

**KLASIFIKASI PENYAKIT DAUN TEH MENGGUNAKAN CNN
ARSITEKTUR RESNET-50 DAN RESNET-101**

(Skripsi)

**Oleh
SISKA HERMAYANTI
NPM 2117051023**



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

ABSTRAK

KLASIFIKASI PENYAKIT DAUN TEH MENGGUNAKAN CNN ARSITEKTUR RESNET-50 DAN RESNET-101

Oleh

SISKA HERMAYANTI

Daun teh merupakan salah satu komoditas perkebunan yang berpengaruh dalam perekonomian di Indonesia, namun produktivitasnya menurun karena serangan penyakit pada daun teh. Beberapa jenis penyakit pada daun teh seperti *Algal Leaf Spot*, *Anthrachnose*, *Bird Eye Spot*, *Brown Blight*, dan *White Leaf Spot*. Oleh karena itu, diperlukan pemanfaatan teknologi untuk membantu klasifikasi penyakit daun teh secara cepat. Penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) dengan dua arsitektur ResNet-50 dan ResNet-101 yang dikenal bagus dalam klasifikasi gambar. Dataset penelitian diperoleh dari platform Kaggle yang sudah dilabel dan akan digunakan lima kelas penyakit daun teh. Tahapan penelitian meliputi *data preprocessing*, pembagian dataset, pelatihan model, serta evaluasi kinerja menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*. Percobaan dilakukan dengan delapan skenario yang memvariasikan rasio pembagian data, ukuran *batch size*, serta penggunaan data dengan maupun tanpa augmentasi. Hasil percobaan menunjukkan bahwa ResNet-50 memperoleh akurasi tertinggi sebesar 93,33% pada skenario keempat, sedangkan ResNet-101 mencapai akurasi tertinggi dengan nilai yang sama pada skenario pertama dan kedua. Secara keseluruhan, ResNet-101 menunjukkan konsistensi performa yang lebih baik dibandingkan ResNet-50, meskipun kedua arsitektur terbukti efektif dalam mengklasifikasikan penyakit daun teh.

Kata Kunci: *Convolutional Neural Network* (CNN), klasifikasi citra, penyakit daun teh, ResNet-50, ResNet-101.

ABSTRACT

CLASSIFICATION OF TEA LEAF DISEASES USING CNN RESNET-50 AND RESNET-101 ARCHITECTURES

By

SISKA HERMAYANTI

Tea leaves are one of the plantation commodities that have a significant impact on the Indonesian economy, but their productivity has declined due to disease attacks on tea leaves. Several types of diseases affecting tea leaves include Algal Leaf Spot, Anthracnose, Bird Eye Spot, Brown Blight, and White Leaf Spot. Therefore, the use of technology is needed to help classify tea leaf diseases quickly. This study employs the Convolutional Neural Network (CNN) method with two ResNet-50 and ResNet-101 architectures, which are known to be effective in image classification. The research dataset was obtained from the Kaggle platform, which is already labeled, and will be used for five classes of tea leaf diseases. The research stages include data preprocessing, dataset division, model training, and performance evaluation using accuracy, precision, recall, and F1-score metrics. The experiments were conducted across eight scenarios that varied the data splitting ratio, batch size, and the use of data with or without augmentation. The results show that ResNet-50 achieved the highest accuracy of 93.33% in the fourth scenario, while ResNet-101 reached the same highest accuracy in the first and second scenarios. Overall, ResNet-101 demonstrated better performance consistency compared to ResNet-50, although both architectures proved effective in classifying tea leaf diseases.

Keywords: *Convolutional Neural Network (CNN), image classification, ResNet-50, ResNet-101, tea leaf disease.*

**KLASIFIKASI PENYAKIT DAUN TEH MENGGUNAKAN CNN
ARSITEKTUR RESNET-50 DAN RESNET-101**

Oleh

SISKA HERMAYANTI

Skripsi

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

Pada

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



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

Judul Skripsi : **KLASIFIKASI PENYAKIT DAUN TEH
MENGUNAKAN CNN ARSITEKTUR
RESNET-50 DAN RESNET-101**

Nama Mahasiswa : **Siska Hermayanti**

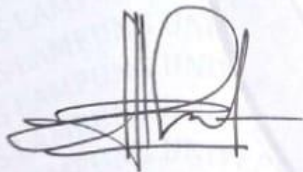
Nomor Pokok Mahasiswa : 2117051023

Program Studi : Ilmu Komputer (S1)

Fakultas : Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing



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

NIP. 197101291997021001



Ridho Sholehurrohman, M. Mat.

NIK. 232111970128101

2. Mengetahui

Ketua Jurusan Ilmu Komputer

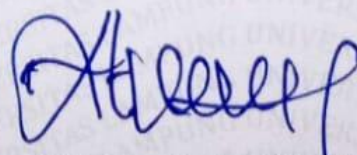


Dwi Sakethi, S.Si., M.Kom.

NIP. 196806111998021001

Ketua Program Studi

S1 Ilmu Komputer



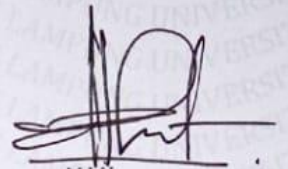
Tristiyanto, M.I.S., Ph.D.

NIP. 198104142005011001

MENGESAHKAN

1. Tim Penguji

Ketua : **Dr. rer. nat. Akmal Junaidi, M.Sc.**



Sekretaris : **Ridho Sholehurrohman, M. Mat.**



Penguji

Bukan Pembimbing : **Favorisen R. Lumbanraja, Ph.D.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



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

NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: 31 Juli 2025

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Siska Hermayanti

NPM : 2117051023

Menyatakan bahwa skripsi saya yang berjudul **“Klasifikasi Penyakit Daun Teh Menggunakan CNN Arsitektur Resnet-50 dan Resnet-101”** merupakan karya saya sendiri dan bukan karya orang lain. Seluruh tulisan yang tertulis dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 26 Agustus 2025

Yang menyatakan



Siska Hermayanti
NPM. 2117051023

RIWAYAT HIDUP



Penulis bernama lengkap Siska Hermayanti, dilahirkan pada tanggal 25 Maret 2003 di OKU Timur Provinsi Sumatra Selatan sebagai anak kedua dari dua bersaudara dari pasangan Bapak Slamet dan Ibu Maimunah. Penulis menempuh pendidikan di Taman Kanak-Kanak (TK) Miftahul Huda pada Tahun 2008 dan melanjutkan pendidikan di SD Negeri 1 Panca Tunggal yang diselesaikan pada Tahun 2015. Kemudian melanjutkan pendidikan menengah pertama di SMP Negeri 1 Belitang Jaya yang diselesaikan pada Tahun 2018 dan pendidikan menengah atas di SMA Negeri 1 Belitang yang diselesaikan pada Tahun 2021.

Pada Tahun 2021 penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SNMPTN. Selama menjadi mahasiswa, penulis melakukan beberapa kegiatan antara lain:

1. Melaksanakan Kerja Praktik di Badan Kepegawaian Daerah Provinsi Lampung pada periode I Tahun 2023.
2. Melaksanakan KKN di Desa Bandar Agung, Kecamatan Bandar Sribhawono, Kabupaten Lampung Timur pada periode II Tahun 2024.
3. Menjadi Asisten Dosen Jurusan Ilmu Komputer Mata Kuliah Sistem Pakar pada Tahun Ajaran 2024/2025.

PERSEMBAHAN

Alhamdulillahirobbil'alamin

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

Kupersembahkan karya ini kepada:

Kedua Orang Tua dan Kakak Tercinta

Yang selalu mendoakan dan selalu berusaha memberikan yang terbaik. Terima kasih atas kasih sayang yang tulus, dukungan yang tak pernah surut, pengorbanan, dan kepercayaan yang menjadi kekuatan dalam menyelesaikan pendidikan ini.

Seluruh Keluarga Besar Ilmu Komputer 2021

Yang senantiasa memberikan semangat dan dukungan.

Almamater Tercinta, Universitas Lampung dan Jurusan Ilmu Komputer

Tempat bertumbuh yang telah memberikan ilmu, pengalaman, dan bekal hidup.

SANWACANA

Puji syukur kehadiran Allah Subhanahu Wa Ta'ala atas segala rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi ini dengan baik dan lancar. Shalawat serta salam senantiasa tercurah kepada Nabi besar Muhammad Shallallahu 'Alaihi Wasallam, yang telah membawa umat manusia menuju jalan kebenaran dan menjadi teladan bagi seluruh kehidupan. Skripsi berjudul **“Klasifikasi Penyakit Daun Teh Menggunakan CNN Arsitektur Resnet-50 Dan Resnet-101”** telah disusun dengan sebaik-baiknya dan sebagai salah satu syarat untuk mendapat gelar sarjana Ilmu Komputer di Universitas Lampung.

Penulis menyadari bahwa dalam proses penyusunan skripsi ini tidak terlepas dari dukungan, bimbingan, serta bantuan dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak, Ibu, kakak, dan keluarga yang selalu memberikan doa terbaik, kasih sayang, dukungan, dan selalu memberikan semangat baik secara moral ataupun material dalam menyelesaikan pendidikan dan skripsi ini.
2. Ibu Prof. Dr. Ir. Lusmelia Afriani, D.E.A., I.P.M. ASEAN Eng. selaku Rektor Universitas Lampung.
3. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
4. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
5. Ibu Yunda Heningtyas, M.Kom. selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
6. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D. selaku Ketua Program Studi Ilmu Komputer FMIPA Universitas Lampung.

7. Bapak Rico Andrian, S.Si, M.Kom. selaku Pembimbing Akademik yang senantiasa memberikan dukungan dan arahan dalam pengembangan akademik selama masa studi.
8. Bapak Dr. rer. nat. Akmal Junaidi, S.Si., M.Sc. selaku dosen Pembimbing Utama yang telah membimbing, memberikan arahan, masukan dan saran yang sangat berarti dalam penyelesaian skripsi ini.
9. Bapak Ridho Sholehurrohman, M. Mat. selaku dosen Pembimbing Kedua yang telah memberikan arahan dan masukan yang sangat membantu dalam proses penyusunan skripsi ini.
10. Bapak Favorisen R. Lumbanraja, Ph.D. selaku dosen Pembahas yang telah memberikan kritik, saran dan masukan yang bermanfaat dalam perbaikan skripsi ini.
11. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu, motivasi selama penulis menempuh pendidikan di Jurusan Ilmu Komputer Universitas Lampung.
12. Seluruh staf dan karyawan Jurusan Ilmu Komputer, Ibu Ade Nora Maela, Bang Zainuddin dan Mas Syam yang telah membantu kelancaran urusan administrasi, laboratorium serta peminjaman ruangan.
13. Teman seperjuangan semasa kuliah, Ika rahma Alia, Aprilia Anggun Sari Rahmawati, dan Kartika Sari yang selalu menemani, mendukung, serta memberikan semangat baik dalam suka maupun duka selama menempuh perkuliahan hingga penyusunan skripsi ini.
14. Anggota Grup Whatsapp Sahabat Jannah, Another Universe, KKN Desa Bandar Agung 2, yang selalu memberikan semangat kepada penulis selama menyelesaikan skripsi ini.
15. Keluarga Besar Ilmu Komputer 2021 yang telah memberikan pengalaman yang sangat berharga selama masa perkuliahan penulis di Jurusan Ilmu Komputer Universitas Lampung.

Penulis menyadari bahwa skripsi ini masih memiliki berbagai kekurangan dan jauh dari kesempurnaan. Namun demikian, penulis berharap skripsi ini dapat memberikan manfaat, khususnya bagi civitas akademika Universitas Lampung serta mahasiswa Ilmu Komputer.

Bandar Lampung, 26 Agustus 2025



Siska Hermayanti
NPM. 2117051023

DAFTAR ISI

	Halaman
DAFTAR GAMBAR.....	vii
DAFTAR TABEL	ix
DAFTAR KODE PROGRAM	x
I. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
II. TINJAUAN PUSTAKA.....	6
2.1. Penelitian Terdahulu	6
2.2. Tanaman Teh.....	10
2.3. Penyakit Daun Teh.....	11
2.4 Pengolahan Citra Digital	15
2.5 <i>Machine Learning</i>	16
2.6 <i>Deep Learning</i>	17
2.7 CNN (<i>Convolutional Neural Network</i>)	18
2.8 ResNet (<i>Residual Network</i>).....	24
2.9 <i>Python</i>	28
2.10 <i>Overfitting dan Underfitting</i>	29
2.11 Augmentasi Data.....	29
2.12 <i>Hyperparameter</i>	30
2.13 Evaluasi Metode.....	33
III. METODE PENELITIAN	37
3.1. Waktu dan Tempat Penelitian	37

3.2.	Perangkat Penelitian.....	39
3.2.1	Perangkat Keras (<i>hardware</i>)	39
3.2.2	Perangkat Lunak (<i>Software</i>).....	39
3.3	<i>Dataset</i>	42
3.4	Tahap Penelitian.....	44
IV.	HASIL DAN PEMBAHASAN	51
4.1.	Pengumpulan <i>Dataset</i>	51
4.2.	<i>Preprocessing</i>	51
4.3.	<i>Splitting</i> Data.....	56
4.4.	Augmentasi Data.....	57
4.5.	Arsitektur Model ResNet	61
4.6.	<i>Hyperparameter Tuning</i>	65
4.7.	<i>Training</i> Model	66
4.8.	Hasil <i>Training</i> Model.....	68
4.9.	Evaluasi Model.....	81
4.10.	Pengujian Model	96
4.11.	Perbandingan Kinerja Model	101
V.	SIMPULAN DAN SARAN	104
5.1.	Simpulan	104
5.2	Saran.....	105
	DAFTAR PUSTAKA	106

DAFTAR GAMBAR

Gambar	Halaman
1. Tanaman Teh (Agnisoonu, 2011).	10
2. Penyakit <i>algal leaf spot</i>	11
3. Penyakit <i>Anthracnose</i>	12
4. Penyakit <i>Bird Eye Spot</i>	13
5. Penyakit <i>Brown Blight</i>	14
6. Penyakit <i>White Leaf Spot</i>	14
7. Representasi Citra.	16
8. Struktur <i>Neural Network Deep Learning</i>	17
9. Arsitektur CNN.	18
10. Ilustrasi <i>Convolution Layer</i>	19
11. Ilustrasi <i>kernel 3x3</i>	20
12. Ilustrasi <i>stride 1</i>	20
13. Ilustrasi <i>padding</i>	21
14. ReLU (<i>Rectified Linear Unit</i>)	22
15. Ilustrasi <i>Pooling Layer</i>	22
16. Ilustrasi <i>Flatten</i>	23
17. Ilustrasi <i>Fully Connected Layer</i>	24
18. <i>Residual Block</i>	26
19. Arsitektur ResNet-50	27
20. Arsitektur ResNet-101	28
21. Ilustrasi <i>overfitting</i> (a) dan <i>underfitting</i> (b)	29
22. Tahap Penelitian.	44
23. Contoh gambar hasil <i>cropping</i>	52
24. Contoh gambar hasil <i>padding</i>	53
25. Contoh pelabelan data.	55

26. Contoh gambar hasil <i>resize</i> .	55
27. Akurasi dan <i>loss training</i> ResNet-50 Skenario 1	69
28. Akurasi dan <i>loss training</i> ResNet-101 Skenario 1	69
29. Akurasi dan <i>loss training</i> ResNet-50 Skenario 2	70
30. Akurasi dan <i>loss training</i> ResNet-101 Skenario 2	71
31. Akurasi dan <i>loss training</i> ResNet-50 Skenario 3	72
32. Akurasi dan <i>loss training</i> ResNet-101 Skenario 3	72
33. Akurasi dan <i>loss training</i> ResNet-50 Skenario 4	73
34. Akurasi dan <i>loss training</i> ResNet-101 Skenario 4	74
35. Akurasi dan <i>loss training</i> ResNet-50 Skenario 5	75
36. Akurasi dan <i>loss training</i> ResNet-101 Skenario 5	75
37. Akurasi dan <i>loss training</i> ResNet-50 Skenario 6	76
38. Akurasi dan <i>loss training</i> ResNet-101 Skenario 6	77
39. Akurasi dan <i>loss training</i> ResNet-50 Skenario 7	78
40. Akurasi dan <i>loss training</i> ResNet-101 Skenario 7	78
41. Akurasi dan <i>loss training</i> ResNet-50 Skenario 8	80
42. Akurasi dan <i>loss training</i> ResNet-101 Skenario 8	80
43. <i>Confusion Matrix</i> skenario 1 ResNet-50 dan ResNet-101	82
44. <i>Confusion Matrix</i> skenario 2 ResNet-50 dan ResNet-101	84
45. <i>Confusion Matrix</i> skenario 3 ResNet-50 dan ResNet-101	85
46. <i>Confusion Matrix</i> skenario 4 ResNet-50 dan ResNet-101	87
47. <i>Confusion Matrix</i> skenario 5 ResNet-50 dan ResNet-101	89
48. <i>Confusion Matrix</i> skenario 6 ResNet-50 dan ResNet-101	91
49. <i>Confusion Matrix</i> skenario 7 ResNet-50 dan ResNet-101	93
50. <i>Confusion Matrix</i> skenario 8 ResNet-50 dan ResNet-101	95
51. Contoh hasil <i>testing</i> prediksi	97
52. Contoh <i>Misclassification</i> ResNet-50	98
53. Contoh <i>Misclassification</i> ResNet-101	98

DAFTAR TABEL

Tabel	Halaman
1. Penelitian Terdahulu Terkait Penelitian yang akan dilakukan.....	6
2. Detail lapisan arsitektur ResNet-50 dan ResNet-101.....	25
3. <i>Multiclass Confusion Matrix</i>	34
4. <i>Timeline</i> Penelitian.....	38
5. <i>Dataset</i>	43
6. Distribusi Kelas Rasio 70:20:10	56
7. Distribusi Kelas Rasio 60:20:20	57
8. Sampel Augmentasi	59
9. Jumlah data <i>train</i> rasio 70:20:10 sebelum & setelah di augmentasi.....	60
10. Jumlah data <i>train</i> rasio 60:20:20 setelah & sebelum augmentasi	60
11. <i>Hyperparameter Tuning</i> Optimal.....	65
12. <i>Classification report</i> skenario 1	82
13. <i>Classification report</i> skenario 2	84
14. <i>Classification report</i> skenario 3	86
15. <i>Classification report</i> skenario 4	87
16. <i>Classification report</i> skenario 5	89
17. <i>Classification report</i> skenario 6	91
18. <i>Classification report</i> skenario 7	93
19. <i>Classification report</i> skenario 8	95
20. Contoh kemiripan gambar kelas asli dengan kelas prediksi	99
21. Perbandingan akurasi model ResNet-50 dan ResNet-101	101
22. Akurasi model penelitian terdahulu dan penelitian yang dilakukan	103

DAFTAR KODE PROGRAM

Kode Program	Halaman
1. <i>Padding</i>	53
2. Augmentasi data.....	58
3. <i>Training Model</i>	67

I. PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan salah satu negara penghasil teh terbesar di dunia, dan komoditas teh memiliki nilai penting bagi perekonomian nasional (Ibrahim *et al.*, 2022). Selain menjadi sumber devisa melalui ekspor, teh juga menyediakan lapangan kerja di sektor perkebunan. Selain manfaat ekonominya, teh juga menjadi bagian dari budaya masyarakat Indonesia, yang membuat keberlanjutan produksi teh sangat penting untuk menjaga kesejahteraan petani dan keberlanjutan industri teh. Namun, meskipun memiliki potensi yang besar, produksi teh di Indonesia mengalami penurunan dalam beberapa tahun terakhir. Berdasarkan data dari Badan Pusat Statistik (BPS), produksi teh Indonesia pada tahun 2022 tercatat sebesar 136.800 ton, mengalami penurunan sebesar 5,72% dibandingkan tahun sebelumnya yang mencapai 145.100 ton (Rizaty, 2023). Penurunan ini menjadi perhatian serius, mengingat dampaknya terhadap pasokan teh domestik dan pendapatan petani yang mengandalkan hasil panen ini untuk keberlangsungan hidup mereka.

Salah satu faktor penyebab terjadinya penurunan hasil produksi teh adalah serangan penyakit pada daun teh. Penyakit-penyakit seperti *Algal Leaf Spot*, *Anthrachnose*, *Bird Eye Spot*, *Brown Blight*, dan *White Leaf Spot* dapat menyebabkan penurunan kualitas dan kuantitas hasil panen. Penyakit-penyakit ini seringkali mengganggu pertumbuhan tanaman teh, menyebabkan daun-daun teh rusak, dan akhirnya menurunkan produktivitas kebun teh secara keseluruhan. Mengidentifikasi sedini mungkin penyakit-penyakit ini sangat penting untuk mengendalikan penyebarannya dan menjaga kualitas produksi teh.

Seiring dengan perkembangan teknologi kecerdasan buatan (*Artificial Intelligence*), terutama dalam bidang *Computer Vision* (CV), teknologi ini semakin banyak diterapkan dalam sektor pertanian untuk mengidentifikasi penyakit pada tanaman. Dengan menggunakan teknik analisis gambar, penyakit pada tanaman dapat diidentifikasi dengan lebih cepat dan akurat, dibandingkan dengan metode tradisional yang sering kali mengandalkan observasi manual yang memakan waktu dan tenaga.

Salah satu teknik *deep learning* yang efektif untuk analisis gambar adalah *Convolutional Neural Network* (CNN) (Alberto & Hermanto, 2023). CNN mampu mengekstraksi fitur-fitur penting dari gambar melalui lapisan konvolusi, sehingga dapat melakukan klasifikasi gambar dengan akurasi tinggi. Ada berbagai macam arsitektur CNN salah satunya adalah *Residual Network*. *Residual Network* (ResNet) memperkenalkan fitur baru bernama *skip connection*, yang memungkinkan jaringan menjadi lebih dalam dan membantu mengurangi nilai *error rate* (Anshori *et al.*, 2022). ResNet-50 memiliki optimisasi yang lebih mudah dan mampu mencapai akurasi tinggi seiring peningkatan kedalaman jaringan (Zahisham *et al.*, 2020). Di samping itu, arsitektur ResNet-101 yang memiliki lapisan konvolusi lebih dalam diharapkan dapat memberikan kinerja yang lebih baik dalam klasifikasi penyakit daun teh karena kedalaman jaringan yang lebih kompleks dan kapasitas pengenalan fitur yang lebih tinggi.

Penelitian terkait dengan menggunakan CNN arsitektur ResNet-50 pada penyakit daun jagung dilakukan oleh (Putra *et al.*, 2022) berjudul "*Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network*". Dengan *dataset* 4.225 gambar dan pengujian berbagai *optimizer*, hasil menunjukkan bahwa *optimizer Adam* dengan ResNet-50 mencapai akurasi tertinggi, yaitu 98,4%. Hasil ini menunjukkan potensi ResNet-50 dalam tugas klasifikasi gambar yang kompleks dan presisi tinggi, yang diharapkan dapat diaplikasikan untuk klasifikasi penyakit daun teh. Sementara itu, penelitian yang dilakukan oleh (Bayu, 2022), berjudul "*Identifikasi Citra Plasmodium Guna Mendeteksi Parasit Penyakit Malaria dengan Menggunakan Algoritma CNN*" bertujuan membandingkan performa

arsitektur ResNet-101 dengan AlexNet dan VGG-19 dalam deteksi parasit malaria. Hasilnya menunjukkan bahwa ResNet-101 memiliki akurasi tertinggi sebesar 97.58%, diikuti oleh VGG-19 dengan akurasi 97.44%, dan AlexNet dengan akurasi 96.18%. Keunggulan arsitektur ResNet-101 dalam mendeteksi gambar menunjukkan bahwa model ini cocok untuk diterapkan pada klasifikasi penyakit daun teh, yang juga memerlukan pengenalan pola kompleks.

Penelitian sebelumnya dengan klasifikasi penyakit daun teh dilakukan oleh (Hidayat & Ernawati, 2020) yang berjudul *"Metode Klasifikasi Algoritma Convolutional Neural Network (CNN) pada Penyakit Daun Teh"*. Penelitian ini berfokus pada deteksi tiga jenis penyakit daun teh, yaitu *Algal Leaf Spot*, *Anthracnose*, dan *Bird Eye Spot*. Dalam penelitian ini, implementasi CNN menunjukkan hasil yang cukup baik dengan akurasi mencapai 79,36%. Selanjutnya penelitian oleh (Lasniari *et al.*, 2022) *"Klasifikasi Citra Daging Babi dan Daging Sapi Menggunakan Deep Learning Arsitektur ResNet-50 dengan Augmentasi Citra"* bertujuan membedakan citra daging sapi dan babi secara otomatis. Menggunakan CNN ResNet-50 dengan augmentasi data, jumlah citra diperbesar dari 457 menjadi 2742 dalam tiga kelas (daging sapi, babi, dan campuran). Model mencapai akurasi tertinggi 87,64%, recall 87,59%, dan precision 90,90%, tanpa indikasi *overfitting*. Augmentasi terbukti meningkatkan akurasi, dengan hasil optimal pada pembagian data latih-uji 90%:10%.

Berdasarkan uraian diatas, penelitian ini bertujuan untuk mengimplementasikan dan membandingkan performa arsitektur CNN ResNet-50 dan ResNet-101 dalam klasifikasi penyakit daun teh. Dengan memanfaatkan kemampuan arsitektur ResNet yang terbukti bagus dalam klasifikasi gambar, diharapkan penelitian ini dapat memberikan kontribusi dalam meningkatkan akurasi dan efisiensi deteksi penyakit daun teh. Harapannya, penelitian ini dapat mendukung petani dalam pengendalian penyakit yang lebih efektif, sehingga kerugian dapat diminimalkan dan produksi teh tetap stabil.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana kinerja metode *Convolutional Neural Network* (CNN) dengan arsitektur ResNet-50 untuk mengklasifikasikan penyakit daun teh?
2. Bagaimana kinerja metode *Convolutional Neural Network* (CNN) dengan arsitektur ResNet-101 untuk mengklasifikasikan penyakit daun teh?
3. Bagaimana perbandingan evaluasi kinerja *Convolutional Neural Network* (CNN) dengan arsitektur ResNet-50 dan arsitektur ResNet-101 dalam klasifikasi penyakit daun teh?

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini yaitu sebagai berikut:

1. Penelitian ini menggunakan *dataset* dari *Kaggle* yang berisi penyakit daun teh dalam bentuk gambar dengan format ekstensi *file* berupa .jpg
2. Terdapat 5 kelas yang akan digunakan pada penelitian ini yaitu, *Algal Leaf Spot*, *Anthrachnose*, *Bird Eye Spot*, *Brown Blight*, dan *White Leaf Spot*.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Mengklasifikasi penyakit daun teh menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur ResNet-50.
2. Mengklasifikasi penyakit daun teh menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur ResNet-101.
3. Membandingkan hasil evaluasi kinerja metode *Convolutional Neural Network* (CNN) dengan arsitektur ResNet-50 dan arsitektur ResNet-101.

1.5 Manfaat Penelitian

Adapun manfaat yang bisa didapatkan dari penelitian ini adalah sebagai berikut:

1. Membantu petani mengidentifikasi penyakit daun teh lebih cepat dan akurat, mengurangi kerugian akibat penurunan hasil panen.
2. Menambah wawasan tentang penggunaan CNN arsitektur ResNet-50 dan arsitektur ResNet-10 dalam klasifikasi gambar penyakit pada daun teh.

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Berikut merupakan rangkuman dari beberapa penelitian terdahulu dengan menggunakan objek dan metode yang dijelaskan pada Tabel 1.

Tabel 1. Penelitian Terdahulu Terkait Penelitian yang akan dilakukan

	Penulis	Judul	Objek	Hasil
1.	(Hidayat & Ernawati, 2020)	Metode Klasifikasi Algoritma <i>Convolutional Neural Network</i> (CNN) pada Penyakit Daun Teh	Penyakit daun teh (3 kelas: <i>Algal Leaf Spot</i> , <i>Anthrachnose</i> , dan <i>Bird Eye Spot</i>)	Menggunakan algoritma CNN, model mencapai akurasi 79,36%.
2.	(Hendrawan <i>et al.</i> , 2023)	Klasifikasi Retak Ban Kendaraan Menggunakan Arsitektur ResNet50	Ban retak dan ban normal	Hasil terbaik ResNet50 Skenario ke-5 (<i>epochs</i> 100, <i>dropout</i> 0.5), mencapai akurasi, presisi, dan <i>recall</i> masing-masing 94% tanpa <i>overfitting</i> atau <i>underfitting</i> .

Tabel 1. (Lanjutan)

	Penulis	Judul	Objek	Hasil
3.	(Oktaviana <i>et al.</i> , 2021)	Klasifikasi Penyakit Padi Berdasarkan Citra Daun Menggunakan Model Terlatih ResNet101	Penyakit daun padi dengan 3 kelas yaitu <i>Bacterial leaf blight, Brown spot</i> , dan <i>Leaf smut</i>	Menggunakan algoritma CNN dengan arsitektur ResNet101 untuk mengklasifikasikan penyakit padi berdasarkan citra daun. Menghasilkan akurasi 100% dengan nilai <i>validation loss</i> 5,61%.
4.	Penelitian yang akan dilakukan	Klasifikasi penyakit daun teh menggunakan CNN arsitektur ResNet-50 dan ResNet-101	Penyakit daun teh 5 kelas	Menggunakan algoritma CNN dan akan membandingkan hasil akurasi arsitektur ResNet-50 dengan ResNet-101. <i>Dataset</i> yang digunakan dari <i>kaggle</i> https://www.kaggle.com/datasets/shashwatwork/identifying-disease-in-tea-leafs

Penelitian terkait penerapan algoritma CNN pada *dataset* penyakit daun teh pernah dilakukan oleh Hidayat & Ernawati (2020) yang bertujuan untuk mengklasifikasikan tiga jenis penyakit daun teh, yaitu *Algal Leaf*, *Antracnose*, dan *Bird Eye Spot*. Penelitian ini menggunakan *dataset* yang bersumber dari *Kaggle* (<https://www.kaggle.com/datasets/shashwatwork/identifying-disease-in-tea-leafs>). Proses penelitian meliputi pengumpulan data gambar, *pre-*

processing, labeling data, pelatihan model menggunakan algoritma *Convolutional Neural Network (CNN)* di *Jupyter Notebook*, dan evaluasi performa model. Dataset dibagi menjadi *training* dan *testing*, dengan pelatihan dilakukan selama 20 *epoch* menggunakan *batch size* sebesar 32. Hasil penelitian menunjukkan bahwa model CNN mampu mengklasifikasikan penyakit daun teh dengan tingkat akurasi sebesar 79,36%, *recall* sebesar 76,16%, dan *F1-score* sebesar 75,90%. Penelitian ini membuktikan bahwa teknologi *deep learning* seperti CNN dapat menjadi solusi untuk membantu petani mengklasifikasi penyakit daun teh dengan lebih cepat dan akurat, sehingga dapat meningkatkan kualitas produksi teh.

Penelitian terkait penerapan arsitektur ResNet-50 pernah dilakukan oleh Hendrawan *et al.*, (2023) yang membahas klasifikasi retak pada ban kendaraan menggunakan arsitektur ResNet50 untuk mengurangi *human error* dalam inspeksi manual. Dataset berasal dari *Kaggle* dengan total 1028 gambar, terdiri dari 376 gambar ban normal dan 327 gambar ban retak, yang dibagi dengan rasio 80:20 untuk data latih dan uji. Proses penelitian mencakup *resizing* gambar menjadi 100×100 piksel, transformasi label menjadi nilai biner, serta *tuning* parameter melalui enam skenario dengan variasi *epochs* dan *dropout*. Pada Skenario 1 (*epochs* 60, *dropout* 0.25), model mencapai akurasi pelatihan 92% dengan *loss* 13%, tetapi validasi hanya mencapai akurasi 56% dengan *loss* 180%, menunjukkan *overfitting*. Skenario 2 (*epochs* 100, *dropout* 0.25) menghasilkan akurasi pelatihan 100% dan *loss* 1%, tetapi validasi hanya mencapai akurasi 53% dengan *loss* 190%, juga menunjukkan *overfitting*. Skenario 3 (*epochs* 150, *dropout* 0.25) menghasilkan akurasi pelatihan 97% dan *loss* 4%, namun validasi hanya mencapai akurasi 70% dengan *loss* 72%, tetap mengalami *overfitting*. Skenario 4 (*epochs* 60, *dropout* 0.5) menunjukkan peningkatan dengan akurasi pelatihan 97% dan *loss* 7%, serta akurasi validasi 96% dengan *loss* 6%, menunjukkan model yang baik. Skenario 5 (*epochs* 100, *dropout* 0.5) menghasilkan hasil terbaik dengan akurasi, presisi, dan *recall* masing-masing 94%, tanpa *overfitting* atau *underfitting*, sementara itu Skenario 6 (*epochs* 150, *dropout* 0.5) mencapai akurasi validasi 100% dengan

loss validasi 1%, tetapi menunjukkan hasil kurang stabil. Kesimpulannya, ResNet50 dengan parameter *tuning* optimal pada Skenario 5 efektif dalam mengenali keretakan pada ban kendaraan.

Penelitian terkait dengan ResNet-101 dilakukan oleh Oktaviana *et al.*, (2021) bertujuan untuk mengidentifikasi tiga jenis penyakit pada daun padi, yaitu *Bacterial Leaf Blight*, *Brown Spot*, dan *Leaf Smut*. Dataset yang digunakan dari *UCI Machine Learning Repository* berisi 120 gambar daun padi, masing-masing 40 gambar untuk setiap jenis penyakit. Dataset ini dibagi menjadi dua bagian, 90% untuk data latih dan 10% untuk validasi. Peneliti memanfaatkan model ResNet101 yang sudah dilatih sebelumnya (*pre-trained*) dan menambahkan beberapa lapisan tambahan untuk meningkatkan akurasi, seperti *Dense Layer*, *BatchNormalization*, dan *Dropout* agar model tidak mudah *overfitting*. Mereka juga melakukan augmentasi data, parameter yang digunakan meliputi *preprocessing_function* yang menggunakan fungsi *preprocess_input* dari *library TensorFlow*, modul *keras-applications-resnet*, *rotation_range = 30*, *zoom_range = 0,4*, dan *horizontal_flip = True*. Untuk data validasi, augmentasi dilakukan dengan sedikit perbedaan, yaitu hanya menggunakan parameter *preprocessing_function* yang sama seperti pada augmentasi data latih. Data latih dan data validasi juga diubah ukurannya (*resize*) menjadi 224 x 224. Setelah melatih model selama 100 *epoch*. Hasilnya mencapai akurasi 100% pada data validasi. Selain itu, *precision*, *recall*, dan *F1-score* juga menunjukkan nilai sempurna, yaitu 1,00 untuk semua kelas. Model ini juga memiliki nilai *validation loss* yang cukup rendah, yaitu 5,61%, menunjukkan kestabilan model dalam pelatihan. Penelitian membuktikan bahwa penggunaan ResNet101 yang dimodifikasi dengan lapisan tambahan seperti *Dense Layer*, *BatchNormalization*, dan *Dropout* sangat efektif untuk meningkatkan performa dalam klasifikasi penyakit daun padi.

2.2. Tanaman Teh

Tanaman teh (*Camellia sinensis*) merupakan jenis tumbuhan yang banyak ditemukan di wilayah tropis. Ciri-cirinya yaitu batang yang tumbuh tegak dengan daun tunggal yang tersusun berselang-seling. Daunnya berbentuk lonjong dengan tulang daun yang meruncing dan dilengkapi bulu-bulu halus di ujungnya. Pada tepi daun terdapat gerigi halus yang tajam. Daun muda memiliki warna lebih cerah, permukaan yang kurang licin, dan ukuran yang lebih besar dibandingkan daun tua, yang berwarna hijau tua dengan tekstur yang lebih licin. Perbedaan ini mencerminkan perubahan pada daun seiring dengan proses pertumbuhannya (Ibrahim *et al.*, 2022). Selain itu, kriteria fisik untuk menentukan daun teh muda adalah daun ketiga hingga kelima dari atas, dengan warna hijau muda. Sedangkan daun tua terdiri dari daun keenam hingga kedelapan yang berwarna hijau tua (Izzreen & Fadzelly, 2013). Tanaman teh sehat dapat dilihat pada Gambar 1.



Gambar 1. Tanaman Teh (Agnisoonu, 2011).

Komposisi teh dipengaruhi oleh berbagai faktor, seperti kondisi iklim, musim panen, metode budidaya yang diterapkan, jenis varietas teh, serta usia tanaman. Teh mengandung berbagai senyawa bioaktif, dengan sekitar sepertiganya berupa *polifenol*. *Polifenol* dalam teh dapat dibedakan menjadi *flavonoid* dan *non-flavonoid*, namun mayoritasnya adalah *flavonoid*. *Polifenol* ini berfungsi sebagai antioksidan yang membantu melawan radikal bebas dalam tubuh dan dapat berperan dalam pencegahan perkembangan sel kanker (Anjarsari, 2022). Selain faktor-faktor tersebut, kualitas daun teh sangat dipengaruhi oleh

kesehatan daun itu sendiri. Daun teh yang terserang penyakit tidak hanya mempengaruhi produktivitas, tetapi juga kandungan senyawa bioaktif yang ada di dalamnya. Oleh karena itu, pemantauan terhadap penyakit daun menjadi langkah penting untuk menjaga kualitas daun, sehingga cita rasa, aroma, dan manfaat kesehatan dari teh dapat tetap optimal.

2.3. Penyakit Daun Teh

Penyakit daun teh merupakan salah satu tantangan utama dalam budidaya tanaman teh (*Camellia sinensis*). Daun teh, yang menjadi komponen utama dalam produksi teh, sangat rentan terhadap berbagai jenis penyakit yang dapat disebabkan oleh jamur, bakteri, maupun faktor lingkungan. Penyakit pada tanaman teh dapat menyebabkan kerugian ekonomi yang signifikan bagi petani. Kerugian ini tidak hanya berasal dari penurunan hasil panen akibat kerusakan pada tanaman, tetapi juga dari biaya tambahan yang harus dikeluarkan untuk menangani dan mengendalikan penyakit tersebut. Berikut 5 penyakit daun teh yang sering dijumpai, data diperoleh dari *Kaggle*.

2.3.1 *Algal leaf spot*

Penyakit *algal leaf spot* disebabkan oleh ganggang *Cephaleuros virescens* yang menimbulkan bercak oranye-cokelat yang dapat melebar pada daun teh. Penyakit ini biasanya muncul saat musim panas dan menyebar di musim hujan, terutama pada tanaman teh yang cenderung tumbuh rapat dan lebat (Mohan & Jayakrishnan, 2024).



Gambar 2. Penyakit *algal leaf spot*.

Gambar 2. merupakan contoh gambar daun teh penyakit *algal leaf spot*. Infeksi ini dapat menghambat fotosintesis karena merusak sebagian atau seluruh permukaan daun. Risiko infeksi meningkat jika tanaman memiliki *drainase* buruk, nutrisi tidak seimbang, serta terpapar suhu dan kelembaban tinggi. Pencegahan dapat dilakukan dengan penggunaan pestisida, teknik budidaya yang sesuai dan pemupukan yang tepat (Mohan & Jayakrishnan, 2024).

2.3.2 *Anthraxnose*

Penyakit antraknosa adalah penyakit yang umum di daerah tropis dan subtropis, disebabkan oleh jamur *Colletotrichum gloeosporioides*. Spora jamur ini dapat tersebar melalui angin, sehingga penyebarannya sangat cepat. Jamur *Colletotrichum* dapat menginfeksi berbagai bagian tanaman, seperti batang, ranting, daun, dan buah (Prihatiningsih *et al.*, 2020).



Gambar 3. Penyakit *Anthraxnose*.

Gambar 3. menampilkan contoh daun teh dengan penyakit *anthracnose*. Kerusakan panen akibat infeksi dapat mencapai 30–75%. Awalnya, daun yang terinfeksi menunjukkan luka basah berwarna gelap, yang kemudian membesar seiring waktu. Jika tidak segera ditangani, penyakit ini akan menyebar ke seluruh kebun. Pemangkasan daun dan ranting mati secara tepat waktu adalah cara efektif untuk menghentikan penyebarannya (Mohan & Jayakrishnan, 2024).

2.3.3 *Bird Eye Spot*

Penyakit bercak mata burung adalah salah satu masalah penting pada tanaman teh karena dapat menurunkan hasil panen dan mempengaruhi kesehatan tanaman. Penyakit ini disebabkan oleh jamur *Cercospora theae petch* dari kelompok *Ascomycetes*. Gejala awalnya muncul pada daun muda, kemudian menyebar ke daun yang lebih tua dan kadang-kadang menyerang tangkai yang menopang tiga pucuk penting dalam proses produksi teh. Jamur ini menyebabkan bercak khas pada daun yang terinfeksi, berwarna coklat hingga hitam dengan pinggiran merah (Gnanamangai *et al.*, 2017).



Gambar 4. Penyakit *Bird Eye Spot*.

Gambar 4. merupakan contoh gambar daun teh dengan penyakit *bird eye spot*. Penyakit ini dipengaruhi oleh iklim dan dapat dikendalikan dengan drainase yang baik, aerasi, pemupukan, dan kalium (Mohan & Jayakrishnan, 2024).

2.3.4 *Brown Blight*

Brown blight adalah penyakit pada daun teh yang dapat mengurangi hasil panen secara signifikan. Ketika *Colletotrichum camelliae*, penyebab utama penyakit ini, menginfeksi tanaman teh daun akan muncul bercak hijau gelap dan berair. Seiring waktu, bercak ini membesar dan berubah menjadi coklat kemerahan atau coklat gelap dengan titik-titik hitam.

Titik-titik hitam ini bisa menghasilkan spora yang mempermudah penyebaran penyakit (Lin *et al.*, 2023).



Gambar 5. Penyakit *Brown Blight*.

Gambar 5. merupakan contoh visual daun teh dengan penyakit *brown blight*. Penyakit ini bisa menyebabkan daun gugur. Pencegahannya bisa dilakukan dengan menjaga jarak tanam untuk sirkulasi udara dan sinar matahari yang cukup.

2.3.5 *White Leaf Spot*

Penyakit *white spot* disebabkan oleh jamur *Phyllosticta theifolia* Hara yang menimbulkan bercak putih pada permukaan daun teh. Seiring waktu, bercak ini dapat menjadi gelap di bagian tepinya dan kehilangan bagian tengah, menyerupai lubang kecil. Penyakit ini berdampak negatif pada proses fotosintesis, hasil panen, dan mutu daun (Mohan & Jayakrishnan, 2024). Gambar 6. contoh visual dari daun teh dengan penyakit *white leaf spot*.



Gambar 6. Penyakit *White Leaf Spot*.

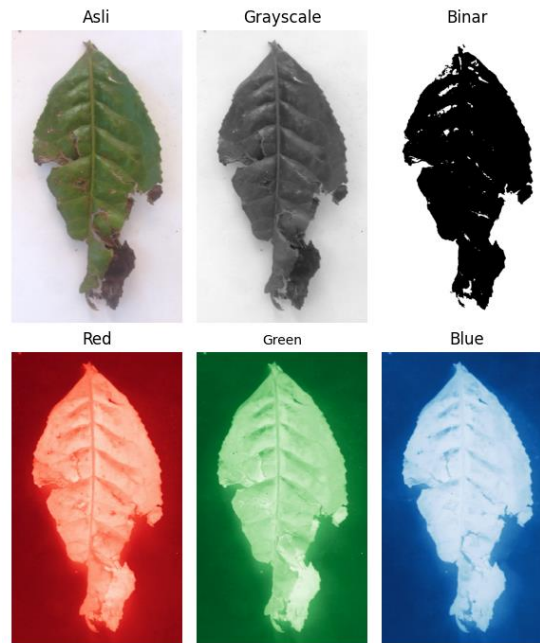
2.4 Pengolahan Citra Digital

Citra digital adalah bentuk visual dunia nyata yang diubah ke dalam format digital sehingga dapat dipahami dan diolah oleh komputer. Gambar ini terdiri dari kumpulan titik-titik kecil bernama piksel yang tersusun dalam pola baris dan kolom. Setiap piksel menyimpan nilai numerik yang merepresentasikan intensitas kecerahan atau warna di lokasi tertentu pada gambar tersebut (Dijaya, 2023). Pengolahan citra digital adalah bidang ilmu yang mempelajari berbagai teknik untuk meningkatkan kualitas gambar, seperti memperbaiki kontras, mengubah warna, atau melakukan restorasi. Selain itu, ilmu ini juga mencakup transformasi gambar, termasuk translasi, rotasi, skala, dan perubahan geometrik. Pengolahan citra juga digunakan untuk memilih fitur gambar (*feature images*) yang paling relevan untuk analisis, mengelola penyimpanan data dengan reduksi dan kompresi, serta mendukung transmisi dan pemrosesan data dengan efisien (Munantri *et al.*, 2020).

Citra digital dapat dibedakan menjadi tiga jenis berdasarkan nilai pikselnya, yaitu (Sholehurrohman, 2021):

1. Citra Biner : Citra ini hanya memiliki dua tingkat intensitas, yaitu hitam dan putih. Piksel dengan nilai 1 merepresentasikan warna hitam, sedangkan piksel bernilai 0 merepresentasikan warna putih.
2. Citra *Grayscale* : Jenis citra ini memiliki tingkat intensitas keabuan tunggal yang berkisar antara 0 hingga 255. Setiap nilai menunjukkan gradasi abu-abu dari gelap ke terang. Nilai 0 adalah warna putih dan nilai 255 adalah warna hitam.
3. Citra Warna (RGB) : Citra ini terdiri dari tiga lapisan warna utama, yaitu merah (*Red*), hijau (*Green*), dan biru (*Blue*). Masing-masing lapisan memiliki rentang nilai intensitas antara 0 hingga 255, yang jika digabungkan membentuk berbagai warna.

Contoh dari representasi citra grayscale, citra binar, dan citra RGB dapat dilihat pada Gambar 7.



Gambar 7. Representasi Citra.

2.5 Machine Learning

Istilah *machine learning* pertama kali diperkenalkan oleh Arthur Samuel pada tahun 1959. *Machine learning* adalah cabang ilmu komputer yang memungkinkan komputer untuk belajar dan memahami sesuatu tanpa pemrograman yang eksplisit (Samuel, 1959). Sederhananya, *machine learning* adalah kemampuan komputer belajar dari data untuk memecahkan masalah. Secara umum *machine learning* dibagi menjadi dua jenis yaitu *supervised learning* dan *unsupervised learning*.

2.5.1 Supervised Learning

Supervised learning adalah metode *machine learning* di mana model dilatih menggunakan data berlabel. Data *training* terdiri dari pasangan *input* dan *output*, sehingga model belajar untuk memetakan *input* ke *output* yang benar. Tujuan utama *supervised learning* adalah membuat prediksi pada data baru berdasarkan pola yang dipelajari. Contoh algoritma *supervised learning* yaitu KNN (*K-Nearest Neighbors*),

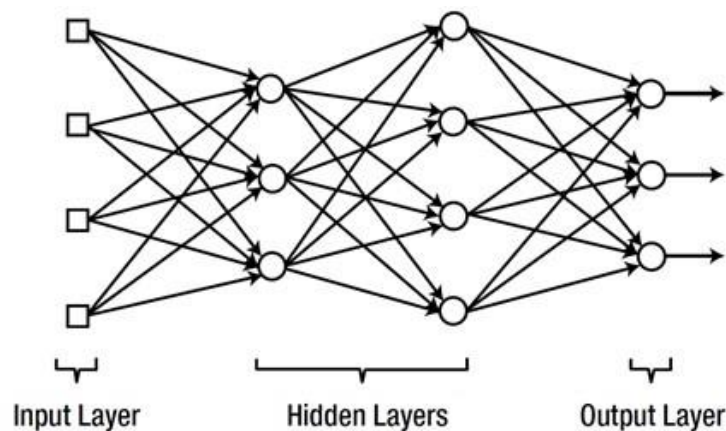
Convolutional Neural Network (CNN), Decision Tree, Naïve Baiyes, Logistic Regression, dan SVM (Santoso et al., 2021).

2.5.2 *Unsupervised Learning*

Unsupervised learning adalah metode *machine learning* yang bekerja dengan data yang tidak memiliki label. Model mencoba menemukan pola tersembunyi dalam data, seperti mengelompokkan data yang mirip satu sama lain. Contoh algoritma yaitu *K-Means* dan DBSCAN (Santoso et al., 2021).

2.6 *Deep Learning*

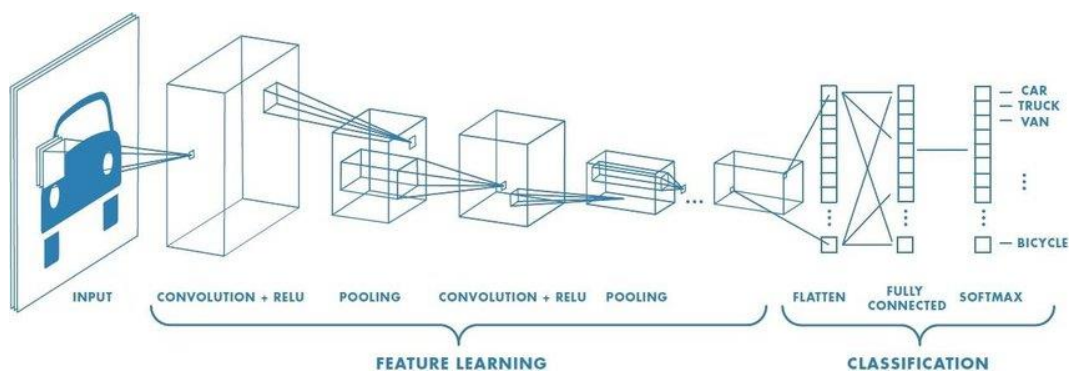
Deep Learning adalah cabang dari *Machine Learning* yang memanfaatkan jaringan neural dengan banyak lapisan (*deep neural network*) untuk menangani masalah pemrosesan data yang kompleks (Adelia et al., 2024). Pembelajaran ini disebut *deep* karena proses pemodelan melibatkan sejumlah lapisan tersembunyi yang memanipulasi data. Pada jenis pemodelan ini, informasi spesifik dikumpulkan dari setiap lapisan. Sebagai contoh, pada lapisan tertentu model dapat mendeteksi tepi gambar, sementara lapisan lainnya mengidentifikasi rona pada citra (Ramadhani et al., 2023). Struktur *Neural Network Deep Learning* dapat dilihat pada Gambar 8.



Gambar 8. Struktur *Neural Network Deep Learning* (Nima & Shila, 2020).

2.7 CNN (*Convolutional Neural Network*)

Convolutional Neural Network (CNN) adalah salah satu jenis *deep neural network* yang dirancang untuk memproses data gambar, memungkinkan pendeteksian dan pengenalan objek secara efektif (Ramadhani *et al.*, 2023). CNN digunakan untuk mengklasifikasi data menggunakan metode *supervised learning*, di mana data sudah memiliki label. Arsitektur *Convolutional Neural Network* (CNN) dapat dilihat pada Gambar 9.



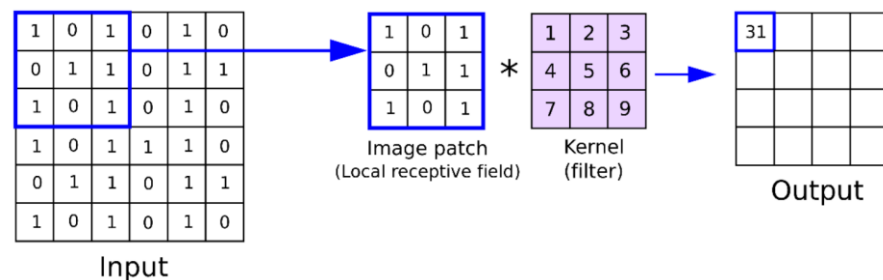
Gambar 9. Arsitektur CNN (Syulistyo *et al.*, 2020).

Ekstraksi fitur pada CNN bekerja secara hierarki artinya *output* tiap lapisan menjadi *input* untuk lapisan berikutnya (Setyawan *et al.*, 2023). Proses dalam *Convolutional Neural Network* ditunjukkan pada Gambar 9. yang terdiri dari dua tahapan utama yaitu *feature learning* dan *classification*. Pada tahap *feature learning*, prosesnya meliputi *convolution layer*, ReLU (*Rectified Linear Unit*), dan *pooling layer*, sedangkan pada tahap *classification* data yang sudah diproses akan melewati *flatten*, *fully-connected layer*, dan menghasilkan prediksi. Setiap *output* dari lapisan-lapisan ini terhubung dan digunakan sebagai *input* untuk lapisan berikutnya.

2.7.1 *Convolution Layer*

Convolution layer merupakan komponen utama dalam CNN (*Convolutional Neural Network*) yang berperan penting dalam proses ekstraksi fitur. Lapisan ini terdiri dari sejumlah *filter* yang melakukan

operasi konvolusi pada gambar *input*. *Filter* dalam lapisan konvolusi terdiri dari kumpulan *neuron* yang diinisialisasi secara acak (Muhammad & Wibowo, 2021) dan tersusun menyerupai matriks dengan ukuran tertentu (panjang dan tinggi dalam piksel) (Anggarkusuma *et al.*, 2024). Nilai *filter* yang diinisialisasi secara acak ini akan diperbarui selama proses *training*, sehingga model dapat semakin optimal dalam mengenali pola pada data *input*. Proses ini menjumlahkan hasil perkalian dari matriks *filter* dengan gambar, *filter* bergerak secara sistematis dari sudut kiri atas hingga ke sudut kanan bawah gambar. Setiap bagian kecil gambar yang dihasilkan dari operasi konvolusi akan dijadikan *input* untuk membentuk representasi fitur baru. Dengan cara ini, CNN mampu mengenali objek pada gambar, bahkan jika posisi objek tersebut bervariasi. Ilustrasi proses dari lapisan konvolusi dapat dilihat pada Gambar 10.



Gambar 10. Ilustrasi *Convolution Layer* (Muhammad & Wibowo, 2021).

Pada *convolution layer*, terdapat tiga elemen penting, yaitu *kernel*, *stride*, dan *padding*.

1. *Kernel*

Kernel pada *Convolutional Neural Network* (CNN) merupakan sekumpulan nilai diskrit yang berfungsi sebagai filter dalam mengekstraksi ciri penting dari data masukan. Bobot awal pada kernel diinisialisasi dengan nilai tertentu, kemudian disesuaikan selama proses pelatihan untuk

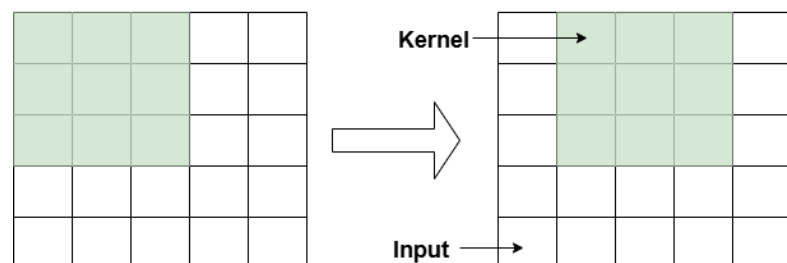
mengenali pola. Dengan demikian, *kernel* berperan dalam mengubah data citra menjadi *feature map* yang merepresentasikan informasi penting, seperti tepi, tekstur, maupun bentuk tertentu (Taye, 2023). Contoh *kernel* berukuran 3x3 CNN dapat dilihat pada Gambar 11.

1	0	-1
1	0	-1
1	0	-1

Gambar 11. Ilustrasi *kernel* 3x3.

2. *Stride*

Stride pada *Convolutional Neural Network* (CNN) adalah parameter yang menentukan jarak pergeseran filter saat proses konvolusi. Nilai *stride* yang kecil menghasilkan *feature map* berukuran lebih besar dengan detail yang tinggi, sedangkan nilai *stride* yang lebih besar memperkecil ukuran output sekaligus mengurangi *overlap* antar area (Taye, 2023). Pengaturan *stride* mempengaruhi ukuran *output* dan detail yang diproses. Ilustrasi perpindahan *kernel* dengan *stride* bernilai 1 dapat dilihat pada Gambar 12.



Gambar 12. Ilustrasi *stride* 1.

3. *Padding*

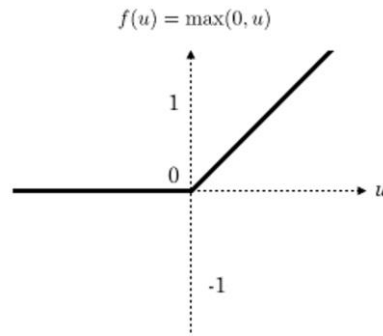
Padding pada CNN adalah teknik menambahkan piksel di tepi citra agar ukuran *output* tidak menyusut dan informasi pada bagian pinggir tetap terjaga. Salah satu bentuk yang paling umum digunakan adalah *zero-padding*, yaitu penambahan piksel bernilai nol untuk menjaga ukuran *feature map* sekaligus mengurangi hilangnya detail di tepi gambar (Taye, 2023). Ilustrasi *input* yang menggunakan *padding* sebanyak 1 dapat dilihat pada Gambar 13.

0	0	0	0	0	0	0
0						0
0						0
0						0
0						0
0						0
0	0	0	0	0	0	0

Gambar 13. Ilustrasi *padding*.

2.7.2 ReLU (*Rectified Linear Unit*)

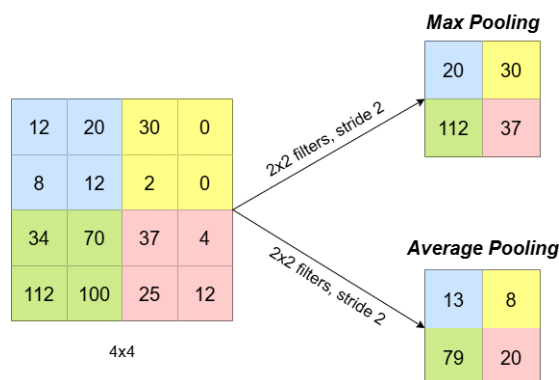
Rectified Linear Unit (ReLU) adalah fungsi aktivasi yang mengubah semua nilai *input* negatif menjadi nol dan *output* yang sebanding dengan *inputnya* (kemiringan 1) jika *input* lebih besar dari nol (Abadi & Wibowo, 2021). ReLU menggunakan fungsi $f(u) = \max(0, u)$, yang berarti jika *input* lebih besar dari nol, ReLU akan mengembalikan nilai *input* tersebut, dan jika *input* lebih kecil dari nol, ReLU akan mengembalikan nol. ReLU digunakan untuk membuat model lebih baik dalam belajar pola yang kompleks. Fungsi ini juga mempercepat pelatihan karena tidak mengalami masalah yang sering muncul pada fungsi aktivasi lain, seperti sulitnya belajar saat nilai *input* terlalu kecil. Ilustrasi ReLU dapat dilihat pada Gambar 14.



Gambar 14. ReLU (*Rectified Linear Unit*) (Abadi & Wibowo, 2021).

2.7.3 Pooling Layer

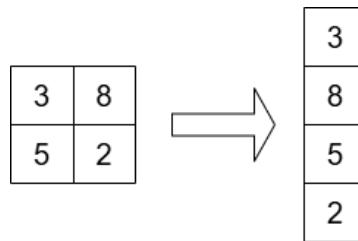
Pooling layer adalah lapisan yang menggunakan filter dengan ukuran tertentu dan bergerak secara bertahap di seluruh area *feature map*. Jenis *pooling* yang umum digunakan adalah *max pooling* dan *average pooling*. Pada *max pooling*, mengambil nilai terbesar, sedangkan *average pooling* mengambil nilai rata-rata (Setyawan *et al.*, 2023). Dengan begitu, *Max pooling* berguna untuk mengekstraksi fitur paling menonjol atau dominan dalam gambar. Sedangkan *average pooling* mempertahankan informasi global dan mengurangi sensitivitas terhadap variasi kecil. *Pooling layer* memperkecil ukuran *feature map*, sehingga mengurangi parameter dan perhitungan, mempercepat pelatihan, mengurangi *overfitting*, serta membuat model lebih tahan terhadap pergeseran atau distorsi kecil pada gambar. Proses *pooling* diilustrasikan pada Gambar 15.



Gambar 15. Ilustrasi *Pooling Layer*.

2.7.4 *Flatten Layer*

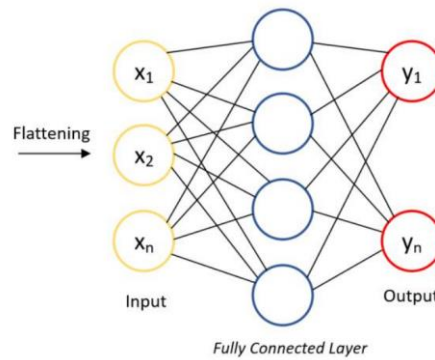
Flatten Layer adalah tahap mengubah data yang awalnya dalam bentuk matriks atau tensor (berdimensi lebih dari satu) menjadi vektor satu dimensi (Abadi & Wibowo, 2021). Proses ini diperlukan agar data yang diproses oleh lapisan sebelumnya, seperti *convolutional* atau *pooling*, dapat diteruskan ke *Fully Connected Layer* yang memerlukan *input* dalam bentuk vektor untuk melakukan klasifikasi atau prediksi. Ilustrasi proses *flatten layer* dapat dilihat pada Gambar 16.



Gambar 16. Ilustrasi *Flatten*.

2.7.5 *Fully Connected Layer*

Fully Connected Layer (FCL) adalah bagian dari *neural network* yang menghubungkan setiap neuron di lapisan sebelumnya dengan setiap neuron di lapisan berikutnya. *Output* dari lapisan *convolution* dan *pooling* berisi *high-level features* atau pola penting yang diekstraksi dari gambar *input* kemudian FCL mengambil data ini dan mengubahnya menjadi vektor dengan panjang N, di mana N adalah jumlah kelas. Tujuan utamanya adalah menentukan kelas yang paling sesuai untuk gambar berdasarkan fitur-fitur yang telah dipelajari selama *training* (Muhammad & Wibowo, 2021). Ilustrasi *fully connected layer* dapat dilihat pada Gambar 17.



Gambar 17. Ilustrasi *Fully Connected Layer* (Handoko *et al.*, 2024).

2.8 ResNet (*Residual Network*)

Residual Network (ResNet) yang dikembangkan oleh Kaiming He dkk, adalah jenis arsitektur jaringan *neural* yang dirancang untuk mengatasi masalah dalam melatih *deep neural networks*, seperti *vanishing gradient* (gradien menjadi terlalu kecil untuk diperbarui) dan penurunan akurasi pelatihan (*degradation problem*). Biasanya, semakin dalam jaringan, semakin sulit untuk melatihnya karena jaringan kesulitan untuk benar-benar memahami pola. Untuk mengatasi ini, ResNet menggunakan pendekatan *residual learning*, di mana setiap lapisan tidak mempelajari fungsi target secara langsung, tetapi mempelajari selisih atau *residual* antara *output* yang diharapkan dan *input* awal. Selisih ini kemudian ditambahkan kembali ke *input* menggunakan *shortcut connections*, yang berfungsi sebagai jalur pintas untuk melewati beberapa lapisan. Pendekatan ini membuat proses pelatihan menjadi lebih sederhana karena jaringan hanya perlu memahami perubahan kecil (*residual*) daripada harus menguasai fungsi yang rumit (He *et al.*, 2016).

Dalam penelitiannya Kaiming He dkk, menguji beberapa jenis ResNet, yaitu ResNet-18, ResNet-34, Resnet-50, ResNet-101, ResNet-152, hingga ResNet-1202. Perbedaan dari setiap arsitektur ResNet tersebut berdasarkan jumlah lapisan yang digunakan. Dalam eksperimennya pada dataset ImageNet, ResNet mencapai kedalaman hingga 152 lapisan, 8 kali lebih dalam dari VGG, tetapi dengan kompleksitas yang lebih rendah. Hasilnya menunjukkan bahwa

jaringan residual lebih mudah dioptimalkan dibandingkan jaringan biasa (*plain networks*), serta memberikan peningkatan akurasi yang signifikan, termasuk kemenangan dalam kompetisi ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*) 2015.

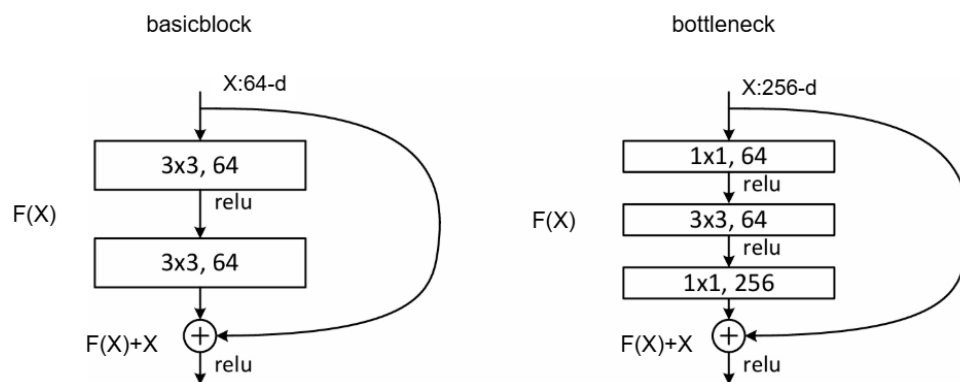
Penelitian ini hanya akan mengimplementasikan arsitektur ResNet-50 dan ResNet-101. Detail lapisan pada arsitektur ResNet-50 dan ResNet-101 berdasarkan penelitian Kaiming He dkk, dapat dilihat pada Tabel 2.

Tabel 2. Detail lapisan arsitektur ResNet-50 dan ResNet-101(He *et al.*, 2016)

Nama <i>Layer</i>	Ukuran <i>Output</i>	ResNet-50	ResNet-101
conv1	112×112	$7 \times 7, 64, \text{stride } 2$	
conv2_x	56×56	$3 \times 3 \text{ max pool, stride } 2$	
		$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4_x	14×14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$
conv5_x	7×7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	<i>Average pool, 1000-d fc, softmax</i>	
FLOPs		3.8×10^9	7.6×10^9

Pada model ResNet yang lebih dalam seperti ResNet-50, ResNet-101, dan ResNet-152, digunakan *bottleneck block* yang terdiri dari tiga lapisan konvolusi (1×1 , 3×3 , dan 1×1). Desain ini berbeda dengan ResNet-18 dan ResNet-34, yang menggunakan dua lapisan konvolusi per blok. Pada *bottleneck block*, lapisan 1×1 pertama digunakan untuk mengurangi dimensi data, lapisan 3×3 untuk ekstraksi fitur, dan lapisan 1×1 terakhir untuk

mengembalikan dimensi fitur ke ukuran semula. Penggunaan blok ini memungkinkan model yang lebih dalam tetap efisien dalam hal komputasi dan jumlah parameter. Pada ResNet-50, ResNet-101, dan ResNet-152, jumlah blok *bottleneck* ditingkatkan, dengan kedalaman masing-masing mencapai 50, 101, dan 152 lapisan, yang memungkinkan peningkatan akurasi tanpa mengorbankan efisiensi pelatihan. *Residual block* memiliki dua jenis jalur utama, yaitu *basic block* dan *bottleneck*. Struktur jaringan pada *residual block* dapat dilihat pada Gambar 18.

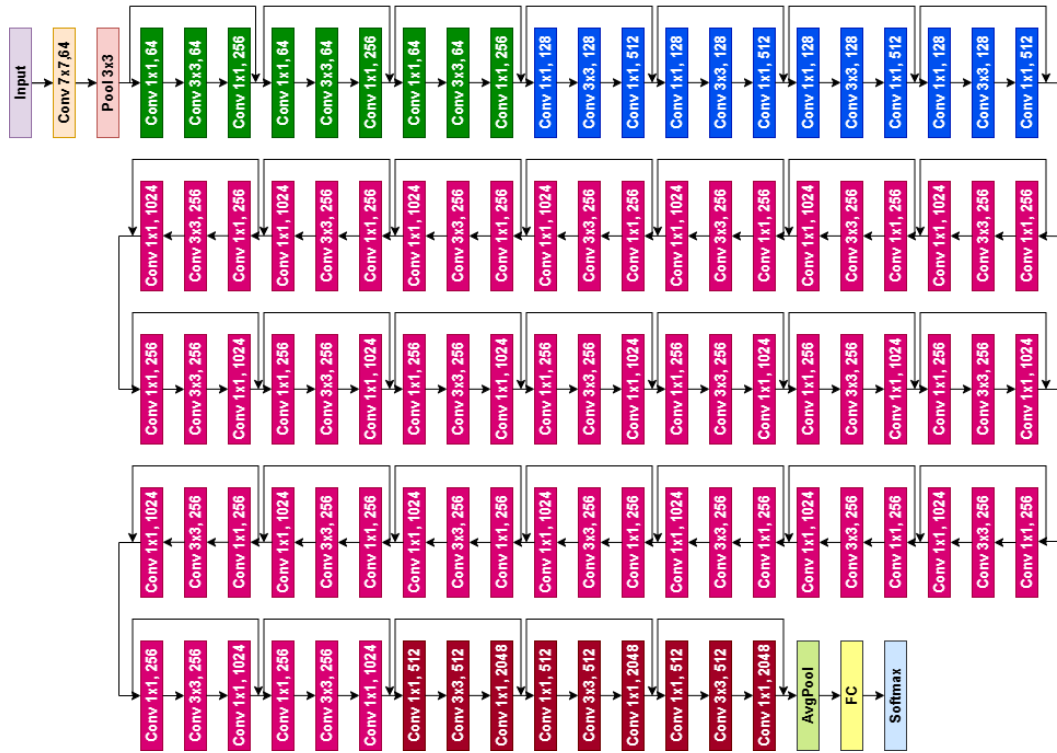


Gambar 18. *Residual Block* (He et al., 2016).

$F(X)$ adalah *output* setelah melewati *residual block*, dan X adalah data *input* asli dari model (melalui *shortcut connections*). Persamaan untuk menghitung *output* terakhir ($H(X)$) adalah: $H(X) = F(X) + X$. Artinya, fungsi ini menambahkan *output* dari *residual block* ($F(X)$) dengan *input* aslinya (X). Penjumlahan sederhana ini membantu mempercepat pelatihan model, membuatnya lebih efisien, dan mencegah masalah seperti penurunan performa jaringan akibat terlalu banyak lapisan (Li et al., 2023).

2.8.1 ResNet-50

ResNet50 menggunakan *bottleneck block* yang terdiri dari tiga lapisan konvolusi (1x1, 3x3, dan 1x1). Lapisan 1x1 berfungsi untuk mengurangi dan mengembalikan dimensi, sementara lapisan 3x3 digunakan untuk ekstraksi fitur. Dengan total 50 lapisan, arsitektur ini dirancang untuk



Gambar 20. Arsitektur ResNet-101.

Gambar 20. merupakan ilustrasi arsitektur ResNet-101 berdasarkan Tabel 2. Seperti *ResNet-50*, arsitektur ini menggunakan blok *residual* berisi tiga layer konvolusi (*Conv 1x1*, *Conv 3x3*, *Conv 1x1*). Bedanya dijumlah blok bagian *conv4_x*, ResNet-101 memiliki 23 blok sedangkan ResNet-50 6 blok, membuat ResNet-101 lebih dalam dan kompleks dengan total 101 layer.

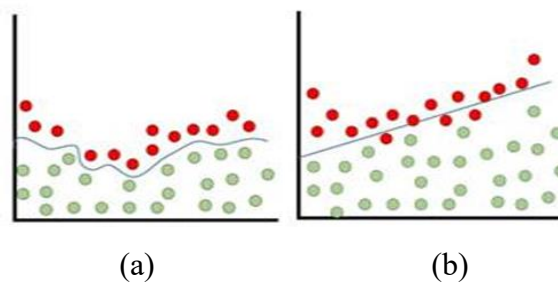
2.9 Python

Python adalah bahasa pemrograman yang menggunakan interpreter untuk menjalankan kode program sehingga dapat menjalankan kode programnya secara langsung. Bahasa ini dikembangkan oleh Guido van Rossum pada tahun 1991 di Stichting Mathematisch Centrum (CWI), Amsterdam. *Python* dirancang agar dapat berjalan di berbagai *platform*, seperti *Windows*, *Linux*, dan lainnya, sehingga menjadikannya salah satu bahasa pemrograman yang paling populer saat ini. *Python* memiliki sintaks yang sederhana dan mudah dipahami, sehingga sangat cocok digunakan oleh pemula maupun pengembang berpengalaman (Rahman *et al.*, 2023). Selain itu, bahasa ini mendukung

berbagai paradigma pemrograman, seperti pemrograman berorientasi objek, fungsional, dan prosedural. Dengan fleksibilitasnya, *Python* banyak digunakan di berbagai bidang, mulai dari pengembangan *web*, analisis data, kecerdasan buatan, hingga otomatisasi sistem.

2.10 *Overfitting dan Underfitting*

Overfitting adalah kondisi di mana model terlalu menyesuaikan diri dengan data latih, sehingga tidak dapat bekerja dengan baik pada data baru. Sedangkan *Underfitting* adalah kondisi di mana model gagal menangkap pola yang cukup dari data, sehingga performanya buruk baik pada data latih maupun data baru (Ganda & Bunyamin, 2021).



Gambar 21. Ilustrasi *overfitting* (a) dan *underfitting* (b) (Taye, 2023).

Gambar 21. memperlihatkan perbedaan antara *overfitting* dan *underfitting*. Pada ilustrasi (a), model mengalami *overfitting* karena terlalu menyesuaikan diri dengan data latih sehingga hasil klasifikasi hanya baik pada data tersebut, namun buruk pada data uji. Sebaliknya pada ilustrasi (b), terjadi *underfitting* di mana model gagal mengenali pola yang sebenarnya ada pada data. *Underfitting* terjadi ketika error pada data latih masih tinggi, sedangkan *overfitting* ditandai dengan adanya perbedaan yang besar antara error data latih dan data uji (Zhang et al., 2019).

2.11 *Augmentasi Data*

Augmentasi data adalah teknik yang digunakan untuk memperluas dataset dengan cara menghasilkan variasi dari data yang sudah ada. Teknik ini

sering digunakan dalam pembelajaran mesin, khususnya dalam *computer vision*, untuk meningkatkan performa model dengan menghasilkan variasi data yang lebih luas tanpa perlu mengumpulkan data tambahan. Augmentasi dapat dilakukan dengan berbagai metode, seperti rotasi, *flipping*, *cropping*, perubahan kecerahan, dan berbagai transformasi lainnya, yang bertujuan untuk membuat model lebih *robust* terhadap variasi di dunia nyata (Shorten & Khoshgoftaar, 2019).

2.12 *Hyperparameter*

Hyperparameter berperan penting dalam menentukan hasil *training* dan akurasi prediksi. *Hyperparameter* adalah nilai-nilai numerik yang tidak dipelajari dari data, tetapi ditentukan sebelum algoritma dijalankan. Penyetelan *hyperparameter* (*hyperparameter tuning*) dilakukan untuk memilih nilai yang memungkinkan model bekerja secara optimal dan efisien, sekaligus mencegah *underfitting* atau *overfitting* dalam proses pelatihan (Hariz *et al.*, 2022). Ada sejumlah parameter yang dapat dioptimalkan melalui *hyperparameter tuning*, berikut beberapa di antaranya.

2.12.1 *Batch Size*

Batch size adalah jumlah sampel data yang digunakan antara pembaruan bobot model dalam satu batch. Pemilihan batch size dipengaruhi oleh ukuran *dataset* dan dapat mempengaruhi kinerja serta waktu pelatihan. *Batch size* yang terlalu besar mempercepat konvergensi, tetapi menghasilkan *set* bobot akhir yang kurang optimal. Sebaliknya, jika *batch size* terlalu kecil, algoritma bisa kehilangan variasi yang seharusnya ada dalam distribusi data, yang dapat menyebabkan proses pelatihan menjadi banyak *noise* (Hariz *et al.*, 2022).

2.12.2 *Optimizer*

Optimizer adalah algoritma yang digunakan untuk memperbarui bobot model guna meminimalkan *loss* antara hasil prediksi dan target sebenarnya. Salah satu *optimizer* yang sering digunakan adalah *Adam (Adaptive Moment Estimation)*, yang menggabungkan kelebihan *Momentum* dan *RMSProp*. Adam dikenal efisien secara komputasi, hemat memori, dan mampu mengoptimalkan fungsi yang kompleks dan *non-convex*, sehingga cocok untuk berbagai kebutuhan dalam pembelajaran mesin (Rahmadhani & Marpaung, 2023). Dalam penggunaannya, Adam memerlukan pengaturan *learning rate*, yaitu parameter penting yang menentukan besarnya langkah koreksi bobot selama proses pelatihan (Antoni *et al.*, 2023).

2.12.3 *Epoch*

Epoch adalah jumlah putaran penuh yang dilakukan selama pelatihan menggunakan seluruh *dataset*. Semakin tinggi jumlah *epoch* yang digunakan, umumnya akurasi model akan meningkat, sedangkan nilai *loss* cenderung menurun. Iterasi adalah satu kali proses *forward pass*, di mana data melewati jaringan untuk menghasilkan prediksi, dan *backward pass*, di mana bobot diperbarui berdasarkan *error* yang dihitung (Muhammad & Wibowo, 2021). Dalam satu *epoch*, terdapat sejumlah iterasi yang sesuai dengan jumlah *batch* yang digunakan. Jumlah *epoch* dan iterasi yang tepat sangat penting untuk memastikan model mencapai performa optimal tanpa mengalami *overfitting* maupun *underfitting*.

2.12.4 *Learning Rate*

Learning rate adalah parameter penting dalam proses pelatihan model yang menentukan seberapa besar perubahan bobot jaringan pada setiap langkah pembelajaran. Jika nilai *learning rate* terlalu kecil, pelatihan akan berlangsung sangat lambat karena perubahan bobotnya sedikit demi sedikit. Sebaliknya, jika nilainya terlalu besar, proses pelatihan bisa menjadi tidak stabil dan model sulit mencapai hasil yang optimal karena perubahan bobotnya terlalu drastis (Rismiyati & Luthfiarta, 2021). Oleh karena itu, pemilihan *learning rate* yang tepat sangat mempengaruhi kecepatan dan hasil akhir dari proses pelatihan model.

2.12.5 *Early Stopping*

Early stopping adalah teknik dalam pelatihan model yang digunakan untuk mencegah *overfitting* dengan cara menghentikan proses pelatihan secara otomatis ketika performa model pada data validasi tidak menunjukkan peningkatan dalam beberapa *epoch* berturut-turut (Bachri & Gunawan, 2024). Jadi, kalau akurasi validasi atau nilai *loss* tidak membaik dalam waktu tertentu, pelatihan akan dihentikan lebih awal sebelum seluruh *epoch* selesai.

2.12.6 *Dropout*

Dropout adalah teknik regularisasi untuk mengurangi *overfitting* pada jaringan saraf tiruan, khususnya pada arsitektur jaringan dalam (*deep neural networks*). Teknik ini bekerja dengan cara menonaktifkan secara acak sejumlah unit neuron beserta koneksinya selama proses pelatihan. Hal ini mencegah *neuron* bergantung satu sama lain, sehingga jaringan menjadi lebih kuat dalam melakukan generalisasi. Saat pengujian, semua *neuron* diaktifkan kembali dan

bobot disesuaikan, menghasilkan prediksi yang merepresentasikan rata-rata dari berbagai jaringan kecil yang dilatih (Srivastava *et al.*, 2014).

2.12.7 *ReduceLROnPlateau*

ReduceLROnPlateau merupakan metode penjadwalan *learning rate* yang bekerja dengan menurunkan nilai *learning rate* secara otomatis ketika nilai *loss* pada data validasi tidak menunjukkan perbaikan selama sejumlah *epoch* tertentu. Berdasarkan penelitian oleh Al-Kababji *et al.*, (2022) metode ini mampu mempercepat proses konvergensi dan menghasilkan nilai *loss* validasi yang kompetitif, bahkan dalam beberapa kasus lebih unggul dibandingkan metode *OneCycleLR*. Untuk mengimplementasikan *ReduceLROnPlateau*, diperlukan beberapa parameter, yaitu nilai awal *learning rate*, metrik evaluasi pada data validasi, nilai *patience* sebagai ambang waktu menunggu sebelum penurunan dilakukan, serta nilai *factor* sebagai pengali untuk menurunkan *learning rate*.

2.13 Evaluasi Metode

Evaluasi metode pada *deep learning* adalah proses menilai kemampuan model yang telah dilatih dalam menyelesaikan tugas tertentu, seperti klasifikasi, deteksi objek, atau segmentasi gambar. Tujuannya adalah untuk mengetahui apakah metode tersebut sudah optimal atau perlu perbaikan. Salah satu cara untuk melakukan evaluasi model klasifikasi adalah dengan menggunakan *confusion matrix*. *Confusion matrix* adalah salah satu metode yang digunakan untuk mengevaluasi kualitas performa model klasifikasi. Penelitian ini cocok menggunakan *multiclass confusion matrix* karena data memiliki lebih dari dua kelas (*multiclass classification*).

Tabel 3. *Multiclass Confusion Matrix* (Markoulidakis *et al.*, 2021)

		Predicted Class			
		C ₁	C ₂	...	C _N
Actual Class	C ₁	C _{1,1}	FP	...	C _{1,N}
	C ₂	FN	TP	...	FN
	...	...	...	...	
	C _N	C _{N,1}	FP	...	C _{N,N}

Keterangan:

- C₁, C₂, ..., C_n : Label kelas (*class labels*) dalam *dataset*
- Diagonal utama (C₁₁, C₂₂, ..., C_{nn}) : Jumlah data yang diprediksi benar atau *True Positive* (TP)
- Sel di luar diagonal (C_{ij}, $i \neq j$) : Jumlah data yang diprediksi salah
- FP (*False Positive*) : Jumlah data yang diprediksi sebagai C_j, tetapi sebenarnya berasal dari kelas lain
- FN (*False Negative*) : Jumlah data yang berasal dari C_i, namun diprediksi sebagai kelas yang berbeda

Tabel 3. menampilkan *multiclass confusion matrix*, *matrix* ini berbentuk NxN, di mana N adalah banyaknya *class* (C₁, C₂, ..., C_N) (Markoulidakis *et al.*, 2021). Pada *confusion matrix*, evaluasi kinerja secara umum dilakukan dengan menghitung nilai akurasi, presisi, *recall*, dan *F1-score* (Handoko *et al.*, 2024).

2.13.1 Akurasi (*Accuracy*)

Akurasi adalah ukuran seberapa banyak data yang berhasil diprediksi dengan benar oleh model dibandingkan dengan seluruh data yang diuji (Markoulidakis *et al.*, 2021). Metrik ini menggambarkan kinerja umum dari model secara keseluruhan. Dinyatakan dengan persamaan (1).

$$Accuracy = \frac{\sum_{i=1}^N TP(C_i)}{\sum_{i=1}^N \sum_{j=1}^N C_{ij}} \dots \dots \dots (1)$$

Keterangan:

$TP(C_i)$: jumlah data kelas C_i yang diprediksi benar

C_{ij} : jumlah data kelas aktual i yang diprediksi sebagai kelas j

2.13.2 Presisi per Kelas (*Positive Predictive Value*)

Precision menunjukkan seberapa tepat model saat memprediksi suatu kelas (Markoulidakis *et al.*, 2021). Artinya, dari semua data yang diprediksi sebagai kelas itu, berapa banyak yang benar-benar berasal dari kelas tersebut. Presisi dinyatakan dengan persamaan (2).

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

Keterangan:

$TP(C_i)$: Prediksi benar untuk kelas C_i

$FP(C_i)$: Data dari kelas lain yang salah diprediksi sebagai C_i

2.13.3 Recall per Kelas (*True Positive Rate*)

Recall menunjukkan seberapa banyak data dari suatu kelas yang berhasil dikenali dengan benar oleh model. Metrik ini penting jika fokusnya adalah menangkap semua data dari kelas tertentu, termasuk yang sulit dikenali (Markoulidakis *et al.*, 2021). *Recall* dinyatakan dengan persamaan (3).

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

Keterangan:

$TP(C_i)$: data kelas C_i yang diprediksi dengan benar

$FN(C_i)$: data kelas C_i yang diprediksi sebagai kelas lain

2.13.4 *F1-Score* per Kelas

F1-score adalah metrik evaluasi yang menggabungkan *precision* dan *recall* dalam satu nilai untuk setiap kelas. Metrik ini digunakan untuk mengukur keseimbangan antara ketepatan model (*precision*) dan kemampuannya dalam mengenali kelas yang benar (*recall*), khususnya saat distribusi data antar kelas tidak merata (Markoulidakis *et al.*, 2021). *F1-Score* dinyatakan dengan persamaan (4).

$$F1(C_i) = \frac{2 \times TPR(C_i) \times PPV(C_i)}{TPR(C_i) + PPV(C_i)} \dots \dots \dots (4)$$

Keterangan:

$TPR(C_i)$: *True Positive Rate* untuk kelas i

$PPV(C_i)$: *Positive Predictive Value* atau *precision* untuk kelas i

III. METODE PENELITIAN

3.1. Waktu dan Tempat Penelitian

3.1.1 Tempat Penelitian

Penelitian dilakukan di Laboratorium Komputasi Dasar Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

3.1.2 Waktu Penelitian

Penelitian ini dilakukan pada semester ganjil tahun akademik 2024/2025 sampai semester genap tahun akademik 2024/2025 dengan perkiraan waktu bulan November 2024 – Juli 2025. Untuk *timeline* penelitian dapat dilihat pada Tabel 4.

Tabel 4. *Timeline* Penelitian[illegible]

3.2. Perangkat Penelitian

3.2.1 Perangkat Keras (*hardware*)

Perangkat keras yang digunakan dalam penelitian ini adalah laptop dengan detail sebagai berikut :

Merk : Asus
 Tipe : Vivobook Flip 14
 Model : TP470EZ
 Processor : Intel® Core™ i7-1165G7
 GPU : Intel® Iris Xe Graphics
 RAM : 16GB LPDDR4X
 Penyimpanan : 512GB M.2 NVMe™ PCIe® 3.0 SSD

3.2.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam proses penelitian ini adalah sebagai berikut :

2. Sistem operasi: *Windows 11 Home 64-bit*

Sistem operasi digunakan sebagai *platform* untuk menjalankan berbagai perangkat lunak yang diperlukan dalam proses penelitian.

3. Text Editor: *Google Colab*

Google Colab adalah *platform* berbasis *cloud* yang memungkinkan untuk menjalankan kode program python. Dilengkapi GPU gratis yang dapat mempercepat proses *Training* model dan integrasi *Google Drive*, *Colab* memudahkan pengembangan proyek *machine learning* tanpa perlu instalasi *software* di komputer (Andarsyah & Yanuar, 2024).

4. Bahasa Pemrograman: *Python* versi 3.11.13

Python adalah bahasa pemrograman yang banyak digunakan di bidang *machine learning*. *Python* menawarkan fitur untuk pustaka seperti TensorFlow, PyTorch, OpenCV, NumPy, Pandas, dan Matplotlib, yang mempermudah pengembangan model klasifikasi objek (Alfarizi *et al.*, 2023).

5. Penyimpanan Data: *Google Drive*

Google Drive merupakan layanan penyimpanan berbasis *cloud* yang diperkenalkan oleh Google pada tahun 2012. Layanan ini memungkinkan pengguna untuk menyimpan berbagai jenis file seperti dokumen, gambar, video, dan presentasi dengan kapasitas awal sebesar 15 GB. *Google Drive* dapat diakses selama perangkat terhubung ke *internet* dan telah terintegrasi dengan berbagai layanan Google lainnya. Dalam penelitian ini, *Google Drive* digunakan sebagai media penyimpanan *dataset* karena mendukung pengelolaan file secara praktis dan memudahkan integrasi dengan *Google Colab* selama proses pelatihan model (Safitri & Nasution, 2023).

6. *Labeling* dan *resize*: *Roboflow*

Roboflow digunakan untuk memberi label pada setiap gambar dalam *dataset* sesuai dengan kelas penyakit daun teh yang dimiliki, serta melakukan *resize* gambar agar ukuran citra konsisten dan sesuai dengan kebutuhan *input* model (Saepudin *et al.*, 2024).

7. *Library* Tensorflow 2.18.0

TensorFlow merupakan *library* utama dalam penelitian ini yang digunakan untuk membangun, melatih, dan mengevaluasi model *deep learning*, khususnya arsitektur *Convolutional Neural Network* (CNN). *Library* ini menyediakan fitur seperti

image preprocessing, *data augmentation*, serta implementasi arsitektur *pretrained* seperti *ResNet*. Selain itu, *TensorFlow* mendukung penggunaan GPU, yang mempercepat proses pelatihan model terutama pada data citra yang berukuran besar dan kompleks (Alfarizi *et al.*, 2023).

8. *Library NumPy 2.0.2*

NumPy digunakan sebagai dasar dalam pengolahan data numerik, terutama untuk mendukung operasi matematika seperti manipulasi *array* dan matriks. Dalam penelitian ini, *NumPy* berperan penting untuk menangani data gambar dalam bentuk numerik agar dapat diproses oleh model *deep learning* (Alfarizi *et al.*, 2023).

9. *Library Matplotlib 3.10.0*

Matplotlib adalah *library* visualisasi 2D dalam Python yang dirancang untuk membuat grafik interaktif maupun siap publikasi. *Library* ini mendukung berbagai jenis plot seperti grafik garis, batang, dan *scatter*, serta mampu menghasilkan output dalam format gambar seperti PNG dan PDF. Dalam penelitian ini, *Matplotlib* digunakan untuk menampilkan grafik akurasi dan *loss* selama pelatihan model, sehingga membantu dalam mengevaluasi performa klasifikasi citra secara visual (Hunter, 2007).

10. *Library Scikit-learn 1.6.1*

Scikit-learn adalah *library Python* yang menyediakan berbagai fungsi untuk analisis data dan *machine learning*. Scikit-learn menyediakan berbagai alat untuk mendukung proses pengolahan data gambar, seperti pembagian *dataset* dan penghitungan metrik evaluasi (misalnya *precision*, *recall*, *F1-score*, atau *confusion matrix*) (Pedregosa *et al.*, 2011).

11. *Library OpenCV 4.11.0*

OpenCV (*Open Source Computer Vision*) adalah *library* yang banyak digunakan untuk mengolah gambar dan video, terutama dalam bidang *computer vision* dan *machine learning*. Dalam klasifikasi gambar, OpenCV sangat berguna untuk tahap awal pemrosesan, seperti mengatur ukuran gambar, mengurangi noise, dan menyesuaikan kontras agar fitur penting lebih mudah dikenali oleh model. Dengan *preprocessing* yang baik, gambar yang diproses jadi lebih jelas, sehingga model klasifikasi bisa bekerja lebih akurat dalam membedakan setiap kelas yang ada (Bradski & Kaehler, 2008).

12. *Draw.io*

Draw.io adalah alat berbasis *web* yang digunakan untuk membuat berbagai jenis diagram seperti *flowchart*, diagram proses, dan diagram sistem. Aplikasi ini tidak memerlukan instalasi karena dapat diakses langsung melalui *browser* dan terhubung dengan *Google Drive*, sehingga memudahkan penyimpanan dan pengelolaan diagram. Dalam penelitian ini, *Draw.io* digunakan untuk memvisualisasikan alur kerja penelitian secara terstruktur, mulai dari tahapan pengumpulan data hingga evaluasi akhir (Harahap, 2018).

3.3 *Dataset*

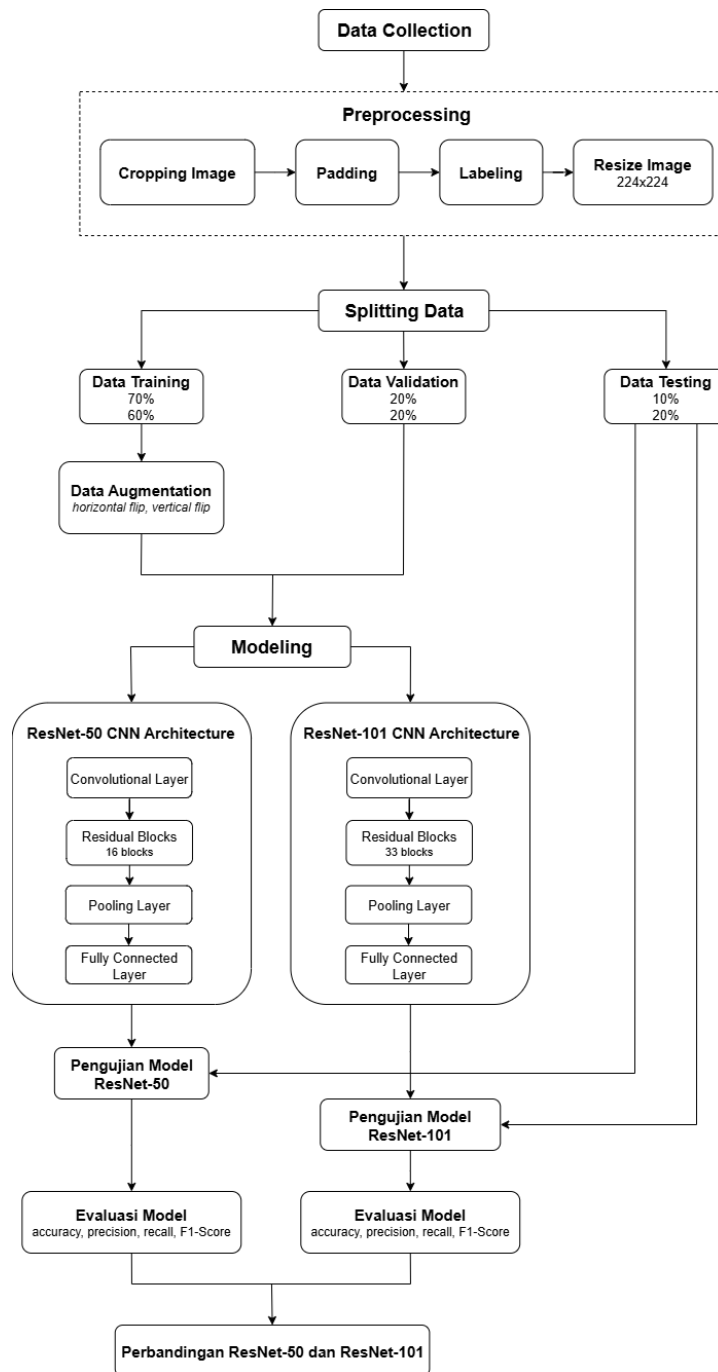
Penelitian ini menggunakan *dataset* “*Identifying Disease in Tea Leaves*”, yang diperoleh dari *Kaggle*. *Dataset* ini terdiri dari gambar daun teh yang terbagi ke dalam lima kelas, setiap kelas mewakili kondisi daun teh yang terinfeksi penyakit tertentu. Data ini akan dipisahkan menjadi data *training*, data *validation*, dan data *testing*. Informasi *dataset* secara detail dapat dilihat pada Tabel 5.

Tabel 5. *Dataset*

No	Nama Penyakit Daun Teh	Sampel 1	Sampel 2	Sampel 3	Jumlah Data	Persen tase
1	<i>Algal leaf spot</i>				113	20,11%
2	<i>Anthracnose</i>				99	17,62%
3	<i>Bird Eye Spot</i>				99	17,62%
4	<i>Brown Blight</i>				111	19,75%
5	<i>White Leaf Spot</i>				140	24,91%
Total					562	100%

3.4 Tahap Penelitian

Tahap penelitian adalah serangkaian langkah yang terstruktur dan berurutan untuk memastikan penelitian berjalan dengan baik dan mencapai hasil yang diinginkan. Tahapan-tahapan ini dapat dilihat pada Gambar 22.



Gambar 22. Tahap Penelitian.

3.4.1 *Data Collection (Pengumpulan Data)*

Dataset pada penelitian ini diperoleh dari *Kaggle* yang berisi gambar-gambar daun teh dengan berbagai kondisi, yang diklasifikasikan berdasarkan jenis penyakitnya. *Dataset* terdiri dari 6 kelas penyakit daun teh, yaitu, *Algal Leaf Spot*, *Anthrachnose*, *Bird Eye Spot*, *Brown Blight*, dan *White Leaf Spot*. Gambar disimpan dalam *google drive* untuk memudahkan penggunaan data ke *google colab*.

3.4.2 *Preprocessing*

Preprocessing adalah langkah-langkah yang dilakukan untuk menyiapkan data agar bisa digunakan dalam proses pelatihan model. Tahap *preprocessing* terdiri dari beberapa langkah berikut:

3.4.2.1 *Cropping Image*

Dilakukan *cropping* gambar secara manual untuk memastikan bahwa objek utama, yaitu daun teh, menjadi fokus utama dalam citra. Banyak gambar dalam dataset awal memiliki latar belakang yang terlalu luas, sehingga objek daun terlihat kecil dan tidak berada di pusat gambar. Hal ini dapat mempengaruhi proses pelatihan model karena fitur penting dari objek menjadi kurang terlihat secara optimal. Oleh karena itu, proses *cropping* dilakukan dengan memotong bagian latar belakang yang tidak diperlukan. Dengan demikian, gambar menjadi lebih seragam dan fokus terhadap objek.

3.4.2.2 *Padding*

Pada tahap pra-pemrosesan data, dilakukan proses *padding* untuk mengubah gambar menjadi berbentuk persegi sebelum di-*resize* ke ukuran 224×224 piksel, sesuai dengan ketentuan input pada model ResNet. Jika gambar langsung diubah ukurannya tanpa penyesuaian, maka bentuk objek di dalam gambar bisa menjadi tidak proporsional atau bahkan terdistorsi. Untuk mengatasi hal tersebut, ditambahkan *padding* pada sisi gambar yang lebih pendek. Dalam penelitian ini, warna padding ditentukan berdasarkan rata-rata warna dari bagian pinggir gambar (atas, bawah, kiri, dan kanan), sehingga hasil padding terlihat lebih menyatu dengan gambar asli.

3.4.2.3 *Labeling*

Pada tahap ini, proses labeling dilakukan untuk mengelompokkan gambar ke dalam kelas-kelas yang sesuai berdasarkan karakteristik visualnya. *Dataset* yang digunakan diperoleh dari Kaggle, di mana gambar-gambar telah tersusun dalam folder yang merepresentasikan masing-masing kelas. Untuk mempermudah pengelolaan dan penyesuaian format dataset, data tersebut kemudian diunggah ke *platform* Roboflow dalam proyek klasifikasi citra. Roboflow secara otomatis mengenali label berdasarkan nama folder, sehingga proses labeling dapat dilakukan secara terstruktur dan efisien. Tahap ini bertujuan untuk memastikan bahwa setiap gambar memiliki label yang tepat sebelum digunakan dalam proses pelatihan model klasifikasi.

3.4.2.4 *Resize Image*

Resize image adalah tahap mengubah ukuran *pixel* citra. Seluruh gambar diubah ukurannya menjadi 224x224 piksel menggunakan fitur *resize* pada Roboflow. Ukuran ini dipilih karena kompatibel dengan arsitektur jaringan konvolusional seperti ResNet, yang digunakan dalam penelitian ini. Proses *resize* ini juga bertujuan untuk menyamakan dimensi semua gambar, sehingga memudahkan proses pelatihan dan meningkatkan konsistensi model dalam mengenali pola visual pada masing-masing kelas.

3.4.3 *Splitting Data*

Dataset yang telah dikumpulkan dan diproses kemudian dibagi ke dalam tiga bagian utama, yaitu data *training*, *validation*, dan *testing*. Pembagian ini dilakukan secara manual menggunakan skrip Python dengan bantuan beberapa pustaka, antara lain *os*, *random*, dan *shutil*. *Dataset* yang digunakan sudah tersusun berdasarkan kelas masing-masing dalam folder terpisah, sehingga proses pembagian bisa dilakukan dengan lebih terstruktur. Dalam penelitian ini digunakan dua skenario pembagian data untuk mengevaluasi pengaruh komposisi data terhadap performa model. Skenario pertama membagi data menjadi 70% untuk *training*, 20% untuk *validation*, dan 10% untuk *testing*. Sedangkan skenario kedua menggunakan proporsi 60% untuk *training*, 20% untuk *validation*, dan 20% untuk *testing*.

3.4.3.1 *Data Training*

Data *training* merupakan data utama yang digunakan dalam proses pelatihan model. Fungsi dari data ini adalah untuk

membantu model mengenali pola, fitur, dan karakteristik visual dari masing-masing kelas. Pada skenario pertama, data *training* terdiri atas 70% dari keseluruhan dataset, sedangkan pada skenario kedua komposisinya menjadi 60%. Dengan jumlah data yang lebih besar, diharapkan model memiliki lebih banyak referensi visual untuk proses pembelajaran dan mampu mencapai generalisasi yang baik.

1. Augmentasi Data

Augmentasi data adalah teknik untuk meningkatkan jumlah dan variasi data pelatihan dengan cara memodifikasi gambar yang ada. Teknik augmentasi yang diterapkan antara lain pembalikan gambar secara horizontal (*horizontal flip*) dan vertikal (*vertical flip*). Teknik ini membantu meningkatkan kemampuan model untuk generalisasi dan mengurangi risiko *overfitting*.

3.4.3.2 Data Validation

Data *validation* digunakan untuk mengevaluasi kinerja model selama proses pelatihan berlangsung. *Dataset* ini tidak digunakan secara langsung dalam proses pembelajaran, melainkan sebagai alat bantu untuk melihat sejauh mana model dapat menggeneralisasi data yang belum dilatih. Baik pada skenario pertama maupun kedua, proporsi data *validation* yang digunakan adalah 20%. Data ini dipisahkan dari sisa *dataset* setelah pembagian data *training*, dan pemisahannya tetap mempertahankan proporsi kelas agar hasil evaluasi tidak bias terhadap kelas tertentu. Hasil dari proses validasi ini digunakan untuk memantau tanda-tanda *overfitting* dan membantu penyesuaian parameter pelatihan.

3.4.3.3 Data *Testing*

Data *testing* digunakan untuk mengukur performa akhir model terhadap data baru. Pada skenario pertama, data *testing* terdiri atas 10% dari total dataset, sedangkan pada skenario kedua 20%. *Testing set* berfungsi untuk menilai kemampuan generalisasi model dalam situasi nyata. Hasil dari pengujian ini digunakan untuk menghitung metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*, yang menggambarkan sejauh mana model mampu mengklasifikasikan citra dengan benar.

3.4.4 *Modeling*

Penelitian ini menggunakan model CNN (*Convolutional Neural Network*) dengan arsitektur ResNet-50 dan ResNet-101. Model ini dibangun menggunakan beberapa komponen utama, yaitu *Convolutional Layer*, *Residual Blocks*, *Average Pooling Layer*, dan *Fully Connected Layer*. Pada arsitektur ini, setiap *Convolutional Layer* tidak hanya melakukan operasi konvolusi untuk mengekstraksi fitur, tetapi juga mencakup *ReLU Activation*, *Batch Normalization*, dan *Max Pooling* untuk membantu proses pelatihan. ResNet-50 memiliki total 50 lapisan dengan 16 *residual block*, sementara ResNet-101 memiliki 101 lapisan dengan 33 *residual block*. *Residual block* ini menggunakan teknik *skip connection*, yang memungkinkan *input* awal blok dijumlahkan langsung dengan *output* blok tersebut. Teknik ini membantu model belajar lebih baik meskipun jumlah lapisannya sangat dalam. Pada bagian akhir model, *Average Pooling Layer* digunakan untuk mereduksi data menjadi ukuran yang lebih kecil, sebelum diteruskan ke *Fully Connected Layer* untuk menghasilkan hasil klasifikasi. Kedua arsitektur ini kemudian akan dibandingkan performanya dalam

mengklasifikasikan penyakit daun teh. Setelah model dibangun, tahap pelatihan dilakukan menggunakan data pelatihan yang telah diproses sebelumnya.

3.4.5 Pengujian Model

Pengujian model ResNet-50 dan ResNet-101 dengan menggunakan data *testing* yang telah dipisahkan sebelumnya. *Testing model* bertujuan untuk mengevaluasi seberapa baik kedua model dalam mengklasifikasikan data selain data *training* dan data *validation*.

3.4.6 Evaluasi Model

Evaluasi model dilakukan untuk mengukur performa model secara keseluruhan. Evaluasi model dilakukan menggunakan *Confusion Matrix* untuk menghitung metrik seperti akurasi, presisi, *recall*, dan *F1-Score*. *Confusion Matrix* membandingkan hasil prediksi model dengan label yang sebenarnya, dan metrik-metrik tersebut dihitung untuk menilai seberapa baik model dalam mengklasifikasikan data.

3.4.7 Perbandingan ResNet-50 & ResNet-101

Pada tahap ini, hasil dari model ResNet-50 dan ResNet-101 akan dibandingkan berdasarkan kinerja masing-masing dalam mengklasifikasikan penyakit daun teh. Perbandingan ini dilakukan dengan melihat hasil nilai akurasi terbaik untuk menentukan model mana yang lebih unggul dalam klasifikasi gambar.

V. SIMPULAN DAN SARAN

5.1. Simpulan

Penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja arsitektur *Convolutional Neural Network* (CNN) yaitu ResNet-50 dan ResNet-101 dalam mengklasifikasikan jenis penyakit pada daun teh. Berdasarkan hasil pengujian yang dilakukan terhadap berbagai skenario rasio data latih, validasi, dan uji variasi *batch size*, diperoleh beberapa kesimpulan sebagai berikut:

1. Arsitektur ResNet-50 menunjukkan kinerja yang cukup baik dalam mengklasifikasikan penyakit daun teh. Dari delapan skenario pengujian yang melibatkan variasi rasio data, *batch size*, dan teknik augmentasi, ResNet-50 mampu mencapai *accuracy* tertinggi sebesar 93,33% pada skenario 4 (rasio 70:20:10, *batch size* 32, dengan augmentasi). Grafik hasil pelatihan juga memperlihatkan tren peningkatan *accuracy* dan penurunan *loss* yang stabil, terutama pada konfigurasi yang melibatkan augmentasi. Hal ini menunjukkan bahwa ResNet-50 memiliki kemampuan yang cukup baik dalam mengenali pola visual, khususnya saat data pelatihan diperluas melalui teknik augmentasi.
2. Arsitektur ResNet-101 menunjukkan performa yang lebih unggul dibandingkan ResNet-50 dalam klasifikasi penyakit daun teh. Model ini mencatat *accuracy* tertinggi sebesar 93,33% pada skenario 1 dan skenario 2, serta secara konsisten menunjukkan nilai *accuracy* yang lebih tinggi di tujuh dari delapan skenario pengujian. Selain itu, grafik hasil pelatihan memperlihatkan bahwa ResNet-101 memiliki tren peningkatan *accuracy* yang lebih cepat serta *loss* yang lebih stabil,

menunjukkan kemampuan pembelajaran dan generalisasi yang lebih baik terhadap variasi citra.

3. Perbandingan kinerja antara ResNet-50 dan ResNet-101 berdasarkan delapan skenario pengujian menunjukkan bahwa ResNet-101 merupakan model yang lebih unggul secara keseluruhan. Berdasarkan evaluasi pada Tabel 21. ResNet-101 mencatat akurasi tertinggi pada sebagian besar skenario dan menunjukkan performa pelatihan yang lebih stabil. Meskipun ResNet-50 kompetitif pada konfigurasi tertentu, seperti skenario 4, konsistensinya masih di bawah ResNet-101. Dengan demikian, ResNet-101 dapat disimpulkan sebagai arsitektur yang lebih andal dan efektif untuk klasifikasi citra penyakit daun teh.

5.2 Saran

Saran yang diberikan untuk penelitian selanjutnya adalah sebagai berikut:

1. Menggunakan *dataset* dengan jumlah data yang lebih besar. Penambahan jumlah data akan memberikan model lebih banyak variasi untuk dipelajari, sehingga dapat meningkatkan akurasi dan kemampuan generalisasi model.
2. Membangun aplikasi berbasis *Android* yang mampu melakukan klasifikasi penyakit daun teh secara langsung melalui kamera perangkat. Dengan adanya aplikasi ini, proses identifikasi penyakit dapat dilakukan secara praktis dan cepat di lapangan.
3. Adanya kemungkinan lebih dari satu penyakit menyerang daun secara bersamaan disarankan untuk mengembangkan model klasifikasi *multi-label* pada penelitian selanjutnya, agar deteksi penyakit lebih akurat dan sesuai kondisi lapangan

DAFTAR PUSTAKA

- Abadi, A. N., & Wibowo, A. T. (2021). Klasifikasi Spesies Anthurium Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *EProceedings of Engineering*, 8(4), 4152–4170.
- Adelia, R., Khairunisa, N., & Zulfiqui, R. (2024). Implementasi Convolutional Neural Network (Cnn) Dalam Mendeteksi Sampah Organik, Plastik, Dan Kertas. *JUTIM (Jurnal Teknik Informatika Musirawas)*, 9(1), 29–37. <https://doi.org/10.32767/jutim.v9i1.2233>
- Agnisoonu. (2011). *Tea leaves: A view from Munnar*. Flickr. <https://www.flickr.com/photos/pensive-reflections/5939414540/>
- Al-Kababji, A., Bensaali, F., & Dakua, S. P. (2022). Scheduling Techniques for Liver Segmentation: ReduceLRonPlateau vs OneCycleLR. *Communications in Computer and Information Science*, 1589 CCIS, 204–212. https://doi.org/10.1007/978-3-031-08277-1_17
- Alberto, J., & Hermanto, D. (2023). Klasifikasi Jenis Burung Menggunakan Metode CNN ResNet-50. *Jurnal Teknik Informatika Dan Sistem Informasi*, 10(3), 34–36.
- Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning. *Karya Ilmiah Mahasiswa Bertauhid (Karimah Tauhid)*, 2(1), 1–6.
- Andarsyah, R., & Yanuar, A. (2024). *Sentimen Analisis Aplikasi Posaja Pada Google Playstore Untuk Peningkatan Pospay Superapp Menggunakan Support Vector Meachine*. 16(2), 1–7. <https://ejurnal.ulbi.ac.id/index.php/informatika/article/view/3533>.

- Anggarkusuma, R., Musdar, I. A., Studi, P., Informatika, T., Studi, P., Informasi, S., Citra, K., & Network, C. N. (2024). *Klasifikasi Citra Komponen Sepeda Motor Menggunakan Algoritma CNN dengan Arsitektur Mobilenet*. 02, 1-12.
- Anjarsari, I. R. D. (2022). Rekayasa budidaya dan penanganan pascapanen untuk meningkatkan kualitas teh Indonesia sebagai minuman fungsional kaya antioksidan. *Kultivasi*, 21(2), 152–158. <https://doi.org/10.24198/kultivasi.v21i2.36027>
- Anshori, R. B., Tsp, H. F., & Siadari, T. S. (2022). Klasifikasi Citra Kanker Serviks Menggunakan Deep Residual Network. *E-Proceeding of Engineering*, 8(6), 3163–3170.
- Antoni, A., Rohana, T., & Pratama, A. R. (2023). Implementasi Algoritma Convolutional Neural Network Untuk Klasifikasi Citra Kemasan Kardus Defect dan No Defect. *Building of Informatics, Technology and Science (BITS)*, 4(4), 1941–1950. <https://doi.org/10.47065/bits.v4i4.3270>
- Bachri, C. M., & Gunawan, W. (2024). JEPIN (Jurnal Edukasi dan Penelitian Informatika) Deteksi Email Spam menggunakan Algoritma Convolutional Neural Network (CNN). *Edukasi Dan Penelitian Informatika*, 10(1), 88–94.
- Bayu, T. (2022). Identifikasi Citra Plasmodium Guna Mendeteksi Parasit Penyakit Malaria dengan Menggunakan Algorithm Convolutional Neural Network. *Science Tech: Jurnal Ilmu Pengetahuan Dan Teknologi*, 8(2), 25–35. <https://doi.org/10.30738/st.vol8.no2.a12796>
- Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library* (M. Loukides (ed.)). O'Reilly Media.
- Das, P. K., & Rupa, S. S. (2023). ResNet for Leaf-based Disease Classification in Strawberry Plant. *International Journal of Applied Methods in Electronics and Computers*, 11(3), 151–157. <https://doi.org/10.58190/ijamec.2023.42>
- Dijaya, R. (2023). *Buku Ajar Pengolahan Citra Digital* (M. K. M. Tanzil M, S.H & M. P. Mahardika D.KW (eds.)). UMSIDA Press.

- Ganda, L. H., & Bunyamin, H. (2021). Penggunaan Augmentasi Data pada Klasifikasi Jenis Kanker Payudara dengan Model Resnet-34. *Jurnal Strategi*, 3(1), 187–193.
- Gnanamangai, B. M., Ponmurugan, P., Jeeva, S. E., Manjugarunambika, K., Elango, V., Hemalatha, K., Kakati, J. P., Mohanraj, R., & Prathap, S. (2017). Biosynthesised silver and copper nanoformulation as foliar spray to control bird's eye spot disease in tea plantations. *IET Nanobiotechnology*, 11(8), 917–928. <https://doi.org/10.1049/iet-nbt.2017.0023>
- Handoko, A. A., Rosid, M. A., & Indahyanti, U. (2024). Implementasi Convolutional Neural Network (CNN) Untuk Pengenalan Tulisan Tangan Aksara Bima. *Smatika Jurnal*, 14(01), 96–110. <https://doi.org/10.32664/smatika.v14i01.1196>
- Harahap, S. H. (2018). Analisis Pembelajaran Sistem Akuntansi Menggunakan Draw.Io Sebagai Perancangan Diagram Alir. *Prosiding Seminar Nasional Multidisiplin Ilmu Universitas Asahan 2018, November*, 101–103.
- Hariz, F. A., Yulita, I. N., & Suryana, I. (2022). Human Activity Recognition Berdasarkan Tangkapan Webcam Menggunakan Metode Convolutional Neural Network (CNN) Dengan Arsitektur MobileNet. *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi*, 3(4), 103–115. <https://doi.org/10.30630/jitsi.3.4.97>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- Hendrawan, I. E., Adam, R. I., & Rozikin, C. (2023). Klasifikasi Retak Ban Kendaraan Menggunakan Arsitektur ResNet50. *SATIN - Sains Dan Teknologi Informasi*, 9(1), 22–32. <https://doi.org/10.33372/stn.v9i1.902>
- Hidayat, A., & Ernawati, T. (2020). Metode Klasifikasi Algoritma Convolutional Neural Network (CNN) Pada Penyakit Daun Teh. 97–102.

- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science and Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- Ibrahim, N., Lestary, G. A., Hanafi, F. S., Saleh, K., Pratiwi, N. K. C., Haq, M. S., & Mastur, A. I. (2022). Klasifikasi Tingkat Kematangan Pucuk Daun Teh menggunakan Metode Convolutional Neural Network. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(1), 162–176. <https://doi.org/10.26760/elkomika.v10i1.162>
- Izzreen, N. Q., & Fadzelly, M. (2013). Phytochemicals and antioxidant properties of different parts of *Camellia sinensis* leaves from Sabah Tea plantation in Sabah, Malaysia. *International Food Research Journal*, 20(1), 307–312.
- Lasniari, S., Jasril, J., Sanjaya, S., Yanto, F., & Affandes, M. (2022). Klasifikasi Citra Daging Babi dan Daging Sapi Menggunakan Deep Learning Arsitektur ResNet-50 dengan Augmentasi Citra. *Jurnal Sistem Komputer Dan Informatika (JSON)*, 3(4), 450–457. <https://doi.org/10.30865/json.v3i4.4167>
- Li, X., Xu, X., He, X., Wei, X., & Yang, H. (2023). Intelligent Crack Detection Method Based on GM-ResNet. *Sensors (Basel, Switzerland)*, 23(20), Article 8369. <https://doi.org/10.3390/s23208369>
- Markoulidakis, I., Rallis, I., Georgoulas, I., Kopsiaftis, G., Doulamis, A., & Doulamis, N. (2021). Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem. *Technologies*, 9(4). <https://doi.org/10.3390/technologies9040081>
- Mohan, A., & Jayakrishnan, A. (2024). Leaf-CAP: A capsule network-based tea leaf disease recognition and detection. In S. Krishnan, A. J. Anand, N. Prasanth, S. Goundar, & C. Ananth (Eds.), *Predictive analytics in smart agriculture*. CRC Press. <https://doi.org/10.1201/9781003391302>
- Muhammad, S., & Wibowo, A. T. (2021). Klasifikasi Tanaman Aglonema Berdasarkan Citra Daun Menggunakan Convolutional Neural Network (CNN). *E-Proceeding of Engineering*, 8(5), 10621–10636. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15673>

- Munantri, N. Z., Sofyan, H., & Florestiyanto, M. Y. (2020). Aplikasi Pengolahan Citra Digital Untuk Identifikasi Umur Pohon. *Telematika*, 16(2), 97. <https://doi.org/10.31315/telematika.v16i2.3183>
- Mustikasari, Ramli, A. R., & Gibran, A. K. (2023). Analisis Kinerja Algoritma Machine Learning Untuk Deteksi Penyakit Daun Teh Dengan Particle Swarm Optimization. *Journal of Embedded Systems, Security and Intelligent Systems*, 4(November), 97–106. <https://doi.org/10.59562/jessi.v4i2.708>
- Nima, R., & Shila, F. (2020). Crack classification in rotor-bearing system by means of wavelet transform and deep learning methods: an experimental investigation. *Journal of Mechanical Engineering, Automation and Control Systems*, 1(2), 102–113. <https://doi.org/10.21595/jmeacs.2020.21799>
- Oktaviana, U. N., Hendrawan, R., Annas, A. D. K., & Wicaksono, G. W. (2021). Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(6), 1216–1222. <https://doi.org/10.29207/resti.v5i6.3607>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Prihatiningsih, N., Djatmiko, H. A., & Erminawati, E. (2020). Komponen epidemi penyakit antraknosa pada tanaman cabai di kecamatan baturaden kabupaten Banyumas. *Jurnal AGRO*, 7(2), 203–212. <https://doi.org/10.15575/8000>
- Putra, I. P., Rusbandi, R., & Alamsyah, D. (2022). Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network. *Jurnal Algoritme*, 2(2), 102–112. <https://doi.org/10.35957/algoritme.v2i2.2360>
- Rahmadhani, U. S., & Marpaung, N. L. (2023). Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN. *Jurnal Informatika: Jurnal Pengembangan IT*, 8(2), 169–173. <https://doi.org/10.30591/jpit.v8i2.5229>

- Rahman, S., Sembiring, A., Siregar, D., Khair, H., Gusti Prahmana, I., Puspadini, R., & Zen, M. (2023). Python : Dasar Dan Pemrograman Berorientasi Objek. In T. Media (Ed.), *Tahta Media Group*.
- Ramadhani, F., Satria, A., & Salamah, S. (2023). Implementasi Algoritma Convolutional Neural Network dalam Mengidentifikasi Dini Penyakit pada Mata Katarak. *Sudo Jurnal Teknik Informatika*, 2(4), 167–175. <https://doi.org/10.56211/sudo.v2i4.408>
- Rismaniyati, & Luthfiarta, A. (2021). VGG16 Transfer Learning Architecture for Salak Fruit Quality Classification. *Telematika*, 18(1), 37–48. <https://doi.org/10.31315/telematika.v18i1.4025>
- Rizaty, M. A. (2023). *Produksi teh di Indonesia turun menjadi 136.800 ton pada 2022*. Dataindonesia.Id. <https://dataindonesia.id/agribisnis-kehutanan/detail/produksi-teh-di-indonesia-turun-menjadi-136800-ton-pada-2022>
- Saepudin, Sujana, N., Mutoffar, M. M., & Haryanto, A. A. (2024). *Analisis Kinerja Yolov8 Optimalisasi Roboflow Untuk Deteksi Ekspresi Wajah Emosional Dengan Machine Learning*. 06(02), 115–124.
- Safitri, K., & Nasution, I. P. (2023). Analisis Penggunaan Aplikasi Google Drive Sebagai Media Penyimpanan Data. *Jurnal Sains Dan Teknologi (JSIT)*, 3(3), 220–223. <https://doi.org/10.47233/jsit.v3i2.891>
- Samuel, A. L. (1959). Some Studies in Machine Learning. *IBM Journal of Research and Development*, 3(3), 210–229.
- Santoso, P., Abijono, H., & Anggreini, N. L. (2021). Algoritma Supervised Learning Dan Unsupervised. *Jurnal Teknologi Terapan*, 4(2), 315–318.
- Setyawan, W. D., Nilogiri, A., & A'yun, Q. (2023). Implementasi Convolution Neural Network (Cnn) Untuk Klasifikasi Pada Citra Ikan Cupang Hias. *JTIK (Jurnal Teknik Informatika Kaputama)*, 7(1), 101–110. <https://doi.org/10.59697/jtik.v7i1.45>
- Sholehurrohman, R. (2021). Pengembangan Yolov3 Dengan Fitur Ekstraktor

MobileNetv2 Untuk Deteksi Dan Klasifikasi Kendaraan Bergerak. *Thesis*.

- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1). <https://doi.org/10.1186/s40537-019-0197-0>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.
- Syulistyo, A. R., Hormansyah, D. S., & Saputra, P. Y. (2020). SIBI (Sistem Isyarat Bahasa Indonesia) translation using Convolutional Neural Network (CNN). *IOP Conference Series: Materials Science and Engineering*, 732(1). <https://doi.org/10.1088/1757-899X/732/1/012082>
- Taye, M. M. (2023). Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. *Computation*, 11(3). <https://doi.org/10.3390/computation11030052>
- Zahisham, Z., Lee, C. P., & Lim, K. M. (2020). Food Recognition with ResNet-50. *IEEE International Conference on Artificial Intelligence in Engineering and Technology, IICAIET 2020*, 0–4. <https://doi.org/10.1109/IICAIET49801.2020.9257825>
- Zhang, H., Zhang, L., & Jiang, Y. (2019). Overfitting and Underfitting Analysis for Deep Learning Based End-to-end Communication Systems. *2019 11th International Conference on Wireless Communications and Signal Processing, WCSP 2019*, 1–6. <https://doi.org/10.1109/WCSP.2019.8927876>