sistem *deep learning* (Rochmawati *et al.*, 2021).

Dalam *deep learning*, *batch size* mengacu pada jumlah data yang diproses sekaligus dalam satu iterasi selama pelatihan model. Pemilihan *batch size* yang sesuai sangat penting karena dapat mempercepat proses training dan membantu mengurangi risiko *overfitting* (Sadewa & Yamasari, 2024).

2.13.2. *Learning Rate*

Learning rate adalah salah satu *hyperparameter* penting dalam proses pelatihan model *machine learning* atau *deep learning*, yang

menentukan seberapa besar langkah pembaruan bobot dilakukan setiap kali model belajar dari kesalahan. Nilai ini secara langsung mempengaruhi kecepatan dan stabilitas proses *training*. Jika *learning rate* diatur terlalu kecil, maka proses pelatihan akan berlangsung sangat lambat karena model melakukan pembaruan bobot dalam langkah yang sangat kecil, sehingga memerlukan banyak iterasi untuk mencapai konvergensi. Sebaliknya, jika *learning rate* terlalu besar, langkah pembaruan bisa menjadi terlalu besar sehingga menyebabkan model melompati titik optimal, bahkan berujung pada ketidakstabilan pelatihan dan kegagalan dalam menemukan solusi terbaik. Oleh karena itu, memilih nilai *learning rate* yang tepat sangat penting agar model dapat belajar secara efisien dan menghasilkan performa optimal (Rismiyati & Luthfiarta, 2021).

2.13.3. *Weight Decay*

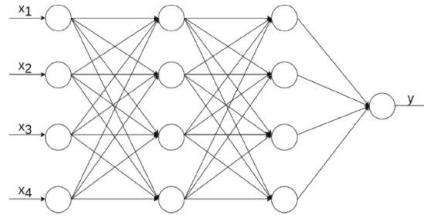
Weight decay adalah teknik yang digunakan dalam pelatihan model *machine learning* untuk mencegah *overfitting* dan meningkatkan kemampuan generalisasi model, terutama saat model berupa kombinasi linear dari beberapa fungsi dasar seperti *neuron* dalam *neural network*. *Weight decay* memberi penalti terhadap bobot yang nilainya terlalu besar. Tujuannya agar model tidak terlalu mengandalkan satu atau dua fitur saja dan tetap belajar dari pola umum dalam data. Penalti ini ditambahkan ke dalam fungsi loss dalam bentuk norma kuadrat dari semua bobot model. Dengan menambahkan penalti ini model akan cenderung memiliki bobot yang kecil sehingga tidak terlalu kompleks. Istilah *weight decay* muncul karena dalam kondisi tanpa fungsi *error* utama, proses penurunan loss hanya berdasarkan penalti bobot akan menyebabkan bobot-bobot tersebut mengecil secara bertahap selama pelatihan (Gnecco & Sanguineti, 2009).

2.13.4. *ReduceLROnPlateau*

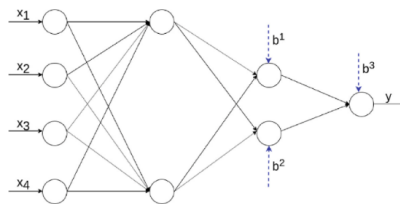
ReduceLROnPlateau adalah sebuah teknik yang digunakan dalam pelatihan model *machine learning* untuk menyesuaikan nilai *learning rate* secara otomatis dan dinamis berdasarkan performa model selama proses training. Mekanisme ini akan mengurangi *learning rate* ketika metrik evaluasi tertentu seperti *validation loss*, tidak menunjukkan perbaikan dalam jumlah *epoch* tertentu. Dengan cara ini, model dapat lebih stabil dalam mencapai konvergensi dan menghindari tersendat di titik minimum lokal akibat *learning rate* yang terlalu besar. Teknik ini sangat berguna untuk memperbaiki proses optimasi ketika model mengalami stagnasi dalam pelatihan (Wardani, 2017).

2.13.5. *DropOut*

Dropout merupakan teknik yang digunakan untuk mengurangi risiko *overfitting* sekaligus mempercepat proses pelatihan model. *Overfitting* sendiri terjadi ketika model terlalu cocok dengan data latih, sehingga performanya menurun saat dihadapkan pada data baru yang belum pernah dilihat. Dalam implementasinya, *dropout* bekerja dengan cara menonaktifkan secara acak sejumlah *neuron* dalam jaringan selama proses pelatihan. Dengan menghilangkan sementara koneksi antar *neuron* ini, kompleksitas model berkurang dan jaringan terdorong untuk belajar representasi yang lebih umum, bukan menghafal data pelatihan (Ahadin *et al.*, 2024). Ilustrasi *neuron* sebelum terdapat *DropOut* dapat dilihat pada Gambar 9 dan ilustrasi *neuron* setelah terdapat *DropOut* dapat dilihat pada gambar 10.



Gambar 9. Ilustrasi *Neuron* Sebelum Terdapat *DropOut* (Ahadin *et al.*, 2024).



Gambar 10. Ilustrasi *Neuron* Setelah Terdapat *DropOut* (Ahadin *et al.*, 2024).

Sejumlah penelitian telah menunjukkan bahwa penerapan *dropout* dapat membantu meningkatkan akurasi dan kemampuan generalisasi pada model CNN. Namun, efektivitasnya terhadap performa CNN masih cepat berbeda-beda, tergantung pada pengaturan konfigurasi yang dipakai serta karakteristik data yang digunakan dalam pelatihan (Yan & Wiliani, 2025).

2.13.6. *Early Stopping*

Early stopping merupakan teknik yang digunakan untuk menghentikan proses pelatihan model secara otomatis apabila tidak terjadi peningkatan pada metrik evaluasi tertentu seperti akurasi atau loss pada data validasi selama jumlah *epoch* berturut-turut. Dengan kata lain, jika model tidak menunjukkan perbaikan performa dalam periode waktu yang ditentukan, maka pelatihan akan dihentikan lebih awal sebelum mencapai jumlah *epoch* maksimum. Tujuan dari teknik ini adalah untuk mencegah *overfitting*, menghemat waktu pelatihan, dan memastikan bahwa

model yang dihasilkan memiliki kemampuan generalisasi yang baik terhadap data baru (Bachri & Gunawan, 2024).

2.13.7. Epoch

Epoch merupakan istilah yang digunakan dalam *machine learning* untuk menunjukkan satu siklus penuh proses pembelajaran suatu algoritma dari data *train*. Satu *epoch* berarti seluruh data *train* telah diproses sekali secara menyeluruh oleh model. Dalam jaringan saraf tiruan, proses ini dilakukan secara berulang dengan tujuan menyesuaikan bobot agar mendekati nilai optimal. Melalui pembelajaran berulang tersebut, model diharapkan mencapai konvergensi, yaitu kondisi ketika perubahan bobot menjadi stabil dan performa model meningkat. Namun, penggunaan *epoch* yang terlalu sedikit dapat menyebabkan model belum mampu mengenali pola dengan baik, sedangkan *epoch* yang terlalu banyak berpotensi membuat model hanya menghafal data *train* (Wasil *et al.*, 2022).

2.14. Data Train

Menurut Wijoyo A *et al.* (2024) Data *train* adalah kumpulan informasi yang digunakan untuk melatih model *machine learning*. Dalam proses pelatihan *machine learning*, digunakan sekumpulan data empiris yang berfungsi sebagai dasar untuk menyesuaikan parameter atau bobot pada model. Proses penyesuaian ini bertujuan agar model mampu merepresentasikan pola dan karakteristik yang terkandung dalam data secara optimal. Data yang digunakan untuk tahap pembelajaran tersebut dikenal dengan istilah data *train*. Melalui data *train*, model belajar mengenali hubungan antara input dan output, sehingga dapat membangun fungsi prediksi atau klasifikasi yang mendekati kondisi nyata (ScienceDirect, 2021).

2.15. Data Validation

Dalam *machine learning*, setelah model melalui tahap pelatihan menggunakan data *train*, diperlukan juga data *validation* sebagai alat evaluasi. Data *validation* berfungsi untuk menilai sejauh mana model mampu mengidentifikasi pola pada data yang belum pernah dilihat sebelumnya. Melalui proses ini, dapat diketahui apakah model memiliki kemampuan generalisasi yang baik atau justru mengalami *overfitting* (Telus Digital, 2025).

2.16. Data Test

Data *testing* adalah bagian penting dalam pengembangan model *machine learning* yang digunakan untuk mengevaluasi kinerja model. Data ini terdiri dari kumpulan data yang tidak memiliki label atau kelas yang diketahui sehingga digunakan untuk menguji kemampuan model dalam membuat prediksi berdasarkan pola yang telah dipelajari selama pelatihan. Selama proses pengujian model, atribut label dari data pengujian tidak ditampilkan agar klasifikasi dapat dilakukan tanpa mengetahui hasil sebenarnya. Setelah klasifikasi selesai label tersebut akan digunakan untuk membandingkan hasil klasifikasi, sehingga dapat dinilai seberapa akurat model tersebut dalam melakukan klasifikasi (Musu *et al.*, 2021).

2.17. Confusion Matrix

Confusion Matrix merupakan teknik evaluasi untuk mengukur performa model klasifikasi dengan membandingkan hasil prediksi dengan data aktual sebenarnya (Alvian *et al.*, 2021). Menurut Purbolaksono *et al.* (2021), *Confusion Matrix* dapat dilihat pada Tabel 2 berikut.

Tabel 2. *Confusion Matrix* (Purbolaksono *et al.*, 2021)

	Data Aktual	
	Positif	Negatif
Data Prediksi Positif	<i>True Positive</i> (TP)	<i>False Positive</i> (FP)
Data Prediksi Negatif	<i>False Negative</i> (FN)	<i>True Negative</i> (TN)

Keterangan:

True Positive (TP) merujuk pada data positif yang diprediksi dengan benar.

True Negative (TN) adalah jumlah data negatif yang terdeteksi dengan akurat.

False Positive (FP) yang juga dikenal sebagai *Type-1 Error*. Adalah data negatif yang salah terdeteksi sebagai positif.

False Negative (FN) atau *Type-2 Error* merupakan kebalikan dari *True Positive*, di mana data positif terdeteksi sebagai negatif.

Menurut Paraijun *et al.* (2022) berikut ini adalah persamaan-persamaan perhitungan dari *Confusion Matrix* yaitu.

2.17.1. Accuracy

Nilai akurasi membantu mengevaluasi seberapa baik model klasifikasi dalam memprediksi secara keseluruhan. Namun perlu diperhatikan bahwa akurasi saja tidak cukup unyuk dataset yang tidak seimbang. Rumus untuk menghitung akurasi dapat dilihat pada Persamaan 1 (Paraijun *et al.*, 2022).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots (1)$$

2.17.2. *Precision*

Pengukuran nilai presisi dilakukan dengan menggunakan rumus berikut. Secara intuitif, presisi menggambarkan kemampuan *classifier* untuk tidak mengklasifikasikan sampel negative sebagai positif. Rumus untuk menghitung presisi dapat dilihat pada Persamaan 2 (Paraijun *et al.*, 2022).

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots (2)$$

2.17.3. *Recall*

Pengukuran nilai *recall* dilakukan dengan menggunakan rumus berikut. Secara intuitif, *recall* mencerminkan kemampuan *classifier* dalam mengidentifikasi semua sampel positif. Rumus untuk menghitung *recall* dapat dilihat pada Persamaan 3 (Paraijun *et al.*, 2022).

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots (3)$$

2.17.4. *F1-Score*

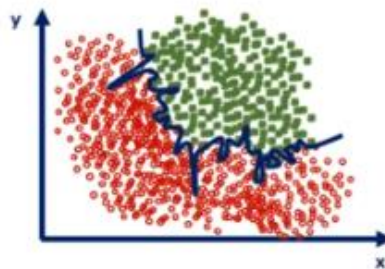
Pengukuran *F1-Score* dilakukan dengan menggunakan rumus berikut. *F1-Score* adalah matrix yang menggabungkan *precision* dan *recall* dalam satu nilai sehingga memberikan keseimbangan antara keduanya. *F1-Score* berguna untuk mengevaluasi model pada dataset yang tidak seimbang karena mempertimbangkan *false positive* dan *false negative*. Rumus untuk menghitung *F1-Score* dapat dilihat pada Persamaan 4 (Paraijun *et al.*, 2022).

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \dots\dots\dots (4)$$

Confusion Matrix berfungsi untuk mengevaluasi kinerja model dan disajikan secara visual dalam bentuk tabel. Ini memberikan wawasan yang lebih mendalam kepada praktisi data mengenai kinerja model, kesalahan, dan kelemahan yang ada. Dengan demikian praktisi data dapat melakukan analisis lebih lanjut terhadap model melalui penyesuaian yang diperlukan. (Fitrianingsih & Zuraeni, 2024).

2.18. *Overfitting*

Dalam metode *machine learning* karakteristik data yang diamati dapat menyebabkan kompleksitas berlebihan pada model, yang beresiko mengakibatkan *overfitting*. Sesuai dengan istilahnya, *overfitting* terjadi ketika algoritma memprediksi terlalu akurat dalam menggambarkan data sampel, sehingga model tersebut hanya menghafal pola yang ada dari data sebelumnya dan tidak benar-benar mempelajari pola yang lebih umum (Firmansyach *et al.*, 2023). Visualisasi *overfitting* dapat dilihat pada Gambar 11.

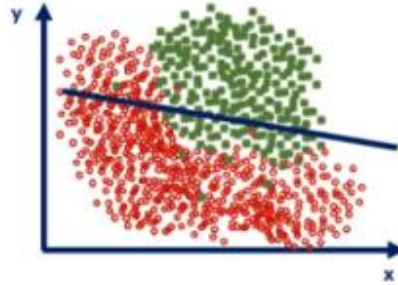


Gambar 11. Ilustrasi *Overfitting* (Fitrah *et al.*, 2021).

2.19. *Underfitting*

Pada kasus ini, *underfitting* model memiliki kondisi ciri-ciri palsu yang berlebihan serta akurasi rendah. *Underfitting* berbeda dengan *overfitting* karena model *underfitting* tidaklah kompleks dan kurang fokus pada data latih. Sehingga, hal ini menyebabkan pola yang sangat buruk ketika menggunakan data latih. Secara umum, *underfitting* terjadi karena model

tidak berhasil merepresentasikan pola-pola yang signifikan dalam data latih (Firmansyach *et al.*, 2023). Visualisasi *underfitting* dapat dilihat pada Gambar 12.



Gambar 12. Ilustrasi *Underfitting* (Fitrah *et al.*, 2021).

2.20. Roboflow

Roboflow adalah sebuah *platform* berbasis *web* yang dirancang untuk menangani pengelolaan dataset secara menyeluruh dalam pengembangan model *computer vision*, mulai dari tahap pengumpulan hingga optimalisasi data. *Platform* ini menyediakan berbagai fitur yang memudahkan proses *preprocessing* seperti augmentasi, *resizing*, anotasi, serta manajemen dataset secara kolaboratif. Selain itu, *roboflow* juga mendukung proses pelatihan model melalui teknik augmentasi otomatis dan kemampuan untuk mengkonversi dataset ke berbagai format yang kompatibel dengan *framework machine learning* populer seperti *TensorFlow*, *PyTorch*, dan *YOLO*. Berkat kelengkapan fitur tersebut, *roboflow* menjadi alat yang efektif untuk membantu pengembang meningkatkan efisiensi kerja, kualitas data, dan akurasi model *computer vision* (Saepudin *et al.*, 2024).

III. METODOLOGI PENELITIAN

3.1. Tempat dan Waktu

3.1.1. Tempat

Penelitian ini dilakukan di laboratorium komputasi dasar gedung MIPA Terpadu Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Laboratorium tersebut berlokasi di Jl. Prof. Dr. Ir. Sumantri Brojonegoro No.1, Gedong Meneng, Kec. Rajabasa, Kota Bandar Lampung, Lampung.

3.1.2. Waktu

Penelitian ini dilakukan pada semester ganjil tahun ajaran 2024/2025 sampai dengan semester genap tahun ajaran 2025/2026 pada bulan November 2024 sampai dengan bulan Juni 2025. Alur pengerjaan penelitian dapat dilihat pada Tabel 3.

3.2. Data dan Alat

3.2.1. Data

Data yang digunakan pada penelitian ini adalah data citra digital yang bersumber dari *Kaggle*. *Kaggle* merupakan *platform* penyedia kumpulan dataset yang dapat diakses secara gratis dan mencakup berbagai bidang seperti pertanian, kesehatan, ekonomi, sosial, olahraga, dan lain-lain. Penelitian ini menggunakan dataset yang

[illegible]

berjudul “*Coffee Plant Disease*” yang berisi citra daun kopi yang berjumlah 1000 gambar RGB dengan 3 kelas yaitu *healthy*, *Rust*, dan *red_spider_mite* dengan ekstensi .jpg. Jumlah dan persentase data masing-masing kelas dapat dilihat pada Tabel 4.

Tabel 4. Jumlah Dan Persentase Data Masing-Masing Kelas

Kelas	Jumlah	Persentase
<i>healthy</i>	435	43,5%
<i>Rust</i>	398	39,8%
<i>Red_spider_mite</i>	167	16,7%
Total	1000	100%

3.2.2. Alat

Alat yang digunakan dalam penelitian ini terbagi menjadi perangkat keras (*hardware*) dan perangkat lunak (*software*)

3.2.2.1. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan pada penelitian ini adalah Laptop dengan spesifikasi sebagai berikut.

- Processor*: 11th Gen Core™ i5-1135G7 @2.40GHz
- Installed RAM*: 8,00 GB
- Storage*: 512 GB

3.2.2.2. Perangkat Lunak (*Software*)

Adapun perangkat lunak yang digunakan pada penelitian ini adalah.

a. Sistem Operasi: *Windows 11 Pro Version 23H2*

Sistem operasi berfungsi sebagai *platform* utama yang mengelola perangkat keras dan perangkat lunak. Selain itu sistem operasi juga berperan dalam penyusunan laporan penelitian.

b. *Google Drive*

Pada penelitian ini, *Google Drive* digunakan untuk menyimpan dataset. *Google Drive* akan dihubungkan dengan *Google Colab* sehingga setiap manipulasi yang dilakukan pada *Google Colab* juga akan terjadi pada *Google Drive*.

c. *Google Colab*

Penelitian ini memanfaatkan *Google Colab* sebagai sarana utama untuk menulis, mengedit, menjalankan, serta melakukan berbagai manipulasi kode *Python* secara langsung melalui *browser*. Penggunaan *Google Colab* memungkinkan proses komputasi dilakukan secara *online* tanpa memerlukan instalasi perangkat lunak tambahan, sehingga mendukung efisiensi dan fleksibilitas dalam analisis data.

d. *Python 3.10.12*

Dalam penelitian ini, *Python* dipilih sebagai bahasa pemrograman karena memiliki kemampuan yang andal dalam mengelola data, serta mendukung berbagai proses yang dibutuhkan dalam *machine learning*. Sifatnya yang fleksibel dan mudah dipahami menjadikan *Python* sangat sesuai untuk digunakan dalam pengembangan dan pelaksanaan dalam penelitian ini.

e. *Roboflow*

Pada penelitian ini, *roboflow* digunakan untuk melakukan anotasi pada objek utama gambar yaitu daun kopi. Selain sebagai anotasi, *roboflow* juga digunakan untuk melakukan *cleaning*, karena pada *roboflow* ketika terdapat duplikasi pada gambar (bukan pada nama file) maka hanya 1 file yang berhasil diunggah.

f. *Package*

Adapun *package* atau *library* yang digunakan dalam penelitian ini adalah sebagai berikut.

- *PyTorch* 0.21.0+cu124

Pada penelitian ini, *PyTorch* digunakan sebagai kerangka kerja utama dalam tahap *modeling* dan pelatihan model *deep learning*. Library ini menyediakan arsitektur model *pretrained SqueezeNet* melalui *models.squeezenet1_1()*, loss function *nn.CrossEntropyLoss()* untuk klasifikasi multi-kelas, optimizer *optim.Adam()* untuk pembaruan bobot, serta *DataLoader()* untuk membaca dataset dalam batch. Selain itu, *torchvision.transforms* digunakan untuk melakukan transformasi data seperti augmentasi, normalisasi, dan konversi gambar ke *tensor* sebelum masuk ke model.

- *Scikit_learn* 1.6.0

Dalam penelitian ini, *Scikit-learn* digunakan pada tahap pembagian data dan evaluasi model. Fungsi *train_test_split()* digunakan untuk membagi dataset

menjadi data latih, validasi, dan uji secara *stratified* agar distribusi kelas tetap seimbang. Selain itu, *classification_report()* dan *confusion_matrix()* digunakan untuk menghitung dan menyajikan metrik evaluasi model seperti *precision*, *recall*, dan *f1-score*.

- *PIL 11.1.0*

Pada penelitian ini, *PIL* berperan penting dalam tahap *preprocessing* gambar, terutama dalam proses manipulasi citra sebelum pelatihan. Fungsi *Image.open()* digunakan untuk membuka gambar, *crop()* untuk memotong gambar sesuai *bounding box* hasil segmentasi, *resize()* untuk menyeragamkan ukuran menjadi 224×224 piksel, serta *Image.new()* dan *paste()* untuk menambahkan *padding* agar gambar berbentuk persegi tanpa mengubah proporsinya.

- *Matplotlib 3.10.0*

Pada penelitian ini, *Matplotlib* digunakan pada tahap visualisasi hasil pelatihan dan evaluasi model. Fungsi *plt.plot()* digunakan untuk menggambarkan grafik akurasi dan loss selama *training*, sedangkan *plt.show()* berfungsi untuk menampilkan grafik tersebut secara interaktif agar proses analisis model lebih mudah dan informatif.

- *Seaborn 0.13.2*

Pada penelitian ini, *Seaborn* digunakan untuk memvisualisasikan *confusion matrix* dalam bentuk *heatmap* melalui fungsi *sns.heatmap()*. Visualisasi

ini membantu dalam memahami pola kesalahan model dan menganalisis prediksi per kelas secara lebih intuitif dibandingkan tampilan tabel numerik biasa.

- *Numpy* 1.26.4

Pada penelitian ini, *NumPy* digunakan dalam berbagai proses numerik pada tahap *preprocessing* dan evaluasi. Fungsinya mencakup konversi data ke *array* (*np.array()*), menghitung akurasi prediksi (*np.mean()*), serta normalisasi atau unnormalisasi nilai piksel gambar. *NumPy* juga digunakan untuk menghitung warna rata-rata gambar yang digunakan sebagai warna *padding* dalam proses penyesuaian rasio gambar.

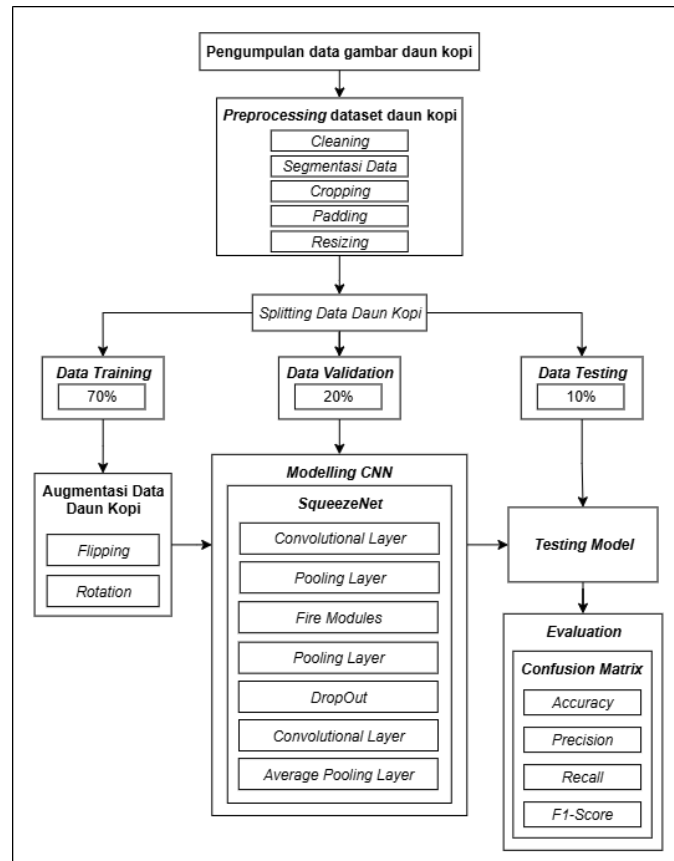
3.3. Metodologi

Penelitian ini memiliki beberapa tahapan yang dapat dilihat pada Gambar 13. Adapun penjelasan dari tahapan tersebut adalah sebagai berikut.

3.3.1. Pengumpulan Data

Penelitian ini menggunakan dataset yang berasal dari *Kaggle* yang berjumlah 1000 citra digital daun kopi dengan ekstensi .jpg dengan format RGB yang memiliki tiga kategori yaitu *Healthy* (sehat) sebanyak 435 gambar, *Rust* (karat daun) sebanyak 398 gambar, dan *red_spider_mite* (tungau merah) sebanyak 167 gambar. Meskipun kelas dan jenis data pada penelitian ini sama dengan data penelitian milik Windiawan & Suharso, 2019 namun dataset yang digunakan tidak sama. Karena pada penelitian yang telah dilakukan sebelumnya dataset yang digunakan berjumlah 1.539

data, sedangkan pada data yang digunakan pada penelitian ini berjumlah 1.000 data. Sampel data masing-masing kelas dapat dilihat pada Tabel 5.



Gambar 13. Tahapan Penelitian Klasifikasi Penyakit Daun Kopi.

Tabel 5. Sampel Data Masing-Masing Kelas

No	Nama Kelas	Sampel Data
1.	<i>healthy</i>	
2.	<i>Rust</i>	

Tabel 5. (lanjutan)

No	Nama Kelas	Sample Data
3.	<i>Red_spider_mite</i>	

3.3.2. *Preprocessing*

Preprocessing merupakan tahap krusial dalam pengolahan data, khususnya dalam penelitian berbasis *machine learning* dan *deep learning*. Tahap ini bertujuan untuk mempersiapkan data mentah agar berada dalam format yang optimal sebelum digunakan dalam proses pelatihan model. Data yang belum melalui *preprocessing* seringkali mengandung *noise*, duplikasi, ukuran yang tidak seragam, atau informasi yang tidak relevan, yang dapat menurunkan kinerja dan akurasi model. Oleh karena itu, *preprocessing* menjadi langkah fundamental untuk memastikan hasil analisis dan prediksi yang lebih akurat dan dapat diandalkan. Adapun tahapan-tahapan *preprocessing* tersebut diantaranya.

a. *Cleaning*

Cleaning merupakan langkah penting dalam tahap *preprocessing*, terutama untuk memastikan kualitas dan keakuratan *dataset* yang digunakan dalam pelatihan model. Dalam penelitian ini, proses *cleaning* dilakukan dengan cara menghapus gambar-gambar duplikat yang terdapat dalam *dataset*. Keberadaan duplikasi dapat menyebabkan bias pada model, karena informasi yang sama dipelajari berulang kali sehingga mengurangi kemampuan model dalam mengenali variasi data secara umum. Selain itu, duplikasi juga dapat memperbesar risiko *overfitting*, di mana model hanya

menghafal data tertentu tanpa benar-benar memahami pola yang mendasarinya. Oleh karena itu, penghapusan duplikasi gambar menjadi langkah penting untuk menjaga keberagaman data, meningkatkan efisiensi pelatihan, serta menghasilkan model yang lebih akurat.

b. Anotasi gambar

Tahap anotasi gambar merupakan proses penting dalam pembangunan *dataset* untuk keperluan pelatihan model berbasis *computer vision*, seperti klasifikasi atau deteksi objek. Anotasi dilakukan dengan memberikan label atau informasi tertentu pada setiap gambar yang menjadi acuan bagi model dalam mempelajari karakteristik masing-masing kelas. Dalam konteks penelitian ini, anotasi digunakan untuk menandai dan mengelompokkan gambar daun kopi berdasarkan jenis kondisi atau penyakitnya, seperti daun sehat, *Rust*, atau *red_spider_mite*. Tanpa anotasi yang akurat, model tidak akan memiliki referensi yang jelas dalam proses pembelajaran, sehingga dapat menyebabkan kesalahan klasifikasi dan penurunan performa. Oleh karena itu, tahap anotasi menjadi komponen krusial dalam memastikan kualitas dataset, serta mendukung proses pelatihan model agar mampu mengenali dan membedakan objek secara tepat.

c. *Cropping*

Tahap *cropping* merupakan proses pemotongan area tertentu dari gambar untuk memfokuskan model pada bagian yang relevan, dalam hal ini objek daun kopi menjadi fokus utama klasifikasi. Dalam penelitian ini, *cropping* dilakukan berdasarkan *bounding box* hasil anotasi guna menghilangkan bagian-bagian gambar yang tidak berkaitan seperti latar belakang atau objek lain di luar daun. Keberadaan elemen yang

tidak relevan dapat mengganggu proses pelatihan model dan menurunkan akurasi prediksi, karena model dapat terdistraksi oleh informasi yang tidak penting, dengan melakukan *cropping*, gambar yang digunakan menjadi lebih terfokus dan bersih, sehingga model dapat belajar lebih efektif terhadap fitur-fitur visual yang membedakan antar kelas. Oleh karena itu, tahap *cropping* berperan penting dalam meningkatkan kualitas *input* data dan mendukung kinerja model secara keseluruhan.

d. Segmentasi

Tahap segmentasi merupakan proses pemisahan objek utama dari latar belakang dalam sebuah gambar dengan tujuan untuk menyoroti area yang benar-benar relevan bagi analisis atau pelatihan model. Dalam konteks penelitian ini, segmentasi dilakukan untuk memisahkan daun kopi dari elemen-elemen lain seperti tanah, rumput, langit, atau latar belakang yang tidak diperlukan. Segmentasi sangat penting karena membantu model fokus hanya pada fitur-fitur visual yang terdapat pada daun, seperti warna, tekstur, atau pola bercak penyakit, tanpa terganggu oleh informasi luar objek. Dengan demikian segmentasi dapat meningkatkan akurasi model dalam mengenali dan membedakan kelas-kelas daun, serta mengurangi risiko kesalahan klasifikasi akibat latar belakang yang bervariasi. Oleh karena itu, tahap segmentasi menjadi bagian krusial dalam *preprocessing* gambar untuk menghasilkan *input* yang lebih bersih, terfokus, dan representatif.

e. *Padding*

Tahap *padding* merupakan proses penambahan ruang kosong di sekitar objek dalam gambar, biasanya dengan warna netral

seperti hitam atau putih, guna menyesuaikan proporsi gambar sebelum dilakukan *resizing*. Dalam penelitian ini, *padding* dilakukan sebelum proses pengubahan ukuran gambar atau *resizing* dengan tujuan untuk menjaga proporsi dan struktur asli daun kopi agar tidak terdistorsi. Tanpa *padding*, proses *resizing* dapat mengubah bentuk daun secara tidak proporsional, seperti merenggangkan atau merapatkan gambar, yang pada akhirnya dapat mempengaruhi kemampuan model dalam mengenali pola visual secara akurat. Dengan menambahkan *padding*, dimensi gambar dapat disesuaikan menjadi bentuk persegi tanpa mengubah aspek rasio objek utama. Oleh karena itu, tahap *padding* memiliki peran penting dalam menjaga keutuhan fitur visual daun, sehingga data yang digunakan tetap representatif dan mendukung performa model secara optimal.

f. *Resizing*

Tahap *resizing* merupakan proses mengubah ukuran gambar ke dimensi tertentu agar seragam dan sesuai dengan kebutuhan arsitektur model *deep learning* yang digunakan. Dalam penelitian ini, semua gambar diubah ukurannya menjadi 224×224 piksel, yang merupakan ukuran standar *input* untuk banyak arsitektur CNN, termasuk *SqueezeNet*. *Resizing* penting dilakukan untuk memastikan bahwa seluruh gambar dalam dataset memiliki dimensi yang konsisten, sehingga dapat diproses oleh model tanpa menimbulkan *error* atau ketidaksesuaian format. Selain itu, ukuran yang seragam juga membantu mempercepat proses pelatihan dan mengurangi beban komputasi. Dengan ukuran 224×224, model dapat tetap mempertahankan informasi visual yang cukup detail tanpa memerlukan kapasitas memori yang terlalu besar. Oleh karena itu, tahap *resizing* menjadi langkah krusial dalam

preprocessing untuk menjamin efisiensi, kompatibilitas, dan konsistensi data dalam pelatihan model.

3.3.3. *Splitting Data*

Tahap *splitting data* merupakan proses penting dalam pengembangan model *machine learning*, karena bertujuan untuk membagi dataset ke dalam beberapa bagian yang memiliki fungsi berbeda dalam proses pelatihan dan evaluasi. Umumnya, dataset dibagi menjadi tiga subset utama, yaitu data *training*, data *validation*, dan data *testing*. Pembagian ini penting untuk memastikan bahwa model tidak hanya mampu mengenali data yang telah dilatih, tetapi juga dapat diuji pada data yang belum pernah dilihat sebelumnya. Dengan demikian, *splitting data* berperan dalam mengukur kemampuan generalisasi model dan mencegah masalah *overfitting*, yaitu kondisi di mana model terlalu baik dalam menghafal data *training* tetapi gagal mengenali pola baru.

Dalam penelitian ini, dataset dibagi menggunakan rasio 70:20:10, di mana 70% data digunakan untuk *training*, 20% untuk *validation*, dan 10% sisanya untuk *testing*. Sebanyak 70% data *training* digunakan untuk melatih model agar dapat mempelajari pola, ciri, dan karakteristik dari masing-masing kelas dalam dataset. Kemudian, 20% data *validation* digunakan untuk memantau dan mengevaluasi kinerja model selama proses pelatihan, serta membantu dalam penyesuaian parameter guna mencegah *overfitting*. Sementara itu, 10% data *testing* digunakan setelah proses pelatihan selesai untuk mengukur performa akhir model terhadap data yang benar-benar baru dan belum pernah digunakan sebelumnya, sehingga dapat memberikan gambaran sejauh mana model mampu melakukan generalisasi. Rasio pembagian data

tersebut dipilih dengan merujuk pada penelitian oleh Dianto *et al.*, (2025) yang menyatakan bahwa proporsi 70:20:10 memberikan performa yang baik, khususnya pada model *EfficientNetB0* yang memiliki karakteristik ringan serupa dengan *SqueezeNet*.

3.3.4. Augmentasi Data

Augmentasi data merupakan salah satu teknik penting yang digunakan untuk memperluas dan memperkaya variasi data pelatihan tanpa harus mengumpulkan data tambahan secara manual. Teknik ini sangat bermanfaat terutama dalam menangani masalah *class imbalance*, yaitu kondisi di mana jumlah data pada setiap kelas tidak seimbang. Ketidakseimbangan ini dapat menyebabkan model cenderung lebih akurat dalam mengenali kelas dengan jumlah data yang lebih banyak, sementara mengabaikan kelas minoritas.

Dalam penelitian ini, augmentasi data diterapkan untuk menyeimbangkan jumlah gambar antar kelas, sehingga model dapat belajar secara adil terhadap semua kelas yang ada. Teknik augmentasi yang digunakan meliputi *flipping* secara horizontal dan vertikal dan rotasi sebesar 15° . Teknik augmentasi tersebut, termasuk derajat rotasi 15° ini dipilih karena merujuk pada penelitian yang dilakukan oleh Anggraini *et al.*, (2025). Dengan menerapkan variasi tersebut, dataset menjadi lebih representatif dan kaya akan pola visual, sehingga membantu meningkatkan kemampuan generalisasi model serta mengurangi risiko *overfitting*.

3.3.5. Modelling CNN dengan Arsitektur SqueezeNet

Modeling dalam konteks *deep learning* merupakan proses perancangan dan pengembangan arsitektur jaringan saraf tiruan

yang terdiri dari banyak lapisan dan dirancang untuk secara otomatis mempelajari serta memahami pola-pola kompleks dari data yang diberikan. Dalam proses ini, model dibangun dengan sejumlah *neuron* buatan yang saling terhubung, di mana setiap lapisan memiliki peran dalam memproses data secara bertahap. Lapisan awal akan mengekstraksi fitur dasar seperti tepi atau warna, sementara lapisan lebih dalam mengekstrak informasi yang lebih kompleks seperti bentuk objek atau pola tertentu dalam gambar atau teks. Tujuan dari *modeling* ini adalah agar sistem dapat mengenali, mengklasifikasikan, atau memprediksi data secara akurat tanpa perlu pemrograman eksplisit untuk setiap jenis pola. Proses *modeling* juga mencakup pemilihan arsitektur, fungsi aktivasi, optimasi parameter, serta evaluasi performa model agar sesuai dengan tujuan dari sistem yang dikembangkan. Pada penelitian kali ini memanfaatkan model *Convolutional Neural Network* (CNN) dengan arsitekturnya yaitu *SqueezeNet*. Lapisan-lapisan model dapat dilihat pada Tabel 6.

Tabel 6. Lapisan Model dengan Arsitektur *SqueezeNet*

<i>Layer</i>	<i>Type</i>	<i>Output Size (H×W×C)</i>	<i>Keterangan</i>
<i>Input</i>	<i>Image</i>	224×224×3	RGB Image
<i>Conv1</i>	64 <i>filters</i> , 3×3, <i>stride</i> =2	111×111×64	<i>Convolution</i> pertama
<i>MaxPool1</i>	3×3, <i>stride</i> =2	55×55×64	<i>Downsampling</i>
<i>Fire2</i>	<i>Squeeze</i> : 16 (1×1) <i>Expand</i> : 64 (1×1), 64 (3×3)	55×55×128	<i>Fire Module</i>
<i>Fire3</i>	<i>Squeeze</i> : 16 (1×1) <i>Expand</i> : 64 (1×1), 64 (3×3)	55×55×128	<i>Fire Module</i>
<i>MaxPool2</i>	3×3, <i>stride</i> =2	27×27×128	<i>Downsampling</i>

Tabel 6. (lanjutan)

<i>Layer</i>	<i>Type</i>	<i>Output Size (H×W×C)</i>	<i>Keterangan</i>
<i>Fire4</i>	<i>Squeeze: 32 (1×1) Expand: 128 (1×1), 128 (3×3)</i>	27×27×256	<i>Fire Module</i>
<i>Fire5</i>	<i>Squeeze: 32 (1×1) Expand: 128 (1×1), 128 (3×3)</i>	27×27×256	<i>Fire Module</i>
<i>MaxPool3</i>	3×3, <i>stride</i> =2	13×13×256	<i>Downsampling</i>
<i>Fire6</i>	<i>Squeeze: 48 (1×1) Expand: 192 (1×1), 192 (3×3)</i>	13×13×384	<i>Fire Module</i>
<i>Fire7</i>	<i>Squeeze: 48 (1×1) Expand: 192 (1×1), 192 (3×3)</i>	13×13×384	<i>Fire Module</i>
<i>Fire8</i>	<i>Squeeze: 64 (1×1) Expand: 256 (1×1), 256 (3×3)</i>	13×13×512	<i>Fire Module</i>
<i>Fire9</i>	<i>Squeeze: 64 (1×1) Expand: 256 (1×1), 256 (3×3)</i>	13×13×512	<i>Fire Module</i>
<i>DropOut</i>	0,3	13×13×512	Regulasi
<i>Conv10</i>	3 <i>filters</i> , 1×1	13×13×3	Klasifikasi 3 kelas
<i>AvgPool</i>	<i>Global Average Pooling (13×13)</i>	1×1×3	Nilai rata-rata

Model *SqueezeNet* menerima *input* berupa citra berukuran 224×224×3 (RGB). Citra ini pertama kali melewati lapisan *Convolutional* pertama (*Conv1*) dengan *kernel* berukuran 3×3, *stride* 2, dan 64 *filter*. Proses ini menghasilkan *feature map* dengan

ukuran $111 \times 111 \times 64$, yang menunjukkan bahwa tinggi dan lebar citra berkurang karena penggunaan *stride* 2. Selanjutnya, *feature map* tersebut diproses oleh *MaxPooling* berukuran 3×3 dengan *stride* 2, sehingga dimensinya semakin menyusut menjadi $55 \times 55 \times 64$.

Kemudian, data citra masuk ke *Fire Module* 2 dan *Fire Module* 3, yang merupakan blok utama *SqueezeNet*. Setiap *fire module* terdiri dari satu lapisan *squeeze* (1×1 , *stride* = 1) yang mengurangi jumlah *channel*, kemudian dua jalur *expand* (1×1 , *stride* = 1 dan 3×3 , *stride* = 1, *padding* = 1) yang memperluas kembali representasi fitur. Setelah *Fire2* dan *Fire4*, ukuran spasial tetap 55×55 , hanya jumlah *channel* yang berubah menjadi 128. Selanjutnya dilakukan *MaxPooling* kembali (3×3 , *stride* 2), yang mereduksi dimensi spasial menjadi $27 \times 27 \times 128$.

Feature map ini kemudian diproses melalui *Fire Module* 4 dan *Fire Module* 5, yang mempertahankan ukuran spasial 27×27 tetapi meningkatkan kedalaman *channel* menjadi 256. Setelah *Fire5*, kembali dilakukan *MaxPooling*, menghasilkan ukuran $13 \times 13 \times 256$. Pada tahap ini, jaringan sudah sangat mengompresi informasi spasial sambil memperkaya representasi fitur.

Berikutnya, *feature map* melalui serangkaian *Fire Module* lagi yaitu *Fire6*, *Fire7*, *Fire8*, dan *Fire9*. Setelah *Fire9*, ukuran spasial tetap 13×13 , dengan kedalaman *channel* mencapai 512. Sesudah *Fire9*, ditambahkan *DropOut* sebesar 0,3. Penambahan ini merupakan modifikasi dari arsitektur *SqueezeNet* 1.1 yang mana pada arsitektur tersebut *DropOut* tidak didefinisikan secara eksplisit namun arsitektur tersebut memuat nilai *DropOut* secara *default* yaitu sebesar 0,5.

Setelah itu, *feature map* diproses oleh *Conv10* dengan 3 *filter*, sesuai dengan jumlah kelas yang dituju. Hasilnya adalah *output* berukuran $13 \times 13 \times 3$. Untuk mengubah menjadi prediksi kelas, dilakukan *Global Average Pooling* yang menghitung rata-rata nilai tiap *channel* sehingga menghasilkan vektor $1 \times 1 \times 3$.

3.3.6. Testing Model

Tahap *testing* model merupakan langkah dalam proses pengembangan model *machine learning* yang berfungsi untuk mengevaluasi performa model terhadap data yang benar-benar belum pernah dilihat sebelumnya. Tidak seperti data *training* dan *validation* yang digunakan selama proses pelatihan dan penyetelan parameter, data *testing* digunakan secara eksklusif untuk mengukur kemampuan generalisasi model. Dengan demikian, hasil evaluasi pada tahap ini memberikan gambaran sejauh mana model mampu mengenali pola dan membuat prediksi yang akurat pada data baru di dunia nyata.

Dalam penelitian ini, gambar-gambar yang salah diklasifikasikan oleh model ditampilkan sebagai bahan evaluasi lebih lanjut untuk memahami kelemahan model, seperti kemungkinan kemiripan visual antar kelas atau kualitas gambar yang buruk. Analisis terhadap hasil prediksi yang keliru ini sangat penting karena dapat memberikan wawasan mendalam mengenai keterbatasan model dan menjadi dasar untuk perbaikan pada tahap pengembangan selanjutnya.

3.3.7. Evaluasi

Evaluasi model merupakan tahap penting dalam penelitian berbasis *machine learning* karena digunakan untuk menilai sejauh mana

model yang telah dilatih mampu memberikan prediksi yang akurat dan andal. Evaluasi dilakukan untuk mengetahui performa model dalam mengklasifikasikan data secara benar, serta mengidentifikasi adanya kesalahan atau kelemahan dalam proses prediksi. Tanpa evaluasi yang menyeluruh, sulit untuk memastikan apakah model benar-benar memahami pola dalam data atau hanya sekadar menghafal data latih.

Dalam penelitian ini, evaluasi model dilakukan dengan menggunakan *confusion matrix*. *Confusion matrix* memberikan gambaran detail mengenai jumlah prediksi yang benar dan salah untuk setiap kelas, sehingga memungkinkan analisis kinerja model secara lebih mendalam. Dari metrik ini dapat dilihat berapa banyak data yang diklasifikasikan dengan benar (*True Positif* dan *True Negatif*), serta jenis kesalahan yang terjadi (*False Positif* dan *False Negatif*). Informasi ini sangat berguna untuk memahami seberapa baik model membedakan antar kelas, serta untuk mengidentifikasi pola kesalahan yang mungkin muncul akibat kemiripan antar kelas atau kualitas data. Dengan demikian, *confusion matrix* menjadi alat evaluasi yang penting dalam menilai akurasi dan ketepatan model secara menyeluruh.

V. SIMPULAN DAN SARAN

5.1. Simpulan

Berdasarkan hasil penelitian, implementasi metode *Convolutional Neural Network* (CNN) dengan arsitektur *SqueezeNet* berhasil diterapkan untuk mengklasifikasikan daun kopi menjadi tiga kelas, yaitu *Rust*, *red_spider_mite*, dan daun sehat, dengan akurasi keseluruhan sebesar 93,68%. Kinerja model menunjukkan variasi, di mana kelas *healthy* memperoleh hasil terbaik dengan *F1-Score* 1,000, diikuti oleh *Rust* dengan nilai *F1-Score* sebesar 0,9231 dan nilai *F1-Score* untuk kelas *red_spider_mite* sebesar 0,8000. Perbedaan performa ini dipengaruhi oleh kemiripan visual antara *Rust* dan *red_spider_mite* yang sama-sama menampilkan diskolorasi kuning kecoklatan, sehingga sering menimbulkan kebingungan pada model. Meskipun demikian, model yang dikembangkan telah memenuhi tujuan penelitian dengan memberikan solusi klasifikasi penyakit daun kopi yang akurat.

5.2. Saran

Adapun saran yang diberikan untuk dapat diterapkan pada penelitian selanjutnya adalah sebagai berikut.

- a. Disarankan untuk menambah jumlah dataset yang lebih seimbang antar kelas. Ketidakseimbangan data pada suatu kelas mengakibatkan performa model kurang optimal. Penambahan dataset yang seimbang diharapkan dapat meningkatkan konsistensi performa model.

- b. Penelitian selanjutnya disarankan menambah kelas penyakit lain yang kerap menyerang daun kopi. hal ini akan membuat model lebih komprehensif dan aplikatif untuk kondisi lapangan yang sesungguhnya.
- c. Disarankan untuk menerapkan pendekatan *multi-label classification* karena dalam kondisi nyata, satu daun dapat terinfeksi lebih dari satu penyakit secara bersamaan. Implementasi ini akan memberikan informasi yang lebih akurat untuk penanganan penyakit yang tepat.

DAFTAR PUSTAKA

- Ahadin, A. I., Hana, F. M., Prihandono, A., & Prayogo Pujiono, I. (2024). Pengembangan Model Klasifikasi Produk Furniture Sebagai Visual Search Menggunakan Algoritma Convolutional Neural Network. *Jurnal SISKOM-KB (Sistem Komputer Dan Kecerdasan Buatan)*, 8(1), 43–51. <https://doi.org/10.47970/siskom-kb.v8i1.707>
- Alvian, V., Hidayatullah, D., Nilogiri, A., Azizah, H., & Faruq, A. (2021). Klasifikasi Siswa Berprestasi Menggunakan Metode K-Nearest Neighbor (KNN) Pada SMA Negeri 2 Situbondo Classification Of Achieving Students Using K-Nearest Neighbor (KNN) Method At SMA Negeri 2 Situbondo. *Jurnal Smart Teknologi*, 1(1), 2774–1702. <http://jurnal.unmuhjember.ac.id/index.php/JST>
- Anggraini, D., Hasibuan, M. S., & Siwi, S. P. (2025). Analisis Citra Medis untuk Mendeteksi Diabetes Menggunakan Metode CNN (Convolutiona Neural Network). 9, 5064–5072.
- Anggraini, R. (2024). Analisis Organoleptik Seduhan Varietas Kopi Lokal Dengan Teknik Manual Brew. *Jurnal Pertanian Dan Pangan*, 6(2), 1–8.
- Anjanadevi, B., Nagesh, V., T., S., & Varma, P. N. M. (2023). An Efficient Model for Forest Fire Detection using Deep Convolutional Neural Networks. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(10s), 177–181. <https://doi.org/10.17762/ijritcc.v11i10s.7617>
- Anwari, N. D., Arifianto, A., & Jondri, J. (2018). Multilabel Image Annotation Menggunakan Metode Convolutional Neural Network. *EProceedings of*

Engineering, 5(1), 1597–1605.

Ariyani, V., Putri, P., Prasetyo, A. B., & Eridani, D. (2022). Perbandingan Kinerja Algoritme Naïve Bayes Dan K-Nearest Neighbor (Knn) Untuk Prediksi Harga Rumah. *Jurnal Ilmiah Teknik Elektro*, 24(2), 162–171. <https://ejournal.undip.ac.id/index.php/transmisi>

Armiady, D., & Muslem, I. R. (2023). Klasifikasi Kualitas Buah Pisang Berdasarkan Citra Buah Menggunakan Stochastic Gradient Descent. *KLIK: Kajian Ilmiah Informatika Dan Komputer*. <http://djournals.com/klik/article/view/1243>

Asaad, R. R., Ismael Ali, R., Arif Ali, Z., & Ahmad Shaaban, A. (2023). Image Processing with Python Libraries. *Academic Journal of Nawroz University*, 12(2), 410–416. <https://doi.org/10.25007/ajnu.v12n2a1754>

Auliaddina, S., & Arifin, T. (2024). Penggunaan Data Augmentasi dan Hyperparameter Tuning dalam Klasifikasi Jenis Batik menggunakan Model CNN. *Januari*, 13(1), 114–128. <http://sistemasi.ftik.unisi.ac.id>

Bachri, C. M., & Gunawan, W. (2024). Deteksi Email Spam menggunakan Algoritma Convolutional Neural Network (CNN). *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 10(1), 88. <https://doi.org/10.26418/jp.v10i1.73306>

Badan Pusat Statistik. (2023). *Statistik Kopi Indonesia 2023*. Badan Pusat Statistik. <https://www.bps.go.id/id/publication/2024/11/29/d748d9bf594118fe112fc51e/statistik-kopi-indonesia-2023.html> "Diakses pada 16-02-2025"

Caron, J., & Markusen, J. R. (2023). Hama dan Penyakit Kopi Arabika (*Coffea Arabika*) di HKM Solok Radjo, Air dingin, Kecamatan Lembah Gumanti, Kabupaten Solk, Provinsi Sumatera Barat. *SYLVA: Jurnal Penelitian Ilmu-Ilmu Kehutanan*, 12(1), 1–23.

- Deng, W., & Wu, R. (2019). Real-Time Driver-Drowsiness Detection System Using Facial Features. *IEEE Access*, 7, 118727–118738. <https://doi.org/10.1109/ACCESS.2019.2936663>
- Dianto, R. A., Anggraeny, T. F., & Maulana, H. (2025). *Analisis Efektifitas Algoritma Mobilenetv3- Large Dan Efficientnet-B0 Untuk Klasifikasi Citra Penyakit Daun Jeruk*. 13(3).
- Dinas Pertanian dan Pangan Kabupaten Bandung (2018). *Mengenal Tanaman Kopi Liberica*. Dinas Pertanian Dan Pangan Kabupaten Bandung. <https://diperpa.badungkab.go.id/Artikel/18070-mengenal-tanaman-kopi-liberica> "Diakses pada 16-02-2025"
- Fahmi, M. N. (2023). Implementasi Mechine Learning menggunakan Python Library : Scikit-Learn (Supervised dan Unsupervised Learning). *Sains Data Jurnal Studi Matematika Dan Teknologi*, 1(2), 87–96. <https://doi.org/10.52620/sainsdata.v1i2.31>
- Fauzan, M., Ilhami, A., & Supriyanto, A. (2023). Deteksi Wajah Jenis Kelamin Dengan Fitur Hijab Dan Tidak Berhijab Menggunakan Jaringan Saraf Konvolusi Gender Face Detection With Hijab and Non-Hijab Features Using Convolutional Neural Networks. *Journal of Information Technology and Computer Science (INTECOMS)*, 6(2).
- Firmansyach, W. A., Hayati, U., & Arie Wijaya, Y. (2023). Analisa Terjadinya Overfitting Dan Underfitting Pada Algoritma Naive Bayes Dan Decision Tree Dengan Teknik Cross Validation. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1), 262–269. <https://doi.org/10.36040/jati.v7i1.6329>
- Fitrah, T. R. M., Nurdin, Y., & Roslidar. (2021). Rancangn Bangun Pengembangan Pintu Otomatis Pendeteksi Masker dan Suhu Tubuh Menggunakan Raspberry Pi 4. *KITEKTRO: Jurnal Komputer, Informasi Teknologi Dan Elektro*, 6(2), 7–14.
- Fitrianingsih, & Zuraeni, B. (2024). Analisis Ramalan Cuaca di Sekupang, Kota Batam Menggunakan Algoritma Decision Tree dan Confusion Matrix. *Jurnal*

- Ekonomi Pembangunan Dan Manajemen*, 1(3), 15–26.
<https://ibnusinapublisher.org/index.php/EKOSPHERE>
- Gnecco, G., & Sanguineti, M. (2009). The weight-decay technique in learning from data: An optimization point of view. *Computational Management Science*, 6(1), 53–79. <https://doi.org/10.1007/s10287-008-0072-5>
- Harahap, H., Rahman, S., & Zen, M. (2024). Analisis Existing Convolutional Neural Network Untuk Klasifikasi Usia Pengunjung Rumah Sakit: Studi Kasus Pemantauan Anak dan Dewasa. *Journal of Computer Science and Information Technology E-ISSN*, 4(1), 16.
- Harni, R., Taufiq, E., & Martono, B. (2015). Ketahanan Pohon Induk Kopi Liberika terhadap Penyakit Karat Daun (*Hemileia vastatrix* B. Et Br.) di Kepulauan Meranti. *Jurnal Tanaman Industri Dan Penyegar*, 2(1), 35. <https://doi.org/10.21082/jtidp.v2n1.2015.p35-42>
- Hendri, H., Hoki, L., Agusman, V., & Aryanto, D. (2021). Penerapan Machine Learning Untuk Mengategorikan Sampah Plastik Rumah Tangga. *Jurnal TIMES*, 10(1), 1–5. <https://doi.org/10.51351/jtm.10.1.2021645>
- Hermanto, K., Salim, D., Wu, B., Salim, O. R., & Gunadi, R. B. (2022). *Penggunaan Python Untuk Menganalisis Pola Penyebaran Covid-19 Di Masa Pandemi*. 2, 120–133.
- Hermawan, A., Suwitno, Junaedi, Kurniawan Maranto, A. R., Sonya Ayu Kumala, Tia Nurapriyanti, & Johan Santoso. (2024). Meningkatkan Pemahaman Pemrograman melalui Pelatihan Python. *Abdi Dharma*, 4(1), 11–20. <https://doi.org/10.31253/ad.v4i1.2806>
- Husna, I. N., Ulum, M., Kurniawan Saputro, A., Tri Laksono, D., & Neipa Purnamasari, D. (2022). Rancang Bangun Sistem Deteksi Dan Perhitungan Jumlah Orang Menggunakan Metode Convolutional Neural Network (CNN). *Seminar Nasional Fortei Regional*, 7, 1–6.

- International Coffee Organization* (2025). *Coffee Market Report January 2025*. International Coffee Organization. <https://www.ico.org/documents/cy2024-25/cmr-0125-e.pdf> "Diakses pada 20-02-2025"
- Jinan, A., Hayadi, B. H., & Utama, U. P. (2022). Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Convolutional Neural Network Melalui Citra Daun (Multilayer Perceptron). *Journal of Computer and Engineering Science*, 1(2), 37–44.
- Kaggle. (2020). *Coffee Plant Disease*. Kaggle. <https://www.kaggle.com/datasets/coffeedisease/coffee-plant-disease> "Diakses pada 12-11-2024"
- Kumaseh, M. R., Latumakulita, L., & Nainggolan, N. (2013). Segmentasi Citra Digital Ikan Menggunakan Metode Thresholding. *Jurnal Ilmiah Sains*, 13(1), 74. <https://doi.org/10.35799/jis.13.1.2013.2057>
- Lin, M., Chen, Q., & Yan, S. (2014). Network in network. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 1–10.
- Magdalena, R., Saidah, S., Pratiwi Caecar, N. K., & Putra Trisnamulya, A. (2021). Klasifikasi Tutupan Lahan Melalui Citra Satelit SPOT-6 dengan Metode Convolutional Neural Network (CNN). *Jurnal Edukasi Dan Penelitian Informatika*, 7(3), 335–339.
- Musu, W., Ibrahim, A., & Heriadi. (2021). Pengaruh Komposisi Data Training dan Testing terhadap Akurasi Algoritma C4.5. *Prosiding Seminar Ilmiah Sistem Informasi Dan Teknologi Informasi*, X(1), 186–195.
- Mutmainah, M., Jusriadi, J., Jeki, J., & Putra, K. (2023). Ketahanan Beberapa Varietas Kopi (*Coffea* sp) Terhadap Penyakit Karat Daun (*Hemileia vastatrix*) Di Kabupaten Sigi. *Fruitset Sains: Jurnal Pertanian Agroteknologi*, 11(5), 357–363.

- Nurdin, A., Satria, D., Kartika, Y., Rezha, A., & Najaf, E. (2024). *Klasifikasi Penyakit Daun Tomat Dengan Metode Convolutional Neural Network Menggunakan Arsitektur Inception-V3. 1*, 1–6.
- Paraijun, F., Aziza, R. N., & Kuswardani, D. (2022). Implementasi Algoritma Convolutional Neural Network Dalam Mengklasifikasi Kesegaran Buah Berdasarkan Citra Buah. *Kilat*, 11(1), 1–9. <https://doi.org/10.33322/kilat.v10i2.1458>
- Purbolaksono, M. D., Irvan Tantowi, M., Imam Hidayat, A., & Adiwijaya, A. (2021). Perbandingan Support Vector Machine dan Modified Balanced Random Forest dalam Deteksi Pasien Penyakit Diabetes. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(2), 393–399. <https://doi.org/10.29207/resti.v5i2.3008>
- Putra, I. N. T. A., Kartini, K. S., Suyitno, Y. K., Sugiarta, I. M., & Puspita, N. K. E. (2023). Penerapan Library Tensorflow, Cvzone, dan Numpy pada Sistem Deteksi Bahasa Isyarat Secara Real Time. *Jurnal Krisnadana*, 2(3), 412–423. <https://doi.org/10.58982/krisnadana.v2i3.335>
- Rachmat, R., Yusuf, M., Abbas, I., & Basmar, M. F. (2022). Segmentasi Citra Menggunakan Algoritma K-Means Berbasis Particle Swarm Optimization. *Metik Jurnal*, 6(2), 146–156. <https://doi.org/10.47002/metik.v6i2.377>
- Raihan, R. M., & Yulianto, S. (2023). Penerapan Pemrograman Python Dalam Menentukan Waktu Overhoul Kondensor Turbin Uap. *Jurnal Konversi Energi Dan Manufaktur*, 8(1), 49–57. <https://doi.org/10.21009/jkem.8.1.6>
- Rasidi, A. I., Pasariru, Y. A. H., Ziqri, A., & Adhinata, F. D. (2022). Klasifikasi Sampah Organik dan Non-Organik Menggunakan Convolutional Neural Network. *Jurnal Teknik Informatika Dan Sistem Informasi*, 8(1), 142–149. <https://doi.org/10.28932/jutisi.v8i1.4314>
- Rismiyati, R., & Luthfiarta, A. (2021). VGG16 Transfer Learning Architecture for Salak Fruit Quality Classification. *Telematika*, 18(1), 37. <https://doi.org/10.31315/telematika.v18i1.4025>

- Rochmawati, N., Hidayati, H. B., Yamasari, Y., Tjahyaningtijas, H. P. A., Yustanti, W., & Prihanto, A. (2021). Analisa Learning Rate dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam. *Journal of Information Engineering and Educational Technology*, 5(2), 44–48. <https://doi.org/10.26740/jieet.v5n2.p44-48>
- Roihan, A., Sunarya, P. A., & Rafika, A. S. (2020). Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 5(1), 75–82. <https://doi.org/10.31294/ijcit.v5i1.7951>
- Sadewa, B. A., & Yamasari, Y. (2024). Implementasi Deep Transfer Learning untuk Klasifikasi Nominal Uang Kertas Rupiah. *Journal of Informatics and Computer Science (JINACS)*, 5(04), 543–551. <https://doi.org/10.26740/jinacs.v5n04.p543-551>
- Saepudin, S., Sujana, N., Mutoffar, M. M., & Haryanto, A. A. (2024). Analisis Kinerja Yolov8 Optimalisasi Roboflow Untuk Deteksi Ekspresi Wajah Emosional Dengan Machine Learning. *Naratif: Jurnal Nasional Riset, Aplikasi Dan Teknik Informatika*, 6(2), 115–124. <https://doi.org/10.53580/naratif.v6i2.292>
- Santoso, S., & Astuti, W. (2019). Ketahanan Empat Kultivar Ubi Kayu Terhadap *Tetranychus kanzawai* Kishida (Acari: Tetranychidae). *Agrovigor: Jurnal Agroekoteknologi*, 12(2), 87–93. <https://doi.org/10.21107/agrovigor.v12i2.5741>
- Santoso, S., Rauf, A., Gultom, N. M., Karmawati, E., & Rumini, W. (2014). Biologi dan kelimpahan tungau merah *Tetranychus* sp. (Acari: Tetranychidae) pada dua kultivar jarak pagar (*Jatropha curcas*). *Jurnal Entomologi Indonesia*, 11(1), 34–42. <https://doi.org/10.5994/jei.11.1.34>
- Sari, I. P., Ramadhani, F., Satria, A., & Apdilah, D. (2023). Implementasi Pengolahan Citra Digital dalam Pengenalan Wajah menggunakan Algoritma PCA dan Viola Jones. *Hello World Jurnal Ilmu Komputer*, 2(3), 146–157.

<https://doi.org/10.56211/helloworld.v2i3.346>

Sari, I. R. (2020). Implementasi Convolutional Neural Networks (Cnn) Untuk Klasifikasi Citra Benih Kacang Hijau Berkualitas. *Engineering, Construction and Architectural Management*, 25(1), 1–9.

ScienceDirect. (2021). *Training and Assessing a Model*. ScienceDirect. <https://www.sciencedirect.com/topics/computer-science/training-data> "Diakses pada 03-02-2025"

Supriyanto, A., R. Rizal Isnanto, & Oky Dwi Nurhayati. (2023). Klasifikasi Penyakit Daun Kopi Robusta Menggunakan Metode SVM dengan Ekstraksi Ciri GLCM. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 12(4), 241–248. <https://doi.org/10.22146/jnteti.v12i4.8044>

Telkom University. (2025). *Belajar ML? Yuk, Kenali Parameter dan Hyperparameter!* Telkom University Surabaya. <https://bds-sby.telkomuniversity.ac.id/belajar-ml-yuk-kenali-parameter-dan-hyperparameter/> "Diakses pada 26-06-2025"

Telus Digital. (2025). *Data validation*. Telus Digital. <https://www.telusdigital.com/glossary/data-validation> "Diakses pada 03-02-2025"

Theerthagiri, P., Ruby, A. U., Chandran, J. G. C., Sardar, T. H., & Shafeeq B. M, A. (2024). Deep SqueezeNet learning model for diagnosis and prediction of maize leaf diseases. *Journal of Big Data*, 11(1). <https://doi.org/10.1186/s40537-024-00972-z>

Toyib, M., Decky, T., & Pratama, K. (2024). Penerapan Algoritma CNN Untuk Mendeteksi Tulisan Tangan AngkaRomawi dengan Augmentasi Data. *Urnal Matematika, Ilmu Pengetahuan Alam, Kebumian Dan Angkasa*, 2(3), 108–120. <https://doi.org/10.62383/algoritma.v2i3.69>

Wardani, P. D. (2017). Klasifikasi Tipe Kematangan Pisang Menggunakan Metode Ensemble Convolutional Neural Network (CNN). *Jurnal*

- Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 8(9), 1–10.
<https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/14157>
- Wasil, M., Harianto, H., & Fathurrahman, F. (2022). Pengaruh Epoch pada Akurasi menggunakan Convolutional Neural Network untuk Klasifikasi fashion dan Furniture. *Infotek: Jurnal Informatika Dan Teknologi*, 5(1), 53–61. <https://doi.org/10.29408/jit.v5i1.4393>
- Widiyani, D. P., & Hartono, J. S. S. (2021). Studi Eksplorasi Agroklimat Tanaman Kopi Robusta (*Coffea canephora*) Kabupaten Tanggamus, Lampung. *Jurnal Agrinika : Jurnal Agroteknologi Dan Agribisnis*, 5(1), 20. <https://doi.org/10.30737/agrinika.v5i1.1523>
- Wijayanto, M. F., Swanjaya, D., & Wulanningrum, R. (2024). Penerapan MobileNet Architecture pada Identifikasi Foto Citra Makanan Indonesia. *Digital Transformation Technology*, 4(1), 652–662. <https://doi.org/10.47709/digitech.v4i1.4449>
- Wijoyo, A., Saputra, A. Y., Ristanti, S., Sya'ban, S. R., Amalia, M., & Febriansyah, R. (2024). Pembelajaran Machine Learning. *OKTAL (Jurnal Ilmu Komputer Dan Science)*, 3(2), 375–380. <https://journal.mediapublikasi.id/index.php/oktal/article/view/2305>
- Windiawan, R., & Suharso, A. (2019). Identifikasi Penyakit pada Daun Kopi Menggunakan Metode Deep Learning VGG16. *Exploreit*, 13(2), 9–16. <https://doi.org/10.35891/explorit>
- Yan, A., & Wiliani, N. (2025). Analisis Pengaruh Dropout Layer pada Convolutional Neural Networks untuk Klasifikasi Gambar Sticky notes. 13(3), 707–714.
- Yann, L., & Yoshua, B. (1995). Convolutional Networks for Images, Speech, and Time-Series. *The Handbook of Brain Theory and Neural Networks*, 255–258. https://www.researchgate.net/profile/Yann_Lecun/publication/2453996_Convolutional_Networks_for_Images_Speech_and_Time-Series/links/0deec519dfa2325502000000.pdf%0Ahttps://www.researchgate.

net/profile/Yann_Lecun/publication/2453996_Convolutional_Networks_fo

Yuhandri. (2019). Perbandingan Metode Cropping pada Sebuah Citra untuk Pengambilan Motif Tertentu pada Kain Songket Sumatera Barat. *Jurnal KomtekInfo*, 6(1), 97–107. <https://doi.org/10.35134/komtekinfo.v6i1.45>