

**IMPLEMENTASI *CONVOLUTIONAL NEURAL NETWORK*
MENGUNAKAN ARSITEKTUR *XCEPTION* UNTUK IDENTIFIKASI
PENYAKIT DAUN PADA TANAMAN DURIAN**

(Skripsi)

Oleh

FARHAT FEBRIANTO

NPM 2115061012



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2025

**IMPLEMENTASI *CONVOLUTIONAL NEURAL NETWORK*
MENGUNAKAN ARSITEKTUR *XCEPTION* UNTUK IDENTIFIKASI
PENYAKIT DAUN PADA TANAMAN DURIAN**

Oleh

FARHAT FEBRIANTO

Skripsi

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

Pada

**Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2025

ABSTRAK

IMPLEMENTASI *CONVOLUTIONAL NEURAL NETWORK* MENGUNAKAN ARSITEKTUR *XCEPTION* UNTUK IDENTIFIKASI PENYAKIT DAUN PADA TANAMAN DURIAN

Oleh

FARHAT FEBRIANTO

Durian (*Durio zibethinus Murr*) merupakan salah satu jenis buah tropis yang tergolong dalam famili *Bombacaceae* yang memiliki nilai ekonomis tinggi. Pengelolaan penyakit tanaman umumnya masih menggunakan metode tradisional yang mengandalkan inspeksi visual oleh ahli, pendekatan ini kurang efektif karena memerlukan tenaga kerja yang banyak, biaya operasional, dan kurang terjangkau oleh petani skala kecil di negara berkembang. Selain itu, keterbatasan pengetahuan petani yang kurang berpengalaman dalam mengklasifikasi penyakit pada tanaman dapat menyebabkan pengambilan keputusan yang kurang tepat, sehingga mengakibatkan produktivitas pertanian menurun. Seiring dengan perkembangan teknologi, pendekatan berbasis *deep learning* menjadi solusi inovatif dalam mendukung bidang pertanian. Pada penelitian ini menggunakan *CNN* dengan arsitektur *xception* yang memanfaatkan *depth-wise separable convolution* untuk menangkap pola dan fitur yang kompleks dengan lebih efisien. Upaya peningkatan performa dilakukan melalui penyesuaian *fully connected layer* pada arsitektur *xception*, optimasi *hyperparameter* serta *fine tuning*. Dataset yang digunakan terdiri dari lima kelas yaitu *Algal leaf spot*, *Allocaridara attack*, *Healthy leaf*, *Leaf blight*, dan *Phomopsis leaf spot*. Hasil penelitian menunjukkan bahwa *fully connected layer* dengan 128 neuron, 60 epoch, batch size 8, learning rate 0.001, dan optimizer adam, serta *fine tuning* memberikan hasil yang paling optimal dengan nilai *accuracy* 95%, *recall* 94%, *precision* 95% dan *f1-score* 95%.

Kata kunci : durian, penyakit daun, *cnn*, *xception*, *hyperparameter*, *fine tuning*, *fully connected layer*.

ABSTRACT

IMPLEMENTATION OF CONVOLUTIONAL NEURAL NETWORK USING XCEPTION ARCHITECTURE TO IDENTIFY LEAF DISEASES IN DURIAN PLANTS

By

FARHAT FEBRIANTO

Durian (Durio zibethinus Murr) is a tropical fruit belonging to the Bombacaceae family and has high economic value. Plant disease management generally still uses traditional methods that rely on visual inspection by experts. This approach is ineffective because it requires a lot of labor, operational costs, and is less affordable for small-scale farmers in developing countries. In addition, the limited knowledge of farmers who lack experience in classifying plant diseases can lead to inaccurate decisions, resulting in decreased agricultural productivity. Along with technological developments, deep learning-based approaches have become an innovative solution to support the agricultural sector. This study uses a CNN with an xception architecture that utilizes depth-wise separable convolution to capture complex patterns and features more efficiently. Efforts to improve performance are carried out through adjusting the fully connected layer in the xception architecture, hyperparameter optimization, and fine tuning. The dataset used consists of five classes: Algal leaf spot, Allocaridara attack, Healthy leaf, Leaf blight, and Phomopsis leaf spot. The results of the study showed that a fully connected layer with 128 neurons, 60 epochs, a batch size of 8, a learning rate of 0.001, and an Adam optimizer, as well as fine tuning, provided the most optimal results with an accuracy value of 95%, a recall of 94%, a precision of 95%, and an f1-score of 95%.

Kata kunci : durian, penyakit daun, cnn, xception, hyperparameter, fine tuning, fully connected layer.

Judul Skripsi : IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK MENGGUNAKAN XCEPTION UNTUK IDENTIFIKASI PENYAKIT DAUN PADA TANAMAN DURIAN

Nama Mahasiswa : Farhat Febrianto

Nomor Pokok Mahasiswa : 2115061012

Program Studi : Teknik Informatika

Jurusan : Teknik Elektro

Fakultas : Teknik

MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping


Yessi Mulyani, S.T., M.T.


Puput Budi Wintoro, S.Kom., M.T.I

NIP. 197312262000122001

NIP. 198410312019031004

2. Mengetahui

Ketua Jurusan

Ketua Program Studi

Teknik Elektro

Teknik Informatika


Herlinawati, S.T., M.T.


Yessi Mulyani, S.T., M.T.

NIP. 197103141999032001

NIP. 197312262000122001

4
MENGESAHKAN

1. Tim Penguji

Ketua

: Yessi Mulyani, S.T., M.T.



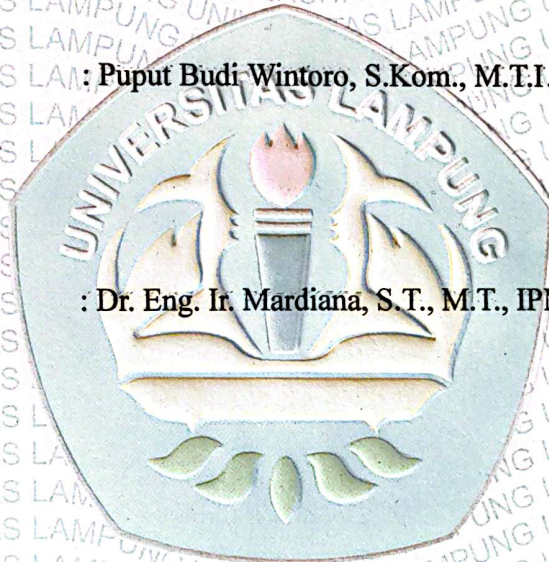
Sekretaris

: Puput Budi Wintoro, S.Kom., M.T.I.



Penguji

: Dr. Eng. Ir. Mardiana, S.T., M.T., IPM.



2. Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.

NIP. 197509282001121002



Tanggal Lulus Ujian Skripsi : 20 Agustus 2025

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi berjudul “Implementasi *Convolutional Neural Network* Menggunakan Arsitektur *Xception* Untuk Identifikasi Penyakit Daun Pada Tanaman Durian” sepenuhnya merupakan hasil karya saya sendiri. Apabila dikemudian hari terbukti pernyataan ini tidak benar, saya siap menerima sanksi sesuai dengan ketentuan hukum yang berlaku.

Bandar Lampung, 20 Agustus 2025

Penulis,



Farhat Febrianto

NPM. 2115061012

RIWAYAT HIDUP



Penulis dilahirkan di Kota Metro, Lampung pada tanggal 14 Februari 2003, sebagai anak kedua dari dua bersaudara, dari Bapak Haryanto dan Ibu Atik Rianawati. Penulis menyelesaikan jenjang pendidikan formal di SD Negeri 5 Metro pada tahun 2015, kemudian melanjutkan ke SMP Negeri 2 Metro dan lulus pada tahun 2018, serta menamatkan pendidikan menengah atas di SMA Negeri 4 Metro pada tahun 2021. Pada tahun 2021, penulis diterima sebagai mahasiswa Program Studi Teknik Informatika Universitas Lampung melalui jalur seleksi SNMPTN. Selama masa perkuliahan, penulis aktif berpartisipasi dalam berbagai kegiatan, antara lain :

1. Mengikuti kegiatan Studi Independen Bersertifikat dari Kementrian Pendidikan dan Budaya di mitra Bangkit Akademi pada tahun 2023.
2. Mengikuti kegiatan pelatihan *Vocational School Graduate Academy* dari Badan Pengembangan Sumber Daya Manusia dan Informatika dibawah Kementrian Komunikasi dan Informatika pada tahun 2022.
3. Menjadi anggota Himpunan Mahasiswa Teknik Elektro pada tahun 2022-2023.

MOTTO

“Nobody cares as much as you think. The spotlight you’re afraid of exists only in your mind”

(Raditya Dika)

“Tidak ada mimpi yang gagal, yang ada hanyalah mimpi yang tertunda. Cuma sekiranya teman-teman merasa gagal dalam mencapai mimpi. Jangan khawatir mimpi-mimpi lain bisa diciptakan”

(Windah Basudara)

“Take the risk or lose your change”

(Penulis)

PERSEMBAHAN

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik, serta hidayah-Nya, saya sebagai penulis dapat menyelesaikan penelitian dan skripsi ini.

Kupersembahkan skripsi ini kepada:

Kedua Orang Tua

“Kedua orang tua dan keluarga yang senantiasa memberikan doa, dukungan, dan nasihat kepada penulis dalam proses penyelesaian skripsi ini”

SANWACANA

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik, serta hidayah-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Implementasi *Convolutional Neural Network* Menggunakan Arsitektur *Xception* Untuk Identifikasi Penyakit Daun Pada Tanaman Durian”. Dalam proses penyusunan skripsi ini, penulis banyak menerima dukungan, bimbingan, serta bantuan baik secara moril maupun materil dari berbagai pihak. Oleh karena itu, dengan penuh rasa hormat dan terima kasih yang mendalam, penulis menyampaikan penghargaan sebesar-besarnya kepada:

1. Kedua orang tua dan keluarga yang senantiasa memberikan doa, dukungan, dan nasihat kepada penulis sepanjang masa perkuliahan;
2. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung;
3. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
4. Ibu Yessi Mulyani, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung, Pembimbing Akademik dan Pembimbing Utama skripsi yang telah memberikan dukungan dan ilmu selama menempuh pendidikan di Fakultas Teknik Universitas Lampung serta telah membimbing dan memberikan saran bagi penulis dalam menyelesaikan penelitian dan penulisan skripsi;
5. Bapak Puput Budi Wintoro, S.Kom., M.T.I selaku Pembimbing Pendamping yang telah membimbing dan memberikan saran bagi penulis dalam menyelesaikan penelitian dan penulisan skripsi;
6. Ibu Dr. Eng. Ir. Mardiana, S.T., M.T., IPM. selaku Penguji yang telah memberikan berbagai saran dan masukan yang membangun terhadap penelitian ini;

7. Seluruh Dosen Akademik Program Studi Teknik Informatika Universitas Lampung yang telah banyak memberikan ilmu kepada penulis selama menempuh pendidikan di Fakultas Teknik Universitas Lampung;
8. Seluruh Staff Administrasi yang telah memberikan bantuan administratif yang diberikan kepada penulis selama menempuh pendidikan di Fakultas Teknik Universitas Lampung;
9. Muhammad Rafi Rizanda, Arbian Alex Pritama, dan Muchammad Raja Haikal Fiaugustian sebagai rekan satu tim dalam penelitian yang telah bekerja sama dan saling mendukung dalam penyelesaian penelitian;
10. Rekan-rekan dalam grup 'Teh Kotak' yang tidak dapat disebutkan satu persatu, yang telah memberikan dukungan besar kepada penulis dalam menyelesaikan skripsi;
11. Seluruh pihak yang terlibat dalam penelitian ini yang tidak dapat penulis sebutkan satu persatu.

Penulis berharap agar laporan ini dapat menjadi referensi bagi pengembangan keilmuan di bidang teknik informatika. Oleh karena itu, semoga penelitian ini bermanfaat bagi yang membacanya.

Bandar Lampung, 20 Agustus 2025

Penulis,

Farhat Febrianto

NPM. 2115061012

DAFTAR ISI

	Halaman
ABSTRAK	i
PERSEMBAHAN.....	viii
SANWACANA	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL	xvii
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	5
1.5 Batasan Masalah.....	5
1.6 Sistematika Penulisan Laporan	5
II. TINJAUAN PUSTAKA.....	7
2.1 Penyakit Daun Durian	7
2.2 <i>Artificial Intelligence</i>	8
2.3 <i>Deep learning</i>	8
2.4 <i>Convolutional neural network</i>	9
2.4.1 <i>Convolutional Layer</i>	10
2.4.2 <i>Pooling Layer</i>	10
2.4.3 <i>Flatten Layer</i>	11
2.4.4 <i>Fully Connected Layer</i>	12
2.4.5 <i>Activation Function</i>	12
2.5 <i>Xception</i>	13
2.6 <i>Hyperparameter Tuning</i>	15
2.6.1 <i>Epoch</i>	16

2.6.2 <i>Batch size</i>	16
2.6.3 <i>Learning rate</i>	17
2.6.4 <i>Optimizer</i>	17
2.7 <i>Fine tuning</i>	17
2.6 <i>Dropout</i>	18
2.8 <i>Transfer Learning</i>	18
2.8 <i>Confusion Matrix</i>	19
2.8 <i>Pyhton</i>	20
2.9 <i>Tensorflow</i>	21
2.10 <i>Google Collaboratory</i>	21
2.11 <i>Augmentasi Data</i>	22
2.12 <i>Depthwise Separable Convolution</i>	22
2.2.1 <i>Penelitian Terkait</i>	24
III. METODE PENELITIAN	31
3.1 <i>Waktu dan Tempat Penelitian</i>	31
3.2 <i>Alat dan Bahan Penelitian</i>	32
3.3 <i>Tahapan Penelitian</i>	34
3.3.1 <i>Identifikasi Masalah</i>	35
3.3.2 <i>Studi Literatur</i>	35
3.3.3 <i>Pengumpulan Dataset</i>	35
3.3.4 <i>Preprocessing Dataset</i>	36
3.3.5 <i>Arsitektur Xception</i>	37
3.3.6 <i>Pelatihan Model</i>	38
3.3.7 <i>Evaluasi Model</i>	39
IV. HASIL DAN PEMBAHASAN.....	40
4.1 <i>Pengumpulan Dataset</i>	40
4.2 <i>Preprocessing Dataset</i>	41
4.2.1 <i>Pembagian Dataset</i>	41
4.2.2 <i>Resize dan Rescale</i>	42
4.2.3 <i>Augmentasi Data</i>	43
4.2.4 <i>Pemuatan Data</i>	47
4.3 <i>Arsitektur Xception</i>	48
4.4 <i>Pelatihan Model</i>	50

4.4.1 Proses <i>Transfer Learning</i>	51
4.4.2 Proses Penyesuaian <i>Fully Connected Layer</i>	52
4.4.3 Proses <i>Fine tuning</i>	56
4.4.4 Pengaruh Augmentasi Data	58
4.4.5 Pengaruh <i>Epoch</i>	61
4.4.5 Pengaruh <i>Batch size</i>	64
4.4.6 Pengaruh <i>Learning rate</i>	67
4.4.7 Pengaruh <i>Optimizer</i>	70
4.4.8 Pengaruh <i>Fine Tuning</i>	73
4.5 Evaluasi Model.....	75
4.5.1 Hasil Uji Arsitektur.....	76
4.5.2 Hasil Uji <i>Epoch</i>	84
4.5.3 Hasil Uji <i>Batch size</i>	95
4.5.4 Hasil Uji <i>Learning Rate</i>	102
4.5.5 Hasil Uji <i>Optimizer</i>	108
4.5.6 Hasil Uji <i>Fine Tuning</i>	114
4.6 Pengujian Model.....	120
4.6.1 Pengujian Model Menggunakan Daun Semangka.....	122
4.6.2 Pengujian Model Menggunakan <i>Leaf Blight</i>	123
4.6.3 Pengujian Model Menggunakan <i>Algal Leaf Spot</i>	124
4.6.4 Pengujian Model Menggunakan <i>Allocardia Attack</i>	125
4.6.5 Pengujian Model Menggunakan <i>Healthy Leaf</i>	126
4.6.6 Pengujian Model Menggunakan <i>Phomopsis Leaf Spot</i>	127
V. SIMPULAN DAN SARAN	128
5.1 Simpulan.....	128
5.2 Saran	129
DAFTAR PUSTAKA	131
LAMPIRAN.....	136

DAFTAR GAMBAR

	Halaman
Gambar 2.1 <i>Durian Leaf Disease</i> [10]	7
Gambar 2.2 Arsitektur <i>Convolutional neural network</i> [13]	10
Gambar 2.3 <i>Convolutional Layer</i> [14]	10
Gambar 2.4 <i>Max Pooling</i> [15]	11
Gambar 2.5 Proses <i>Flatten Layer</i>	11
Gambar 2.6 <i>Fully Connected Layer</i> [17]	12
Gambar 2.7 Arsitektur <i>Xception</i> [19]	15
Gambar 2.8 Konsep <i>Hyperparameter Tuning</i> [21]	16
Gambar 2.9 <i>Transfer Learning</i> [23]	19
Gambar 3.1 Tahapan Penelitian	34
Gambar 3.2 Dataset [41]	36
Gambar 3.3 Arsitektur <i>Xception</i>	37
Gambar 4.1 Alur <i>Preprocessing</i> Dataset	41
Gambar 4.2 Hasil Augmentasi Data	45
Gambar 4.3 Hasil Arsitektur <i>Xception</i>	50
Gambar 4.4 Grafik Akurasi dan <i>Loss</i> Satu Lapisan <i>Dense</i> dengan 64 <i>Neuron</i>	53
Gambar 4.5 Grafik Akurasi dan <i>Loss</i> Satu Lapisan <i>Dense</i> dengan 128 <i>Neuron</i> ...	53
Gambar 4.6 Grafik Akurasi dan <i>Loss</i> Satu Lapisan <i>Dense</i> dengan 256 <i>Neuron</i> ...	53
Gambar 4.7 Grafik Akurasi dan <i>Loss</i> Dua Lapisan <i>Dense</i> dengan 64 dan 128 <i>Neuron</i>	54
Gambar 4.8 Grafik Akurasi dan <i>Loss</i> Tiga Lapisan <i>Dense</i> dengan 64, 128, dan 256 <i>Neuron</i>	54
Gambar 4.9 Diagram Batang Perbandingan Arsitektur.....	55
Gambar 4.10 Grafik Akurasi dan <i>Loss</i> Tidak Menggunakan Augmentasi.....	58
Gambar 4.11 Grafik Akurasi dan <i>Loss</i> Augmentasi secara <i>On the fly</i>	59
Gambar 4.12 Grafik Akurasi dan <i>Loss</i> Augmentasi Dataset Seimbang.....	59

Gambar 4.13 Diagram Batang Perbandingan Augmentasi	60
Gambar 4.14 Grafik Akurasi dan <i>Loss</i> 10 <i>Epoch</i>	61
Gambar 4.15 Grafik Akurasi dan <i>Loss</i> 20 <i>Epoch</i>	61
Gambar 4.16 Grafik Akurasi dan <i>Loss</i> 30 <i>Epoch</i>	62
Gambar 4.17 Grafik Akurasi dan <i>Loss</i> 40 <i>Epoch</i>	62
Gambar 4.18 Grafik Akurasi dan <i>Loss</i> 50 <i>Epoch</i>	62
Gambar 4.19 Grafik Akurasi dan <i>Loss</i> 60 <i>Epoch</i>	63
Gambar 4.20 Diagram Batang Perbandingan <i>Epoch</i>	63
Gambar 4.21 Grafik Akurasi dan <i>Loss Batch Size</i> 8	65
Gambar 4.22 Grafik Akurasi dan <i>Loss Batch Size</i> 16	65
Gambar 4.23 Grafik Akurasi dan <i>Loss Batch Size</i> 32	65
Gambar 4.24 Grafik Akurasi dan <i>Loss Batch Size</i> 64	66
Gambar 4.25 Diagram Batang Perbandingan <i>Batch Size</i>	66
Gambar 4.26 Grafik Akurasi dan <i>Loss Learning Rate</i> 0.01	68
Gambar 4.27 Grafik Akurasi dan <i>Loss Learning Rate</i> 0.001	68
Gambar 4.28 Grafik Akurasi dan <i>Loss Learning Rate</i> 0.0001	68
Gambar 4.29 Diagram Batang Perbandingan <i>Learning Rate</i>	69
Gambar 4.30 Grafik Akurasi dan <i>Loss Optimizer Adam</i>	70
Gambar 4.31 Grafik Akurasi dan <i>Loss Optimizer Rmsprop</i>	71
Gambar 4.32 Grafik Akurasi dan <i>Loss Optimizer Nadam</i>	71
Gambar 4.33 Diagram Batang Perbandingan <i>Optimizer</i>	72
Gambar 4.34 Grafik Akurasi dan <i>Loss Fine Tuning</i> Membuka 20 Lapisan.....	73
Gambar 4.35 Grafik Akurasi dan <i>Loss Fine Tuning</i> Membuka 40 Lapisan.....	73
Gambar 4.36 Grafik Akurasi dan <i>Loss Fine Tuning</i> Membuka Semua Lapisan...	74
Gambar 4.37 Diagram Batang Perbandingan <i>Fine Tuning</i>	74
Gambar 4.38 <i>Confusion Matrix</i> Satu Lapisan <i>Dense</i> Dengan 64 <i>Neuron</i>	76
Gambar 4.39 <i>Confusion Matrix</i> Satu Lapisan <i>Dense</i> Dengan 128 <i>Neuron</i>	77
Gambar 4.40 <i>Confusion Matrix</i> Satu Lapisan <i>Dense</i> Dengan 256 <i>Neuron</i>	79
Gambar 4.41 <i>Confusion Matrix</i> Dua Lapisan <i>Dense</i> Dengan 64 dan 128 <i>Neuron</i>	80
Gambar 4.42 <i>Confusion Matrix</i> Tiga Lapisan <i>Dense</i> Dengan 64, 128 dan 256 <i>Neuron</i>	82
Gambar 4.43 Diagram Batang Metrik Evaluasi Hasil Uji Arsitektur	83

Gambar 4.44 <i>Confusion Matrix 10 Epoch</i>	85
Gambar 4.45 <i>Confusion Matrix 20 Epoch</i>	86
Gambar 4.46 <i>Confusion Matrix 30 Epoch</i>	88
Gambar 4.47 <i>Confusion Matrix 40 Epoch</i>	89
Gambar 4.48 <i>Confusion Matrix 50 Epoch</i>	91
Gambar 4.49 <i>Confusion Matrix 60 Epoch</i>	92
Gambar 4.50 Diagram Batang Perbandingan Matrik Evaluasi Hasil Uji <i>Epoch</i> ..	94
Gambar 4.51 <i>Confusion Matrix Batch Size 8</i>	95
Gambar 4.52 <i>Confusion Matrix Batch Size 16</i>	97
Gambar 4.53 <i>Confusion Matrix Batch Size 32</i>	98
Gambar 4.54 <i>Confusion Matrix Batch Size 64</i>	100
Gambar 4.55 Diagram Perbandingan Metrik Evaluasi Hasil Uji Batch size	101
Gambar 4.56 <i>Confusion Matrix Learning Rate 0.01</i>	103
Gambar 4.57 <i>Confusion Matrix Learning Rate 0.001</i>	104
Gambar 4.58 <i>Confusion Matrix Learning Rate 0.0001</i>	106
Gambar 4.59 Diagram Batang Perbandingan Matrik Evaluasi Hasil Uji <i>Learning Rate</i>	107
Gambar 4.60 <i>Confusion Matrix Optimizer Adam</i>	109
Gambar 4.61 <i>Confusion Matrix Optimizer Rmsprop</i>	110
Gambar 4.62 <i>Confusion Matrix Optimizer Nadam</i>	112
Gambar 4.63 Diagram Batang Metrik Evaluasi Hasil Uji <i>Optimizer</i>	113
Gambar 4.64 <i>Confusion Matrix</i> Membuka 20 Lapisan Terakhir	115
Gambar 4.65 <i>Confusion Matrix</i> Membuka 40 Lapisan Terakhir	116
Gambar 4.66 <i>Confusion Matrix</i> Membuka Seluruh Lapisan	118
Gambar 4.67 Diagram Batang Perbandingan Metrik Evaluasi Hasil Uji <i>Optimizer</i>	119
Gambar 4.68 Hasil Pengujian Model Menggunakan Daun Semangka.....	122
Gambar 4.69 Hasil Pengujian Model Menggunakan <i>Leaf Blight</i>	123
Gambar 4.70 Hasil Pengujian Model Menggunakan <i>Algal Leaf Spot</i>	124
Gambar 4.71 Hasil Pengujian Model Menggunakan <i>Allocaridara Attack</i>	125
Gambar 4.72 Hasil Pengujian Model Menggunakan <i>Healthy Leaf</i>	126
Gambar 4.73 Hasil Pengujian Model Menggunakan <i>Phomopsis Leaf Spot</i>	127

DAFTAR TABEL

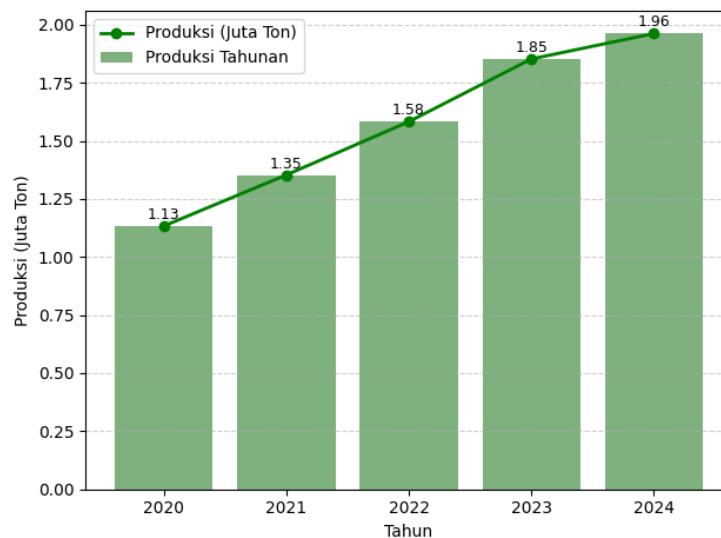
	Halaman
Tabel 2.1 Penelitian Terkait.....	28
Tabel 3.1 Waktu Penelitian.....	31
Tabel 3.2 Alat Penelitian	32
Tabel 3.3 Bahan Penelitian.....	33
Tabel 4.1 Total Dataset.....	40
Tabel 4.2 Distribusi Dataset	42
Tabel 4.3 Augmentasi Data	44
Tabel 4.4 Dataset Seimbang.....	46
Tabel 4.5 Summary Arsitektur <i>Xception</i>	49
Tabel 4.6 Tabel Perbandingan Arsitektur	55
Tabel 4.7 Perbandingan Augmentasi.....	60
Tabel 4.8 Tabel Perbandingan <i>Epoch</i>	64
Tabel 4.9 Perbandingan <i>Batch size</i>	67
Tabel 4.10 Perbandingan <i>Learning rate</i>	69
Tabel 4.11 Perbandingan <i>Optimizer</i>	72
Tabel 4.12 Perbandingan <i>Fine tuning</i>	75
Tabel 4.13 Metrik Evaluasi Model Satu Lapisan <i>Dense</i> Dengan 64 <i>Neuron</i>	76
Tabel 4.14 Metrik Evaluasi Model Satu Lapisan <i>Dense</i> Dengan 128 <i>Neuron</i>	78
Tabel 4.15 Metrik Evaluasi Model Satu Lapisan <i>Dense</i> Dengan 128 <i>Neuron</i>	79
Tabel 4.16 Metrik Evaluasi Model Dua Lapisan <i>Dense</i> Dengan 64 dan 128 <i>Neuron</i>	81
Tabel 4.17 Metrik Evaluasi Model Tiga Lapisan <i>Dense</i> Dengan 64, 128 dan 256 <i>Neuron</i>	82
Tabel 4.18 Analisis Perbandingan Metrik Evaluasi Hasil Uji Arsitektur.....	84
Tabel 4.19 Metrik Evaluasi Model 10 <i>Epoch</i>	85

Tabel 4.20 Metrik Evaluasi Model 20 <i>Epoch</i>	87
Tabel 4.21 Metrik Evaluasi Model 30 <i>Epoch</i>	88
Tabel 4.22 Metrik Evaluasi Model 40 <i>Epoch</i>	90
Tabel 4.23 Metrik Evaluasi Model 50 <i>Epoch</i>	91
Tabel 4.24 Metrik Evaluasi Model 60 <i>Epoch</i>	93
Tabel 4.25 Analisis Perbandingan Metrik Evaluasi Hasil Uji <i>Epoch</i>	94
Tabel 4.26 Metrik Evaluasi Model <i>Batch size</i> 8	96
Tabel 4.27 Metrik Evaluasi Model <i>Batch Size</i> 16.....	97
Tabel 4.28 Metrik Evaluasi Model <i>Batch size</i> 32	99
Tabel 4.29 Metrik Evaluasi Model <i>Batch size</i> 64	100
Tabel 4.30 Analisis Perbandingan Metrik Evaluasi Hasil Uji <i>Batch size</i>	102
Tabel 4.31 Metrik Evaluasi Model <i>Learning rate</i> 0.01.....	103
Tabel 4.32 Metrik Evaluasi Model <i>Learning rate</i> 0.001.....	105
Tabel 4.33 Metrik Evaluasi Model <i>Learning rate</i> 0.0001.....	106
Tabel 4.34 Analisis Perbandingan Matrik Evaluasi Hasil Uji <i>Learning rate</i>	108
Tabel 4.35 Metrik Evaluasi Model <i>Optimizer Adam</i>	109
Tabel 4.36 Metrik Evaluasi Model <i>Optimizer Rmsprop</i>	111
Tabel 4.37 Metrik Evaluasi Model <i>Optimizer Nadam</i>	112
Tabel 4.38 Analisis Perbandingan Metrik Evaluasi Hasil Uji <i>Optimizer</i>	114
Tabel 4.39 Metrik Evaluasi Model Membuka 20 Lapisan Terakhir	115
Tabel 4.40 Metrik Evaluasi Model Membuka 40 Lapisan Terakhir	117
Tabel 4.41 Metrik Evaluasi Model Membuka Semua Lapisan	118
Tabel 4.42 Analisis Perbandingan Matrix Evaluasi Hasil Uji <i>Fine Tuning</i>	120

I. PENDAHULUAN

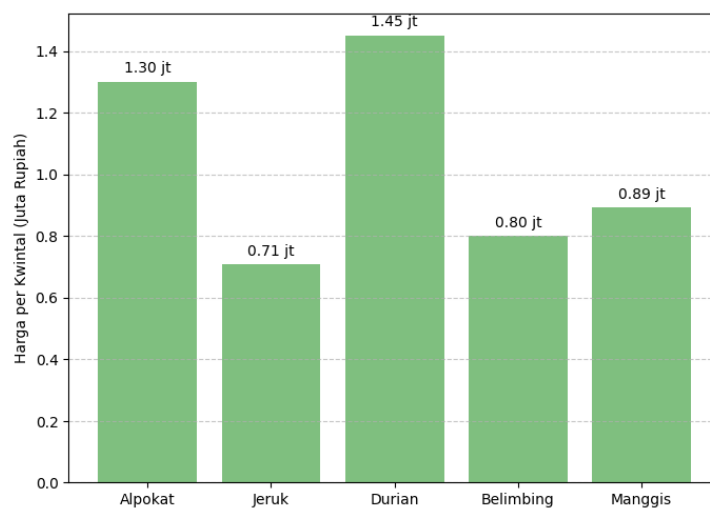
1.1 Latar Belakang

Durian atau disebut dalam bahasa latin yaitu *Durio zibethinus Murr* merupakan salah satu jenis buah tropis yang tergolong dalam famili *Bombacaceae* yang memiliki nilai ekonomis tinggi [1]. Berdasarkan data yang dirilis oleh Badan Pusat Statistik, produksi durian di Indonesia mengalami peningkatan yang konsisten dari tahun ke tahun. Pada tahun 2020, total produksi durian tercatat sebesar 1.133.194 ton. Jumlah tersebut meningkat menjadi 1.352.957 ton pada tahun 2021, kemudian naik menjadi 1.582.171 ton pada tahun 2022. Tren kenaikan ini berlanjut pada tahun 2023 dengan total produksi sebesar 1.852.045 ton, dan pada tahun 2024 mencapai sekitar 1.961.486 ton. Berdasarkan gambar 1.1, dapat diamati bahwa produksi durian di Indonesia menunjukkan tren peningkatan yang signifikan selama lima tahun terakhir. Kenaikan ini mencerminkan pertumbuhan yang konsisten dan positif dalam sektor produksi durian nasional [2].



Gambar 1.1 Peningkatan Produksi Durian Di Indonesia Tahun 2020 - 2024

Peningkatan produksi durian menunjukkan dampak ekonomi yang signifikan bagi perekonomian di Indonesia. Seperti yang ditunjukkan pada gambar 1.2 harga jual durian yang relatif tinggi dibandingkan dengan komoditas buah lainnya menjadikannya sebagai salah satu peluang agribisnis yang potensial dalam meningkatkan pendapatan petani. Tingkat produksi durian yang terus meningkat setiap tahunnya membuka peluang untuk ekspansi ke pasar internasional [3]. Selain sebagai komoditas pertanian, durian juga dimanfaatkan sebagai daya tarik agrowisata yang menggabungkan konsumsi buah dengan wisata edukatif, budaya, dan alam di sejumlah daerah di Indonesia, seperti Pasuruan, Banyuwangi, Medan, Bogor, Polewali Mandar, dan masih banyak lagi [4].



Gambar 1.2 Perbandingan Harga Jual Durian dan Komoditas Buah lain (per kwintal)

Untuk mendukung keberlanjutan produksi durian yang meningkat, pengendalian penyakit tanaman menjadi aspek yang penting. Deteksi dini penyakit tanaman merupakan langkah strategis untuk menjaga kesehatan tanaman dan memastikan produktivitas tanaman menjadi optimal. Metode tradisional untuk pengelolaan penyakit mengandalkan inspeksi visual oleh ahli, namun pendekatan ini kurang efektif karena memerlukan tenaga kerja yang banyak, biaya operasional, dan kurang terjangkau oleh petani skala kecil di negara berkembang [5]. Selain itu, keterbatasan pengetahuan petani yang kurang berpengalaman dalam mengklasifikasi penyakit pada tanaman dapat menyebabkan pengambilan

keputusan yang kurang tepat, sehingga mengakibatkan produktivitas pertanian menurun [6]. Hal ini juga menjadi tantangan bagi masyarakat awam yang ingin memulai budidaya durian, karena tanpa pengetahuan dan perawatan yang tepat, tanaman durian rentan terserang penyakit yang dapat menyebabkan menghambat pertumbuhan atau bahkan menyebabkan kematian. Dalam upaya deteksi dini penyakit pada tanaman durian, daun dipilih sebagai indikator utama karena merupakan bagian tanaman yang paling mudah diamati serta kerap menunjukkan gejala awal serangan penyakit, sehingga memungkinkan klasifikasi lebih cepat dan akurat terhadap kondisi kesehatan tanaman.

Dalam menghadapi tantangan tersebut, seiring dengan perkembangan teknologi saat ini mendapatkan peluang besar dalam mengembangkan solusi inovatif pada bidang pertanian. Penerapan *deep learning* telah menghasilkan peningkatan luar biasa dalam akurasi pengenalan gambar termasuk dalam tugas *image classification* yang berkontribusi terhadap peningkatan koordinasi dan akurasi sistem dalam mendeteksi dan klasifikasi penyakit daun tanaman [7] [8]. Dalam merancang suatu model pembelajaran mesin untuk klasifikasi penyakit daun pada tanaman, pemilihan arsitektur model *Convolutional neural network* yang tepat menjadi aspek yang penting untuk mencapai kinerja optimal. Di antara berbagai pilihan arsitektur lainnya seperti *VGG-16*, *VGG-19*, *InceptionV3*, *MobileNetV2*, *MobileNetV3*, *AlexNet*, *ResNet18*, *Xception* menjadi pilihan yang sangat tepat untuk tugas *image classification*, termasuk deteksi dan kategorisasi penyakit pada tanaman, karena arsitektur yang mengandalkan konvolusi terpisah secara mendalam (*depth-wise separable convolutions*) memungkinkan model ini menangkap pola dan fitur yang kompleks dengan lebih efisien, sehingga menghasilkan akurasi yang tinggi dalam mendeteksi penyakit daun [9].

Dengan mengimplementasikan teknologi ini, penelitian ini bertujuan untuk merancang model klasifikasi penyakit pada daun tanaman durian secara cepat dan akurat sebagai tahap awal dalam pengembangan aplikasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya maka rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana membuat kecerdasan buatan dengan model *Convolutional neural network* berbasis arsitektur *Xception* dalam mengklasifikasikan penyakit daun pada tanaman durian ?
2. Bagaimana performa dari model *Convolutional neural network* berbasis arsitektur *Xception* dalam mengklasifikasikan penyakit daun tanaman durian ?
3. Bagaimana pengaruh penyesuaian arsitektur *Xception* dan pengaturan hyperparameter pelatihan untuk menghasilkan performa terbaik dalam mengklasifikasi penyakit daun pada tanaman durian ?
4. Bagaimana pengaruh peningkatan jumlah *layer* yang di-*fine-tuning* pada arsitektur *Xception* terhadap peningkatan performa model dalam mengklasifikasikan penyakit daun pada tanaman durian?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Mengembangkan model kecerdasan buatan menggunakan *Convolutional neural network* berbasis arsitektur *Xception* dalam mengklasifikasikan penyakit daun pada tanaman durian.
2. Menganalisis dan mengevaluasi performa model *Convolutional neural network* berbasis arsitektur *Xception* dalam mengklasifikasikan penyakit daun tanaman durian.
3. Menganalisis dan mengevaluasi pengaruh penyesuaian arsitektur dan *hyperparameter* pelatihan terhadap performa model dalam mengklasifikasi penyakit daun pada tanaman durian.
4. Menganalisis dan mengevaluasi pengaruh variasi jumlah *layer* yang di-*fine-tuning* terhadap performa model dalam mengklasifikasikan penyakit daun tanaman durian.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

1. Menjadi referensi dan menambah literatur mengenai penerapan arsitektur *Xception* dalam mengklasifikasi penyakit daun pada tanaman durian.
2. Menjadi tahap awal dalam pengembangan aplikasi berbasis kecerdasan buatan untuk mengklasifikasi penyakit daun pada tanaman durian secara cepat dan akurat.

1.5 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah:

1. Penelitian ini menggunakan metode *Convolutional neural network (CNN)* menggunakan arsitektur *Xception* dan tidak membandingkan hasil dengan metode lain.
2. Dataset menggunakan dataset publik yang tersedia di platform kaggle.com, yaitu "Durian Leaf Disease Dataset" yang dikembangkan oleh pengguna dengan nama akun cthng123.

1.6 Sistematika Penulisan Laporan

Adapun sistematika yang digunakan dalam penulisan penelitian ini adalah sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini secara umum meliputi latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, serta sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Pada bab ini berisi tentang dasar teori dan teori pendukung serta penelitian sebelumnya yang berkaitan dengan penelitian ini.

BAB III : METODOLOGI PENELITIAN

Pada bab ini berisi mengenai waktu penelitian, tempat penelitian, alat dan bahan penelitian serta tahapan yang dilakukan dalam penelitian ini.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini berisikan mengenai hasil dan pembahasan dari penelitian yang telah dilakukan serta penjelasan hasil dari pengujian model.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini berisikan kesimpulan yang diperoleh dari hasil penelitian yang telah dilakukan serta berisikan saran yang dapat digunakan sebagai acuan untuk penelitian selanjutnya.

II. TINJAUAN PUSTAKA

2.1 Penyakit Daun Durian

Tanaman durian (*Durio zibethinus*) rentan terhadap berbagai gangguan pada daun yang dapat mengganggu kesehatan dan produktivitasnya. Gangguan tersebut dapat diklasifikasikan ke dalam lima kategori utama, yaitu: *Algal leaf spot*, *Allocaridara attack*, *Healthy leaf*, *Leaf blight*, dan *Phomopsis leaf spot*. *Algal leaf spot* disebabkan oleh *Cephaleuros Virescens*, yang ditandai dengan munculnya bintik berwarna oranye karat pada permukaan atas daun, infeksi ini mengganggu fotosintesis dan dapat menyebabkan kerontokan daun. *Allocaridara attack* memperlihatkan gejala akibat serangan kutu loncat *Allocaridara malayensis*, berupa daun menggulung, menguning, serta adanya kerusakan fisik memicu pertumbuhan jamur jelaga. *Healthy leaf* menunjukkan daun yang tampak sehat, berwarna hijau mengilap, dan bebas dari gejala penyakit atau kerusakan. *Leaf blight* digambarkan oleh daun yang memiliki lesi besar berwarna cokelat pucat dengan tepi tidak beraturan, yang dapat disebabkan oleh patogen seperti *Rhizoctonia solani AG 1-ID* dan dapat mengakibatkan kelayuan hingga kematian daun. *Phomopsis leaf spot* disebabkan oleh *Diplodia heoromae* dan *C. Gloeosporioides*, yang ditandai dengan munculnya bintik melingkar berwarna cokelat berdiameter sekitar 1 mm dengan tepi gelap dan lingkaran kuning pada daun [10].



Gambar 2.1 *Durian Leaf Disease* [10]

2.2 Artificial Intelligence

Dengan kata lain, kecerdasan buatan merancang sistem yang dapat meniru cara berpikir dan bertindak manusia dalam menyelesaikan berbagai tugas. Teknologi ini tidak hanya meniru kecerdasan manusia, tetapi juga dapat berperilaku seperti manusia dalam melakukan pekerjaan tertentu [11].

Dalam implementasinya *Artificial Intelligence* memiliki beberapa tahapan yang harus diikuti untuk memastikan model *AI* tersebut berhasil dan efektif. Prosesnya dimulai dengan menentukan tujuan dan mengidentifikasi masalah yang ingin diselesaikan. Langkah ini penting untuk memahami kebutuhan dan tantangan yang akan diatasi menggunakan *AI*. Setelah itu, pemilihan jenis *AI* yang sesuai menjadi tahap berikutnya, di mana kita harus memilih metode atau teknik *AI* yang cocok dengan masalah yang dihadapi.

Tahap selanjutnya adalah mempelajari secara mendalam model atau algoritma yang telah dipilih, memastikan algoritma yang dipilih tepat dan efektif. Setelah itu, dilakukan pemrograman atau pembuatan *software* yang menjadi basis dari *AI* yang sedang dikembangkan. Langkah selanjutnya adalah penyempurnaan dan optimasi dari algoritma atau model yang sudah dibuat agar *AI* dapat bekerja dengan lebih optimal dan efisien. Terakhir, dilakukan penanaman *software* ke dalam perangkat atau mesin, sehingga komputer atau mesin tersebut dapat beroperasi dengan kecerdasan yang telah dibangun.

2.3 Deep learning

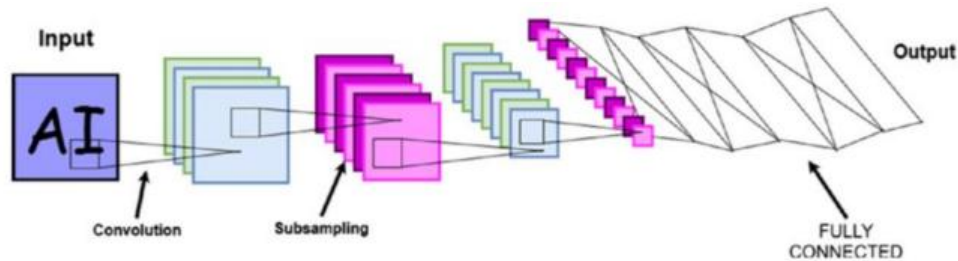
Deep learning adalah salah satu konsep pembelajaran mesin yang didasarkan pada jaringan syaraf tiruan (JST). Untuk banyak aplikasi, model *deep learning* mengungguli model machine learning dangkal (*shallow*) dan pendekatan analisis data tradisional. [12]. Pendekatan ini terinspirasi dari struktur dan cara kerja sistem saraf biologis, khususnya otak manusia, dalam memproses informasi secara bertingkat. Dalam konteks teknis, *Deep learning* memanfaatkan banyak lapisan tersembunyi (*hidden layers*) dalam jaringan saraf yang memungkinkan sistem untuk melakukan ekstraksi fitur secara otomatis dari data mentah, tanpa

memerlukan intervensi manusia secara eksplisit dalam proses feature engineering. Dengan kemampuannya untuk membangun representasi data secara hierarkis, *Deep learning* terbukti efektif dalam mengolah berbagai jenis data tidak terstruktur seperti citra digital, sinyal suara, dan teks.

Keunggulan utama *Deep learning* terletak pada kapasitasnya dalam menangani kompleksitas data dalam skala besar, serta kemampuannya untuk belajar dari pola-pola yang tidak dapat dikenali oleh algoritma pembelajaran tradisional. Teknologi ini telah menjadi landasan bagi berbagai inovasi dalam kecerdasan buatan (artificial intelligence), antara lain dalam bidang pengenalan wajah (facial recognition), pemrosesan bahasa alami (natural language processing), sistem rekomendasi, kendaraan otonom, serta aplikasi medis seperti deteksi penyakit berbasis citra radiologi. Namun demikian, implementasi *Deep learning* juga memiliki sejumlah tantangan, di antaranya adalah kebutuhan akan data pelatihan yang sangat besar, tingginya beban komputasi, serta risiko terjadinya *overfitting* jika tidak dilakukan regularisasi dan optimisasi yang tepat. Oleh karena itu, penggunaan *Deep learning* dalam praktik membutuhkan pemahaman mendalam terhadap teori jaringan saraf, teknik optimasi, serta infrastruktur komputasi yang memadai.

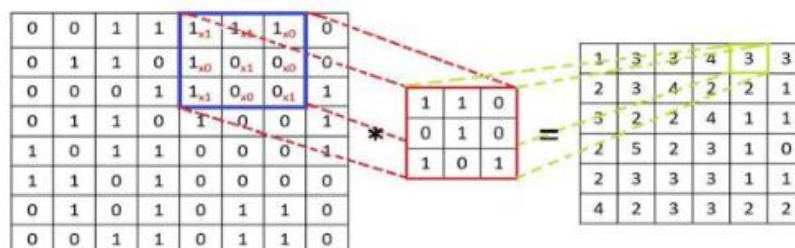
2.4 Convolutional neural network

Dalam *Convolutional neural network* adalah jenis jaringan saraf tiruan feed-forward yang mendalam, yang dirancang untuk memproses data gambar. *Convolutional neural network* bekerja dengan mempertahankan struktur, yaitu mempelajari fitur-fitur dari yang sederhana hingga kompleks. Hal ini membuat *Convolutional neural network* sangat efektif untuk tugas-tugas pengenalan objek dan berbagai permasalahan dalam bidang *computer vision* contohnya untuk pengenalan wajah, klasifikasi gambar, pemrosesan bahasa alami dan pengenalan suara [13]. Secara umum, arsitektur CNN terdiri atas lapisan input, diikuti oleh sejumlah *convolutional layer*, *pooling layer*, *fully connected layer*, dan *output layer*.

Gambar 2.2 Arsitektur *Convolutional neural network* [13]

2.4.1 Convolutional Layer

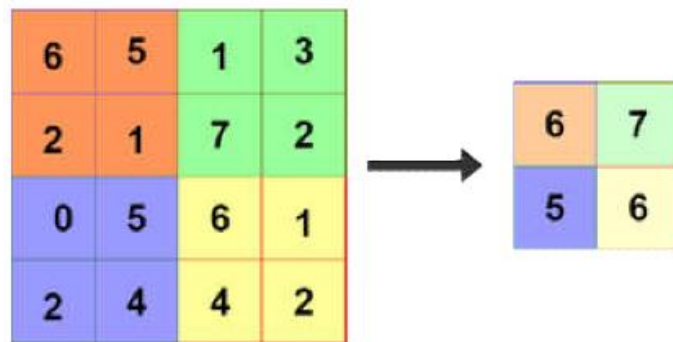
Convolutional layer adalah lapisan yang menerapkan operasi konvolusi terhadap input-nya menggunakan sebuah filter (kernel). Proses konvolusi memanfaatkan apa yang disebut sebagai filter. Seperti layaknya gambar, filter memiliki ukuran tinggi, lebar, dan tebal tertentu. Lapisan konvolusional menggunakan filter kernel untuk menghitung konvolusi gambar input, dan mengekstraksi fitur-fitur mendasar [14]. Filter ini akan diinisialisasi dengan suatu nilai yang akan menjadi sebuah parameter yang terus diperbarui dalam proses *learning*. Dalam proses ini, filter yang telah diinisialisasi dengan bobot awal akan digeser (*di-slide*) melintasi seluruh area input. Pada setiap posisi, dilakukan operasi perkalian elemen-sejajar (*element-wise multiplication*) antara nilai-nilai filter dan bagian input yang dilaluinya, kemudian hasilnya dijumlahkan menjadi satu nilai yang mana akan menghasilkan matriks baru yang disebut *feature map*.

Gambar 2.3 *Convolutional Layer* [14]

2.4.2 Pooling Layer

Pooling layer berperan dalam mengurangi dimensi representasi yang dihasilkan dari proses ekstraksi fitur, sehingga kompleksitas komputasi model dapat ditekan. Dengan demikian, pooling layer sangat efektif untuk mengurangi jumlah dimensi

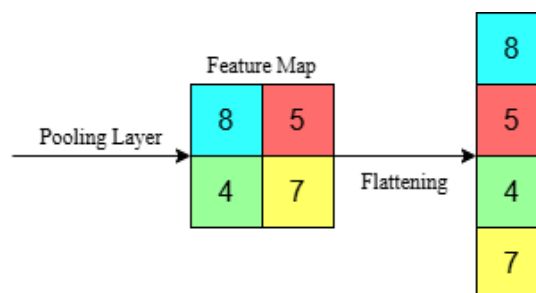
sekaligus mempertahankan fitur-fitur penting yang telah diekstraksi. Metode pooling yang sering digunakan antara lain Max Pooling, yang mengambil nilai maksimum dari area tertentu, dan Average Pooling, yang mengambil nilai rata-rata. Hal ini membantu mengurangi jumlah fitur dan mempertahankan fitur yang penting. Konfigurasi ini memungkinkan pengurangan ukuran representasi fitur secara efisien, yang tidak hanya mempercepat proses komputasi, tetapi juga membantu mengurangi risiko *overfitting* pada model [15].



Gambar 2.4 *Max Pooling* [15]

2.4.3 *Flatten Layer*

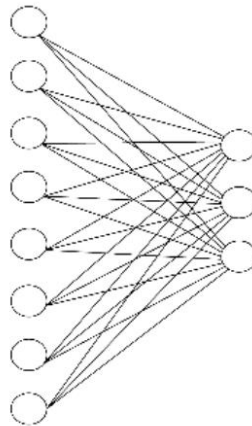
Dalam arsitektur *Convolutional neural network* *Flatten Layer* memiliki peran penting sebagai jembatan antara *Convolutional Layer* dan *Fully Connected Layer*. Lapisan ini bertugas mengubah output multidimensi dari proses konvolusi dan *pooling*, yang umumnya berbentuk vektor tiga dimensi, menjadi vektor satu dimensi yang dapat digunakan sebagai vektor fitur untuk *fully connected layer* [16]. Dengan demikian, *Flatten Layer* tidak hanya menyederhanakan struktur data, tetapi juga menjamin kompatibilitas arsitektural antara lapisan-lapisan dalam CNN secara keseluruhan.



Gambar 2.5 Proses *Flatten Layer*

2.4.4 *Fully Connected Layer*

Fully connected layer merupakan komponen penting dalam arsitektur jaringan saraf tiruan. Lapisan ini umumnya digunakan pada bagian akhir model untuk melakukan proses klasifikasi berdasarkan fitur-fitur yang telah diekstraksi oleh lapisan-lapisan sebelumnya [17]. Pada layer ini setiap *Neuron* terhubung keseluruhan *Neuron* dilapisan sebelumnya, hal ini memungkinkan untuk mengintegrasikan semua informasi yang telah dipelajari dari fitur-fitur sebelumnya dan mendapatkan output yang sesuai seperti kelas yang terdapat dalam tugas klasifikasi. *Fully connected layer* memiliki peran penting dalam proses pengambilan keputusan akhir pada jaringan saraf dengan menggabungkan seluruh informasi yang telah diproses sebelumnya untuk menghasilkan output akhir model.



Gambar 2.6 *Fully Connected Layer* [17]

2.4.5 *Activation Function*

Fungsi aktivasi merupakan komponen penting dalam arsitektur jaringan saraf tiruan. Fungsinya adalah untuk mengubah output dari setiap *Neuron* atau lapisan *Neuron* sebelum diteruskan ke lapisan berikutnya. Tanpa fungsi aktivasi, jaringan saraf hanya akan mampu memetakan hubungan linier antara input dan output, yang tentu saja membatasi kemampuannya dalam mempelajari pola kompleks atau *nonlinier*.

Salah satu alasan utama penggunaan fungsi aktivasi nonlinier adalah karena ia memungkinkan jaringan dengan dua atau lebih lapisan tersembunyi untuk

memodelkan fungsi-fungsi *nonlinier* secara efektif. Dengan kata lain, fungsi aktivasi berperan dalam memberikan *non-linear transformation* terhadap data, sehingga jaringan dapat menyelesaikan tugas-tugas yang lebih rumit seperti klasifikasi gambar, pengenalan suara, atau deteksi objek [18]. Secara umum terdapat dua jenis fungsi aktivasi dalam jaringan saraf yaitu untuk lapisan tersembunyi dan untuk lapisan keluaran, berikut ini contoh fungsi aktivasi :

- 1) ReLU, Fungsi ReLU yang didefinisikan dengan rumus $f(x) = \max(x, 0)$ digunakan dalam jaringan saraf tiruan untuk menentukan sebuah *Neuron* harus aktif atau tidak dalam memproses informasi. Jika output fungsi ReLU adalah 0, maka *Neuron* tersebut dianggap tidak aktif atau "mati" dan tidak berkontribusi dalam proses pengolahan informasi. Keaktifan *Neuron* sangat penting karena hanya *Neuron* yang aktif yang akan berkontribusi dalam menghasilkan output dan memperbarui bobot selama proses pelatihan. Dengan kata lain, fungsi aktivasi membantu jaringan memilih *Neuron-Neuron* mana yang relevan terhadap input tertentu, sehingga memungkinkan jaringan untuk mempelajari pola-pola penting dalam data secara lebih efisien dan efektif. Mekanisme ini juga berperan dalam meningkatkan kemampuan jaringan untuk melakukan generalisasi terhadap data baru.
- 2) Softmax, Fungsi *softmax* memiliki keunggulan utama dalam menghasilkan output berupa probabilitas yang bernilai antara 0 hingga 1, dengan total jumlah seluruh probabilitas sama dengan satu. Fungsi aktivasi ini umumnya digunakan dalam model klasifikasi multikelas, karena mampu memberikan distribusi probabilitas untuk setiap kelas dan kelas target akan memperoleh nilai probabilitas tertinggi dibandingkan kelas lainnya.

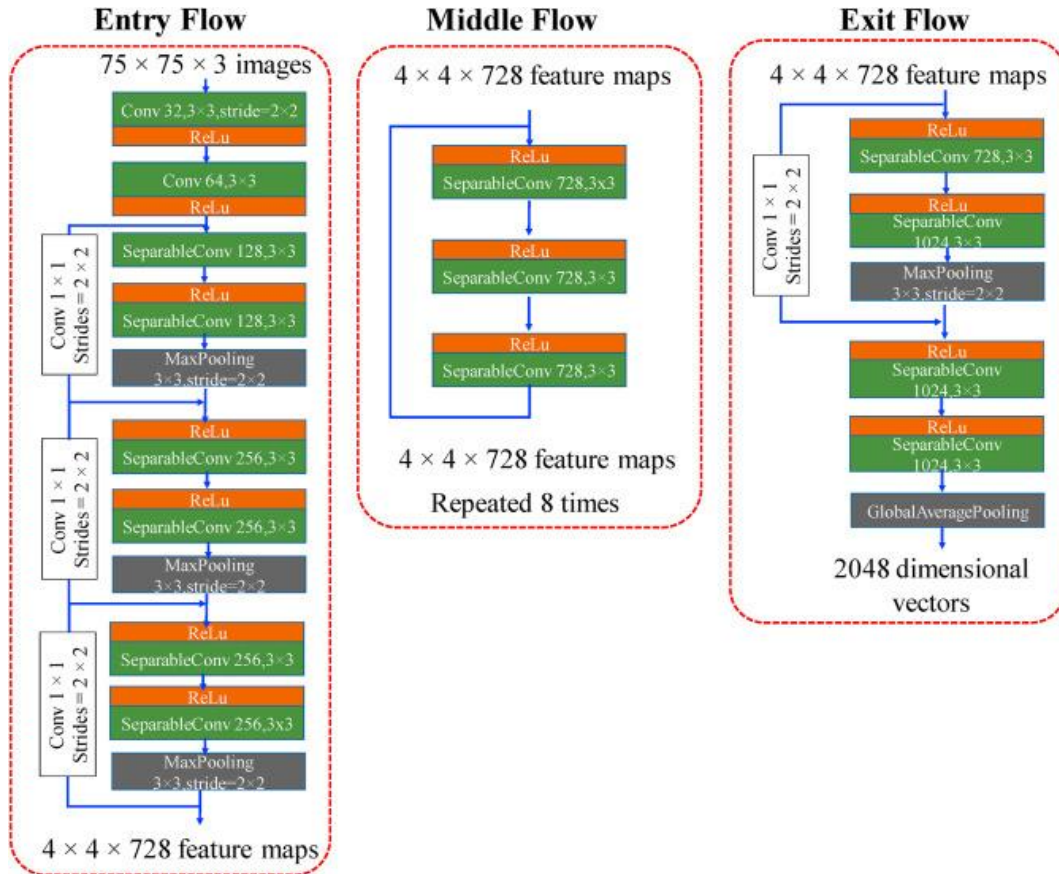
2.5 Xception

Istilah "*Xception*" adalah singkatan dari "*Extreme Inception*," yang mencerminkan desainnya sebagai iterasi arsitektur Inception dan diperkenalkan oleh Francois Chollet pada tahun 2017, yang juga dikenal sebagai pengembang utama dari Keras, sebuah *library Deep learning*. Arsitektur ini terdiri dari 36 *convolutional layers* ekstraksi fitur utama dari citra yang dikelompokkan kedalam tiga bagian utama

yaitu *entry flow*, *middle flow*, dan *exit flow*. Pada tahap *entry flow* bertugas mengekstraksi fitur dasar dari citra input RGB melalui dua lapisan konvolusi awal, masing-masing dengan 32 dan 64 filter, yang berfungsi untuk mengekstraksi fitur dasar seperti tepi dan tekstur. Selanjutnya terdapat tiga blok *depthwise separable convolution* yang menggabungkan *depthwise* (untuk mengekstraksi informasi spasial dari masing-masing channel) dan *pointwise convolution* (untuk menggabungkan informasi antar-channel), serta aktivasi ReLU dan MaxPooling untuk reduksi dimensi spasial.

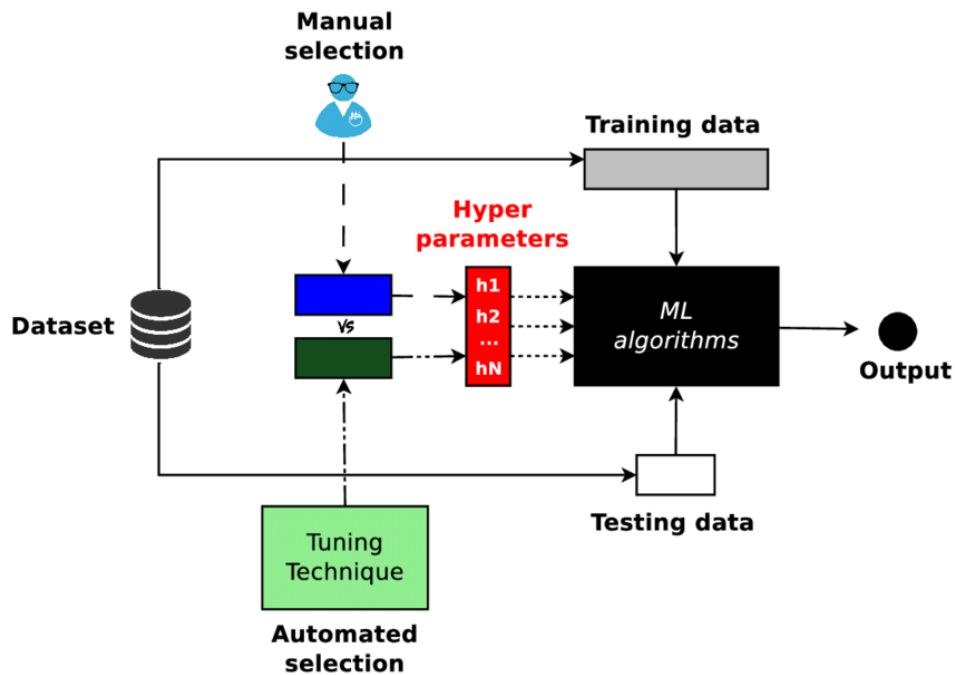
Pada tahap *middle flow* berperan mengekstraksi fitur tingkat tinggi tanpa mengubah ukuran spasial melalui delapan blok identik yang masing-masing terdiri dari tiga lapisan *SeparableConv2D* dan dilengkapi *residual connection* yang membantu menjaga stabilitas pelatihan dan mempermudah proses propagasi gradien. Pada tahap *exit flow* menyempurnakan ekstraksi fitur dengan dua tahap *SeparableConv2D*, *MaxPooling*, dan *Global Average Pooling* untuk mengubah feature map menjadi vektor satu dimensi. Vektor ini kemudian diproses oleh *Fully Connected Layer* yang terdiri dari lapisan *Dense* yang berfungsi sebagai penggabung informasi akhir dari seluruh fitur dan *dropout batch* untuk mengurangi *overfitting*, dan lapisan *output* menggunakan fungsi aktivasi *softmax* untuk menghasilkan probabilitas dari masing-masing kelas, dalam hal ini lima kategori klasifikasi daun durian berdasarkan kondisi kesehatannya.

Arsitektur *Xception* dirancang dengan memanfaatkan *depthwise separable convolution* secara menyeluruh, menggantikan lapisan konvolusional tradisional yang umum digunakan dalam arsitektur CNN konvensional. Pendekatan ini memungkinkan *Xception* untuk mencapai tingkat akurasi yang tinggi, sekaligus secara signifikan mengurangi jumlah parameter dan kompleksitas komputasi yang diperlukan, dibandingkan dengan arsitektur konvolusional sebelumnya [19].

Gambar 2.7 Arsitektur *Xception* [19]

2.6 Hyperparameter Tuning

Hyperparameter Tuning merupakan proses penentuan nilai parameter yang ditetapkan sebelum pelatihan model dimulai dan tidak diperoleh langsung dari data pelatihan. Pemilihan nilai *hyperparameter* yang tepat sangat berpengaruh terhadap kinerja model, karena dapat meningkatkan akurasi serta efisiensi proses pelatihan. Dalam konteks *Convolutional neural network*, penentuan *hyperparameter* sangat berpengaruh terhadap performa model [20]. Contoh *hyperparameter* yang umum digunakan antara lain *batch size*, yang menentukan jumlah data yang diproses sebelum pembaruan bobot; *learning rate*, yang mengatur besar langkah dalam proses pembelajaran; serta *epoch*, yang menunjukkan berapa kali seluruh data pelatihan digunakan dalam proses pelatihan. Penyesuaian nilai-nilai tersebut secara tepat dapat menghasilkan performa model yang optimal.



Gambar 2.8 Konsep *Hyperparameter Tuning* [21]

2.6.1 Epoch

Epoch adalah jumlah iterasi penuh di mana seluruh dataset dilatih melalui model. Pemilihan jumlah *epoch* yang tepat penting untuk memastikan bahwa model memiliki cukup waktu untuk belajar tanpa menjadi *overfitting* [18]. Jika jumlah *epoch* terlalu sedikit, model belum sempat memahami pola dalam data dan cenderung mengalami *underfitting*. Sebaliknya, jika terlalu banyak, model bisa terlalu menyesuaikan diri terhadap data pelatihan dan gagal melakukan generalisasi pada data baru.

2.6.2 Batch size

Batch Size adalah sebagian kecil data dengan ukuran tetap yang diproses secara berurutan dalam pelatihan. Setiap batch melalui proses *forward pass*, perhitungan *loss*, *backward pass* dan optimasi. *Batch size* merupakan faktor krusial dalam pelatihan model *machine learning*, terutama dengan dataset yang besar. Peningkatan ukuran umumnya akan meningkatkan akurasi dan mengurangi *loss*, tetapi ada titik dimana akurasi berkurang jika ukuran batch terus ditingkatkan [18].

Batch yang terlalu besar juga dapat memperlambat proses pelatihan dibandingkan dengan ukuran batch yang lebih kecil. Pada akhirnya, pemilihan ukuran batch yang optimal seringkali melibatkan uji coba yang disesuaikan dengan data dan model spesifik yang digunakan.

2.6.3 *Learning rate*

Learning rate adalah parameter utama yang dapat disesuaikan pada *optimizer*. Fungsinya adalah menentukan seberapa besar “langkah” yang diambil model saat menyesuaikan bobot dan bias berdasarkan gradien untuk mengurangi *loss*. Penyesuaian, perencanaan, ataupun penerapan *learning rate* awal yang tepat sangat penting untuk meningkatkan kinerja model mencapai batas maksimal [18]. Ini seringkali memerlukan eksperimen, *learning rate* yang optimal memungkinkan model untuk belajar dengan cepat dan mencapai performa prediksi terbaik pada dataset yang diberikan.

2.6.4 *Optimizer*

Optimizer adalah algoritma yang bertanggung jawab untuk memperbarui bobot dan bias jaringan secara iteratif berdasarkan gradien dari fungsi kerugian. Berbagai jenis *optimizer* memiliki keunggulan masing-masing tergantung pada jenis data dan model yang digunakan [18]. Training dengan *Hyperparameter* yang sesuai menggunakan *Adam Optimizer* dengan *Learning rate* kecil (0.0001) sangat relevan dalam konteks *transfer learning* atau *fine-tuning* model yang sudah menggunakan *pre-trained weights* (bobot yang sudah dilatih sebelumnya).

2.7 *Fine tuning*

Fine-tuning merupakan salah satu teknik dalam *transfer learning* yang bertujuan untuk menyesuaikan model prelatih dengan dataset baru yang lebih spesifik. Pada tahap awal, lapisan dasar dari model dibekukan untuk mempertahankan representasi fitur umum yang telah dipelajari dari dataset *Imagenet*. Selanjutnya, sebagian lapisan atas model dibuka untuk dilatih kembali sehingga dapat

menangkap pola yang lebih relevan dengan permasalahan yang sedang diteliti. Teknik *fine tuning* terbukti dapat meningkatkan kemampuan model untuk beradaptasi dengan tugas-tugas tertentu, tetapi *fine tuning* yang berlebihan dapat menurunkan kinerja akibat *overfitting* terutama pada set data yang tidak seimbang [22]. Dalam teknik *fine tuning* dapat dilakukan dengan cara otomatis yaitu sebuah skema untuk secara adaptif membekukan bagian dari model yang paling dekat dengan konvergensi selama penyempurnaan membuat pelatihan lebih cepat namun dalam hal akurasi masih sedikit lebih rendah dibandingkan *full fine tuning* [23]

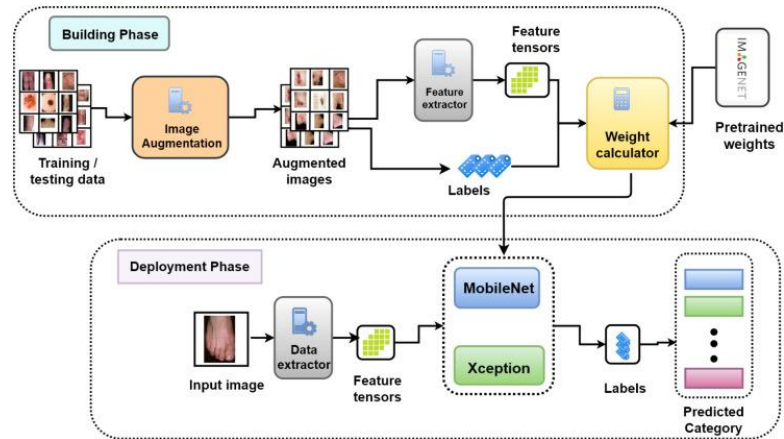
2.6 Dropout

Lapisan dropout adalah teknik regularisasi dalam jaringan saraf yang bertujuan mencegah *overfitting*. Lapisan ini akan menonaktifkan beberapa *Neuron* untuk mencegah jaringan saraf menjadi terlalu bergantung pada *Neuron* tertentu, atau agar *Neuron* tidak terlalu diandalkan sepenuhnya dalam suatu kasus spesifik (yang umum terjadi jika model overfit pada data pelatihan) [18]. Selain itu, dropout dapat membantu mengatasi noise dan gangguan lain dalam data pelatihan dengan cara acak menonaktifkan *Neuron* pada tingkat yang ditentukan disetiap *forward pass*, memaksa jaringan untuk belajar bagaimana membuat prediksi yang akurat hanya dengan sebagian *Neuron* yang tersisa. Dropout memaksa model untuk menggunakan lebih banyak *Neuron* untuk tujuan yang sama, menghasilkan peluang lebih tinggi untuk mempelajari fungsi dasar yang menjelaskan data. Karena *Neuron* yang aktif selalu berubah, dropout juga membantu mencegah *overfitting*, karena model tidak dapat menggunakan *Neuron* tertentu untuk menghafal sampel tertentu.

2.8 Transfer Learning

Transfer learning adalah teknik pembelajaran mesin yang pengetahuan yang diperoleh dari suatu domain atau tugas tertentu, seperti klasifikasi gambar menggunakan dataset *Imagenet*, digunakan kembali untuk menyelesaikan tugas lain yang serupa. Tujuan utama dari teknik ini adalah untuk meningkatkan efisiensi proses pelatihan dan kinerja model, khususnya ketika jumlah data pelatihan pada tugas baru terbatas atau tidak sebesar dataset awal. Dengan menggunakan *transfer*

learning, model seperti *Xception* dapat memanfaatkan bobot dan representasi fitur yang telah dipelajari sebelumnya, sehingga mereka dapat melakukan klasifikasi pada data baru tanpa perlu melalui proses pelatihan dari awal [24].



Gambar 2.9 *Transfer Learning* [24]

2.8 Confusion Matrix

Pada evaluasi model menggunakan confusion matrix sebagai alat untuk mengukur kinerja suatu sistem klasifikasi. Alat ini memberikan wawasan mendalam tentang bagaimana model klasifikasi melakukan prediksi dan seberapa baik model tersebut dalam mengklasifikasikan data [25]. Confusion matrix terdiri dari empat variabel kunci, yaitu: *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*. Confusion matrix membantu dalam memahami sejauh mana model klasifikasi rentan terhadap jenis kesalahan tertentu. Informasi dari TP, TN, FP, dan FN menjadi dasar untuk menghitung berbagai metrik evaluasi kinerja seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Pada dataset yang tidak seimbang, metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score* dianggap lebih representatif dibandingkan akurasi karena dapat memberikan gambaran yang lebih jelas mengenai kekuatan dan kelemahan model dalam menangani kelas minoritas [26].

Accuracy adalah seberapa tepat model dalam memprediksi yang benar dari seluruh data yang diuji. Ini mengukur seberapa sering model memprediksi dengan tepat, baik kelas positif maupun negatif.

$$Accuracy = \frac{True\ Prediction}{Total\ Prediction}$$

Precision adalah seberapa tepat model dalam memprediksi positif yang benar dari seluruh prediksi positif yang dibuat model. Ini menunjukkan seberapa akurat model dalam memprediksi kelas positif.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Recall adalah seberapa banyak model berhasil menemukan data positif yang sebenarnya dari seluruh data positif yang ada. Ini menunjukkan seberapa lengkap model dalam mengidentifikasi kelas positif.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

F1 Score adalah seberapa baik model dalam membuat prediksi positif yang benar secara konsisten, dengan mempertimbangkan baik ketepatan (*precision*) maupun kelengkapan (*recall*).

$$F1\ Score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

2.8 Python

Python adalah bahasa pemrograman yang sangat populer, dikenal dengan sintaksis yang sederhana, mudah dipahami, dan fleksibel. Hal ini membuatnya cocok baik untuk pemula maupun pengembang berpengalaman. Python digunakan di berbagai bidang, seperti pengembangan perangkat lunak, kecerdasan buatan (AI), pengembangan web, dan analisis data [27]. Keunggulan Python terletak pada pustaka dan framework yang kuat, seperti Keras, TensorFlow, PyTorch, dan scikit-learn, yang memudahkan pengembangan aplikasi dan penelitian. Keras mempermudah pembuatan jaringan saraf dengan antarmuka yang intuitif, sementara TensorFlow dan PyTorch digunakan untuk proyek machine learning dan *Deep learning* yang lebih kompleks. Scikit-learn cocok untuk tugas machine learning tradisional seperti regresi dan klasifikasi.

2.9 Tensorflow

Implementasi *deep learning* menggunakan *framework tensorflow* memberikan hasil yang sangat baik karena kemampuannya dalam melakukan simulasi, pelatihan, dan klasifikasi [28]. TensorFlow memungkinkan pembuatan model neural network yang dapat mensimulasikan pola-pola dalam data, seperti dalam pengenalan gambar atau pengolahan data sekuensial. Selama pelatihan, TensorFlow memanfaatkan berbagai algoritma optimisasi dan dukungan GPU/TPU untuk mempercepat proses belajar, sementara *API Keras* yang terintegrasi mempermudah pembangunan dan pelatihan model. Setelah pelatihan, TensorFlow juga memungkinkan model untuk mengklasifikasikan data baru dengan akurat, menggunakan lapisan klasifikasi seperti softmax dan fungsi *loss* seperti categorical crossentropy. Dengan berbagai fitur ini, TensorFlow menjadi alat yang sangat efektif dalam membangun model *deep learning* untuk berbagai aplikasi, mulai dari klasifikasi gambar hingga analisis teks.

2.10 Google Collaboratory

Google Collaboratory atau *Google Colab*, adalah platform berbasis *cloud* yang memungkinkan pengguna menulis, menjalankan, dan berbagi kode *Python* langsung melalui web browser. Dirancang khusus untuk *data science* dan *machine learning*, platform ini menyediakan lingkungan komputasi yang fleksibel, mudah diakses, dan gratis. Salah satu fitur utama *Google Colab* adalah dukungannya terhadap *Jupyter Notebook*, sebuah aplikasi *web open-source* yang memungkinkan integrasi kode, teks terformat, dan visualisasi data dalam satu dokumen interaktif. Dengan *Google Colab*, pengguna dapat menjalankan *Jupyter Notebook* tanpa memerlukan instalasi atau konfigurasi tambahan, membuatnya ideal bagi analis, pengembang, peneliti, dan pendidik yang membutuhkan solusi komputasi praktis dan efisien [29].

2.11 Augmentasi Data

Augmentasi data merupakan sebuah teknik yang digunakan dalam pembelajaran mesin untuk meningkatkan ukuran dan keragaman dataset pelatihan. Dengan menghasilkan sampel data baru dari data yang sudah ada, augmentasi data secara signifikan meningkatkan ukuran dan keragaman dataset. Teknik ini memiliki peran penting untuk mengatasi *overfitting* dikarenakan metode ini dapat meningkatkan generalisasi data [30]. Berikut ini penjelasan mengenai beberapa metode augmentasi data yang akan digunakan

1. *Rescale 1./255* : Proses menormalisasi nilai piksel dalam sebuah gambar. Setiap piksel memiliki tiga saluran warna (merah, hijau, dan biru), dan setiap saluran memiliki nilai intensitas dari 0 hingga 255. Dengan membagi setiap nilai piksel dengan 255, mengubah rentang nilai tersebut menjadi antara 0 dan 1. Tujuannya adalah agar model dapat memproses data gambar dengan lebih cepat dalam pelatihan model.
2. *Width and shift range* : Proses menggeser gambar secara horizontal dan vertikal. Teknik ini membantu meningkatkan variasi data agar dapat membantu model mengenali objek yang tidak selalu berada di tengah gambar.
3. *Rotation range* : Proses memutar gambar secara acak agar model tetap dapat mengenali objek meskipun dalam posisi yang berputar atau miring.
4. *Shear range* : Proses untuk menciptakan atau memperbaiki sudut pandang, digunakan untuk meningkatkan gambar agar model dapat melihat sebagaimana manusia dapat melihat dari berbagai sudut.
5. *Zooming* : Proses memperbesar gambar
6. *Horizontal flip* : Proses membalik gambar secara horizontal (seperti melihat bayangan di cermin) sehingga dapat menggandakan variasi data pelatihan.

2.12 Depthwise Separable Convolution

Depthwise separable convolution diperkenalkan oleh Sifre dan diterapkan untuk tujuan klasifikasi citra. Memfaktorkan operasi konvolusi adalah bentuk dari *depthwise separable convolution*, dimana konvolusi biasa diubah menjadi operasi

dengan dua tahap, yaitu tahap pertama *depthwise separable convolution*, dan tahap kedua *pointwise convolution*. Proses konvolusi ini menggantikan konvolusi biasa dengan dua tahap, yaitu *depthwise convolution* yang berfungsi sebagai penyaring spasial (filter yang bekerja pada dimensi ruang dari gambar, yaitu tinggi dan lebar), dan *pointwise convolution* yang berfungsi untuk menggabungkan informasi antar channel hasil dari *depthwise convolution*. Pendekatan ini mampu secara signifikan mengurangi jumlah parameter dan ukuran model tanpa mengorbankan kualitas representasi fitur [31].

Kernel konvolusi reguler biasanya memiliki tiga parameter, yaitu tinggi (h), lebar (w), dan kanal input (C_{in}) dari sebuah peta fitur input (I). Lapisan konvolusi yang dihasilkan ($h \times w \times C_{in}$) diterapkan sebagai $K \times K \times C_{in} \times C_{out}$ di mana C_{out} adalah jumlah kanal output.

Pada operasi *depthwise separable convolution* memiliki dua operasi konvolusi yaitu *depthwise separable convolution* dan *pointwise convolution*. Secara matematis, *depthwise separable convolution* dapat dituliskan sebagai berikut:

$$G(y, x, j) = \sum_{u=1}^k \sum_{v=1}^k K(u, v, j) \times I(y + u - 1, x + v, j)$$

di mana K merepresentasikan kernel dari operasi *depthwise separable convolution* dengan ukuran $K \times K \times C_{in}$. Filter ke- n^{th} pada kernel K diterapkan pada kanal input ke- n^{th} dari peta fitur I untuk merekonstruksi peta fitur output G . Saat merekonstruksi fitur baru, digunakan *pointwise convolution*. Secara matematis, *pointwise convolution* dapat dituliskan sebagai berikut:

$$O(y, x, l) = \sum_{j=1}^{C_{in}} G(y, x, j) \times P(j, l)$$

dimana ukuran kernel operasi konvolusi titik adalah $1 \times 1 \times C_{in} \times C_{out}$

2.2.1 Penelitian Terkait

Penelitian ini tidak lepas dari penelitian yang sudah dilakukan, sehingga penelitian yang dilakukan memiliki persamaan dan perbedaan objek yang diteliti. Ringkasan dari penelitian dapat dilihat dibawah ini:

1. A. L. Sabarre, A. S. Navidad, D. S. Torbela, and J. J. Adtoon, “*Development of Durian Leaf Disease Detection on Android device*” diterbitkan pada tahun 2021. Penelitian ini mengembangkan sistem deteksi penyakit daun durian berbasis Android dengan menggunakan metode *MobileNetV2*, yang merupakan arsitektur *deep learning* ringan dan efisien. Tujuan utama dari penelitian ini adalah untuk membantu petani mengenali penyakit daun durian secara cepat dan akurat melalui aplikasi *mobile*. Dataset yang digunakan terdiri dari gambar daun durian yang dikategorikan menjadi daun sehat, daun terkena antraknosa, dan daun terkena penyakit lain. Gambar-gambar tersebut diproses dan dilatih menggunakan *MobileNetV2*, kemudian diintegrasikan ke dalam aplikasi *Android* melalui *TensorFlow Lite*. Hasil pengujian menunjukkan bahwa model mampu mengklasifikasikan penyakit daun durian dengan akurasi sebesar 91%. Aplikasi ini memberikan solusi praktis dalam deteksi dini penyakit tanaman dan berpotensi meningkatkan efisiensi pengelolaan kebun durian secara digital [32].
2. V. Gonzalez-Huitron, J. A. León-Borges, A. E. Rodriguez-Mata, L. E. Amabilis-Sosa, B. Ramírez-Pereda, and H. Rodriguez, “*Disease detection in Tomato Leaves via CNN with Lightweight Architectures Implemented in Raspberry Pi 4*” diterbitkan pada tahun 2021. Penelitian ini bertujuan mengembangkan sistem deteksi penyakit daun tomat menggunakan arsitektur *Convolutional neural network* yang ringan dan dapat dijalankan secara efisien pada perangkat *Raspberry Pi 4*. Dengan menggunakan dataset *PlantVillage* yang terdiri dari 11 kelas (termasuk daun sehat dan berbagai jenis penyakit), penulis menguji beberapa arsitektur *Convolutional neural network* seperti *MobileNetV2*, *NASNetMobile*, *EfficientNetB0*, *Xception* dan lainnya. Hasil penelitian menunjukkan bahwa *Xception* memberikan performa terbaik dengan akurasi sebesar 100% dan *MobileNetV2* dengan akurasi sebesar 98,76%. Penelitian ini membuktikan bahwa penerapan model *deep learning* ringan di perangkat edge seperti *Raspberry Pi* memungkinkan deteksi penyakit tanaman secara real-time,

efisien, dan potensial untuk diterapkan di lingkungan pertanian yang terbatas akses teknologinya [33].

3. Md. M. Islam et al., “*A Deep Learning Model for Cotton Disease Prediction Using Fine-Tuning With Smart Web Application in Agriculture*,” diterbitkan pada tahun 2023. Penelitian ini mengusulkan pendekatan deteksi penyakit berbasis *deep learning* yang menyelidiki kinerja beberapa algoritma *transfer learning*. Dalam pemrosesan terlebih dahulu kumpulan data gambar menggunakan pengubahan ukuran, penajaman, penskalaan ulang, pemotongan, pembesaran, dan pembalikan horizontal agar sesuai dengan proses pemisahan untuk membagi kumpulan data gambar. Kemudian, beberapa algoritma *transfer learning* yang terkenal termasuk *VGG-16*, *VGG-19*, *Inception-V3* dan *Xception* diadopsi untuk dilatih dengan kumpulan data gambar kapas untuk prediksi penyakit. Selain itu, metrik kinerja seperti presisi, penarikan kembali dan pengukuran F1 digunakan untuk mengevaluasi pengklasifikasi dengan kumpulan data gambar uji. Akhirnya, model *Xception* telah dipilih karena memberikan tingkat akurasi tertinggi sebesar 98,70% untuk mengembangkan aplikasi pintar berbasis web untuk prediksi penyakit kapas di kehidupan nyata. Dalam penelitian ini berharap bahwa para profesional di bidang pertanian dapat menggunakannya untuk validasi diri, konsultasi, dan alasan pertanian lainnya [24].
4. L. Mkonyi et al., “*Early identification of TutaAbsoluta in Tomato Plants Using Deep Learning*,” diterbitkan pada tahun 2020. Penelitian ini mengusulkan model *deep learning* untuk mengidentifikasi hama *T. absoluta* pada tanaman tomat. Penelitian ini menggunakan *transfer learning* melalui model *VGG16*, *VGG19*, dan *ResNet50*, yang telah dilatih sebelumnya pada *Imagenet*, untuk melatih pengklasifikasi pada kumpulan data kami. Pelatihan model dilakukan menggunakan kumpulan data dengan 2145 gambar dengan gambar daun yang sehat dan terinfeksi yang dikumpulkan dari eksperimen internal. Model yang berkinerja tinggi adalah *VGG16*, yang mencapai akurasi keseluruhan sebesar 91,9% dalam klasifikasi 66 gambar yang sebelumnya tidak terlihat dari set pengujian. Hasilnya menunjukkan bahwa transfer learning adalah metode yang ampuh yang dapat mencapai akurasi tinggi dalam identifikasi hama *T. absoluta*

dari gambar daun tanaman tomat. Metode ini melakukan ekstraksi fitur otomatis, demi menghemat waktu peneliti dari tugas ekstraksi fitur yang padat karya yang biasanya menghasilkan hasil yang salah. Di masa mendatang, para akademisi dapat mengumpulkan lebih banyak kumpulan data untuk meningkatkan kinerja model kami untuk mengklasifikasikan gambar yang tidak terlihat. Akibatnya, model tersebut akan ditingkatkan dengan kemampuan mengukur tingkat keparahan invasi *T. absoluta* di lahan pertanian. Oleh karena itu, penelitian ini sebagai sistem pendukung keputusan untuk memungkinkan petani mengambil tindakan yang tepat untuk menyelamatkan lahan pertanian setelah mendeteksi invasi pada tahap awal pertumbuhan tanaman tomat [35].

5. T. Sultan *et al.*, “*LeafDNet: Transforming Leaf Disease Diagnosis Through Deep Transfer Learning*,” diterbitkan pada tahun 2024. Penelitian ini berhasil mengembangkan model deteksi penyakit tanaman berbasis deep transfer learning menggunakan arsitektur *Xception* yang telah dimodifikasi. Penambahan lapisan konvolusional dan *Dense* yang dapat dilatih, serta penerapan teknik regularisasi dan dropout, terbukti meningkatkan kemampuan model dalam mengekstraksi fitur kompleks dari citra daun tanaman. Dengan tingkat akurasi sebesar 98%, presisi 99%, *recall* 98%, dan F1-score 98%, model ini menunjukkan kinerja yang unggul dibandingkan metode tradisional maupun pendekatan *deep learning* lainnya. Hasil ini menunjukkan bahwa arsitektur yang diusulkan memiliki potensi besar untuk diterapkan secara luas dalam sistem deteksi dini penyakit tanaman, serta memberikan kontribusi nyata terhadap pengelolaan kesehatan tanaman yang lebih efisien dan berkelanjutan [36].
6. M. Mat Daud, A. Abualqumssan, F. ‘Atyka N. Rashid, and M. H. Md Saad, “*Durian Disease Classification Using Transfer Learning For Disease Management System*,” diterbitkan pada tahun 2023. Penelitian ini berhasil mengembangkan model deteksi penyakit daun pada tanaman durian berbasis transfer learning menggunakan arsitektur *VGG-19* dan *ResNet-9* dengan menggunakan beberapa kombinasi *hyperparameter*. Hasil ini menunjukkan bahwa *ResNet-9* mencapai tingkat akurasi yang mengesankan sebesar 99,1%

saat menggunakan pengoptimal *Adam* dengan laju pembelajaran maksimum 0,001 [37].

7. S. V, A. Bhagwat, and V. Laxmi, “*LeafSpotNet: A deep learning framework for Detecting Leaf Spot Disease in Jasmine Plants*,” diterbitkan pada tahun 2024. Penelitian ini mengembangkan sebuah pengklasifikasi berbasis *MobileNetV3* untuk tanaman melati, yang memanfaatkan *Convolutional neural network* ringan untuk mengidentifikasi tahap-tahap penyakit secara akurat. Model ini mengintegrasikan depth-wise convolution layers dan max pooling layers untuk ekstraksi fitur yang lebih baik, dengan fokus pada fitur tingkat rendah yang penting dalam mendeteksi penyakit. Melalui teknik pra-pemrosesan, termasuk augmentasi data menggunakan Conditional GAN dan seleksi fitur dengan Particle Swarm Optimization, model ini menunjukkan performa yang tangguh. Evaluasi pada dataset yang telah dikurasi menunjukkan akurasi pelatihan sebesar 97%, mencerminkan efektivitasnya. Pengujian di dunia nyata dengan kondisi yang beragam seperti sudut kamera ekstrem dan pencahayaan yang bervariasi menunjukkan ketahanan model, dengan akurasi pengujian antara 94% hingga 96%. Pendekatan komprehensif ini menegaskan pentingnya pengklasifikasi yang diusulkan dalam mengatasi tantangan penyakit bercak daun pada tanaman komersial [38].
8. Y. D. Wulandari, L. C. Munggaran, F. A. Setiawan, and I. A. Satya, “*Chili Leaf Health Classification Using Xception Pretrained Model*,” diterbitkan pada tahun 2024. Dalam penelitian ini berhasil mengklasifikasikan gambar daun cabai sehat dan tidak sehat menggunakan model CNN pretrained *Xception*. Dengan dataset berisi 2136 gambar, model ini mencapai akurasi 91% secara keseluruhan, dan menunjukkan tingkat keberhasilan 100% saat mengklasifikasikan 10 gambar uji secara tepat. Hasil ini membuktikan bahwa model *Xception* yang dilatih sebelumnya pada dataset *Imagenet* mampu secara efektif mengidentifikasi kondisi daun cabai [39].
9. J. O. Carnagie, A. R. Prabowo, E. P. Budiana, and I. K. Singgih, “*Essential Oil Plants Image Classification Using Xception Model*,” diterbitkan pada tahun 2022. Penelitian ini mengembangkan model klasifikasi gambar tanaman penghasil minyak atsiri menggunakan metode *transfer learning* berbasis

arsitektur *Xception*. Model ini berhasil mengklasifikasikan 10 jenis tanaman minyak atsiri dengan performa yang baik, mencapai akurasi training sebesar 93,36% dan validation sebesar 77,16% [40].

10. Y. Zhu, H. JiaYI, Y. Li, and W. Li, “*Image Identification of Cashmere and Wool Fibers Based on the Improved Xception Network*,” diterbitkan pada tahun 2022. Penelitian ini mengidentifikasi gambar serat kasmir dan wol dan mengatasi masalah fitur yang tidak mencukupi dan *overfitting* dalam pelatihan jaringan. Penelitian ini menggunakan jaringan *Xception* menggunakan fungsi aktivasi *swish* yang ditingkatkan untuk mengurangi *overfitting*. Hasil penelitian ini menunjukkan bahwa akurasi identifikasi model adalah 98,95% [41].

Tabel 2.1 Penelitian Terkait

Penulis	Tahun	Judul	Kesimpulan
A. L. Sabarre, A. S. Navidad, D. S. Torbela, and J. J. Adtoon. [32]	2021	<i>Development of durian leaf disease detection on Android device</i>	Penelitian ini mengembangkan sistem deteksi penyakit daun durian berbasis android menggunakan metode <i>MobileNetV2</i> dan menghasilkan akurasi sebesar 91%
V. Gonzalez-Huitron, J. A. León-Borges, A. E. Rodriguez-Mata, L. E. Amabilis-Sosa, B. Ramírez-Pereda, and H. Rodriguez. [33]	2021	<i>Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4</i>	Penelitian ini mengembangkan sistem deteksi penyakit daun tomat menggunakan berbagai arsitektur seperti <i>MobileNetV2</i> , <i>EfficientNetB0</i> , <i>Xception</i> , dan lainnya. <i>Xception</i> memberikan performa terbaik dibandingkan arsitektur lainnya.
Md. M. Islam et al. [34]	2023	<i>A deep learning model for cotton disease prediction using fine-tuning with smart web application in agriculture</i>	Penelitian ini mengembangkan sistem deteksi penyakit pada tanaman kapas menggunakan beberapa algoritma transfer learning seperti <i>VGG16</i> , <i>Inception</i> , dan <i>Xception</i> . Hasil penelitian ini menunjukan model <i>Xception</i> memiliki performa yang tinggi sebesar 98,70%

L. Mkonyi <i>et al.</i> , [35]	2020	<i>Early identification of Tuta absoluta in tomato plants using deep learning</i>	Penelitian ini mengembangkan sistem untuk mengidentifikasi hama <i>T. absoluta</i> pada tanaman tomat. Hasil penelitian ini menunjukkan bahwa arsitektur <i>VGG16</i> memiliki akurasi yang tinggi sebesar 91.9%
T. Sultan <i>et al.</i> , [36]	2024	<i>LeafDNet: Transforming Leaf Disease Diagnosis Through Deep Transfer Learning</i>	Penelitian ini mengembangkan deteksi penyakit tanaman berbasis deep transfer learning menggunakan arsitektur <i>Xception</i> dan menghasilkan akurasi sebesar 98%.
M. Mat Daud, A. Abualqumssan, F. 'Atyka N. Rashid, and M. H. Md Saad, [37]	2023	<i>Durian Disease Classification Using Transfer Learning For Disease Management System</i>	Penelitian ini menggunakan <i>deep learning</i> dengan arsitektur <i>VGG-19</i> dan <i>ResNet-9</i> mencapai tingkat akurasi sebesar 99,1% saat menggunakan <i>optimizer Adam</i> dengan <i>learning rate</i> 0.001 untuk mengklasifikasikan penyakit daun pada tanaman durian.
S. V, A. Bhagwat, and V. Laxmi, [38]	2024	<i>LeafSpotNet: A deep learning framework for detecting leaf spot disease in jasmine plants</i>	Penelitian ini mengembangkan sistem klasifikasi berbasis <i>MobileNetV3</i> untuk tanaman melati dan menunjukkan akurasi pelatihan sebesar 97%
Y. D. Wulandari, L. C. Munggaran, F. A. Setiawan, and I. A. Satya [39]	2024	<i>Chili Leaf Health Classification using Xception Pretrained Model</i>	Penelitian ini mengembangkan sistem klasifikasi untuk daun cabai sehat dan tidak sehat menggunakan model <i>CNN pretrained Xception</i> dan mencapai akurasi sebesar 91%.
J. O. Carnagie, A. R. Prabowo, E. P. Budiana, and I. K. Singgih [40]	2022	<i>Essential Oil Plants Image Classification Using Xception Model</i>	Penelitian ini mengembangkan model klasifikasi gambar tanaman penghasil minyak astri menggunakan metode <i>transfer learning</i> berbasis arsitektur <i>Xception</i> . Model ini mencapai akurasi

			<i>training</i> sebesar 93,36% dan <i>validation</i> sebesar 77,16%
Y. Zhu, H. JiaYI, Y. Li, and W. Li [41]	2022	<i>Image identification of cashmere and wool fibers based on the improved Xception network</i>	Penelitian ini mengidentifikasi gambar serat kasmir dan wol dan mengatasi masalah fitur yang tidak mencukupi dan <i>overfitting</i> dalam pelatihan jaringan. Penelitian ini menggunakan jaringan <i>Xception</i> menggunakan fungsi aktivasi swish yang ditingkatkan untuk mengurangi <i>overfitting</i> . Hasil penelitian ini menunjukkan bahwa akurasi identifikasi model adalah 98,95%

III. METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Pelaksanaan penelitian dan pengembangan alat ini dilakukan pada waktu dan tempat yang telah ditentukan sebagai berikut :

Waktu : Maret 2025 - Juli 2025

Tempat : Lab. Komputer Universitas Lampung

Tabel 3.1 Waktu Penelitian

No.	Aktivitas	Maret 2025	April 2025	Mei 2025	Juni 2025	Juli 2025
1.	Identifikasi Masalah					
2.	Studi Literatur					
2	Pengumpulan Dataset					
3	<i>Preprocessing</i> Dataset					
4	Perancangan Model					
5	Pelatihan Model					
6	Evaluasi Model					
6	Penulisan Laporan					

3.2 Alat dan Bahan Penelitian

Adapun alat dalam penelitian ini adalah sebagai berikut :

Tabel 3. 2 Alat Penelitian

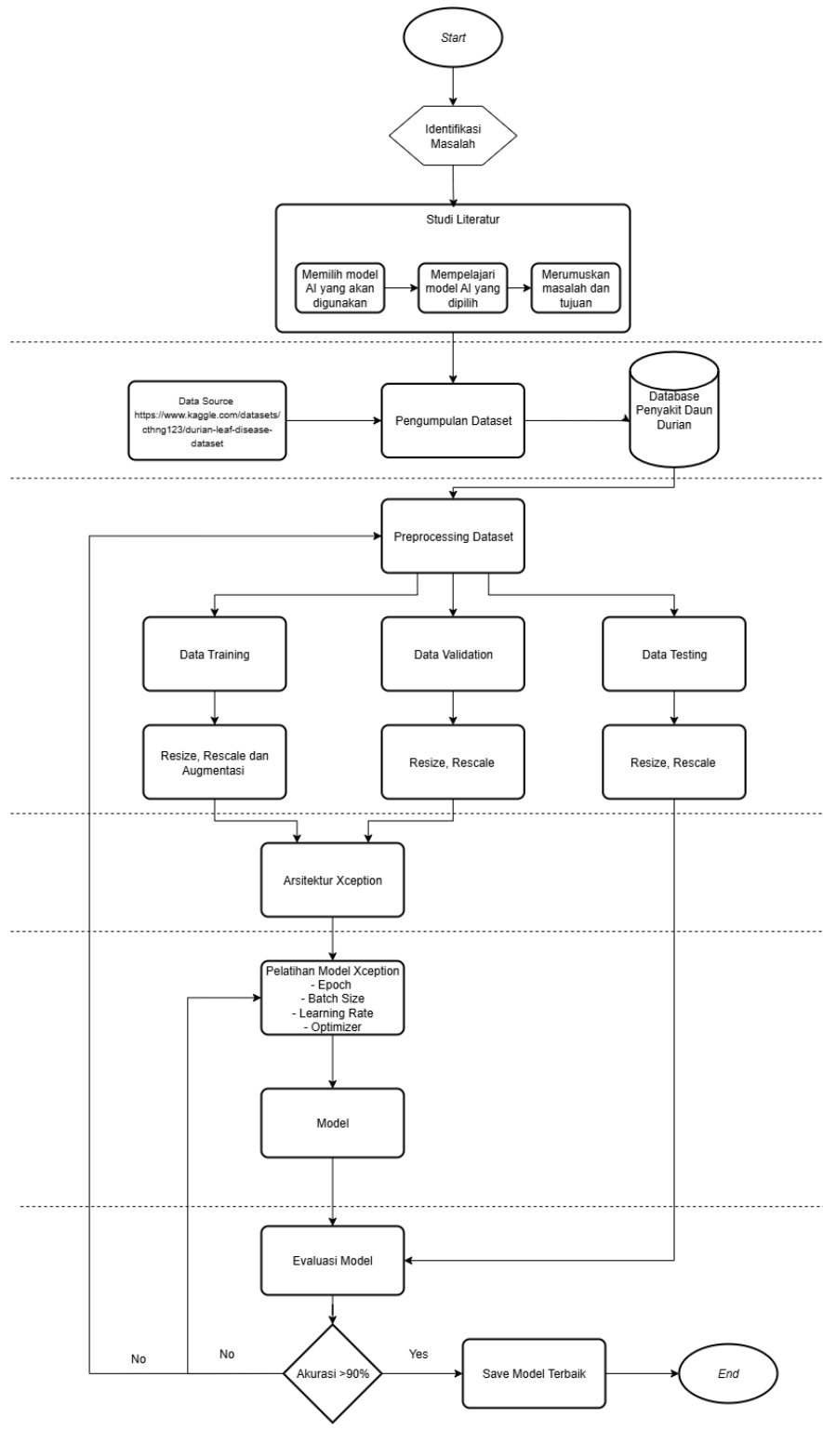
No.	Nama Alat	Spesifikasi	Deskripsi
1.	Laptop Tuf F15	Intel® Core™ i5-10300H, Memory 8GB DDR4, Storage 512 GB, Windows 11 64bit, NVIDIA® GeForce® GTX 1650 4GB.	Perangkat keras yang digunakan untuk seluruh proses penelitian sampai dengan selesai
2.	Google Collaboratory	-	Layanan berbasis cloud yang disediakan oleh Google, memungkinkan pengguna untuk menulis dan menjalankan kode Python melalui browser.
3.	Python	v3.12.11	Bahasa pemrograman yang digunakan dalam pengembangan aplikasi berbasis machine learning dan <i>deep learning</i> .
4.	Tensorflow & Keras	v2.18.0	TensorFlow adalah pustaka open source yang dikembangkan oleh Google untuk komputasi numerik dan machine learning, sedangkan Keras merupakan antarmuka pemrograman tingkat tinggi di atas TensorFlow yang dirancang untuk

			memudahkan pembangunan dan pelatihan model <i>deep learning</i> .
5.	Google Drive	-	Layanan penyimpanan berbasis <i>cloud</i> yang disediakan oleh Google. Alat ini digunakan untuk tempat penyimpanan dataset gambar penyakit daun tanaman durian
6	Draw.io	-	Layanan berbasis website yang digunakan untuk membuat diagram
7	Notion	-	Layanan berbasis website yang digunakan untuk melakukan pekerjaan tim

Tabel 3. 3 Bahan Penelitian

No.	Nama Bahan	Spesifikasi	Deskripsi
1.	Dataset <i>Durian Leaf Disease</i>	Dataset mencakup lima kelas dengan total gambar sebesar 4437	Data digunakan untuk membangun model yang datanya diambil dari : https://www.kaggle.com/datasets/ct hng123/durian-leaf-disease-dataset

3.3 Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian

3.3.1 Identifikasi Masalah

Tahap awal dalam penelitian ini adalah melakukan identifikasi masalah yang akan menjadi fondasi utama dalam keseluruhan proses penelitian. Proses ini mencakup eksplorasi berbagai fenomena terkait tantangan dalam identifikasi penyakit daun pada tanaman durian yang memerlukan keahlian khusus. Pendekatan ini bertujuan untuk memastikan bahwa penelitian memiliki arah yang sesuai dengan kebutuhan nyata terhadap permasalahan yang sedang dihadapi.

3.3.2 Studi Literatur

Tahap studi literatur dilakukan untuk memilih model *AI* yang tepat digunakan untuk mengklasifikasikan penyakit daun pada tanaman durian. Berdasarkan hasil kajian tersebut, arsitektur *Xception* dipilih karena menunjukkan performa unggul dalam hal akurasi dibandingkan arsitektur lain seperti *VGG16*, *ResNet*, dan *Inception*, khususnya dalam konteks klasifikasi penyakit daun pada tanaman [9]. Studi literatur ini juga mencakup pemahaman mendalam mengenai penerapan *Convolutional neural network* khususnya *Xception*. Penelusuran referensi dilakukan melalui berbagai sumber terpercaya, seperti jurnal ilmiah dari database *IEEE Xplore*, *ScienceDirect*, dan *Google Scholar*, serta beberapa situs website. Selain itu, langkah langkah berikutnya adalah merumuskan masalah secara spesifik dan merumuskan tujuan penelitian yang akan menjadi fokus pembahasan selanjutnya.

3.3.3 Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini merupakan dataset publik yang tersedia di platform *kaggle.com*, yaitu "Durian Leaf Disease Dataset" yang dikembangkan oleh pengguna dengan nama akun *ĐứcThắng123* [42]. Dataset ini dilisensikan di bawah *Creative Commons Attribution 4.0 International License (CC BY 4.0)*. Secara keseluruhan, dataset mencakup 4.437 citra daun durian yang terbagi ke dalam lima kelas, yakni: *algal leaf spot*, *allocaridara attack*, *healthy leaf*, *leaf blight*, dan *phomopsis leaf spot*. Anotasi pada dataset ini dilakukan oleh dua orang pakar pertanian berpengalaman dengan keahlian khusus dalam identifikasi penyakit

daun durian. Para pakar ini terlebih dahulu mempelajari definisi kelas dan contoh gambar, kemudian secara independen memberi label pada seluruh dataset. Contoh sampel gambar dari setiap kelas yang telah dikumpulkan dapat dilihat pada gambar 3.2.



Gambar 3.2 Dataset [42]

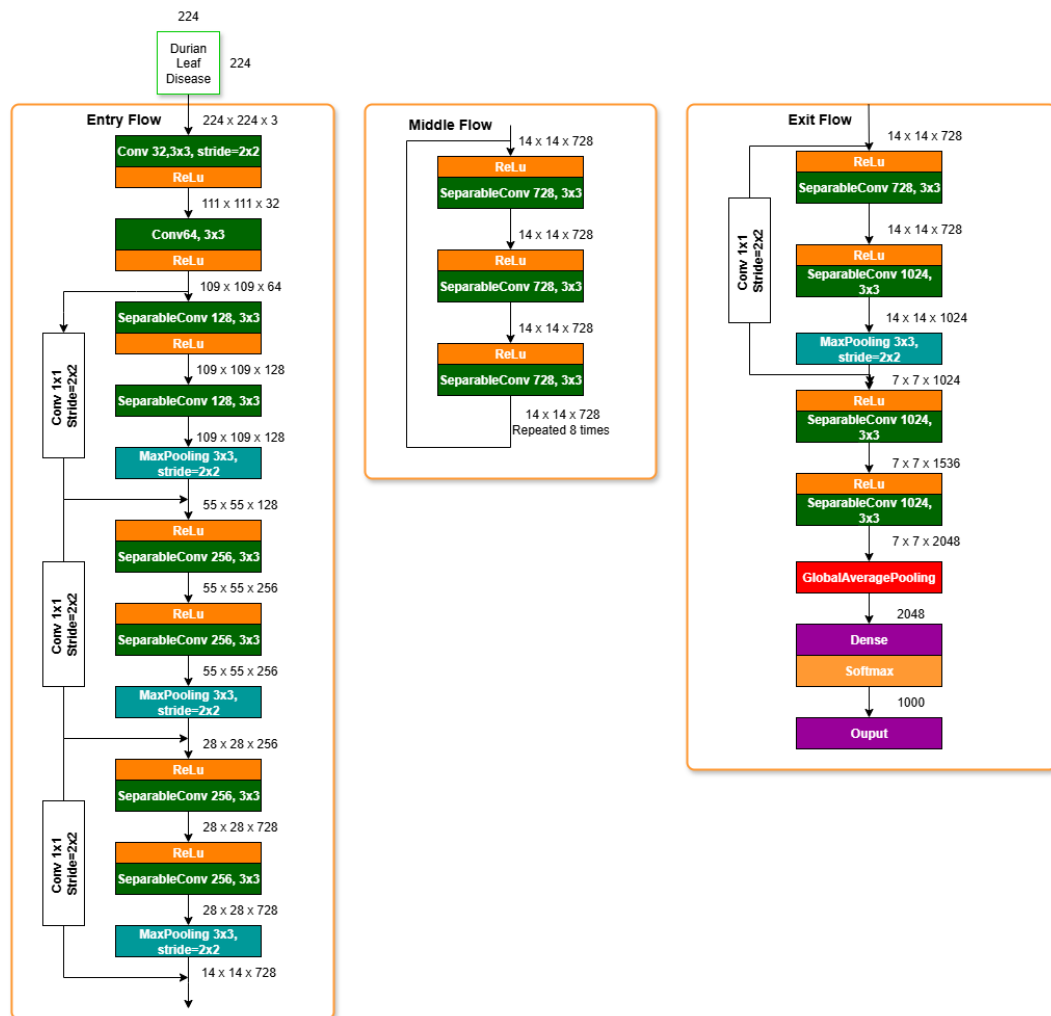
3.3.4 Preprocessing Dataset

Sebelum dataset digunakan dalam proses pelatihan model, terlebih dahulu melalui tahapan *preprocessing* agar memastikan data siap digunakan untuk membuat model. Dalam dataset ini, gambar dibagi menjadi tiga subset utama yaitu data pelatihan 70% (data yang digunakan untuk melatih model), data validasi 10% (data yang digunakan untuk mengukur performa model saat melakukan pelatihan), dan data pengujian 20% (data yang digunakan untuk evaluasi model untuk melihat performa yang dilakukan model). Gambar-gambar yang sudah dikumpulkan diubah ukurannya menjadi 224x 224 piksel agar sesuai dengan arsitektur *Xception* yang digunakan. Selanjutnya, seluruh nilai piksel dari gambar dinormalisasi dengan mereskalanya ke rentang $[0, 1]$ dengan cara $rescale=1./255$. dilakukan untuk meningkatkan performa dan kestabilan pelatihan model.

Selanjutnya, dilakukan proses augmentasi data pada subset data pelatihan secara *on the fly* selama proses pelatihan berlangsung menggunakan pustaka *ImageDataGenerator* dan secara menyeimbangkan dataset. Augmentasi ini bertujuan untuk meningkatkan jumlah variasi data sehingga model dapat belajar dari berbagai kondisi visual dan mengurangi terjadinya *overfitting* dan *underfitting*. Teknik augmentasi yang diterapkan meliputi *width and shift range*, *rotation range*, *shear range*, *zooming*, dan *horizontal flip*. Proses ini kemudian menghasilkan *batch* gambar yang akan digunakan dalam pelatihan model. Data pelatihan dan validasi

masing-masing dibaca dari direktori tersendiri, lalu diubah ke format *one-hot encoding* dengan pengaturan *class_mode='categorical'*. Untuk data validasi, hanya dilakukan normalisasi tanpa augmentasi agar evaluasi model dilakukan pada data sebenarnya tanpa perubahan. Dengan pendekatan ini, proses *preprocessing* tidak hanya mempersiapkan data secara teknis, tetapi juga meningkatkan kualitas pembelajaran model selama pelatihan berlangsung.

3.3.5 Arsitektur *Xception*



Gambar 3.3 Arsitektur *Xception*

Arsitektur model yang digunakan dalam penelitian ini adalah *Xception* yang menggunakan teknik *depthwise separable convolution* untuk meningkatkan

efisiensi dan akurasi. Arsitektur ini terdiri atas 36 lapisan konvolusionel utama yang dibagi kedalam tiga bagian yaitu *entry flow*, *middle flow*, dan *exit flow*, serta *fully connected* dengan 1.000 *Neuron (Dense layer)*, dan fungsi aktivasi *softmax* yang menghasilkan prediksi untuk 1.000 kelas pada dataset *Imagenet*. Dalam penelitian ini model *Xception* yang digunakan dimuat dengan bobot awal dari pra-pelatihan pada dataset *Imagenet*. Lapisan akhir klasifikasi asli pada model ini kemudian diganti dan disesuaikan dengan jumlah kelas pada dataset penyakit pada daun durian, yaitu lima kelas. Untuk mendukung performa klasifikasi yang optimal, beberapa konfigurasi pada lapisan *Dense* ditambahkan, termasuk penggunaan lapisan *dropout* untuk mencegah *overfitting*.

3.3.6 Pelatihan Model

Pada proses pelatihan dimulai dengan memuat model *Xception* dengan bobot yang telah dilatih sebelumnya pada dataset *Imagenet*, yang berisi jutaan gambar dari 1000 kelas. Langkah ini memanfaatkan pengetahuan awal dalam mengenali fitur visual umum seperti tepi, tekstur dan bentuk dasar. Setelah itu, lapisan klasifikasi asli pada bagian akhir model kemudian diganti dengan lapisan baru yang sesuai dengan jumlah kelas dan dataset penyakit daun serta ditambahkan lapisan *Dense* dan *dropout*.

Pada tahap awal pelatihan, seluruh lapisan konvolusionel *Xception* dibekukan (*frozen*) agar hanya lapisan baru yang dilatih terlebih dahulu agar penyesuaian awal terhadap data baru tidak merusak fitur yang sudah dipelajari dari *Imagenet*. Selanjutnya, dilakukan pelatihan awal terhadap model dengan kondisi lapisan *convolutional* tetap dibekukan. Setelah model menunjukkan konvergensi awal pada data baru, dilakukan pelatihan lanjutan dengan membuka kembali sebagian atau seluruh lapisan konvolusionel, bersama dengan lapisan klasifikasi. Strategi ini bertujuan untuk memungkinkan penyesuaian bobot secara lebih menyeluruh, sehingga model dapat mengekstraksi fitur yang lebih spesifik dan relevan terhadap dataset penyakit daun durian.

Untuk memperoleh performa terbaik proses pelatihan model dilakukan secara iteratif (pelatihan yang dijalankan berulang kali) hal ini untuk mengamati performa

model disetiap siklus pelatihan. Dalam proses ini, struktur arsitektur dasar dari *Xception* tidak dimodifikasi, sehingga dapat dilakukan serangkaian eksperimen pada pengaturan hyperparameter untuk memperoleh performa terbaik. *Hyperparameter* tersebut meliputi jumlah *epoch*, *batch size*, *learning rate*, dan *optimizer*. Eksperimen dilakukan dengan mencoba berbagai kombinasi dari *hyperparameter* tersebut guna mengevaluasi pengaruhnya terhadap performa model, khususnya terhadap metrik akurasi. Setiap kombinasi diuji untuk mengamati perubahan hasil yang diperoleh, sehingga dapat ditentukan konfigurasi terbaik yang menghasilkan akurasi tertinggi dan kinerja paling optimal dari model *Xception* pada tugas klasifikasi citra daun durian.

3.3.7 Evaluasi Model

Tahapan evaluasi dimulai dengan mempersiapkan data pengujian yang sebelumnya telah dipisahkan dari dataset utama. Data ini tidak digunakan selama proses pelatihan model, sehingga hasil evaluasinya mencerminkan kemampuan model dalam menghadapi data baru yang belum pernah dikenalnya. Evaluasi dilakukan untuk mengetahui seberapa baik model mampu mengklasifikasikan gambar daun durian ke dalam kategori yang benar, seperti daun sehat atau berbagai jenis penyakit. Pada tahap ini, digunakan beragam metrik evaluasi, yaitu: *accuracy*, *precision*, *recall*, dan *f1-score*.

Selain metrik tersebut, digunakan juga *confusion matrix* yaitu berupa tabel yang memperlihatkan perbandingan antara hasil prediksi model dan label sebenarnya dari data uji. Dari sini dapat diketahui kelas mana yang sering mengalami salah prediksi. Analisis terhadap hasil evaluasi ini penting karena dapat menunjukkan kelebihan dan kekurangan model.

V. SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil dan analisis dari penelitian yang telah dilakukan dapat disimpulkan sebagai berikut :

1. Model berhasil dikembangkan menggunakan arsitektur *Xception* dengan akurasi sebesar 95% dalam mengklasifikasikan jenis penyakit daun pada tanaman durian.
2. Model *Xception* yang dikembangkan menunjukkan performa tinggi pada tugas klasifikasi penyakit daun pada tanaman durian, dengan akurasi sebesar 95% pada data uji, *recall* 94%, *precision* 95% dan *f1-score* 95%. Analisis *confusion matrix* menunjukkan bahwa model mengalami kelemahan dalam mengenali kelas *Phomopsis leaf spot*. Kelemahan ini disebabkan oleh dua faktor utama: kemiripan ciri fisik dengan penyakit lain seperti *Algal leaf spot*, dan sebagian besar citra penyakit ini masih menunjukkan area daun yang sehat, sehingga model sering salah mengklasifikasikannya sebagai kelas *healthy leaf*.
3. Penyesuaian arsitektur pada lapisan akhir model *Xception* terbukti efektif dalam meningkatkan kinerja. Hal ini dilakukan dengan menambahkan satu lapisan dense berukuran 128 *neuron* pada lapisan terakhir. Penambahan kompleksitas arsitektur tidak memberikan peningkatan yang signifikan, bahkan cenderung menurunkan performa model. Selain itu, kinerja model juga dipengaruhi oleh pengaturan *hyperparameter* pelatihan. Konfigurasi terbaik diperoleh dengan menggunakan 60 *epoch*. Peningkatan jumlah *epoch* dapat meningkatkan performa model, tetapi setelah titik tertentu, model menunjukkan kecenderungan jenuh sehingga penambahan *epoch* tidak lagi memberikan peningkatan yang berarti. Ukuran *batch size* 8 juga ditemukan sebagai yang paling optimal. Hal ini karena *batch size* yang besar cenderung menyebabkan model mengalami *overfitting* dikarenakan model belajar dengan cakupan data

yang terlalu luas pada setiap langkah. Selanjutnya, nilai *learning rate* 0.001 terbukti paling efektif. Nilai yang terlalu besar menyebabkan pelatihan yang tidak stabil, sementara nilai yang terlalu kecil menyebabkan *overfitting*. Terakhir, penggunaan *optimizer Adam* menghasilkan akurasi tertinggi dengan loss yang rendah, menunjukkan performa yang lebih unggul dibandingkan dengan *optimizer* lain seperti *Nadam* dan *Rmsprop*.

4. Peningkatan jumlah lapisan yang dibuka saat *fine-tuning* meningkatkan performa model, dimana membuka seluruh lapisan terbukti paling efektif untuk menyesuaikan representasi fitur, meningkatkan akurasi, dan memperkuat kemampuan generalisasi dalam klasifikasi citra daun durian.

5.2 Saran

Adapun saran yang dapat diberikan berdasarkan penelitian yang telah dilakukan untuk penelitian selanjutnya adalah sebagai berikut :

1. Fokus pada peningkatan kinerja klasifikasi untuk kelas *Phomopsis Leaf Spot* dengan penambahan jumlah dataset, dikarenakan pada kelas ini menunjukkan performa yang lebih rendah dibandingkan kelas lainnya.
2. Melakukan eksplorasi terhadap arsitektur *Convolutional neural network* lainnya dengan memanfaatkan kombinasi *hyperparameter* optimal yang telah ditemukan pada penelitian ini untuk memperoleh performa yang lebih baik.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [1] N. Najira, E. Selviyanti, Y. Br. Tobing, K. Kasmawati, R. Sianturi, and A. B. Suwardi, “Diversitas Kultivar tanaman Durian (*Durio zabethinus* Murr.) Ditinjau dari Karakter Morfologi,” *JBT*, vol. 20, no. 2, pp. 185–193, May 2020, doi: 10.29303/jbt.v20i2.1871.
- [2] Badan Pusat Statistik, “Produksi Tanaman Buah-buahan dan Sayuran Tahunan Menurut Provinsi dan Jenis Tanaman, 2024.” [Online]. Available: <https://www.bps.go.id/id/statistics-table/2/NjIjMg==/produksi-tanaman-buah-buahan.html>
- [3] Badan Pusat Statistik, “Rata-Rata Harga Buah-Buahan di Kabupaten Ngawi.” [Online]. Available: <https://ngawikab.bps.go.id/id/statisticstable/2/MTczIzI=/rata-rata-harga-buah-buahan-di-kabupaten-ngawi.html>
- [4] Ni Luh Made Pertiwi F and Yana Gabriella Wijaya, “10 Tempat Wisata Durian.” [Online]. Available: <https://travel.kompas.com/read/2020/05/10/194000327/catat-10-tempat-wisata-durian-petik-dan-makan-sepuasnya?page=all>
- [5] F. S. Alamri, T. Sadad, A. S. Almasoud, R. A. Aurangzeb, and A. Khan, “Mango Disease Detection Using Fused Vision Transformer with ConvNeXt Architecture,” *CMC*, vol. 83, no. 1, pp. 1023–1039, 2025, doi: 10.32604/cmc.2025.061890.
- [6] S. Gupta and A. K. Tripathi, “Fruit and Vegetable Disease Detection and Classification: Recent Trends, Challenges, and Future Opportunities,” *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108260, Jul. 2024, doi: 10.1016/j.engappai.2024.108260.
- [7] A. J., J. Eunice, D. E. Popescu, M. K. Chowdary, and J. Hemanth, “Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications,” *Agronomy*, vol. 12, no. 10, p. 2395, Oct. 2022, doi: 10.3390/agronomy12102395.

- [8] Y. M. Abd Algani, O. J. Marquez Caro, L. M. Robladillo Bravo, C. Kaur, M. S. Al Ansari, and B. Kiran Bala, “*Leaf Disease Identification and Classification Using Optimized Deep Learning*,” *Measurement: Sensors*, vol. 25, p. 100643, Feb. 2023, doi: 10.1016/j.measen.2022.100643.
- [9] M. R. A. Rashid *et al.*, “*An Ensemble Learning Framework with Explainable AI for Interpretable Leaf Disease Detection*,” *Array*, vol. 26, p. 100386, Jul. 2025, doi: 10.1016/j.array.2025.100386.
- [10] M. M. Daud, A. Abualqumssan, F. ‘Atyka N. Rashid, M. H. M. Saad, W. M. D. W. Zaki, and N. S. M. Satar, “*Durian Disease Classification using Vision Transformer for Cutting-Edge Disease Control*,” *IJACSA*, vol. 14, no. 12, 2023, doi: 10.14569/ijacsa.2023.0141246.
- [11] T. Wahyudi, “Studi Kasus Pengembangan dan Penggunaan Artificial Intelligence (AI) Sebagai Penunjang Kegiatan Masyarakat Indonesia,” vol. 9, no. 1, 2023.
- [12] C. Janiesch, P. Zschech, and K. Heinrich, “*Machine Learning and Deep Learning*,” *Electron Markets*, vol. 31, no. 3, pp. 685–695, Sep. 2021, doi: 10.1007/s12525-021-00475-2.
- [13] R. Budi, *Deep Learning dengan Python*. Semarang: Yayasan Prima Agus Teknik, 2022.
- [14] P. Purwono, A. Ma’arif, W. Rahmaniari, H. I. K. Fathurrahman, A. Z. K. Frisky, and Q. M. U. Haq, “*Understanding of Convolutional Neural Network (CNN): A Review*,” *IJRCS*, vol. 2, no. 4, pp. 739–748, Jan. 2023, doi: 10.31763/ijrcs.v2i4.888.
- [15] G. W. Intyanto, “*Klasifikasi Citra Bunga dengan Menggunakan Deep Learning: CNN (Convolution Neural Network)*,” *JAEL*, vol. 7, no. 3, p. 80, Dec. 2021, doi: 10.19184/jaei.v7i3.28141.
- [16] A. Balasundaram, P. Sundaresan, A. Bhavsar, M. Mattu, M. S. Kavitha, and A. Shaik, “*Tea Leaf Disease Detection Using Segment Anything Model and Deep Convolutional Neural Networks*,” *Results in Engineering*, vol. 25, p. 103784, Mar. 2025, doi: 10.1016/j.rineng.2024.103784.
- [17] R. A. Saputra, D. R. R. Putra, and M. A. Asyrofi, “*Implementasi Convolutional Neural Network (CNN) Untuk Mendeteksi Pengguna Masker Pada Gambar*,”

- vol. Vol. 11 No. 3, pISSN: 2303-0577 eISSN: 2830-7062, doi: <https://doi.org/10.23960/jitet.v11i3.3286>.
- [18] H. K. Daniel Kukiela, *Neural Network from Scratch in Python*. Sentdex, Kinsley Enterprises, 2020. [Online]. Available: <https://nnfs.io/>
 - [19] M. A. Hasan *et al.*, “Prediction of Fetal Brain Gestational Age Using Multihead Attention With Xception,” *Computers in Biology and Medicine*, vol. 182, p. 109155, Nov. 2024, doi: 10.1016/j.compbiomed.2024.109155.
 - [20] Afis Julianto, Andi Sunyoto, and Ferry Wahyu Wibowo, “Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi,” *teknimedia*, vol. 3, no. 2, pp. 98–105, Dec. 2022, doi: 10.46764/teknimedia.v3i2.77.
 - [21] Reyvan Maulid, “Konsep Hyperparameter Tuning pada Machine Learning.” [Online]. Available: <https://dqlab.id/konsep-hyperparameter-tuning-pada-machine-learning>
 - [22] T. M. Adepoju, M. O. Oladele, M. O. Akintunde, and A. M. Ogunleye, “Effect of Fine-tuning Transfer Learning Layers: A Case Study of Breast Cancer Classification,” *fuoyejet*, vol. 9, no. 4, pp. 660–668, Feb. 2025, doi: 10.4314/fuoyejet.v9i4.15.
 - [23] Y. Liu, S. Agarwal, and S. Venkataraman, “AutoFreeze: Automatically Freezing Model Blocks to Accelerate Fine-tuning,” Apr. 03, 2021, *arXiv*: arXiv:2102.01386. doi: 10.48550/arXiv.2102.01386.
 - [24] R. Sadik, A. Majumder, A. A. Biswas, B. Ahammad, and Md. M. Rahman, “An in-Depth Analysis of Convolutional Neural Network Architectures with Ttransfer Learning for Skin Disease Diagnosis,” *Healthcare Analytics*, vol. 3, p. 100143, Nov. 2023, doi: 10.1016/j.health.2023.100143.
 - [25] S. Suhendar, A. Purnama, and E. Fauzi, “Deteksi Penyakit Pada Daun Tanaman Ubi Jalar Menggunakan Metode Convolutional Neural Network,” *J. Ilm. Inform. Glob.*, vol. 14, no. 3, pp. 62–67, Dec. 2023, doi: 10.36982/jiig.v14i3.3478.
 - [26] S. Timilsina, S. Sharma, and S. Kondo, “Advancements in Maize Leaf Disease Detection, Segmentation and Classification: A Review,” *Biosystems*

- Engineering*, vol. 255, p. 104162, Jul. 2025, doi: 10.1016/j.biosystemseng.2025.104162.
- [27] Angelina M. T. I. Sambu Ua *et al.*, “Penggunaan Bahasa Pemrograman Python Dalam Analisis Faktor Penyebab Kanker Paru-Paru,” *jupti*, vol. 2, no. 2, pp. 88–99, Jul. 2023, doi: 10.55606/jupti.v2i2.1742.
- [28] M. A. Abu, N. H. Indra, A. H. A. Rahman, N. A. Sapiee, and I. Ahmad, “A study on Image Classification based on Deep Learning and Tensorflow,” vol. 12, no. 4, 2019.
- [29] R. Gelar Guntara, “Deteksi Atap Bangunan Berbasis Citra Udara Menggunakan Google Colab dan Algoritma Deep Learning YOLOv7,” *JMASIF*, vol. 2, no. 1, pp. 9–18, May 2023, doi: 10.59431/jmasif.v2i1.156.
- [30] R. Poojary, R. Raina, and A. Kumar Mondal, “Effect of Data-Augmentation on Fine-Tuned CNN Model Performance,” *IJ-AI*, vol. 10, no. 1, p. 84, Mar. 2021, doi: 10.11591/ijai.v10.i1.pp84-92.
- [31] W. Muhammad, S. Aramvith, and T. Onoye, “Multi-scale Xception Based Depthwise Separable Convolution for Single Image Super-Resolution,” *PLoS ONE*, vol. 16, no. 8, p. e0249278, Aug. 2021, doi: 10.1371/journal.pone.0249278.
- [32] A. L. Sabarre, A. S. Navidad, D. S. Torbela, and J. J. Adtoon, “Development of Durian Leaf Disease Detection on Android Device,” *IJECE*, vol. 11, no. 6, p. 4962, Dec. 2021, doi: 10.11591/ijece.v11i6.pp4962-4971.
- [33] V. Gonzalez-Huitron, J. A. León-Borges, A. E. Rodriguez-Mata, L. E. Amabilis-Sosa, B. Ramírez-Pereda, and H. Rodriguez, “Disease Detection in Tomato Leaves via CNN With Lightweight Architectures Implemented in Raspberry Pi 4,” *Computers and Electronics in Agriculture*, vol. 181, p. 105951, Feb. 2021, doi: 10.1016/j.compag.2020.105951.
- [34] Md. M. Islam *et al.*, “A Deep Learning Model for Cotton Disease Prediction Using Fine-Tuning With Smart Web Application in Agriculture,” *Intelligent Systems with Applications*, vol. 20, p. 200278, Nov. 2023, doi: 10.1016/j.iswa.2023.200278.

- [35] L. Mkonyi *et al.*, “Early Identification of *Tuta Absoluta* in Tomato Plants Using Deep Learning,” *Scientific African*, vol. 10, p. e00590, Nov. 2020, doi: 10.1016/j.sciaf.2020.e00590.
- [36] T. Sultan *et al.*, “LeafDNet: Transforming Leaf Disease Diagnosis Through Deep Transfer Learning,” *Plant Direct*, vol. 9, no. 2, p. e70047, Feb. 2025, doi: 10.1002/pld3.70047.
- [37] M. Mat Daud, A. Abualqumssan, F. ‘Atyka N. Rashid, and M. H. Md Saad, “Durian Disease Classification Using Transfer Learning For Disease Management System,” *JISTM*, vol. 8, no. 33, pp. 67–77, Dec. 2023, doi: 10.35631/JISTM.833006.
- [38] S. V. A. Bhagwat, and V. Laxmi, “LeafSpotNet: A Deep Learning Framework for Detecting Leaf Spot Disease in Jasmine Plants,” *Artificial Intelligence in Agriculture*, vol. 12, pp. 1–18, Jun. 2024, doi: 10.1016/j.aiia.2024.02.002.
- [39] Y. D. Wulandari, L. C. Munggaran, F. A. Setiawan, and I. A. Satya, “Chili Leaf Health Classification using Xception Pretrained Model,” *SISTEMASI*, vol. 13, no. 3, p. 1084, May 2024, doi: 10.32520/stmsi.v13i3.3943.
- [40] J. O. Carnagie, A. R. Prabowo, E. P. Budiana, and I. K. Singgih, “Essential Oil Plants Image Classification Using Xception Model,” *Procedia Computer Science*, vol. 204, pp. 395–402, 2022, doi: 10.1016/j.procs.2022.08.048.
- [41] Y. Zhu, H. JiaYI, Y. Li, and W. Li, “Image Identification of Cashmere and Wool Fibers Based on the Improved Xception Network,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 9301–9310, Nov. 2022, doi: 10.1016/j.jksuci.2022.09.009.
- [42] ĐứcThắng123, “Durian Leaf Disease Dataset.” [Online]. Available: <https://www.kaggle.com/datasets/cthng123/durian-leaf-disease-dataset>