



**ANALISIS MODEL MOBILENETV2 UNTUK KLASIFIKASI  
TANAMAN BERPOLEN**



Program Studi

S1 Sistem Informasi

Oleh:

UNIVERSITAS  
**Dinamika**

**MUHAMMAD HARIS MAULANA KOLIQ ABIANSYAH**

**21410100029**

---

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2025**

**ANALISIS MODEL MOBILENETV2 UNTUK KLASIFIKASI  
TANAMAN BERPOLEN**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana**



UNIVERSITAS  
**Dinamika**  
Oleh:  
Nama : Muhammad Haris M.K.A  
NIM : 21410100029  
Program Studi : S1 Sistem Informasi

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
UNIVERSITAS DINAMIKA  
2025**

## TUGAS AKHIR

### ANALISIS MODEL MOBILENETV2 UNTUK KLASIFIKASI TANAMAN BERPOLEN

Dipersiapkan dan disusun Oleh  
**Muhammad Haris M.K.A**  
**NIM: 21410100029**

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahasan  
Pada: 12, Agustus 2025

#### Susunan Dewan Pembahasan

##### Pembimbing

- I. Vivine Nurcahyawati, M.Kom.  
NIDN. 0723018101
- II. Sri Hariani Eko Wulandari, S.Kom., M.MT.  
NIDN. 0726017801

  
Digitally signed  
by Vivine  
Nurcahyawati  
Date: 2025.09.01  
10:42:45 +07'00'  
  
  
\_\_\_\_\_  
  
  
Digitally signed  
by Slamet A.  
Date:  
2025.09.03  
15:24:01 +07'00'

##### Pembahasan

- I. Slamet M.T  
NIDN. 0701127503

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar sarjana



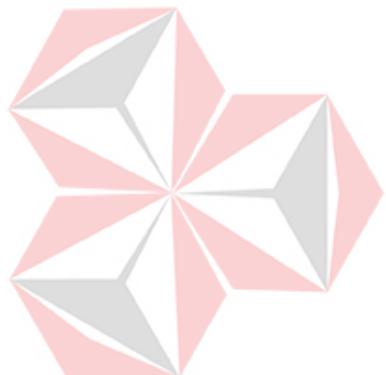
Digitally signed  
by Tan Amelia  
Date: 2025.09.05  
00:12:13 +07'00'

Tan Amelia, S.Kom., M.MT.

NIDN. 0728017602

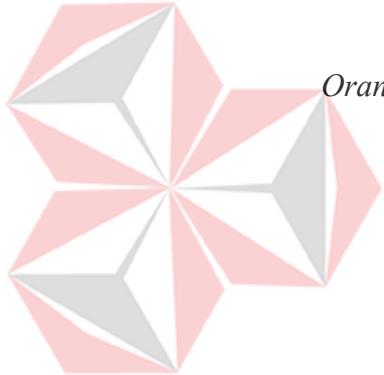
Dekan Fakultas Teknologi dan Informatika  
UNIVERSITAS DINAMIKA

*"When you know about something it stops being a nightmare. When you know how  
to fight it, it stops being so threatening"*



-Geralt of Rivia-

UNIVERSITAS  
**Dinamika**



*Tugas Akhir ini*

*Saya persembahkan kepada*

*Orang Tua, Keluarga Besar, Dosen Pembimbing, Dosen Wali,*

*dan Teman-teman*

UNIVERSITAS  
**Dinamika**

**PERNYATAAN**  
**PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama : **Muhammad Haris Maulana Koliq Abiansyah**  
NIM : **21410100029**  
Program Studi : **S1 Sistem Informasi**  
Fakultas : **Fakultas Teknologi dan Informatika**  
Jenis Karya : **Laporan Tugas Akhir**  
Judul Karya : **ANALISIS MODEL MOBILENETV2 UNTUK KLASIFIKASI TANAMAN BERPOLLEN**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 31 Juli 2025



M. Haris M.K.A  
NIM : 21410100029

## ABSTRAK

Terdapat beberapa tanaman di Surabaya yang menghasilkan serbuk sari (*polen*) yang dapat memicu alergi, alergi bunga sangat mebahayakan karena dapat menyebabkan asma kambuh, gangguan tidur, bersin-bersin, dan hidung tersumbat. Namun belum tersedia sistem yang memetakan sebarannya secara otomatis dan partisipatif. Sehingga masyarakat dengan alergi dapat mengantisipasi atas keberadaan tanaman tersebut. Penelitian ini bertujuan untuk menganalisis kinerja model *MobileNetV2* yang dikenal ringan dan efisien untuk perangkat dengan sumber daya terbatas, dalam mengklasifikasikan lima jenis tanaman berpolen. Model dikembangkan menggunakan pendekatan *transfer learning* dengan data sebanyak 250 citra tanaman dari dokumentasi lokal dan sumber terbuka, serta diperluas melalui augmentasi. Metode meliputi preprocessing, pelatihan model, dan evaluasi menggunakan *confusion matrix*. Hasil menunjukkan akurasi validasi mencapai 89%, dengan F1-score tertinggi 1.00 pada kelas *Hippobroma longiflora*. Temuan ini menunjukkan bahwa *MobileNetV2* efisien untuk klasifikasi tanaman berpolen meski dengan data terbatas.

**Kata Kunci:** MobileNetV2, Klasifikasi Citra, Tanaman Berpolen



## KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "ANALISIS MODEL MOBILENETV2 UNTUK KLASIFIKASI TANAMAN BERPOLEN" dengan baik dan tepat waktu. Penulisan Tugas Akhir ini merupakan salah satu syarat yang harus dipenuhi untuk menyelesaikan Program Sarjana pada Program Studi S1 Sistem Informasi, Fakultas Teknologi dan Informatika, Universitas Dinamika.

Penulis menyadari bahwa penyusunan Tugas Akhir ini tidak akan terwujud tanpa adanya bimbingan, dukungan, dan doa dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang tulus kepada:

1. Orang tua dan keluarga besar tercinta yang senantiasa memberikan doa, dukungan moril, maupun materiil yang tak terhingga.
2. Ibu Tan Amelia, S.Kom., M.MT., selaku Dekan Fakultas Teknologi dan Informatika Universitas Dinamika.
3. Bu Endra Rahmawati, M.Kom., selaku kaprodi S1 Sistem Informasi Universitas Dinamika.
4. Ibu Vivine Nurcahyawati, M.Kom., dan Ibu Sri Hariani Eko Wulandari, S.Kom., M.MT., selaku dosen pembimbing yang telah memberikan bimbingan, waktu, saran, dan dukungan yang sangat berharga selama proses penggeraan Tugas Akhir ini.
5. Bapak Slamet M.T, selaku dosen pembahas yang telah memberikan masukan dan wawasan yang membangun untuk penyempurnaan laporan ini.
6. Seluruh dosen Program Studi S1 Sistem Informasi Universitas Dinamika yang telah memberikan bekal ilmu pengetahuan yang bermanfaat.
7. Teman-teman Fanbase botak serta semua pihak yang telah membantu dan memberikan semangat yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang. Semoga Tugas Akhir ini dapat memberikan manfaat

dan kontribusi bagi pengembangan ilmu pengetahuan dan menjadi referensi bagi penelitian selanjutnya.

Surabaya, 07 Juli 2025

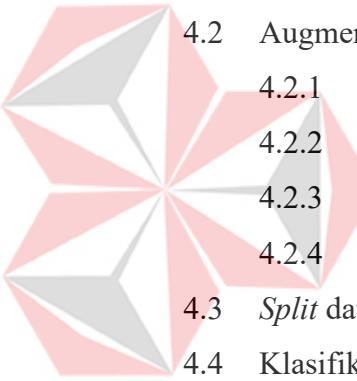
Penulis



## DAFTAR ISI

Halaman

ABSTRAK .....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI .....	x
DAFTAR GAMBAR .....	xii
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN .....	xiv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	2
1.5 Manfaat.....	2
BAB II LANDASAN TEORI .....	4
2.1 Penelitian Terdahulu .....	4
2.2 Bunga.....	6
2.3 MobileNetV2.....	6
2.3.1 Input Citra .....	8
2.3.2 <i>Initial Covolution</i> .....	8
2.3.3 <i>Inverted Residual Block</i> .....	9
2.3.4 <i>Last Convolution Layer (1x1)</i> .....	13
2.3.5 <i>Average Global Pool</i> .....	13
2.3.6 <i>1x1 Covolution Layer</i> .....	13
2.3.7 <i>Classification</i> .....	13
2.3.8 <i>Flatten Layer</i> .....	14
2.3.9 <i>Fully Connected Layer</i> .....	14
2.3.10 <i>Softmax Activation</i> .....	14
2.4 <i>Confusion Matrix</i> .....	15
2.4.1 Akurasi .....	16
2.4.2 Presisi .....	16
2.4.3 <i>Recall</i> .....	16



2.4.4	F1-Score .....	16
BAB III	METODOLOGI PENELITIAN.....	17
3.1	Akuisisi Citra.....	18
3.2	Augmentasi Citra .....	18
3.3	<i>Resize</i> Citra .....	19
3.4	<i>Split</i> data .....	19
3.5	Klasifikasi .....	19
3.5.1	Arsitektur MobileNetV2 .....	20
3.5.2	Proses Pelatihan dan Kompilasi Model .....	21
3.5.3	Output Model .....	21
3.6	Evaluasi.....	21
BAB IV	HASIL DAN PEMBAHASAN.....	23
4.1	Akuisisi Citra.....	23
4.2	Augmentasi Citra .....	23
4.2.1	<i>Rescale</i> .....	24
4.2.2	Rotasi .....	24
4.2.3	<i>Zoom</i> .....	25
4.2.4	<i>Flipping horizontal</i> .....	26
4.3	<i>Split</i> data .....	27
4.4	Klasifikasi .....	28
4.5	<i>Resize</i> Citra .....	31
4.6	Evaluasi.....	31
BAB V	PENUTUP.....	36
5.1	Kesimpulan.....	36
5.2	Saran .....	36
DAFTAR	PUSTAKA .....	38
LAMPIRAN	.....	40

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Arsitektur MobileNetV2.....	7
Gambar 2.2 <i>Inverted Residual Block</i> .....	10
Gambar 3.1 Tahap Penelitian .....	17
Gambar 3.2 Augmentasi Citra.....	19
Gambar 4.1 Kode Augmentasi Data.....	23
Gambar 4.2 <i>Rescale</i> Citra .....	24
Gambar 4.3 Rotasi Citra.....	25
Gambar 4.4 <i>Zoom</i> Citra.....	26
Gambar 4.5 <i>Mirror</i> Citra.....	26
Gambar 4.6 Gambar Hasil Augmentasi .....	27
Gambar 4.7 Konfigurasi <i>Split Dataset</i> .....	28
Gambar 4.8 Inisialisasi MobileNetV2 .....	28
Gambar 4.9 Pembekuan Parameter Model.....	29
Gambar 4.10 Perakitan Model Fungsional .....	29
Gambar 4.11 Kompilasi Model .....	29
Gambar 4.12 <i>Fitting</i> Model.....	30
Gambar 4.13 Grafik Akurasi .....	30
Gambar 4.14 Grafik <i>Loss</i> .....	31
Gambar 4.15 Konfigurasi <i>Resize</i> dan pemuatan citra .....	31
Gambar 4.16 Visualisasi <i>Confusion Matrix</i> .....	32
Gambar 4.17 <i>Classification Report</i> .....	35



## **DAFTAR TABEL**

	Halaman
Tabel 2.1 Penelitian Terdahulu .....	4
Tabel 2.2 Model <i>Summary</i> MobileNetV2 .....	7
Tabel 4.1 Jumlah Citra Dataset .....	23



## **DAFTAR LAMPIRAN**

	Halaman
Lampiran 1. Contoh <i>Dataset Gambar</i> .....	40
Lampiran 2. Plagiasi.....	43
Lampiran 3. Surat Bimbingan.....	44
Lampiran 4. Biodata.....	45



## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Kota Surabaya sebagai salah satu kota metropolitan di Indonesia memiliki banyak ruang terbuka hijau, taman kota, trotoar, dan kawasan permukiman yang dipenuhi berbagai jenis tanaman hias dan peneduh. Tanaman-tanaman tersebut memberikan manfaat ekologis dan estetika yang signifikan. Namun demikian, sebagian tanaman diketahui sebagai penghasil serbuk sari (polen) yang dapat memicu reaksi alergi, terutama bagi individu dengan sistem imun yang sensitif. Gejala seperti bersin, mata berair, gatal, hingga sesak napas sering kali muncul akibat paparan polen (Agung et al., 2025).

Di Indonesia, prevalensi rinitis alergi dilaporkan mencapai 53%, dengan serbuk sari sebagai salah satu pemicu utama (Prihatini, 2022). Mengingat dampaknya terhadap kesehatan masyarakat, identifikasi dini terhadap jenis-jenis tanaman yang berpotensi menghasilkan polen tinggi menjadi hal yang penting, khususnya di daerah perkotaan seperti Surabaya yang memiliki keragaman vegetasi cukup tinggi. Beberapa tanaman umum yang ditemukan di ruang terbuka Surabaya dan diketahui sebagai penghasil polen antara lain: *Delonix regia*, *Euphorbia milii*, *Hippobroma longiflora*, *Lagerstroemia indica*, dan *Wrightia antidysenterica*.

Dalam upaya mendukung proses identifikasi tanaman secara otomatis, teknologi deep learning menjadi salah satu pendekatan yang relevan dan efektif, khususnya dalam klasifikasi citra tumbuhan. Salah satu arsitektur deep learning yang banyak digunakan untuk klasifikasi gambar adalah MobileNetV2. Model ini dirancang untuk efisiensi tinggi dengan ukuran parameter yang kecil dan kecepatan inferensi yang cepat, sehingga cocok untuk digunakan pada perangkat dengan sumber daya terbatas seperti smartphone (Khasanah, 2021).

Penelitian ini berfokus pada analisis kinerja model MobileNetV2 dalam mengklasifikasikan gambar tanaman penghasil polen. Pemilihan model ini didasarkan pada keunggulannya dalam menangani permasalahan klasifikasi dengan efisiensi komputasi yang optimal tanpa mengorbankan akurasi secara signifikan. Dengan menganalisis performa model pada dataset tanaman lokal yang umum

ditemukan di Surabaya, penelitian ini bertujuan untuk mengkaji sejauh mana MobileNetV2 dapat digunakan sebagai solusi otomatisasi dalam identifikasi visual tanaman berpolen tinggi.

## 1.2 Rumusan Masalah

Bagaimana melakukan analisis model MobileNetV2 untuk mengidentifikasi tanaman lokal berpolen tinggi di Surabaya?.

## 1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, diterapkan beberapa batasan masalah sebagai berikut:

1. Penelitian ini menggunakan model MobileNetV2.
2. Objek penelitian hanya berfokus pada lima kelas tanaman lokal di surabaya yaitu *Delonix regia*, *Euphorbia milii*, *Hippobroma longiflora*, *Lagerstroemia indica*, dan *Wrightia antyidysenterica*.
3. Data yang digunakan berasal dari dokumentasi lokal dan *Dataset opensource*.
4. Pengukuran kinerja menggunakan confusion metriks yaitu akurasi, *precision*, *recall*, dan F1-score

## 1.4 Tujuan

Penelitian ini bertujuan untuk mengintegrasikan model *deep learning* MobileNetV2. Model akan digunakan untuk mengklasifikasikan lima jenis tanaman lokal di Surabaya.

## 1.5 Manfaat

Adapun dari Tugas Akhir ini dapat diperoleh manfaat sebagai berikut:

1. Mendukung pengembangan keilmuan di bidang Sistem Informasi, khususnya dalam teknologi *deep learning*.
2. Penelitian ini menambah literatur mengenai penerapan arsitektur MobileNetV2 dalam klasifikasi citra, khususnya untuk objek tanaman lokal berpolen tinggi. Hal ini memberikan kontribusi terhadap pengembangan studi kecerdasan buatan dalam bidang klasifikasi visual berbasis CNN ringan.

3. Menjadi referensi bagi penelitian selanjutnya dalam topik klasifikasi citra tanaman lokal dan penerapan model pretrained seperti MobileNetV2 dengan pendekatan *low-code*.
4. Penelitian ini mengkaji performa MobileNetV2 dalam situasi nyata dengan jumlah data terbatas dan variasi citra yang tinggi. Analisis ini penting untuk memahami batasan dan kekuatan model ketika diimplementasikan dalam skenario dunia nyata yang tidak selalu ideal.



## BAB II

### LANDASAN TEORI

#### 2.1 Penelitian Terdahulu

Beberapa penelitian terdahulu yang relevan menjadi acuan dalam studi ini. Tinjauan ini membantu memahami metode, model, dan hasil yang telah ada untuk memposisikan kebaruan penelitian ini. Rangkuman penelitian tersebut disajikan dalam Tabel 2.1 berikut:

Tabel 2.1 Penelitian Terdahulu

Penulis	Judul	Hasil	Perbedaan
(Wang et al., 2022)	Dynamic ensemble selection of Convolutional neural networkss and its application in flower classification.	Penelitian ini mengembangkan sebuah metode untuk seleksi ansambel dinamis pada model CNN untuk meningkatkan akurasi klasifikasi bunga. Secara spesifik, tiga belas model MobileNet yang berbeda dilatih menggunakan <i>Dataset</i> bunga. Dari kumpulan model tersebut, sebuah sub-set pengklasifikasi dipilih secara dinamis untuk mengidentifikasi setiap citra bunga. Metode yang diusulkan ini berhasil mencapai akurasi sebesar 95.50% dalam mengklasifikasikan lima spesies bunga yang berbeda.	Perbedaannya terletak pada pendekatan yang digunakan; penelitian Wang et al. menggunakan pendekatan menggabungkan beberapa model, sedangkan penelitian ini berfokus pada analisis satu arsitektur spesifik, yaitu MobileNetV2.
(Rajkomar & Pudaruth, 2023)	A Mobile App for the Identification of Flowers Using Deep learning.	Penelitian ini berhasil mengembangkan sebuah aplikasi seluler inovatif bernama "FloralCam" yang bertujuan untuk mengidentifikasi spesies bunga yang umum ditemukan di Mauritius. Mereka membangun <i>Dataset</i> kustom bernama "FlowerNet" yang terdiri dari 11.660 citra untuk 106 spesies bunga. Dengan menggunakan transfer learning, beberapa model CNN di-fine-tune, dan kinerja terbaik dicapai oleh model MobileNetV2 dengan akurasi 99.74%. Model terbaik ini kemudian diubah ke format TensorFlow Lite dan diintegrasikan ke dalam aplikasi seluler yang dibangun menggunakan Flutter.	Penelitian Rajkomar & Pudaruth menggunakan 106 spesies bunga di Mauritius jumlah kelasnya juga berbeda, yaitu 106 spesies bunga di Mauritius dibandingkan dengan 5 spesies bunga berpolen di Surabaya.

<b>Penulis</b>	<b>Judul</b>	<b>Hasil</b>	<b>Perbedaan</b>
<b>(Gunawan &amp; Setiawan, 2022)</b>	<i>Convolutional neural networks dalam Analisis Citra Medis</i>	Dokumen ini merupakan sebuah review paper yang merangkum penggunaan <i>deep learning</i> , khususnya metode <i>Convolutional neural networks</i> (CNN), dalam domain analisis citra medis. Dijelaskan bahwa para peneliti telah mendapatkan hasil yang memuaskan dalam berbagai tugas seperti klasifikasi, deteksi, dan segmentasi pada beragam jenis citra medis, termasuk Magnetic Resonance Imaging (MRI), Computer Tomography (CT), dan X-Rays.	Perbedaan yang paling mendasar adalah domain aplikasi. Penelitian Gunawan & Setiawan berfokus sepenuhnya pada ranah analisis citra medis , sedangkan penelitian yang diajukan berada dalam ranah klasifikasi bunga.
<b>(Khasanah, 2021)</b>	Implementasi Arsitektur MovenetV2 Untuk Klasifikasi Citra Beras Impor	Penelitian ini berhasil mengimplementasikan arsitektur MobileNetV2 untuk mengklasifikasikan lima jenis beras impor: Basmati, Bulgur, Japonica, Jasmine, dan Kimbapssal. <i>Dataset</i> awal sebanyak 180 citra diperbanyak melalui augmentasi menjadi 1.980 citra. Model yang dikembangkan berhasil mencapai akurasi sebesar 96,4% dan nilai loss 13,6%. Hasil akhir dari penelitian ini kemudian diimplementasikan dalam bentuk website.	Perbedaan utama terletak pada objek penelitian. Penelitian Khasanah mengklasifikasikan jenis beras impor , sementara penelitian yang diajukan mengklasifikasikan tanaman lokal berpolen tinggi..
<b>(Simanjutak et al., 2025)</b>	Menggunakan algoritma <i>Convolutional neural networks</i> (CNN) di desa bintang kecamatan sidikalang	Penelitian ini berhasil mengembangkan sebuah sistem untuk mengidentifikasi dan mengklasifikasikan enam kelas penyakit pada tanaman cabai rawit di Desa Bintang, Kecamatan Sidikalang. Metode yang digunakan adalah CNN dengan arsitektur LeNet-5. Kinerja model yang dihasilkan menunjukkan akurasi 86%, presisi 87%, dan recall 86%	Arsitektur yang digunakan berbeda; penelitian ini memakai LeNet-5, sedangkan penelitian yang diajukan menggunakan MobileNetV2 yang lebih modern dan efisien. Tujuannya juga berbeda; penelitian ini berfokus pada identifikasi penyakit tanaman , sedangkan penelitian yang diajukan bertujuan untuk analisis model MobileNetV2.

## 2.2 Bunga

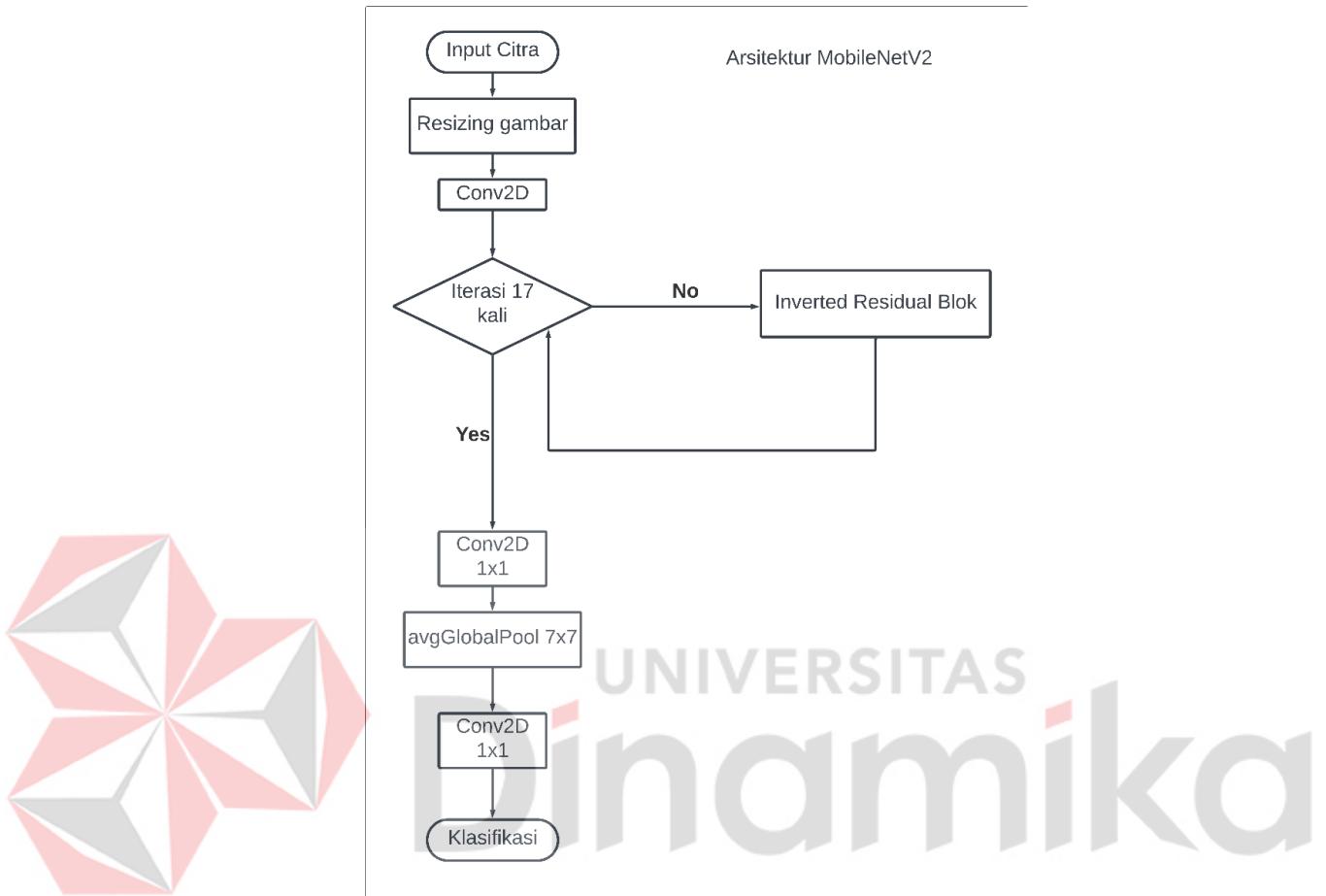
Bunga merupakan organ reproduksi pada tumbuhan yang memiliki variasi morfologi, seperti bentuk, warna, dan jenis yang sangat beragam. Keanekaragaman ini menjadikan bunga sebagai objek yang menarik dalam penelitian berbasis citra digital. Berdasarkan karakteristik reproduksinya, bunga dapat dikelompokkan menjadi dua jenis utama, yaitu bunga berpolen dan bunga tidak berpolen. Bunga berpolen, seperti *Delonix regia*, *Euphorbia millii*, *Hippobroma longiflora*, *Lagerstroemia indica*, *Wrightia antidysenterica*, memiliki butiran polen yang merupakan fitur mikro unik dan dapat terlihat pada citra digital, menjadikannya penting untuk dianalisis (Sukiatmodjo, 2024). Bunga sering dijadikan sebagai data uji dalam bidang pengolahan citra untuk pengenalan dan klasifikasi menggunakan metode kecerdasan buatan, seperti *Convolutional Neural Networks* (CNN) (Intyanto, 2021) dan arsitektur ringan seperti MobileNetV2.

## 2.3 MobileNetV2

MobileNetV2 merupakan arsitektur Convolutional Neural Network (CNN) yang dikembangkan untuk mengatasi keterbatasan perangkat dengan sumber daya komputasi rendah, seperti smartphone, perangkat IoT, dan aplikasi berbasis web. Arsitektur ini memanfaatkan *depthwise separable convolution* sebagai teknik utama untuk mengurangi jumlah parameter dan kompleksitas perhitungan, sehingga proses pelatihan maupun inferensi dapat berjalan lebih cepat dan efisien. Dengan cara ini, MobileNetV2 mampu menjaga keseimbangan antara kinerja komputasi dan tingkat akurasi yang tetap kompetitif dibandingkan arsitektur CNN yang lebih kompleks.

Selain itu, MobileNetV2 memperkenalkan konsep *Inverted Residual Blocks* dan *linear bottlenecks* yang berfungsi untuk mempertahankan informasi penting dalam representasi fitur sambil tetap menjaga efisiensi pemodelan. Kombinasi dari teknik tersebut menjadikan MobileNetV2 memiliki ukuran model yang ringan, penggunaan memori yang rendah, serta waktu komputasi yang singkat. Karakteristik ini membuatnya sangat sesuai diterapkan pada berbagai tugas klasifikasi citra, terutama pada aplikasi nyata yang menuntut pemrosesan data

secara cepat, efisien, dan dapat dijalankan pada perangkat dengan kapasitas terbatas (Khasanah, 2021). Arsitektur MobileNetV2 ditunjukkan gambar 2.2 di bawah ini:



Gambar 2.1 Arsitektur MobileNetV2  
(Sumber: Tragoudaras et al., 2022)

Arsitektur MobileNetV2 berisi *convolution layer* dengan *filter* 32 dan memiliki 17 *bottleneck* dengan rincian pada Tabel 2.2:

Tabel 2.2 Model *Summary* MobileNetV2

Input	Operator	<i>t</i>	<i>c</i>	<i>n</i>	<i>s</i>
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	<i>bottleneck</i>	1	16	1	1
$112^2 \times 16$	<i>bottleneck</i>	6	24	2	2
$56^2 \times 24$	<i>bottleneck</i>	6	32	3	2
$28^2 \times 32$	<i>bottleneck</i>	6	64	4	2
$14^2 \times 64$	<i>bottleneck</i>	6	96	3	1

Input	Operator	<i>t</i>	<i>c</i>	<i>n</i>	<i>s</i>
$14^2 \times 96$	<i>bottleneck</i>	6	160	3	2
$7^2 \times 160$	<i>bottleneck</i>	6	320	1	1
$7^2 \times 320$	<i>conv2d 1x1</i>	-	1280	1	1
$7^2 \times 1280$	<i>Avgpool 7x7</i>	-	-	1	-
$1 \times 1 \times 1280$	<i>conv2d 1x1</i>	-	<i>k</i>	-	-

(Sumber: Khasanah, 2021)

### 2.3.1 Input Citra

Input citra merupakan tahap awal dalam proses klasifikasi menggunakan arsitektur MobileNetV2. Model ini dirancang untuk menerima masukan berupa citra berukuran  $224 \times 224$  piksel dengan 3 kanal warna (RGB). Ukuran tersebut merupakan standar input yang digunakan oleh arsitektur MobileNetV2 yang telah dilatih sebelumnya (*pre-trained*) pada dataset *ImageNet*. Oleh karena itu, seluruh citra yang digunakan dalam proses pelatihan dan inferensi perlu disesuaikan ukurannya agar sesuai dengan spesifikasi ini.

Selain penyesuaian ukuran, input citra juga harus melalui proses *rescaling*, yaitu mengubah rentang nilai piksel dari 0–255 menjadi 0–1. Hal ini dilakukan dengan membagi nilai piksel dengan 255 agar data berada dalam skala yang sesuai untuk pelatihan model berbasis jaringan saraf, serta membantu dalam konvergensi yang lebih stabil dan cepat.

Proses praproses input citra secara umum meliputi:

1. *Resize*: Mengubah ukuran gambar menjadi  $224 \times 224$  piksel.
2. *Rescale*: Mengubah skala nilai piksel dari [0–255] menjadi [0–1].
3. Augmentasi (opsional): Menambahkan variasi seperti rotasi, *flipping*, dan *zoom* untuk meningkatkan generalisasi model.

### 2.3.2 Initial Covolution

Tahap awal dari MobileNetV2 adalah konvolusi standar dengan kernel berukuran  $3 \times 3$ , stride sebesar 2, dan padding untuk menjaga dimensi. Lapisan ini bertujuan untuk mengurangi resolusi spasial input sekaligus mengekstrak fitur awal dari citra.

Rumus perhitungan output feature map (Nelson, 2023):

$$W_{out} = \frac{(W_{in} - F + 2P)}{S} + 1 \quad (1)$$

$$H_{out} = \frac{(H_{in} - F + 2P)}{S} + 1 \quad (2)$$

Di mana:

- $W_{in}, H_{in}$  : Ukuran matriks input.
- $F$  : Ukuran matriks filter.
- $P$  : Jumlah *padding*.
- $S$  : Ukuran *stride*.

Rumus Operasi Konvolusi:

Operasi konvolusi untuk menghasilkan nilai pada titik  $(i,j)$  di *feature map* dihitung dengan (Zhang, 2020):



UNIVERSITAS  
Dinamika

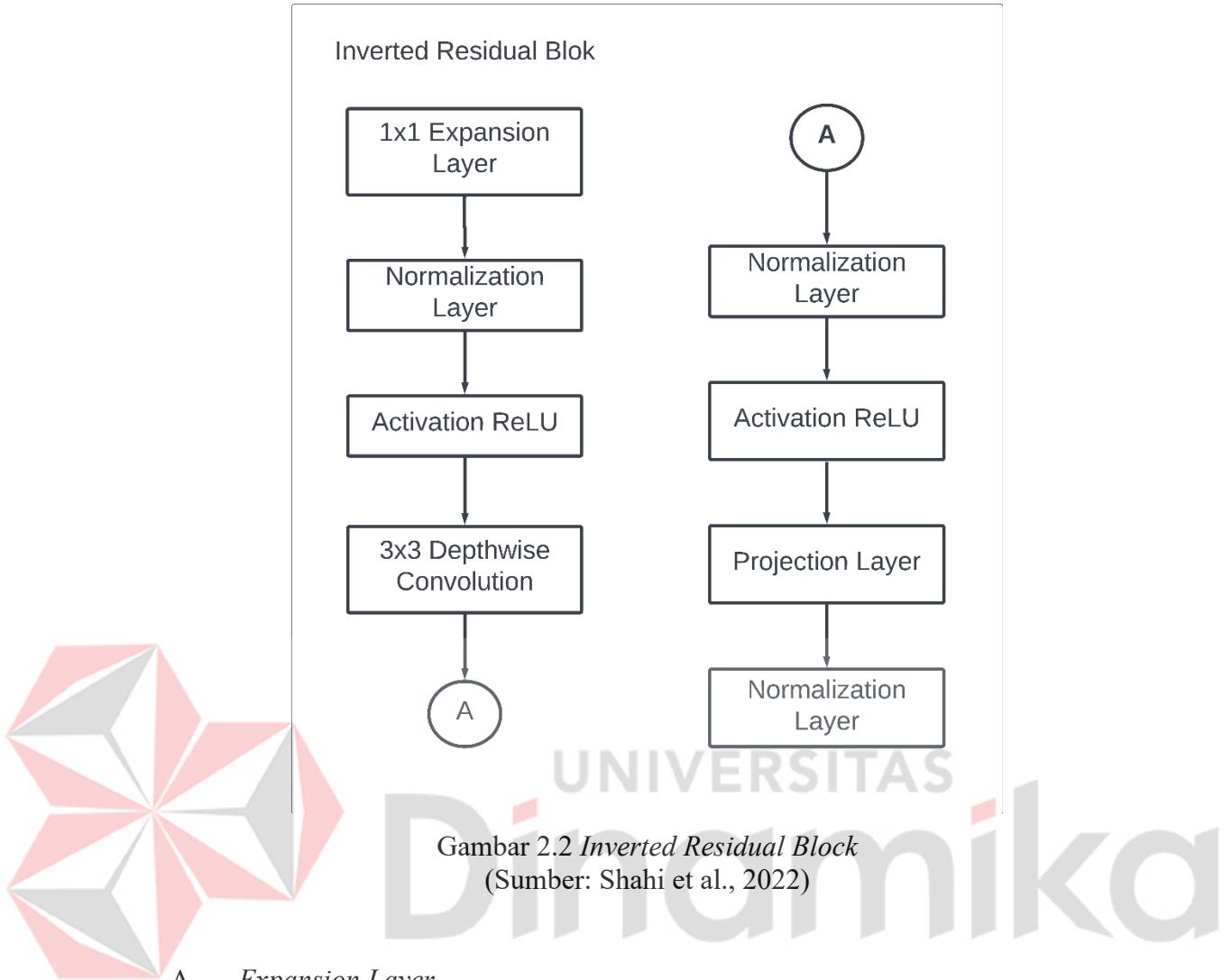
$$(f * g)(i, j) = \sum_a \sum_b f(a, b) \cdot g(i - a, j - b). \quad (3)$$

Di mana:

- $f$  : Matriks input (misalnya, gambar).
- $g$  : Matriks filter atau kernel.
- $m, n$  : Dimensi dari filter atau kernel
- $b$  : Nilai piksel pada posisi  $(i,j)$  di matriks output hasil konvolusi.

### 2.3.3 Inverted Residual Block

Blok ini merupakan tahap utama dalam arsitektur MobileNetV2. Setiap blok terdiri dari tiga tahapan konvolusi: *expansion*, *depthwise*, dan *projection*, dengan kemungkinan *residual connection*, untuk alur *Inverted Residual Block* dapat dilihat pada gambar 2.3:



#### A. Expansion Layer

Lapisan ini menerima masukan berupa peta fitur yang "sempit" (memiliki sedikit saluran) dan berfungsi untuk memperluas dimensinya secara signifikan. Proses ini diikuti oleh *Normalization Layer* untuk menstabilkan proses pelatihan dan *Activation Layer (ReLU)* untuk memperkenalkan non-linearitas. Rumus berikut merupakan formulasi matematis dari proses ekspansi fitur melalui konvolusi  $1 \times 1$  dan fungsi aktivasi ReLU, sebagaimana diterapkan pada blok Inverted Residual dalam arsitektur MobileNetV2 (Sandler et al., 2018).

#### B. Normalization Layer

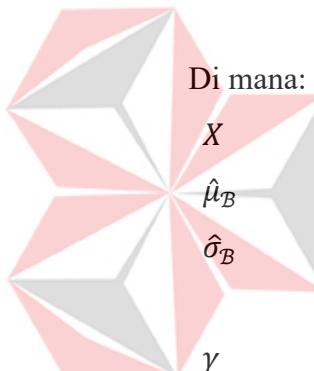
Setiap tahapan dalam *Inverted Residual Block*, seperti *expansion*, *depthwise convolution*, dan *projection*, dilengkapi dengan proses normalisasi, terutama menggunakan *Batch Normalization*. Lapisan ini bertujuan untuk menstabilkan dan

mempercepat proses pelatihan jaringan dengan menjaga distribusi output dari masing-masing lapisan tetap konsisten. Dengan normalisasi, model dapat belajar lebih cepat dan mengurangi risiko *overfitting*.

*Batch Normalization* bekerja dengan menghitung rata-rata dan standar deviasi dari output *layer* sebelumnya untuk kemudian menormalkan nilai-nilai tersebut ke dalam distribusi standar (mean = 0, standar deviasi = 1). Setelah proses normalisasi, dilakukan transformasi linear menggunakan parameter skala ( $\gamma$ ) dan pergeseran ( $\beta$ ), yang dapat dipelajari selama pelatihan.

Rumus Operasi *Batch Normalization* (Zhang, 2020):

$$BN(x) = \gamma \odot \frac{X - \hat{\mu}_B}{\hat{\sigma}_B} + \beta \quad (4)$$



- $X$  : Masukan ke lapisan Normalisasi *Batch*.
- $\hat{\mu}_B$  : Rata-rata (*mean*) dari *mini-batch* saat ini (B).
- $\hat{\sigma}_B$  : Standar deviasi (*standard deviation*) dari *mini-batch* saat ini (B).
- $\gamma$  : Faktor penskalaan (*scaling factor*) yang dapat dipelajari (diinisialisasi sebagai 1).
- $\beta$  : Faktor *offset* atau pergeseran (*offset factor*) yang dapat dipelajari (diinisialisasi sebagai 0).
- $\odot$  : Simbol ini menunjukkan perkalian elemen-demi-elemen (*element-wise multiplication*).
- $BN(x)$  : *Output* yang dinormalisasi, diskalakan, dan digeser dari lapisan Normalisasi *Batch*.

### C. Activation Layer (ReLU)

Setelah operasi konvolusi, setiap elemen pada *feature map* dilewatkan melalui sebuah fungsi aktivasi non-linear. Tujuannya adalah untuk memperkenalkan non-linearitas ke dalam model, yang memungkinkan jaringan untuk mempelajari pola yang lebih kompleks. Tanpa fungsi aktivasi, seluruh jaringan akan berperilaku

seperti satu filter linear saja. Fungsi aktivasi yang paling umum digunakan dalam CNN adalah *Rectified Linear Unit* (ReLU).

Perhitungan ReLU (Zhang, 2020):

ReLU mengubah semua nilai negatif menjadi nol dan mempertahankan nilai positif.

$$f(x) = \max(0, x) \quad (5)$$

Di mana  $x$  adalah nilai input dari sebuah elemen di *feature map*. Keunggulan ReLU adalah sederhana secara komputasi dan membantu mengatasi masalah *vanishing gradient*.

#### D. Depthwise Convolution

Lapisan ini melakukan konvolusi per kanal secara terpisah menggunakan kernel  $3 \times 3$  untuk melakukan pemfilteran spasial secara efisien pada peta fitur yang telah diperluas.

Rumus Biaya Komputasi (Dong et al., 2020): Biaya komputasi untuk tahap ini dihitung dengan:

$$Cost_{dw} = D_K \cdot D_K \cdot W \cdot D_W \cdot D_H \quad (6)$$

Di mana:

$Cost_{dw}$  : Total biaya komputasi untuk tahap *depthwise*

$D_K$  : Dimensi kernel ( $3 \times 3$ ).

$W$  : Lebar spasial dari peta fitur keluaran.

$D_W$  : Jumlah saluran masukan (input channels).

$D_H$  : Tinggi spasial dari peta fitur keluaran

#### E. *Projection Layer*

Lapisan ini merupakan sebuah konvolusi  $1 \times 1$  yang berfungsi memproyeksikan kembali peta fitur yang lebar ke representasi "sempit" dengan jumlah saluran yang lebih sedikit. Lapisan ini diikuti oleh *Normalization Layer* namun tanpa fungsi aktivasi (linier).

#### **2.3.4 Last Convolution Layer ( $1 \times 1$ )**

Setelah melewati rangkaian *Inverted Residual Block* terakhir, peta fitur diproses oleh sebuah *Convolution layer*  $1 \times 1$ . Lapisan ini berfungsi untuk memperluas dimensi fitur secara signifikan (dari 320 menjadi 1280 saluran) untuk menciptakan representasi fitur yang lebih kaya sebelum tahap agregasi.

#### **2.3.5 Average Global Pool**

Tahap ini bertujuan untuk mengurangi dimensi spasial dari peta fitur menjadi sebuah vektor fitur tunggal. Operasi ini mengambil nilai rata-rata dari seluruh piksel untuk setiap saluran, yang efektif untuk mengurangi jumlah parameter dan membuat model lebih robust terhadap variasi posisi objek pada gambar.

#### **2.3.6 $1 \times 1$ Covolution Layer**

Lapisan ini berfungsi sebagai pengklasifikasi akhir yang ekuivalen dengan *fully connected layer* pada arsitektur CNN konvensional. *Convolution layer*  $1 \times 1$  ini menerima vektor fitur berukuran  $1 \times 1 \times 1280$  dari tahap sebelumnya dan memetakannya ke sejumlah skor mentah (*logits*) yang sesuai dengan jumlah kelas target. Dalam konteks penelitian ini, lapisan ini akan menghasilkan 5 keluaran yang merepresentasikan lima kelas tanaman yang akan diidentifikasi.

#### **2.3.7 Classification**

Tahap klasifikasi pada *Convolutional neural networks* (CNN) merupakan bagian akhir dari proses pengolahan citra, setelah fitur-fitur penting berhasil diekstraksi pada tahap *feature learning*. Tujuan utama tahap ini adalah untuk menentukan kelas dari objek yang terdapat dalam citra masukan berdasarkan

representasi fitur yang telah diperoleh. Proses klasifikasi terdiri dari tiga komponen utama, yaitu *flatten layer*, *fully connected layer*, dan *softmax layer*.

### 2.3.8 *Flatten Layer*

Fungsinya sangat sederhana namun krusial: mengubah data dari format matriks 2D (atau 3D jika memperhitungkan kedalaman) menjadi format vektor 1D.

**Proses Flatten:** Lapisan ini mengambil *feature map* terakhir yang dihasilkan oleh blok konvolusi/pooling dan "meratakannya" menjadi satu kolom vektor.

**Contoh Perhitungan:** Jika *feature map* terakhir memiliki dimensi  $7 \times 7 \times 64$  (Tinggi x Lebar x Kedalaman), maka setelah melalui *flatten layer*, outputnya akan menjadi sebuah vektor dengan panjang  $7 \times 7 \times 64 = 3136$ . Vektor inilah yang menjadi input untuk lapisan berikutnya.

### 2.3.9 *Fully Connected Layer*

Lapisan ini bekerja seperti jaringan saraf tiruan konvensional. Setiap neuron di lapisan ini terhubung ke semua output dari *flatten layer*. Tujuannya adalah untuk menggunakan kombinasi dari fitur-fitur yang telah dipelajari untuk membuat keputusan klasifikasi.

### 2.3.10 *Softmax Activation*

Ini adalah *fully connected layer* terakhir yang berfungsi untuk menghasilkan output akhir dari jaringan. Untuk tugas klasifikasi multi-kelas, fungsi aktivasi *Softmax* digunakan untuk mengubah skor mentah (*logits*) menjadi distribusi probabilitas.

Perhitungan *Softmax* (Belagatti Pavan, 2024): Probabilitas untuk setiap kelas  $i$  dari total  $C$  kelas dihitung sebagai:

$$P(y = i|Z) = \text{softmax}(Z_i) = \frac{e^{Z_i}}{\sum_{j=1}^C e^{Z_j}} \quad (7)$$

$Z_i$ : Skor output dari neuron untuk kelas ke- $i$

Hasilnya adalah vektor probabilitas di mana jumlah semua elemennya adalah 1. Kelas dengan probabilitas tertinggi dipilih sebagai prediksi akhir.

#### 2.4 *Confusion Matrix*

*Confusion matrix* merupakan sebuah tabel evaluasi yang digunakan untuk mengukur performa model klasifikasi. Matriks ini menunjukkan perbandingan antara label sebenarnya dan hasil prediksi model dalam bentuk empat komponen utama. Berdasarkan nilai-nilai tersebut, dapat dihitung metrik evaluasi lain seperti akurasi, presisi, *recall*, dan *F1-score* yang memberikan gambaran menyeluruh tentang kinerja model (Simanjutak et al., 2025).

*Confusion matrix* sangat berguna untuk mengetahui sejauh mana model mampu melakukan klasifikasi secara benar serta mengidentifikasi jenis kesalahan yang terjadi, baik pada kelas positif maupun negatif. Kinerja model dievaluasi berdasarkan empat kemungkinan hasil yang menjadi komponen utama matriks tersebut. Dua komponen pertama merepresentasikan hasil prediksi yang benar:

1. *True Positive* (TP), yaitu saat model dengan tepat memprediksi data yang sebenarnya positif sebagai kelas positif.
2. *True Negative* (TN), yaitu saat model dengan tepat memprediksi data yang sebenarnya negatif sebagai kelas negatif.

Dua komponen lainnya mengidentifikasi jenis-jenis kesalahan spesifik yang dilakukan oleh model:

1. *False Positive* (FP), yang terjadi ketika data yang sebenarnya negatif keliru diprediksi oleh model sebagai kelas positif. Kesalahan ini juga dikenal sebagai "Error Tipe I".
2. *False Negative* (FN), yang terjadi ketika data yang sebenarnya positif keliru diprediksi oleh model sebagai kelas negatif. Kesalahan ini juga dikenal sebagai "Error Tipe II".

Keempat komponen ini—TP, TN, FP, dan FN—kemudian menjadi dasar untuk menghitung metrik-metrik evaluasi kuantitatif yang memberikan gambaran utuh tentang performa model.

#### 2.4.1 Akurasi

Akurasi mengukur proporsi total prediksi yang benar (TP dan TN) dari keseluruhan data. Metrik ini memberikan gambaran umum tentang seberapa sering model melakukan prediksi dengan benar.

Rumus Perhitungan:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

#### 2.4.2 Presisi

Presisi mengukur tingkat ketepatan dari prediksi positif yang dibuat oleh model. Metrik ini menjawab pertanyaan: "Dari semua data yang diprediksi sebagai kelas positif, berapa persen yang benar-benar positif?"

Rumus Perhitungan:

$$Presisi = \frac{TP}{TP + FP} \quad (9)$$

#### 2.4.3 Recall

*Recall*, yang juga dikenal sebagai sensitivitas atau *True Positive Rate* (TPR), mengukur kemampuan model untuk menemukan kembali semua data yang sebenarnya positif. Metrik ini menjawab pertanyaan: "Dari semua data yang seharusnya positif, berapa persen yang berhasil teridentifikasi oleh model?"

Rumus Perhitungan:

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

#### 2.4.4 F1-Score

*F1-Score* adalah rata-rata harmonik (*harmonic mean*) dari Presisi dan *Recall*. Metrik ini sangat berguna ketika terdapat distribusi kelas yang tidak seimbang (*imbalanced Dataset*), karena ia memberikan satu angka tunggal yang menyeimbangkan kedua metrik tersebut. Nilai F1-Score yang tinggi menunjukkan bahwa model memiliki Presisi dan *Recall* yang sama-sama baik.

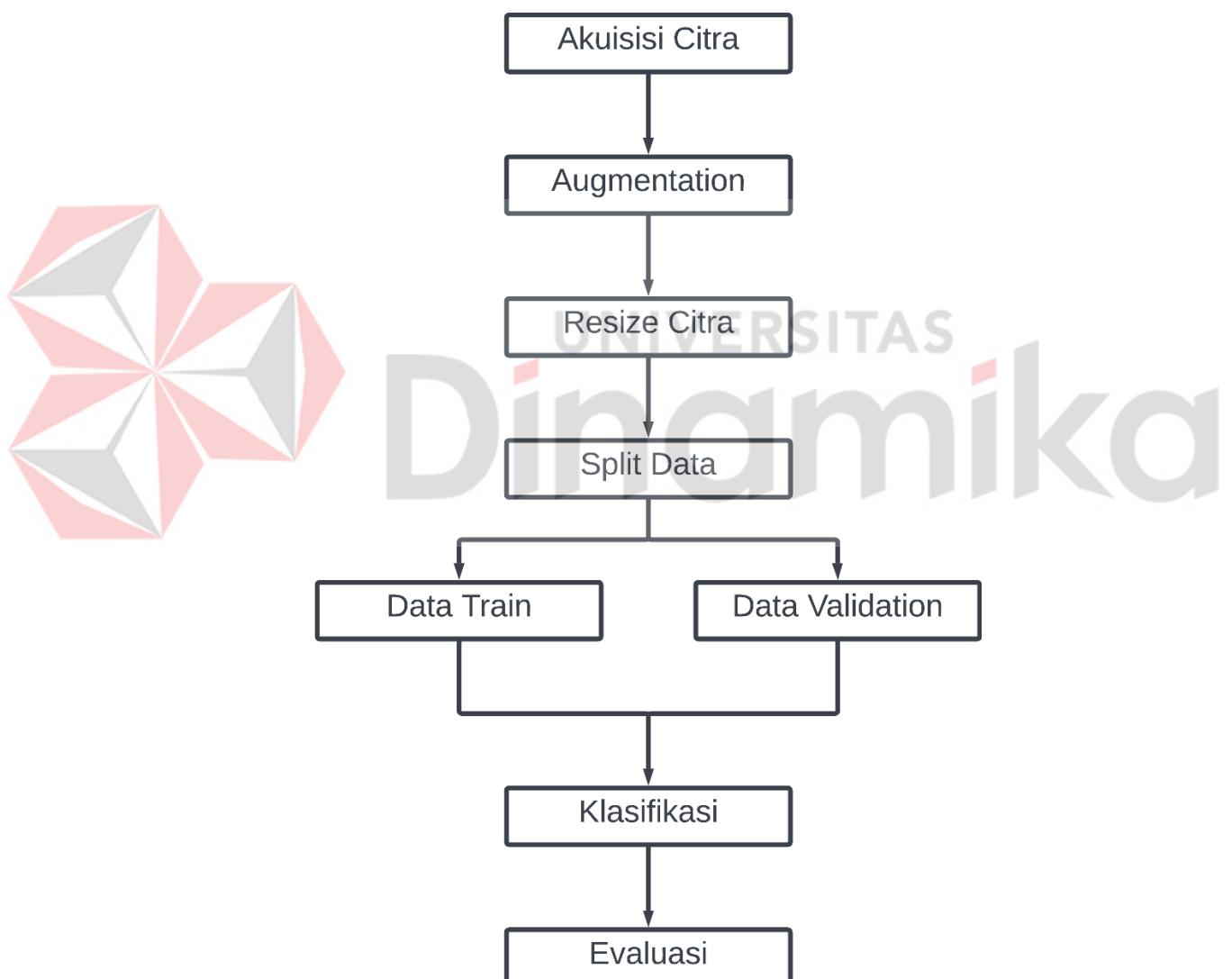
Rumus Perhitungan:

$$F1 - Score = \frac{Presisi \times Recall}{Presisi + Recall} \quad (11)$$

### BAB III

## METODOLOGI PENELITIAN

Penelitian ini bertujuan untuk mengembangkan dan menganalisis model klasifikasi citra tanaman berpolen tinggi menggunakan arsitektur MobileNetV2. Proses metodologi dibagi menjadi lima tahap utama, yaitu: Akuisisi Citra, Augmentasi Data, *Resize Citra*, *Split data*, Klasifikasi, dan Evaluasi, seperti pada gambar 3.1:



Gambar 3.1 Tahap Penelitian

### 3.1 Akuisisi Citra

Akuisisi citra dilakukan dengan mengumpulkan gambar lima jenis tanaman lokal. *Dataset* diperoleh dari dokumentasi lapangan dan sumber *opensource*.

Berikut untuk Sumber data dan Atribut:

Sumber Data Bunga:

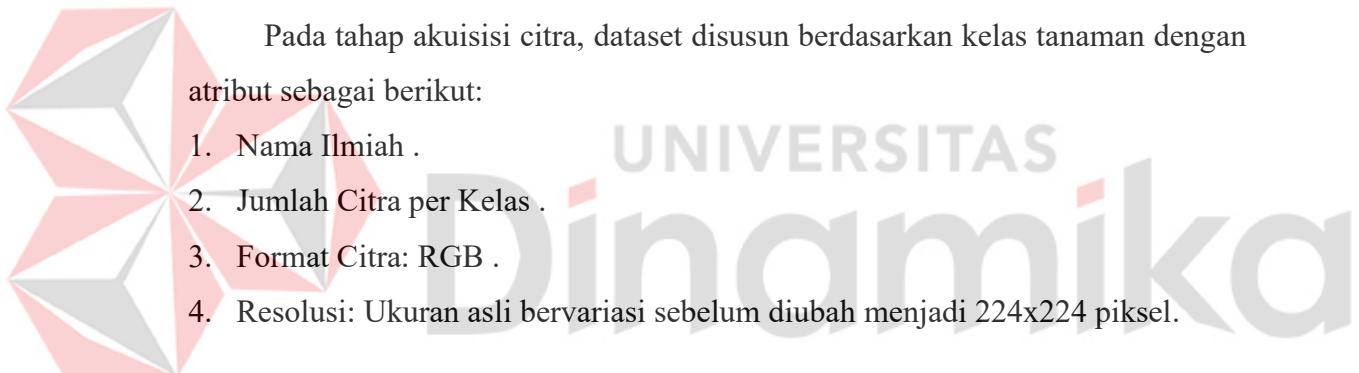
Data citra bunga diperoleh dari dua sumber utama data primer dan data sekunder, yaitu:

1. Data Primer: Dokumentasi Bunga berpolen yang diambil secara langsung di taman bunga bratang.
2. Data Sekunder: Bunga berpolen yang diambil dari internet ([Data Opensource](#)).

Atribut Data pada Akuisisi Citra:

Pada tahap akuisisi citra, dataset disusun berdasarkan kelas tanaman dengan atribut sebagai berikut:

1. Nama Ilmiah .
2. Jumlah Citra per Kelas .
3. Format Citra: RGB .
4. Resolusi: Ukuran asli bervariasi sebelum diubah menjadi 224x224 piksel.



### 3.2 Augmentasi Citra

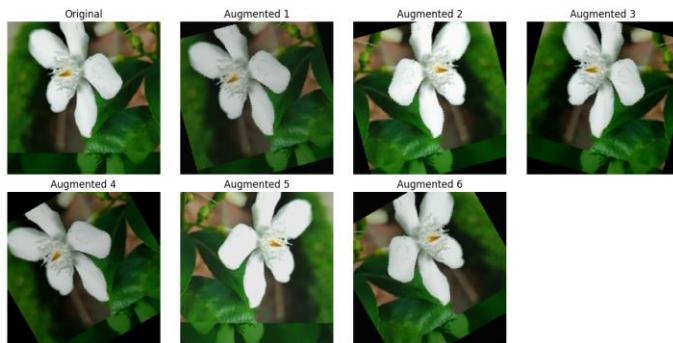
Augmentasi dilakukan untuk memperbanyak variasi data dan membantu model belajar dari kondisi gambar yang berbeda-beda. Augmentasi diterapkan hanya pada data training agar validasi tetap objektif.

Jenis augmentasi yang digunakan meliputi:

1. Normalisasi nilai piksel ke rentang [0,1] (dengan membagi piksel dengan 255).
2. Rotasi acak ( $\pm 20^\circ$  Derajat),
3. *Flip horizontal* (Citra dicerminkan secara horizontal),
4. *Zoom* acak hingga 20%,

Implementasi dilakukan menggunakan *ImageDataGenerator* dari pustaka *tensorflow.keras.preprocessing.image*.

Gambar augmentasi seperti gambar 3.3:



Gambar 3.2 Augmentasi Citra

### 3.3 *Resize* Citra

*Resize* citra dilakukan sebagai tahap awal *preprocessing* untuk menyeragamkan ukuran gambar. Seluruh gambar diubah ukurannya menjadi  $224 \times 224$  piksel, yang merupakan ukuran input standar untuk arsitektur MobileNetV2. *Resize* dilakukan sebelum augmentasi agar proses transformasi (rotasi, *zoom*, dan *flipping*) berlaku secara konsisten pada dimensi input yang sudah ditentukan.

### 3.4 *Split* data

*Dataset* dibagi menjadi dua bagian:

1. *Training set* sebanyak 80% dari total data,
2. *Validation set* sebanyak 20%.

Pembagian dilakukan secara otomatis melalui parameter *validation\_split* pada *ImageDataGenerator*. Proses ini juga secara otomatis membaca label dari nama folder dan mengubahnya menjadi one-hot encoding, yang dibutuhkan dalam klasifikasi multi-kelas.

### 3.5 Klasifikasi

Model klasifikasi yang digunakan dalam penelitian ini adalah MobileNetV2, sebuah arsitektur jaringan syaraf konvolusional (CNN) ringan dan efisien, yang

dikembangkan untuk digunakan pada perangkat dengan sumber daya terbatas seperti smartphone dan sistem berbasis web. MobileNetV2 diimplementasikan menggunakan pendekatan transfer learning, di mana bobot awal model diambil dari hasil pelatihan pada *Dataset* besar *ImageNet*, kemudian disesuaikan (*fine-tuning*) untuk klasifikasi lima kelas tanaman berpolen tinggi.

### 3.5.1 Arsitektur MobileNetV2

Arsitektur MobileNetV2 digunakan sebagai ekstraktor fitur (*base\_model*) untuk mengekstraksi pola visual dari citra tanaman. Untuk adaptasi terhadap tugas klasifikasi lima kelas tanaman lokal ini, model MobileNetV2 diinisialisasi tanpa lapisan klasifikasi bawaannya (*include\_top=False*), sehingga bagian atas model tersebut tidak disertakan.

Seluruh bobot pada *base\_model* dibekukan (*base\_model.trainable = False*) agar tidak ikut dilatih ulang. Pembekuan ini bertujuan untuk mempertahankan kemampuan ekstraksi fitur yang sudah dipelajari dari *ImageNet*, mengingat ukuran dataset penelitian yang relatif kecil. Setelah fitur diekstraksi oleh *base\_model*, beberapa lapisan kustom ditambahkan untuk mengadaptasi model pada klasifikasi lima kelas tanaman. Lapisan-lapisan tambahan ini meliputi:

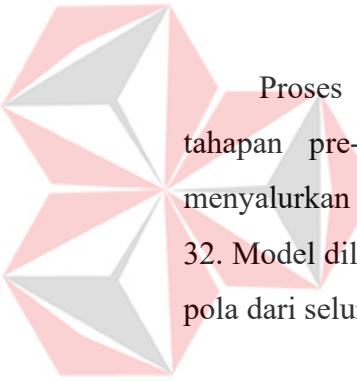
1. ***Global Average Pooling 2D (GlobalAveragePooling2D)***: Lapisan ini berfungsi untuk mereduksi dimensi spasial dari peta fitur yang dihasilkan oleh MobileNetV2 menjadi sebuah vektor fitur tunggal dengan mengambil rata-rata setiap kanal.
2. ***Dropout Layer (Dropout(0.5))***: Untuk mengurangi risiko *overfitting* selama proses pelatihan, *dropout layer* dengan tingkat 0.5 ditambahkan setelah *Global Average Pooling*.
3. ***Dense Layer (Dense(5, activation='softmax'))***: Sebagai lapisan keluaran (*output layer*) dan pengklasifikasi akhir, digunakan satu lapisan *Dense* dengan 5 neuron, yang merepresentasikan lima kelas tanaman target. Fungsi aktivasi.

softmax diterapkan pada lapisan ini untuk menghasilkan probabilitas prediksi untuk masing-masing kelas.

### 3.5.2 Proses Pelatihan dan Kompilasi Model

Model yang telah dirakit dengan penyesuaian arsitektur ini kemudian dikompilasi untuk mempersiapkan proses pelatihan. Konfigurasi kompilasi yang diterapkan adalah sebagai berikut:

1. **Fungsi Loss:** *categorical\_crossentropy* dipilih sebagai fungsi *loss* untuk mengukur kesalahan prediksi model, yang sesuai untuk tugas klasifikasi multi-kelas dengan label *one-hot encoding*.
2. **Optimizer:** Adam digunakan sebagai *optimizer* untuk memperbarui bobot model selama pelatihan, dipilih karena stabilitas dan efisiensinya dalam proses konvergensi.
3. **Metrik Evaluasi:** Metrik *accuracy* digunakan untuk memantau performa model selama pelatihan dan validasi.



Proses pelatihan model dilakukan menggunakan data yang telah melalui tahapan pre-processing dan augmentasi. *train\_generator* digunakan untuk menyalurkan data latih, dan *val\_generator* untuk data validasi, dengan *batch size* 32. Model dilatih selama 10 *epoch*, di mana pada setiap *epoch* model mempelajari pola dari seluruh data latih dan mengevaluasi kinerjanya pada data validasi.

### 3.5.3 Output Model

Setelah seluruh proses pelatihan yang telah dikonfigurasi pada 10 *epoch* diselesaikan, hasil akhirnya adalah model klasifikasi MobileNetV2 yang telah terlatih dan dioptimalkan berdasarkan dataset citra tanaman berpolen yang telah diproses. Model yang telah belajar mengekstraksi dan mengklasifikasikan pola visual ini kemudian disimpan sebagai berkas (.h5) untuk tujuan evaluasi lebih lanjut dan sebagai artefak yang siap digunakan dalam tahap inferensi pada citra tanaman baru.

## 3.6 Evaluasi

Evaluasi model dilakukan dengan menggunakan *confusion matrix* dan *classification report* untuk mengukur performa sistem klasifikasi citra tanaman.

*Confusion matrix* digunakan untuk membandingkan hasil prediksi model dengan label sebenarnya, sehingga dapat diketahui jumlah prediksi yang benar TP dan TN (*true positive* dan *true negative*) maupun yang salah FP dan FN(*false positive* dan *false negative*) untuk masing-masing kelas.

Melalui analisis ini, dapat terlihat pola kesalahan yang terjadi serta efektivitas model dalam membedakan antar kelas. Sementara itu, *classification report* memberikan metrik evaluasi lebih rinci seperti *precision*, *recall*, F1-score, dan akurasi. *Precision* menunjukkan tingkat ketepatan model dalam memprediksi setiap kelas, *recall* mengukur kemampuan model dalam menemukan seluruh anggota suatu kelas, F1-score menjadi pengukur keseimbangan antara *precision* dan *recall*, dan akurasi menyatakan proporsi total prediksi yang benar dari keseluruhan data. Kedua metode ini digunakan secara komplementer untuk memberikan gambaran menyeluruh terhadap kinerja model klasifikasi yang telah dikembangkan.



## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Akuisisi Citra

Akuisisi data dilakukan dengan mengumpulkan citra dari lima jenis tanaman lokal yang diketahui memiliki produksi serbuk sari tinggi. *Dataset* diperoleh dari dokumentasi lapangan dan sumber terbuka (*opensource*). Setiap jenis tanaman disimpan dalam direktori terpisah dengan struktur Tabel 4.1:

Tabel 4.1 Jumlah Citra Dataset

Nama Tanaman	Nama Ilmiah	Jumlah Citra
Kamboja Jepang	<i>Delonix Regia</i>	50
Pakis Giwang	<i>Euphorbia milii</i>	50
Bunga Kitolod	<i>Hippobroma longiflora</i>	50
Bungur Jepun	<i>Lagerstroemia indica</i>	50
Melati Anting	<i>Wrightia Antyidysenterica</i>	50
Total		250

*Dataset* bersifat RGB dengan ukuran asli bervariasi.

#### 4.2 Augmentasi Citra

Untuk memperluas keragaman data pelatihan dan meningkatkan generalisasi model, dilakukan augmentasi pada citra. Augmentasi ini hanya diterapkan pada data pelatihan (*training set*), sementara data validasi dibiarkan dalam kondisi asli untuk menjaga objektivitas evaluasi.

Konfigurasi augmentasi menggunakan *ImageDataGenerator* seperti pada gambar 4.1:

```
train_datagen = ImageDataGenerator(  
    rescale=1./255,  
    validation_split=0.2,  
    rotation_range=20,  
    zoom_range=0.2,  
    horizontal_flip=True  
)
```

Gambar 4.1 Kode Augmentasi Data

Jenis augmentasi yang digunakan meliputi:

1. Normalisasi nilai piksel ke rentang [0,1] (dengan membagi piksel dengan 255).
2. Rotasi acak ( $\pm 20^\circ$  Derajat),
3. *Flip horizontal* (Citra dicerminkan secara horizontal),
4. *Zoom* acak hingga 20%,

#### **4.2.1 Rescale**

Proses *rescale* dilakukan dengan tujuan untuk menormalkan nilai piksel pada setiap citra. Nilai piksel gambar digital umumnya berada dalam rentang 0 hingga 255. Model *deep learning* seperti MobileNetV2 bekerja lebih optimal ketika nilai input berada dalam rentang 0 hingga 1. Oleh karena itu, dilakukan proses pembagian setiap nilai piksel dengan angka 255, sehingga nilai piksel akan berada pada rentang 0.0 hingga 1.0.

Proses ini dilakukan melalui parameter `rescale=1./255` pada objek `ImageDataGenerator`. Secara visual, hasil *rescale* tidak menunjukkan perubahan yang dapat dilihat secara kasat mata, karena transformasi hanya terjadi pada skala nilai, bukan warna atau bentuk citra seperti pada gambar 4.2:



Gambar 4.2 *Rescale* Citra

#### **4.2.2 Rotasi**

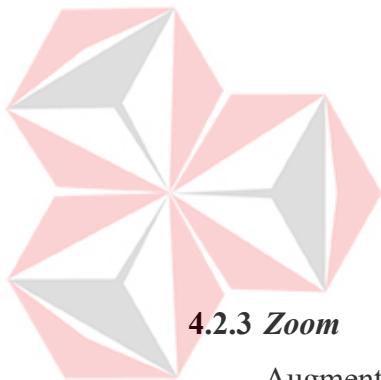
Augmentasi rotasi digunakan untuk memberikan variasi sudut pandang terhadap objek dalam citra. Dalam konteks klasifikasi tanaman, rotasi berguna untuk mengantisipasi kondisi saat citra tanaman diambil dari sudut yang berbeda,

baik miring ke kanan atau ke kiri. Model yang dilatih dengan citra yang telah mengalami rotasi akan memiliki ketahanan yang lebih baik terhadap orientasi objek.

Parameter yang digunakan adalah *rotation\_range*=20, yang berarti citra dapat diputar secara acak dalam rentang -20 hingga 20 derajat. Augmentasi ini membantu model untuk lebih adaptif terhadap variasi orientasi citra pada data baru yang belum pernah dilihat sebelumnya, untuk contoh dapat dilihat pada gambar 4.3:



Gambar 4.3 Rotasi Citra



#### 4.2.3 Zoom

Augmentasi *zoom* diterapkan untuk memberikan variasi dalam ukuran objek yang ada di dalam citra. Parameter yang digunakan adalah *zoom\_range*=0.2, yang berarti citra dapat diperbesar atau diperkecil secara acak hingga 20% dari ukuran aslinya. Dengan adanya *zoom*, model dapat mempelajari fitur-fitur tanaman dari berbagai tingkat kedekatan. Misalnya, citra yang diperbesar dapat memberikan detail lebih pada bagian bunga, sementara citra yang *dizoom-out* memberikan konteks terhadap keseluruhan struktur tanaman. Hal ini meningkatkan generalisasi model terhadap variasi ukuran dan skala objek, untuk contoh citra zoom dapat dilihat pada gambar 4.4:



Gambar 4.4 Zoom Citra

#### 4.2.4 *Flipping horizontal*

*Flip horizontal* atau pencerminan citra secara horizontal dilakukan untuk menambahkan variasi dalam arah tampilan objek. Dengan menggunakan parameter *horizontal\_flip=True*, citra tanaman dapat dicerminkan ke arah kiri atau kanan. Augmentasi ini berguna dalam kasus di mana bentuk dan arah objek tidak memiliki keterikatan arah tertentu. Dalam klasifikasi tanaman, pencerminan gambar tidak mengubah identitas objek, sehingga teknik ini aman untuk diterapkan tanpa mengganggu label kelas. *Flip horizontal* memungkinkan model mengenali objek dengan sudut pandang yang berlawanan secara simetris, untuk hasil dari *Flip horizontal* dapat dilihat pada gambar 4.5:



Gambar 4.5 Mirror Citra

Hasil augmentasi memperkaya variasi latar, sudut, dan bentuk tanaman dalam *Dataset* pelatihan, sehingga membantu model mengenali pola yang lebih umum, untuk hasil dari augmentasi citra dapat dilihat pada gambar 4.6:



Gambar 4.6 Gambar Hasil Augmentasi

Gambar di atas menampilkan hasil augmentasi citra yang diterapkan pada dataset tanaman lokal. Augmentasi ini dilakukan untuk menambah keragaman data pelatihan secara sintetik, sehingga model dapat lebih adaptif terhadap variasi bentuk, orientasi, dan pencahayaan dari objek citra. Variasi augmentasi seperti rotasi, pembesaran, dan *flipping horizontal* menciptakan versi alternatif dari citra asli tanpa mengubah labelnya. Hal ini penting untuk meningkatkan kemampuan generalisasi model dan mencegah *overfitting*, terutama pada dataset dengan jumlah citra terbatas. Dengan pendekatan ini, model MobileNetV2 dapat mengenali objek tanaman dengan lebih baik meskipun terdapat perbedaan posisi atau sudut pandang saat pengambilan gambar di dunia nyata.

### 4.3 *Split* data

*Dataset* dibagi secara otomatis menjadi dua subset:

1. *Training set* sebesar 80%
2. *Validation set* sebesar 20%

Pembagian ini menggunakan parameter *validation\_split*=0.2 dalam *ImageDataGenerator* seperti gambar 4.7. Sistem juga secara otomatis membaca

label dari struktur direktori dan menerapkannya sebagai *one-hot encoding*, yang diperlukan untuk klasifikasi multi-kelas.

```
train_datagen = ImageDataGenerator(  
    rescale=1./255,  
    validation_split=0.2, # 80% train, 20% validation  
    rotation_range=20,  
    zoom_range=0.2,  
    horizontal_flip=True  
)
```

Gambar 4.7 Konfigurasi *Split Dataset*

#### 4.4 Klasifikasi

Proses klasifikasi dalam penelitian ini dilakukan menggunakan arsitektur MobileNetV2 sebagai model dasar, yang telah dilatih sebelumnya pada dataset *ImageNet*. MobileNetV2 dipilih karena efisien dan ringan untuk tugas klasifikasi citra, terutama saat jumlah data relatif kecil seperti dalam penelitian ini. Model diinisialisasi tanpa bagian atas (*fully connected layer* bawaan), sehingga memungkinkan penyesuaian klasifikasi terhadap lima kelas tanaman lokal berpolen tinggi, seperti terlihat pada Gambar 4.8.

```
IMG_SHAPE = (224,224,3)  
  
base_model= MobileNetV2(input_shape=IMG_SHAPE,  
    include_top=False,  
    weights='imagenet')
```

Gambar 4.8 Inisialisasi MobileNetV2

Langkah pertama adalah memuat model MobileNetV2 dan mengatur *include\_top=False* agar tidak menyertakan layer klasifikasi bawaan. Selain itu, parameter trainable diset ke *False* agar seluruh bobot pada MobileNetV2 tidak dilatih ulang. Hal ini dilakukan agar model tetap memanfaatkan representasi fitur yang sudah dipelajari dari *ImageNet* dan hanya melatih bagian klasifikasi baru. Proses ini ditampilkan dalam Gambar 4.9.

```
base_model.trainable = False
```

Gambar 4.9 Pembekuan Parameter Model

Setelah model dasar dibekukan, ditambahkan beberapa *layer* untuk klasifikasi akhir. Data citra yang masuk akan diproses oleh MobileNetV2 untuk mengekstraksi fitur visual, kemudian hasilnya diringkas menggunakan *GlobalAveragePooling2D*. Agar model lebih tahan terhadap *overfitting*, ditambahkan *Dropout* sebesar 0.5. Terakhir, digunakan *Dense layer* dengan 5 neuron dan fungsi aktivasi *softmax* sebagai *output layer* untuk mengklasifikasikan ke lima kelas. Rangkaian arsitektur ini ditunjukkan dalam Gambar 4.10.

```
global_average_layer = GlobalAveragePooling2D()
inputs = tf.keras.Input(shape=IMG_SHAPE)
x = base_model(inputs, training=False)
x = global_average_layer(x)
x = tf.keras.layers.Dropout(0.5)(x)
outputs = prediction_layer(x)
model = Model(inputs, outputs)
model.summary()
```

Gambar 4.10 Perakitan Model Fungsional

Model yang telah dibentuk kemudian dikompilasi menggunakan fungsi *loss categorical\_crossentropy*, yang berguna untuk mengukur kesalahan pada klasifikasi multi-kelas, serta *optimizer Adam*, yang digunakan untuk mempercepat proses pembaruan bobot model selama pelatihan. Data yang telah melalui tahap augmentasi dan normalisasi sebelumnya disalurkan ke dalam model dalam bentuk *batch* citra berukuran  $224 \times 224$  piksel dengan format *array* empat dimensi. Kompilasi ini menyiapkan model agar siap dilatih menggunakan konfigurasi yang sesuai dengan kebutuhan klasifikasi. Proses ini ditampilkan pada Gambar 4.11.

```
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

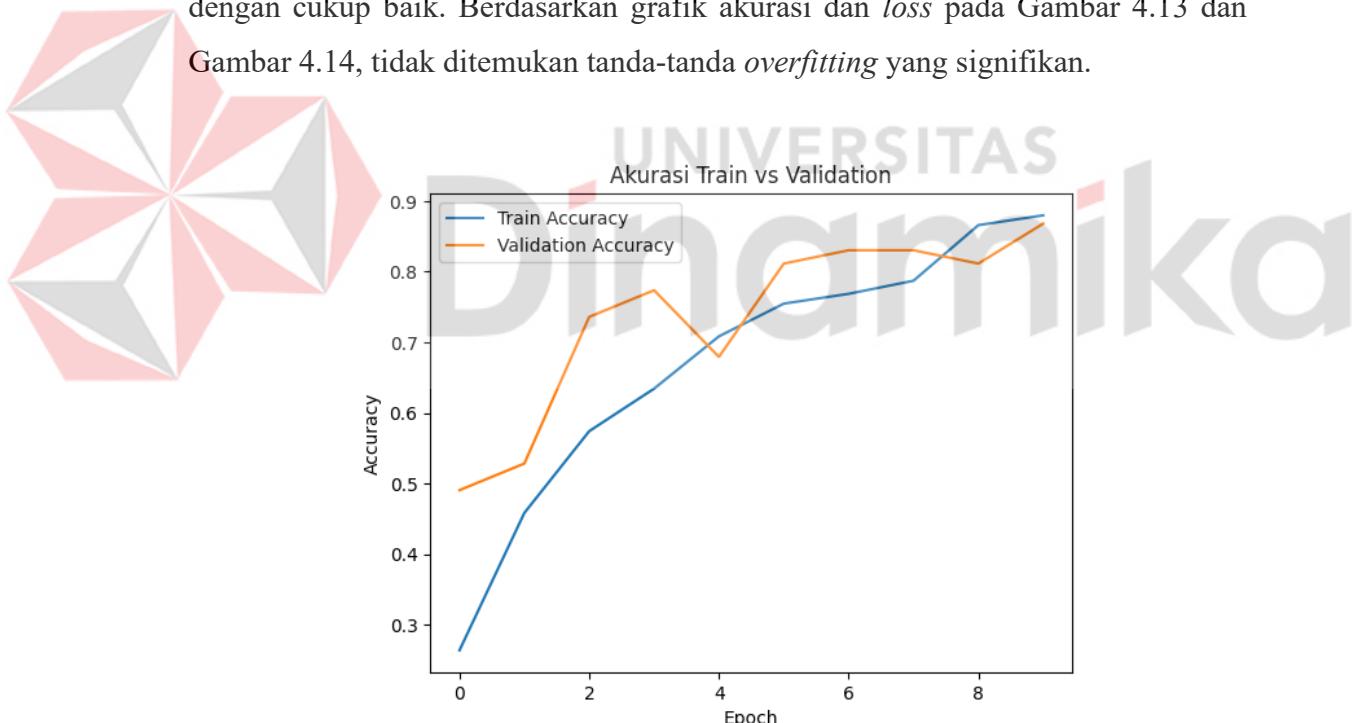
Gambar 4.11 Kompilasi Model

Model dilatih selama 10 *epoch* dengan metode `model.fit()`, menggunakan `train_generator` untuk data latih dan `val_generator` untuk data validasi. Proses pelatihan ditampilkan dalam Gambar 4.12.

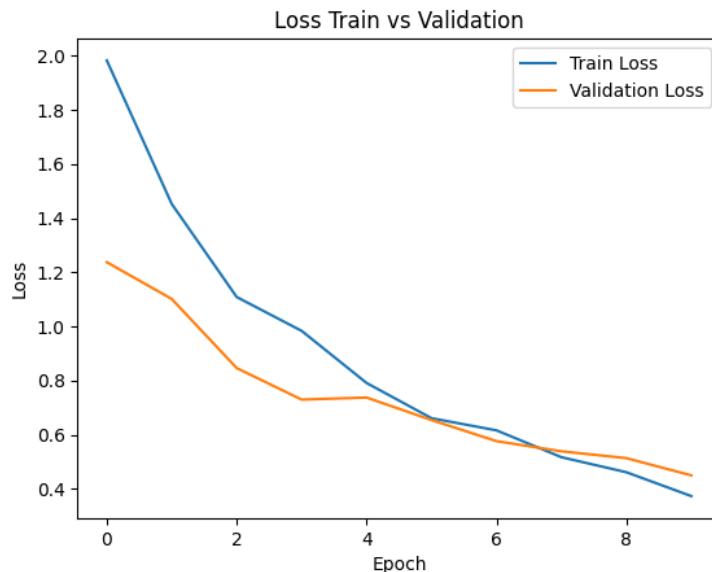
```
History = model.fit(  
    train_generator,  
    validation_data = val_generator,  
    epochs=10  
)
```

Gambar 4.12 *Fitting Model*

Hasil pelatihan menunjukkan bahwa akurasi pelatihan mencapai 81%, sedangkan akurasi validasi mencapai sekitar 86% pada *epoch* ke-10. Ini menunjukkan bahwa model mampu mempelajari pola visual dari citra tanaman dengan cukup baik. Berdasarkan grafik akurasi dan *loss* pada Gambar 4.13 dan Gambar 4.14, tidak ditemukan tanda-tanda *overfitting* yang signifikan.



Gambar 4.13 Grafik Akurasi



Gambar 4.14 Grafik Loss

#### 4.5 Resize Citra

Seluruh citra hasil akuisisi memiliki ukuran dan resolusi yang bervariasi. Agar dapat diproses oleh arsitektur MobileNetV2, semua citra di-*resize* menjadi ukuran  $224 \times 224$  piksel, sesuai dengan format input default MobileNetV2.

*Resize* dilakukan sebelum augmentasi, agar seluruh transformasi data (rotasi, *zoom*, *flipping*) dilakukan terhadap citra berukuran seragam. Proses resize ini secara otomatis diterapkan melalui *ImageDataGenerator* dengan parameter *target\_size* seperti pada gambar 4.15.

```
IMG_SHAPE = (224,224,3)

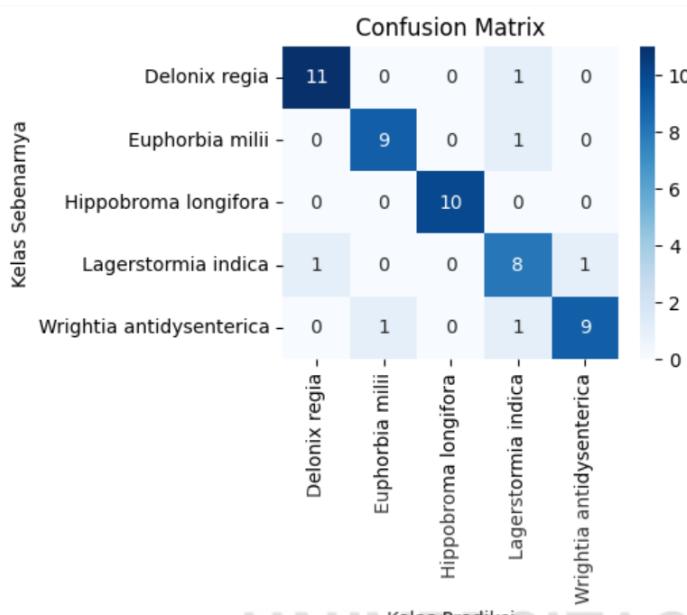
base_model= MobileNetV2(input_shape=IMG_SHAPE,
    include_top=False,
    weights='imagenet')
```

Gambar 4.15 Konfigurasi *Resize* dan pemuatan citra

#### 4.6 Evaluasi

Evaluasi model dilakukan untuk mengukur performa sistem klasifikasi citra tanaman menggunakan metrik yang dihitung dari *confusion matrix*. Matriks ini membandingkan hasil prediksi model dengan label yang sebenarnya, sehingga

dapat diketahui jumlah prediksi yang benar (*True Positive*) dan prediksi yang salah (*True Positive*), maupun yang salah (*False Positive* dan *False Negative*) untuk setiap kelas tanaman seperti gambar 4.16 visualisasi *confusion matrix*.



Gambar 4.16 Visualisasi *Confusion Matrix*

Perhitungan Metrik Evaluasi:

Berdasarkan data dari gambar *Confusion Matrix*, berikut adalah rincian perhitungan untuk setiap kelas:

Akurasi Keseluruhan Model:

Akurasi mengukur proporsi total prediksi yang benar (TP dan TN) dari keseluruhan data. Metrik ini memberikan gambaran umum tentang seberapa sering model melakukan prediksi dengan benar di seluruh kelas, dapat dilihat pada gambar 4.17.

Jumlah prediksi benar seluruh kelas (*True Positive + True Negative*): 47

Jumlah prediksi salah seluruh kelas (*False Positive + False Negative*): 6

Total sampel: 53

$$Akurasi = \frac{47}{47+6} = \frac{47}{53} \approx 89\%$$

Perhitungan Metrik Per Kelas:

Metrik *Precision*, *Recall*, dan F1-Score memberikan gambaran kinerja model secara lebih rinci untuk setiap kelas secara individual, yang menjelaskan mengapa nilainya bervariasi antar kelas.

### ***Delonix regia***

*True Positive* (TP): 11

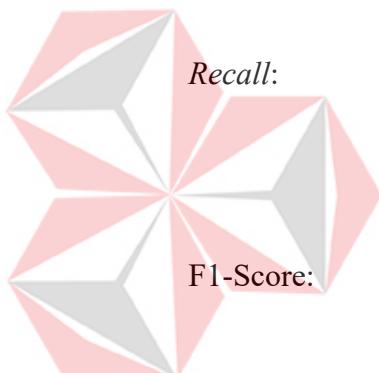
*True Negative* (TN): 0

*False Positive* (FP): 1

*False Negative* (FN): 1

*Precision*:

$$\text{Precision} = \frac{10}{10 + 1} = \frac{10}{11} \approx 92\%$$



UNIVERSITAS  
**Dinamika**

$$\text{Recall} = \frac{10}{10 + 1} \approx 92\%$$

$$F1 - Score = 2 \times \frac{92\% \times 92\%}{92\% + 92\%} \approx 92\%$$

### ***Euphorbia milii***

*True Positive* (TP): 9

*False Positive* (FP): 1

*False Negative* (FN): 1

*Precision*:

$$\text{Precision} = \frac{9}{9 + 1} = 90\%$$

*Recall*:

$$\text{Recall} = \frac{9}{9 + 1} = 90\%$$

F1-Score:

$$F1 - Score = 2 \times \frac{90\% \times 90\%}{90\% + 90\%} = 90\%$$

### *Hippobroma longiflora*

*True Positive* (TP): 10

*False Positive* (FP): 0

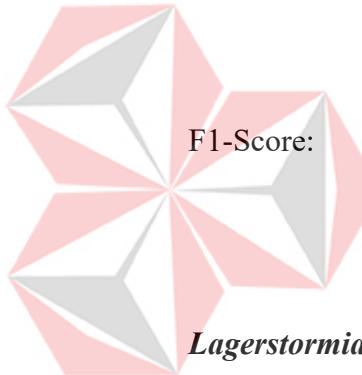
*False Negative* (FN): 0

*Precision*:

$$Precision = \frac{10}{10} = 100\%$$

*Recall*:

$$Recall = \frac{10}{10} = 100\%$$



*Lagerstromia indica*

$$F1 - Score = 2 \times \frac{100\% \times 100\%}{100\% + 100\%} = 100\%$$

*True Positive* (TP): 8

*False Positive* (FP): 3

*False Negative* (FN): 2

*Precision*:

$$Precision = \frac{8}{8 + 3} \approx 73\%$$

*Recall*:

$$Precision = \frac{8}{8 + 2} = 80\%$$

F1-Score:

$$F1 - Score = 2 \times \frac{73\% \times 80\%}{73\% + 80\%} \approx 76\%$$

*Wrightia antidysenterica*

*True Positive* (TP): 9

*False Positive* (FP): 1

*False Negative* (FN): 2

*Precision:*

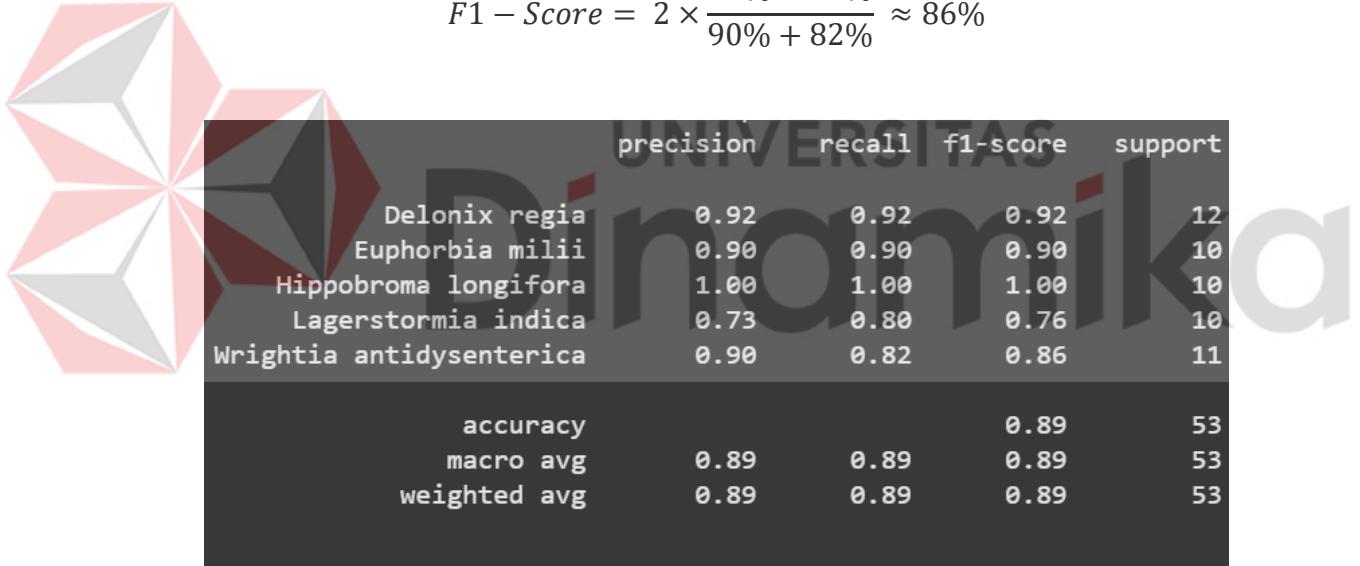
$$\text{Precision} = \frac{9}{9 + 1} = 90\%$$

*Recall:*

$$\text{Recall} = \frac{9}{9 + 2} \approx 82\%$$

F1-Score:

$$\text{F1 - Score} = 2 \times \frac{90\% \times 82\%}{90\% + 82\%} \approx 86\%$$



Gambar 4.17 *Classification Report*

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan hasil dan pembahasan pada bab sebelumnya, dapat disimpulkan bahwa:

1. Model MobileNetV2 berhasil mengembangkan untuk klasifikasi lima jenis tanaman lokal berpolen tinggi, dengan pendekatan *transfer learning* dan pelatihan pada *Dataset* terbatas. Arsitektur disesuaikan dengan menambahkan *Global Average Pooling*, *dropout*, dan *dense layer* sebagai klasifikasi akhir.
2. Model menunjukkan performa klasifikasi yang bervariasi antar kelas. Kelas *Hippobroma longiflora* diklasifikasikan dengan sangat baik ( $F1\text{-score} = 1.00$ ), namun kelas seperti *Euphorbia milli* memiliki recall yang rendah (0.70), menunjukkan bahwa model belum mampu membedakan seluruh kelas secara konsisten.
3. Akurasi validasi tertinggi tercapai pada 86%, sedangkan akurasi pelatihan hanya 81% pada data training *epoch* ke 10 dapat dilihat pada gambar 4.13. Selisih ini masih dapat diterima, namun menunjukkan bahwa generalisasi model belum sepenuhnya optimal.

Dengan demikian, dapat disimpulkan bahwa MobileNetV2 cocok digunakan untuk klasifikasi citra tanaman lokal berpolen tinggi pada kondisi data terbatas, namun hasil klasifikasi masih belum sepenuhnya merata di semua kelas dan memerlukan peningkatan pada tahap data dan pelatihan lanjutan.

#### 5.2 Saran

1. Penambahan jumlah data pelatihan sangat disarankan, khususnya pada kelas yang memiliki performa evaluasi rendah, seperti *Euphorbia milli*. Semakin banyak variasi citra akan meningkatkan kemampuan generalisasi model.
2. Eksperimen dengan *fine-tuning* beberapa *layer* awal MobileNetV2 dapat dilakukan agar fitur lokal yang lebih spesifik pada tanaman tropis Indonesia dapat lebih dikenali oleh model.

3. Penerapan validasi silang (*cross-validation*) dapat digunakan dalam penelitian lanjutan untuk mendapatkan hasil evaluasi yang lebih *robust* terhadap distribusi data.
4. Sistem dapat dikembangkan lebih lanjut dengan fitur deteksi objek (*object detection*), sehingga pengguna tidak hanya mengklasifikasikan tanaman, tetapi juga mengetahui posisi tanaman dalam gambar.
5. Penerapan model ke dalam aplikasi mobile perlu dipertimbangkan untuk memperluas jangkauan pengguna dan mempermudah pelaporan tanaman langsung dari lapangan.



## DAFTAR PUSTAKA

- Belagatti Pavan. (2024, May 12). *Understanding the Softmax Activation Function: A Comprehensive Guide.* <https://www.singlestore.com/blog/a-guide-to-softmax-activation-function/>
- Dong, K., Zhou, C., Ruan, Y., & Li, Y. (2020). MobileNetV2 Model for Image Classification. *Proceedings - 2020 2nd International Conference on Information Technology and Computer Application, ITCA 2020*, 476–480. <https://doi.org/10.1109/ITCA52113.2020.00106>
- Gunawan, D., & Setiawan, H. (2022). Convolutional Neural Network dalam Citra Medis. *KONSTELASI: Konvergensi Teknologi Dan Sistem Informasi*, 2(2), 376–390. <https://doi.org/10.24002/konstelasi.v2i2.5367>
- Intyanto, G. W. (2021). Klasifikasi Citra Bunga dengan Menggunakan Deep Learning: CNN (Convolution Neural Network). *Jurnal Arus Elektro Indonesia*, 7(3), 80. <https://doi.org/10.19184/jaei.v7i3.28141>
- Khasanah, N. (2021). *IMPLEMENTASI ARSITEKTUR MOBILENETV2 UNTUK KLASIFIKASI CITRA BERAS IMPOR TESIS*. 1–23.
- Nelson, N. (2023). *Implementasi Siamese Convolutional Neural Network pada Citra Jerawat untuk Klasifikasi Jenis Jerawat.* 5–10. <https://kc.umn.ac.id/id/eprint/25550>
- Rajkomar, G., & Pudaruth, S. (2023). A Mobile App for the Identification of Flowers Using Deep Learning. *International Journal of Advanced Computer Science and Applications*, 14(5), 76–102. <https://doi.org/10.14569/IJACSA.2023.0140508>
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>
- Shahi, T. B., Sitaula, C., Neupane, A., & Guo, W. (2022). Fruit classification using attention-based MobileNetV2 for industrial applications. *PLoS ONE*, 17(2 February), 1–21. <https://doi.org/10.1371/journal.pone.0264586>
- Simanjutak, T. J., S, K. S., Syahputra, H., Iskandar, S., Idrus, A., & Febrian, D. (2025). *MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK ( CNN ) DI DESA BINTANG KECAMATAN SIDIKALANG.* 9(1), 584–591.
- Sukiatmodjo, J. (2023). *Sistem Pengenalan Jenis Bunga Menggunakan Deep Learning dengan CNN.*

Tragoudaras, A., Stoikos, P., Fanaras, K., Tziouvaras, A., Floros, G., Dimitriou, G., Kolomvatsos, K., & Stamoulis, G. (2022). Design Space Exploration of a Sparse MobileNetV2 Using High-Level Synthesis and Sparse Matrix Techniques on FPGAs. *Sensors*, 22(12), 1–16. <https://doi.org/10.3390/s22124318>

Wang, Z., Wang, K., Wang, X., Pan, S., & Qiao, X. (2022). Dynamic ensemble selection of convolutional neural networks and its application in flower classification. *International Journal of Agricultural and Biological Engineering*, 15(1), 216–223. <https://doi.org/10.25165/j.ijabe.20221501.6313>

Zhang, A. (2020). Dive Into Deep Learning. In *Journal of the American College of Radiology* (Vol. 17, Issue 5). <https://doi.org/10.1016/j.jacr.2020.02.005>

