

**TESIS**

**KLASIFIKASI JENIS BIJI KOPI SANGRAI MENGGUNAKAN METODE  
CONVOLUTIONAL NEURAL NETWORK (CNN)**



Disusun oleh:

**Nama : Hallfa Sekar Metha**  
**NIM : 21.51.2100**  
**Konsentrasi : Business Intelligence**

**PROGRAM STUDI S2 INFORMATIKA**  
**PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA**  
**YOGYAKARTA**

**2024**

**TESIS**

**KLASIFIKASI JENIS BIJI KOPI SANGRAI MENGGUNAKAN METODE  
CONVOLUTIONAL NEURAL NETWORK (CNN)**

**CLASSIFICATION OF ROASTED COFFEE BEANS USING THE  
CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

**Nama : Halffa Sekar Metha**  
**NIM : 21.51.2100**  
**Konsentrasi : Business Intelligence**

**PROGRAM STUDI S2 INFORMATIKA**  
**PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA**  
**YOGYAKARTA**

**2024**

**HALAMAN PENGESAHAN**

**KLASIFIKASI JENIS BIJI KOPI SANGRAI MENGGUNAKAN METODE  
CONVOLUTIONAL NEURAL NETWORK (CNN)**

**CLASSIFICATION OF ROASTED COFFEE BEANS USING THE  
CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD**

Dipersiapkan dan Disusun oleh

**Halifa Sekar Metha**

**21.51.2100**

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Senin, 5 Agustus 2024

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer

Yogyakarta, 5 Agustus 2024

**Rektor**

**Prof. Dr. M. Suyanto, M.M.**

**NIK. 190302001**

## HALAMAN PERSETUJUAN

### KLASIFIKASI JENIS BIJI KOPI SANGRAI MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

### CLASSIFICATION OF ROASTED COFFEE BEANS USING THE CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD

Dipersiapkan dan Disusun oleh

**Halfa Sekar Metha**

**21.51.2100**

Telah Ditujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Senin, 5 Agustus 2024

**Pembimbing Utama**

**Prof. Dr. Kusriani, M.Kom.**  
NIK. 190302106

**Pembimbing Pendamping**

**Dhani Ariatmanto, M.Kom., Ph.D.**  
NIK. 190302197

**Anggota Tim Penguji**

**Dr. Kumara Ari Yuana, S.T., M.T.**  
NIK. 190302575

**Dr. Ferry Wahyu Wibowo, S.Si, M.Cs.**  
NIK. 190302235

**Prof. Dr. Kusriani, M.Kom.**  
NIK. 190302106

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer

Yogyakarta, 5 Agustus 2024  
**Direktur Program Pascasarjana**

**Prof. Dr. Kusriani, M.Kom.**  
NIK. 19030210

## HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Halifa Sekar Methu  
NIM : 21.51.2100  
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:  
**Klasifikasi Jenis Biji Kopi Sangrai Menggunakan Metode Convolutional Neural Network (CNN)**

Dosen Pembimbing Utama : Prof. Dr. Kusriani, M.Kom.  
Dosen Pembimbing Pendamping : Dhani Ariatmanto, M.Kom., Ph.D.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 5 Agustus 2024  
Yang Menyatakan,

  
Halifa Sekar Methu

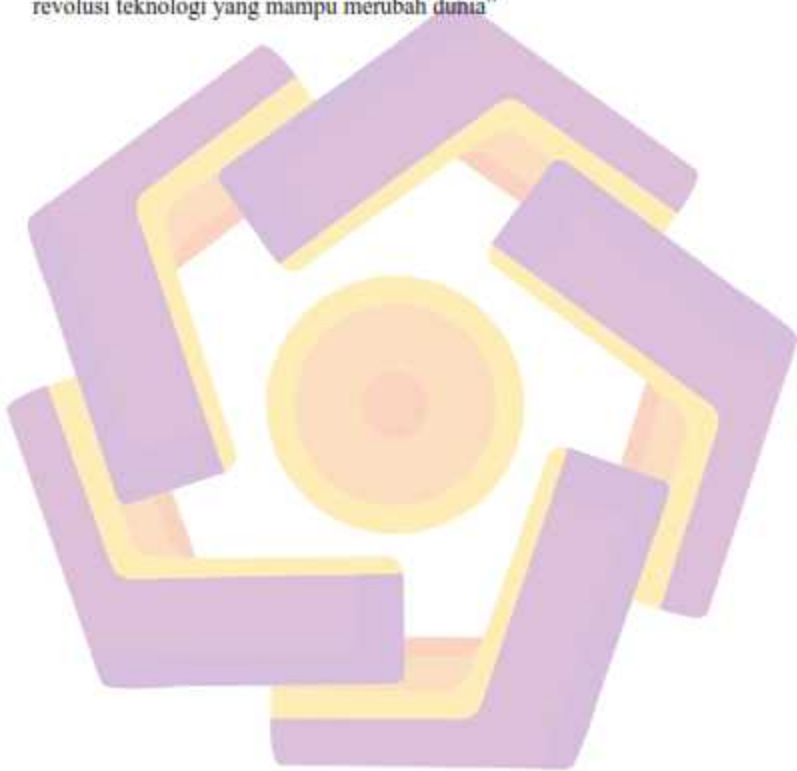
## HALAMAN PERSEMBAHAN

Segala puji bagi Allah Subhanahu wa ta'ala atas segala limpahan Rahmat karunia serta hidayahNya sehingga tesis ini dapat terselesaikan. Hasil ini saya persembahkan untuk :

1. Kedua orang tua saya yang saya cintai, dengan segala cinta dan kasih sayangnya, doa, harapan serta perjuangannya.
2. Saudara saya yang selalu mendukung dan mendoakan.
3. Teman saya yang menjadi support sistem terbaik.
4. Untuk diriku sendiri yang telah kuat berjuang sampai sejauh ini melewati berbagai macam tantangan kehidupan.
5. Untuk seorang laki-laki yang sedang menungguku, mungkin aku mengenalmu, mungkin tidak, mungkin kita pernah bertemu, mungkin juga belum. Tak apa saat ini belum waktunya, ada saatnya kita akan bersatu.

## HALAMAN MOTTO

“Dalam setiap baris kode dan setiap algoritma yang dikembangkan, tertanam harapan untuk menemukan jawaban atas tantangan zaman, menyongsong revolusi teknologi yang mampu merubah dunia”





## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah Subhanahu wa ta'ala atas limpahan rahmat, hidayah serta inayah-Nya, penulis masih diberi kesempatan dan kemudahan untuk menyelesaikan tesis ini.

Penulis mengucapkan terimakasih yang sebesar-besarnya, kepada :

1. Bapak Prof. Dr. M.Suyanto, MM, selaku rektor Universitas AMIKOM Yogyakarta
2. Ibu Prof. Dr. Kusriani, M.Kom, selaku pembimbing utama yang telah mendampingi dalam penyusunan tesis ini.
3. Bapak Dhani Ariatmanto, M.Kom., Ph.d, selaku pembimbing pendamping yang selalu memberi arahan kepada penulis dalam pengerjaan tesis ini.
4. Kedua orang tua tercinta yang tak pernah lelah dalam memberikan dukungan dan doanya.

Semoga tesis ini dapat menambah pengetahuan dan memberi manfaat bagi para pembacanya serta dapat digunakan sebagai salah satu referensi untuk penelitian selanjutnya.

Yogyakarta, 5 Agustus 2024

Penulis



## DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiii
INTISARI.....	xv
<i>ABSTRACT</i> .....	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1. Tinjauan Pustaka.....	6
2.2. Keaslian Penelitian.....	10

2.3. Landasan Teori.....	15
2.3.1. Klasifikasi.....	15
2.3.2. Biji Kopi Sangrai.....	15
2.3.3. Convolutional Neural Network (CNN).....	15
2.3.4. Gambaran Arsitektur Convolutional Neural Network (CNN).....	16
2.3.5. Layer.....	17
2.3.6. Hyperparameter.....	20
2.3.7. MobileNetV2.....	22
2.3.8. InceptionV3.....	23
2.3.9. Metode Evaluasi.....	23
<b>BAB III METODE PENELITIAN.....</b>	<b>26</b>
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	26
3.2. Metode Pengumpulan Data.....	26
3.3. Metode Analisis Data.....	26
3.4. Alur Penelitian.....	27
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....</b>	<b>32</b>
4.1. Pengumpulan Data.....	32
4.2. Preprocessing.....	34
4.2.1. Split Data.....	34
4.2.2. Augmentasi.....	35
4.3. Proses Klasifikasi menggunakan CNN.....	40
4.3.1. MobileNetV2.....	40
4.3.2. InceptionV3.....	41

4.4. Skenario Percobaan.....	43
4.4.1. Skenario Percobaan Pertama.....	44
4.4.2. Skenario Percobaan Kedua.....	44
4.4.3. Skenario Percobaan Ketiga.....	45
4.4.4. Skenario Percobaan Keempat.....	47
4.5. Proses Pengujian.....	48
4.5.1. Pengujian Skenario Pertama.....	48
4.5.2. Pengujian Skenario Kedua.....	51
4.5.3. Pengujian Skenario Ketiga.....	53
4.5.4. Pengujian Skenario Keempat.....	55
4.6. Evaluasi Hasil Penelitian.....	59
4.6.1. Hasil Akurasi MobileNetV2.....	67
4.6.2. Hasil Akurasi InceptionV3.....	68
4.6.3. Perbandingan Hasil Akurasi Kedua Arsitektur.....	68
<b>BAB V PENUTUP.....</b>	<b>71</b>
5.1. Kesimpulan.....	71
5.2. Saran.....	72
<b>DAFTAR PUSTAKA.....</b>	<b>73</b>

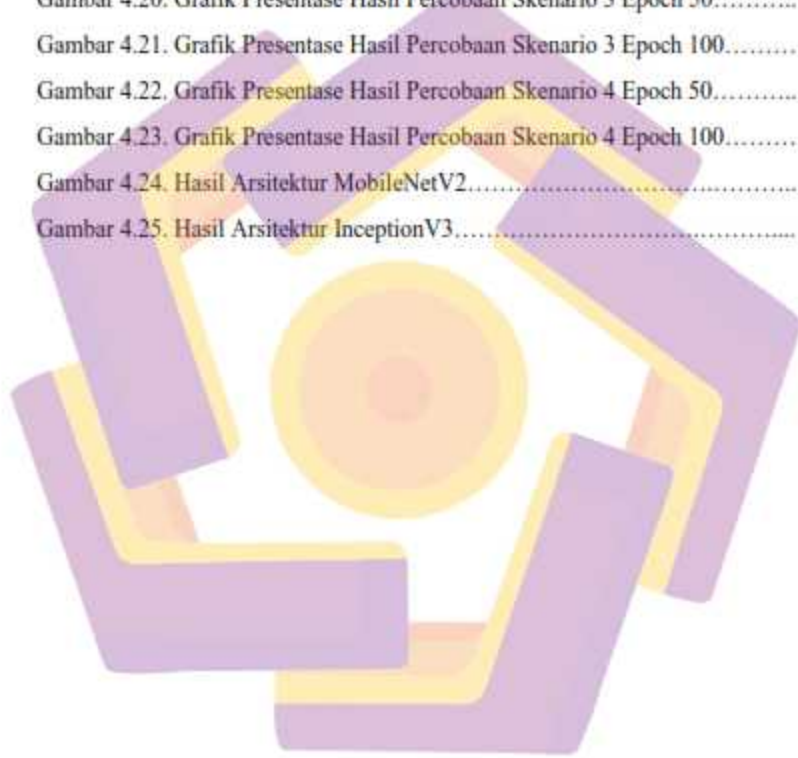
## DAFTAR TABEL

Tabel 2.1. Matriks literatur review dan posisi penelitian.....	10
Tabel 2.2. Arsitektur MobileNetV2 .....	23
Tabel 2.3. Tabel Pengujian Confusion Matrix.....	24
Tabel 3.1. Detail Pembagian Dataset .....	29
Tabel 4.1. Contoh Sampel Biji Kopi.....	33
Tabel 4.2. Skenario Percobaan .....	43
Tabel 4.3. Percobaan Skenario 1.....	44
Tabel 4.4. Percobaan Skenario 2.....	44
Tabel 4.5. Percobaan Skenario 3.....	46
Tabel 4.6. Percobaan Skenario 4.....	47
Tabel 4.7. Hasil Pengujian.....	65

## DAFTAR GAMBAR

Gambar 2.1. Arsitektur Convolutional Neural Network.....	16
Gambar 2.2. Convolution Layer.....	17
Gambar 2.3. ReLu Function.....	18
Gambar 2.4. Pooling Layer.....	19
Gambar 2.5. Fully Connected Layer.....	20
Gambar 2.6. Alur Proses InceptionV3.....	23
Gambar 3.1. Alur Penelitian.....	27
Gambar 3.2. Tahapan Preprocessing.....	28
Gambar 4.1. Library Import Dataset.....	33
Gambar 4.2. Split Dataset.....	34
Gambar 4.3. Sampel Biji Kopi Sangrai.....	35
Gambar 4.4. List Kode Proses Augmentasi (Flips).....	35
Gambar 4.5. Hasil Augmentasi (Flips).....	36
Gambar 4.6. List Kode Proses Rotations.....	37
Gambar 4.7. Hasil Augmentasi (Rotations).....	37
Gambar 4.8. List Kode Proses Zoom.....	38
Gambar 4.9. Hasil Augmentasi (Zoom).....	38
Gambar 4.10. List Kode Proses Strecth.....	39
Gambar 4.11. Hasil Augmentasi (Strecth).....	40
Gambar 4.12. List Kode Fungsi Arsitektur MobileNetV2.....	40
Gambar 4.13. List Kode Fungsi Arsitektur InceptionV3.....	42
Gambar 4.14. Kode Penambahan Hyperparameter pada Skenario Percobaan.....	45

Gambar 4.15. Hasil Pengujian MobileNetV2.....	49
Gambar 4.16. Hasil Pengujian InceptionV3.....	51
Gambar 4.17. Hasil Pengujian MobileNetV2+Hyperparameter.....	53
Gambar 4.18. Hasil Pengujian InceptionV3+Hyperparameter.....	56
Gambar 4.19. Grafik Presentase Hasil Percobaan Skenario 1 dan 2.....	59
Gambar 4.20. Grafik Presentase Hasil Percobaan Skenario 3 Epoch 50.....	60
Gambar 4.21. Grafik Presentase Hasil Percobaan Skenario 3 Epoch 100.....	61
Gambar 4.22. Grafik Presentase Hasil Percobaan Skenario 4 Epoch 50.....	62
Gambar 4.23. Grafik Presentase Hasil Percobaan Skenario 4 Epoch 100.....	63
Gambar 4.24. Hasil Arsitektur MobileNetV2.....	67
Gambar 4.25. Hasil Arsitektur InceptionV3.....	68





## INTISARI

Di antara hasil perkebunan, kopi memiliki nilai ekonomi yang relatif tinggi dibandingkan dengan tanaman perkebunan lainnya dan merupakan sumber devisa yang signifikan. Salah satu kunci dalam meningkatkan daya saing produk biji kopi adalah proses pengolahan biji kopi yang dimulai dari proses pemetikan biji kopi dan berlanjut hingga biji kopi siap disangrai.

Awalnya, klasifikasi biji kopi dilakukan berdasarkan pengalaman dan pengetahuan manusia dengan cara *human visual system* (HVS). Namun, dalam prosesnya manusia memiliki keterbatasan stamina sehingga mempengaruhi keakuratan dalam mengklasifikasikan biji kopi, terutama dalam membedakan jenis kopi yang sudah disangrai. Sehingga, teknologi pengenalan citra gambar menjadi salah satu cara untuk membantu proses klasifikasi.

Pada penelitian ini, penulis melakukan klasifikasi pada biji kopi sangrai jenis Arabika. Ada 4 tingkatan yaitu tingkatan berwarna hijau dengan nama Green Laos Typica Bolaven, tingkatan terang dengan nama Light Laos Typica Bolaven, kemudian tingkatan medium dengan nama Doi Chaang, yang terakhir tingkatan dark dengan Brazil Cerrado dengan menggunakan arsitektur CNN yaitu MobileNetV2 dan InceptionV3 dengan tujuan untuk mengetahui dan membandingkan tingkat akurasi dari arsitektur CNN yang digunakan. Dari hasil penelitian didapatkan nilai akurasi tertinggi pada penelitian ini terhadap klasifikasi biji kopi sangrai adalah arsitektur MobileNetV2 dengan penerapan *hyperparameter* pada proses *training*, mendapatkan nilai akurasi sebesar 98.12%. Sedangkan pada penggunaan InceptionV3 mendapatkan nilai akurasi sebesar 96.88%. Kinerja MobileNetV2 lebih baik dari InceptionV3 sehingga penggunaan MobileNetV2 dengan *adjusting hyperparameter* bisa dipertimbangkan untuk penelitian berikutnya.

Kata kunci: Klasifikasi, Arsitektur CNN, Biji Kopi Sangrai, Perbandingan Akurasi



## **ABSTRACT**

*Among plantation products, coffee has relatively high economic value compared to other plantation crops and is a significant source of foreign exchange. One of the keys to increasing the competitiveness of coffee bean products is the coffee bean processing process which starts from the process of picking the coffee beans and continues until the coffee beans are ready to be roasted.*

*Initially, coffee bean classification was carried out based on human experience and knowledge using the human visual system (HVS). However, in the process humans have limited stamina which affects the accuracy in classifying coffee beans, especially in distinguishing types of coffee that have been roasted. So, image recognition technology is one way to help the classification process.*

*In this research, the author classified Arabica roasted coffee beans. There are 4 levels, namely the green level with the name Green Laos Typica Bolaven, the light level with the name Light Laos Typica Bolaven, then the medium level with the name Doi Chaang, finally the dark level with Brazil Cerrado using the CNN architecture, namely MobileNetV2 and InceptionV3 with the aim of finding out and compare the accuracy level of the CNN architecture used. From the research results, it was found that the highest accuracy value in this research for the classification of roasted coffee beans was the MobileNetV2 architecture with the application of hyperparameters in the training process, obtaining an accuracy value of 98.12%. Meanwhile, when using InceptionV3, you get an accuracy value of 96.88%. The performance of MobileNetV2 is better than InceptionV3 so that the use of MobileNetV2 with adjusting hyperparameters can be considered for future research.*

*Keyword: Classification, CNN Architecture, Roasted Coffee Beans, Accuracy Comparison*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Di antara hasil perkebunan, kopi memiliki nilai ekonomi yang relatif tinggi dibandingkan dengan tanaman perkebunan lainnya dan merupakan sumber devisa yang signifikan (Teniro & Zainudin, 2022). Salah satu kunci dalam meningkatkan daya saing produk biji kopi adalah proses pengolahan biji kopi yang dimulai dari proses pemetikan biji kopi dan berlanjut hingga biji kopi siap disangrai (Muammar Khadafi Ihsan, 2021). Suhu memainkan peran penting dalam proses pemanggangan karena panas mengubah bentuk biji kopi dan melepaskan senyawa yang berkontribusi pada rasa dan aromanya (Santoso et al., 2022).

Awalnya, klasifikasi biji kopi dilakukan berdasarkan pengalaman dan pengetahuan manusia dengan cara *human visual system* (HVS). Namun, dalam prosesnya manusia memiliki keterbatasan stamina sehingga mempengaruhi keakuratan dalam mengklasifikasikan biji kopi, terutama dalam membedakan jenis kopi yang sudah disangrai. Sehingga, teknologi pengenalan citra gambar menjadi salah satu cara untuk membantu proses klasifikasi.

Beberapa studi telah dilakukan untuk menangani permasalahan mengenai klasifikasi jenis biji kopi yang telah disangrai, menggunakan *Convolutional Neural Network* (CNN). Metode ini dipilih karena kemampuannya dalam memproses data berupa piksel dan citra visual dengan spesifik (Wirogatama, 2019). Metode ini juga memiliki keunggulan dalam mengenali objek pada posisi yang spesifik dan

meningkatkan efisiensi waktu pemrosesan (Setiawan, 2020). Hal ini menjadikan *Convolutional Neural Network* (CNN) sebagai metode yang sangat efisien dan efektif dalam klasifikasi gambar.

Beberapa penelitian yang membahas tentang klasifikasi di bidang pertanian sebelumnya sudah pernah dilakukan. Penelitian (Rivalto et al., 2020) *Classification of Indonesian Coffee Types with Deep Learning*. Penelitian ini menggunakan algoritma *Deep Learning* dan juga metode *Convolutional Neural Network* (CNN). Pengumpulan data dilakukan dengan mengambil gambar biji kopi menggunakan kamera. Pengumpulan data ini berisi 4 jenis kopi asal Indonesia (Garut, Gayo, Kerinci, Temanggung) dengan 617 gambar biji kopi. Setelah pengujian, sistem dapat mengenali objek dengan akurasi 74,26%.

Penelitian selanjutnya *Alexnet Convolutional Neural Network to Classify The Types of Indonesian Coffee Beans* (Hendrawan et al., 2021) Penelitian ini bertujuan untuk mengklasifikasikan tiga jenis biji kopi Arabika Indonesia yaitu Gayo Aceh, Kintamani Bali, dan Toraja Tongkonan. Metode klasifikasi yang digunakan adalah *AlexNet* jaringan saraf konvolusional dengan analisis sensitivitas menggunakan beberapa variasi pengoptimal seperti SGDm, Adam, dan RMSProp serta *learning rate* 0,00005 dan 0,0001. Setiap jenis kopi menggunakan 500 data untuk pelatihan dan validasi dengan sebaran 70% pelatihan dan 30% validasi. Pada pengujian *Confusion Matrix*, akurasinya mencapai 99,6%.

Penelitian Klasifikasi Jenis Biji Kopi menggunakan *Convolutional Neural Network* dan *Transfer Learning* pada model VGG16 dan MobileNetV2. Dalam pengujian yang melibatkan tiga model, yaitu model CNN, model CNN-transfer

learning VGG16, dan MobileNetV2, hasilnya menunjukkan bahwa akurasi tertinggi tercapai saat menggunakan model CNN-transfer learning MobileNetV2 untuk mengklasifikasikan citra biji kopi, yakni mencapai 96% (Murinto et al., 2023).

Dalam penelitian ini dilakukan penerapan 2 model arsitektur yaitu MobileNetV2 dan InceptionV3 untuk mengklasifikasi 4 jenis tingkat biji kopi sangrai dari jenis biji kopi Arabika. Ada 4 tingkatan kopi biji sangrai yaitu tingkatan berwarna hijau dengan nama Green Laos Typica Bolaven, tingkatan terang dengan nama Light Laos Typica Bolaven, kemudian tingkatan medium dengan nama Doi Chaang, yang terakhir tingkatan dark dengan Brazil Cerrado. Penelitian ini dilakukan untuk mengukur dan membandingkan mana tingkat akurasi tertinggi dari 2 model arsitektur CNN MobileNetV2 dan InceptionV3.

### **1.2. Rumusan Masalah**

Bagian ini memuat permasalahan yang akan dirumuskan adalah sebagai berikut :

- a. Seberapa tinggi tingkat akurasi yang dihasilkan dari klasifikasi jenis biji kopi sangrai menggunakan arsitektur CNN model MobileNetV2 dan InceptionV3?
- b. Faktor apa saja yang mempengaruhi tingkat akurasi yang dihasilkan dari kedua arsitektur CNN model MobileNetV2 dan InceptionV3?

### **1.3. Batasan Masalah**

Batasan masalah dalam penelitian ini adalah sebagai berikut :

- a. Jenis biji kopi yang digunakan untuk data penelitian ini adalah tingkatan berwarna hijau dengan nama Green Laos Typica Bolaven, tingkatan terang



dengan nama Light Laos Typica Bolaven, kemudian tingkatan medium dengan nama Doi Chaang, dan yang terakhir tingkatan dark dengan nama Brazil Cerrado yang terdiri dari 1600 gambar dan masing-masing berjumlah 400.

- b. Data objek gambar biji kopi sangrai untuk proses training dan testing berasal dari *Kaggle.com* (<https://www.kaggle.com/datasets/gpiosenka/coffee-bean-dataset-resized-224-x-224>)
- c. Dilakukan *preprocessing* data dengan melakukan augmentasi, serta pembagian data *training* dan data *testing* secara manual.
- d. Ekstraksi fitur gambar berdasarkan ciri ciri bentuk dari objek citra biji kopi sangrai dengan menggunakan algoritma *Convolutional Neural Network* (CNN).
- e. Model arsitektur yang digunakan adalah MobileNetV2 dan InceptionV3.
- f. Maksimal nilai epoch adalah 100.
- g. Proses evaluasi model klasifikasi menggunakan *Confusion Matrix*.

#### **1.4. Tujuan Penelitian**

Tujuan penelitian ini adalah sebagai berikut :

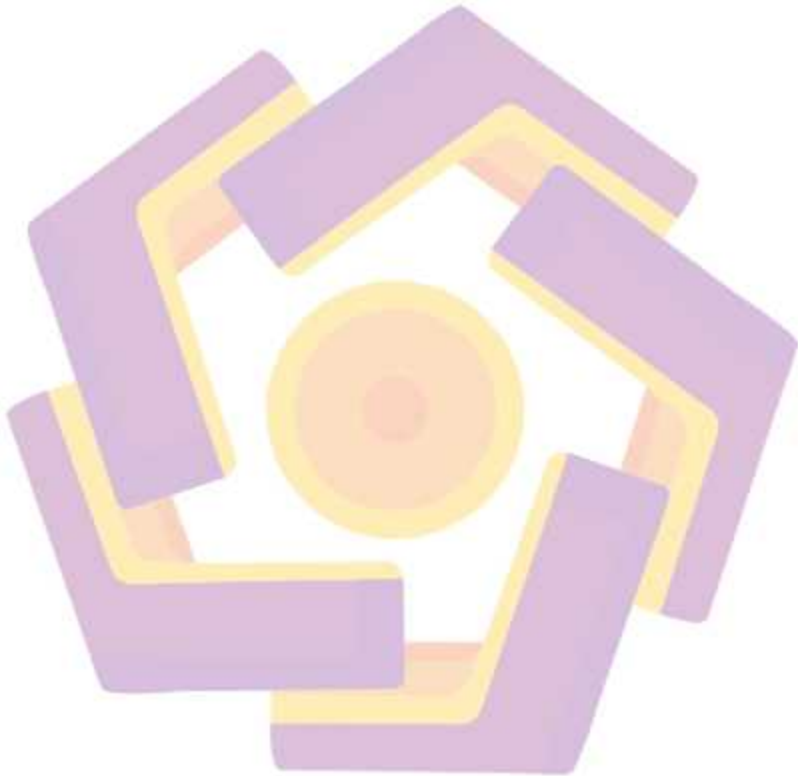
Untuk mengetahui tingkat akurasi dari arsitektur CNN yang digunakan pada klasifikasi biji kopi sangrai.

#### **1.5. Manfaat Penelitian**

Manfaat dari penelitian ini adalah sebagai berikut :

- a. Rancangan arsitektur yang diterapkan dapat dijalankan dan dapat mengklasifikasikan gambar biji kopi sangrai dengan akurasi yang tinggi sesuai dengan batasan yang telah ditentukan.

- b. Memberikan kontribusi penelitian terhadap biji kopi sangrai menggunakan metode CNN dengan dua model arsitektur terhadap nilai akurasi yang dihasilkan.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Tinjauan Pustaka

Penelitian yang dilakukan oleh (Yunita, 2022) Implementasi Metode *Linear Discriminan Analysis* untuk Klasifikasi Biji Kopi. Penelitian ini melakukan identifikasi jenis biji kopi dengan algoritma *Linear Discriminan Analysis* (LDA) untuk proses klasifikasi tiga jenis biji kopi arabika, biji kopi robusta dan biji kopi liberika. Pada penelitian ini terdapat tahapan akuisisi citra, *preprocessing*, ekstraksi fitur, penerapan LDA, dan evaluasi. Dalam proses klasifikasi menggunakan algoritma LDA dengan menggunakan tiga kali percobaan dengan pengujian menggunakan 80% citra latih 20% citra uji, 70% citra latih 30% citra uji, dan 60% citra latih 40% citra uji. Hasil dari penelitian ini dengan menggunakan 150 dataset dapat menghasilkan akurasi sebesar 84% untuk performa lainnya dengan nilai precision 95%, recall 88%, dan F-Measure 86%.

Penelitian yang berjudul *Classification of Roasted Coffee Beans with Principal Component Analysis and Random Forest* (Miftahuddin & Rais, 2024). Penelitian ini bertujuan untuk mengidentifikasi dan mengklasifikasikan jenis Robusta Ciwidey atau Robusta Toraja biji kopi sangrai menggunakan algoritma *Random Forest* dan PCA. Tahap *preprocessing* yaitu *greyscaling* dan *resize*, ekstraksi fitur dengan *Principal Component Analysis* dan klasifikasi dengan *Random Forest*. Dengan dataset 2000 data gambar yang terbagi menjadi 1000 gambar biji kopi Ciwidey dan 1000 gambar biji kopi toraja dengan komposisi 800



gambar latih dan 200 gambar uji. Dengan metode *Random Forest* mendapatkan akurasi tertinggi sebesar 98,05% dengan nilai  $n = 20$  dan 360 data diuji.

Penelitian sebelumnya yaitu oleh (Cahyono et al., 2023) mengangkat judul *Klasifikasi Jenis Biji Kopi dengan menggunakan Metode Gray Level Co-occurrence Matrix (GLCM)*. Tujuan dari penelitian ini adalah untuk mengetahui akurasi mengklasifikasikan jenis biji kopi dengan menggunakan metode *Gray Level Co-occurrence Matrix (GLCM)*. Pada penelitian ini dataset yang digunakan adalah biji kopi jenis *Robusta Tanggul dan Raung*, serta *Arabika Ijen dan Bali* yang telah di sangrai. Tingkat kematangan sangrai biji kopi yang digunakan pada penelitian ini adalah coklat muda (*light roast*) untuk jenis *Arabika* dan *medium roast* untuk jenis *robusta*. Hasil yang diperoleh dari akurasi klasifikasi jenis biji kopi dengan menggunakan metode *GLCM* sebesar 99% untuk klasifikasi tipe kernel atau powder, 93% untuk klasifikasi jenis *Robusta* atau *Arabika*, dan 56% untuk klasifikasi daerah asal yaitu (*Tanggul, Raung, Ijen, dan Bali*) dengan citra data uji sebanyak 80 citra.

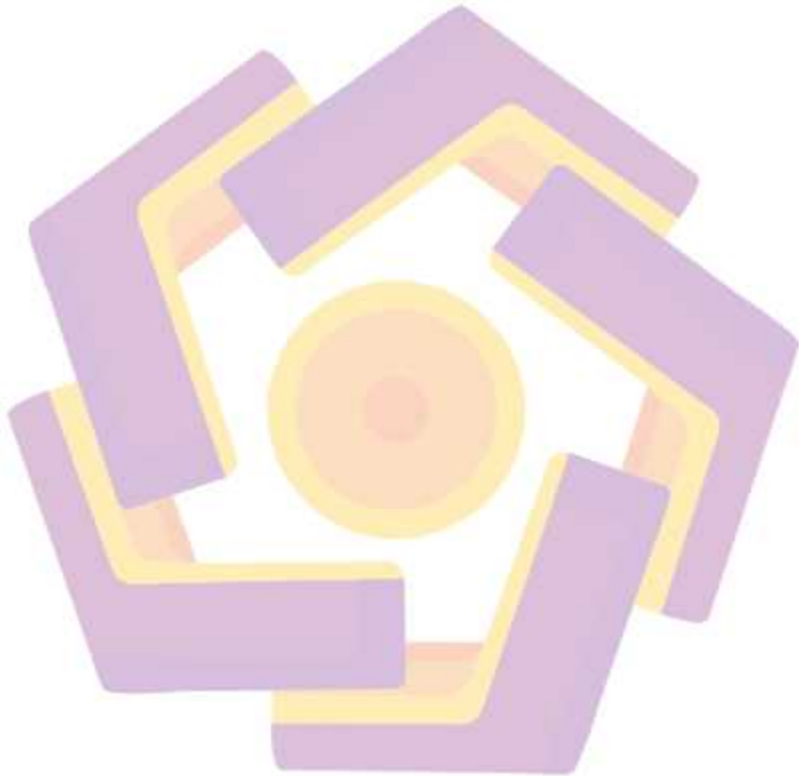
Penelitian selanjutnya oleh (Waliyansyah & Umar Hafidz Asy'ari Hasbullah, 2021) dengan judul *Comparison of Tree Method, Support Vector Machine, Naïve Bayes, and Logistic Regression on Coffee Bean Image* berisi tentang pengklasifikasian biji kopi yang dilakukan dengan pengolahan citra digital. Parameter yang digunakan adalah analisis tekstur menggunakan metode *Gray Level Cooccurrence Matrix (GLCM)* dengan 4 fitur yaitu *Energy, Correlation, Homogeneity & Contrast*. Untuk ekstraksi fitur menggunakan algoritma klasifikasi yaitu *Naïve Bayes, Tree, Support Vector Machine (SVM)* dan *Logistic Regression*.

Evaluasi model klasifikasi biji kopi menggunakan parameter sebagai berikut: AUC, F1, CA, presisi & recall. Dataset yang digunakan adalah 29 gambar biji kopi Arabika dan 29 gambar biji Robusta. Untuk menguji keakuratan model menggunakan *Cross Validation*. Hasil yang diperoleh akan dievaluasi menggunakan Matriks Konfusi. Berdasarkan hasil pengujian dan evaluasi model diperoleh metode SVM yang terbaik dengan nilai AUC = 1, CA = 0.983, F1 = 0.983, Precision = 0.983 dan Recall = 0.983.

Penelitian (Amtate & Teferi, 2022) *Multiclass classification of Ethiopian coffee bean using deep learning*. Penelitian ini bertujuan untuk mengembangkan model yang mengklasifikasikan biji kopi dari enam asal berbeda di Ethiopia (Jimma, Limmu, Nekemte, Yirgacheffe, Bebeke, dan Sidama) ke dalam sembilan kelas. Dataset untuk penelitian ini dikumpulkan dari Pusat Inspeksi dan Lelang Kualitas Kopi Ethiopia (ECQIAC). Total dataset 8646 gambar biji kopi dikumpulkan dan 1190 gambar ditambahkan menggunakan augmentasi sehingga total kumpulan data menjadi 9836. Penelitian dilakukan menggunakan CNN dengan normalisasi *batch* dan augmentasi, dataset dibagi menjadi 80% untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian. Adapun akurasi klasifikasi 99,89% dengan *log-loss* sebesar 0,92%.

Penelitian (Janandi & Cenggoro, 2020) dengan judul *An Implementation of Convolutional Neural Network for Coffee Beans Quality Classification in a Mobile Information System*. Penelitian menggunakan arsitektur ResNet-152 dan VGG16. Dataset diambil dengan cara mengumpulkan gambar biji kopi dan kemudian memberi label dan sebanyak 160 gambar. Pelabelan dari Standar Nasional

Indonesia (Indonesian National Standard). Hasilnya menunjukkan bahwa ResNet-152 mencapai akurasi tertinggi sebesar 73,3%.



## 2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian  
Klasifikasi Jenis Biji Kopi Sangrai menggunakan Metode Convolutional Neural Network (CNN)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran	Perbandingan
1	Klasifikasi Varietas Kopi berdasarkan <i>Green Bean Coffee</i> menggunakan Metode <i>Machine Learning</i>	M. Agung Nugroho & Maria Mediatrix Sebatubun. <i>Joism : Jurnal Of Information System Management</i> . 2020	Mengidentifikasi varietas kopi arabika agar dapat digunakan sebagai second opinion untuk mengidentifikasi varietas kopi arabika.	Akurasi yang didapat adalah varietas ciwangi red bourbon sebesar 71,4%, ciwangi catimor 85,7%, dan rasuna sigarar utang 80%.	Agar diperbanyak lagi data varietasnya untuk mencapai keberhasilan proses klasifikasinya.	Penelitian yang dilakukan termasuk kedalam deep learning dan menggunakan jenis kopi arabika yang didalamnya ada 4 jenis tingkat sangrai yaitu tidak di sangrai atau yang masih hijau, sangrai ringan, sedang, dan gelap. Sedangkan Pada penelitian yang dirujuk ini menggunakan metode machine learning. Dengan meneliti tiga varietas green bean kopi yaitu Ciwangi Catimor, Ciwangi Redbourbon, dan Rasuna Sigararutang.

Tabel 2.1. Matriks literatur review dan posisi penelitian  
 Klasifikasi Jenis Biji Kopi Sangrai menggunakan Metode Convolutional Neural Network (CNN) (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran	Perbandingan
2	<i>Classification of Indonesian Coffee Types with Deep Learning</i>	Alfan Rivalto, Pranowo, & Albertus Joko Santoso, AIP Conference Proceedings, 2020	Mengklasifikasikan jenis biji kopi Indonesia menggunakan Deep Learning.	Penelitian ini menggunakan 4 jenis kopi asal Indonesia (Garut, Gayo, Kerinci, Temanggung) dengan 617 gambar biji kopi. Dan setelah pengujian akurasi yang di dapat adalah sebesar 74,26%.	Jika menambahkan jumlah dataset yang lebih banyak kemungkinan akan menghasilkan hasil yang lebih baik lagi. Dan juga sebaiknya menggunakan model arsitektur lainnya seperti AlexNet, Resnet, Inception.	Sama sama menggunakan algoritma Convolutional Neural Network (CNN) akan tetapi penelitian yang akan di lakukan memakai arsitektur MobileNetV2 dan InceptionV3. Penelitian yang akan dilakukan hanya mencari tingkat akurasi terbaik dari klasifikasi biji kopi sangrai. Sedangkan penelitian yang dirujuk dapat diimplementasikan ke dalam aplikasi mobile dan diimplementasikan secara real-time.



Tabel 2.1. Matriks literatur review dan posisi penelitian  
Klasifikasi Jenis Biji Kopi Sangrai menggunakan Metode Convolutional Neural Network (CNN) (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran	Perbandingan
3	<i>An Implementation of Convolutional Neural Network for Coffee Beans Quality Classification in a Mobile Information System</i>	Robby Janandi & Tjeng Wawan Cenggoro. <i>International Conference on Information Management and Technology (ICIMTech)</i> . IEEE, 2020.	Mengklasifikasikan biji kopi berdasarkan kualitasnya dan mengembangkan aplikasi seluler yang didukung oleh deep-learning berdasarkan model untuk secara otomatis mengklasifikasikan kualitas biji kopi melalui kamera ponsel.	ResNet-152 lebih baik di bandingkan VGG16 untuk kasus klasifikasi kualitas biji kopi yaitu sebesar 73,3%.	Penelitian ini mungkin akan lebih baik lagi jika menggunakan Hyperparameter kedalam eksperimennya.	Penelitian yang dilakukan yaitu mengklasifikasikan jenis biji kopi sangrai lalu setelah itu mencari model mana yang terbaik. Sedangkan penelitian yang dirujuk menggunakan aplikasi seluler untuk mengetahui informasi tentang kualitas biji kopi.
4	<i>Smart agriculture: real-time classification of green coffee beans by using a convolutional neural network</i>	Nen-Fu Huang dkk. <i>IET Smart Cities</i> . 2020	Untuk memecahkan masalah manual yang memakan waktu pemilihan biji kopi menggunakan <i>Deep Learning</i> dan <i>Convolutional Neural Network</i> .	Akurasi klasifikasi biji kopi secara keseluruhan adalah 93%.	Sistem ini hanya dapat menangani satu sisi biji kopi. Jika cacat muncul di sisi belakang peneliti tidak akan melakukannya. Dan di penelitian ini perlu melatih model yang mampu mendeteksi cacat pada kedua sisi biji kopi.	Penelitian tidak menggunakan sistem real time. Sedangkan penelitian yang dirujuk menggunakan real time yaitu secara langsung dapat mengetahui jenis biji kopi.

Tabel 2.1. Matriks literatur review dan posisi penelitian  
 Klasifikasi Jenis Biji Kopi Sangrai menggunakan Metode Convolutional Neural Network (CNN) (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran	Perbandingan
5	<i>Classification of Arabica Coffee Green Beans Using Digital Image Processing Using the K-Nearest Neighbor Method</i>	Nurun Najmi Aminina & Galuh Wilujeng Saraswat, Journal of Applied Intelligent System (JAIS), 2022	Melakukan klasifikasi menggunakan metode K-Nearest Neighbor dan ekstraksi fitur menggunakan rata-rata nilai warna Merah-Hijau-Biru (RGB) dalam menentukan mutu dan kualitas biji kopi menurut gradenya sehingga bisa mendapatkan harga jual yang tinggi.	Dalam penelitian ini menggunakan sebanyak 150 data citra latih dan 150 data citra uji, hasil akurasi klasifikasi ini adalah 80%.	Untuk menghasilkan klasifikasi yang baik sebaiknya menggunakan beberapa model yang berbeda dengan memanfaatkan metode kecerdasan buatan.	Perbedaan antara penelitian sebelumnya dengan penelitian yang akan dilakukan, yaitu pada model K-Nearest Neighbor. Sedangkan penelitian yang akan dilakukan menggunakan model Convolutional Neural Network.



Tabel 2.1. Matriks literatur review dan posisi penelitian  
 Klasifikasi Jenis Biji Kopi Sangrai menggunakan Metode Convolutional Neural Network (CNN) (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran	Perbandingan
6	Klasifikasi Jenis Biji Kopi menggunakan Convolutional Neural Network dan Transfer Learning pada model VGG16 dan MobileNetV2	Murinto, Mifiaharahma Rosyd & Martania Melany. JRST (Jurnal Riset Sains dan Teknologi). 2023	Mengusulkan model Convolutional Neural Network (CNN) Transfer Learning untuk diimplementasikan pada sistem cerdas untuk proses klasifikasi citra jenis biji kopi.	Dari hasil pengujian yang dilakukan pada 3 model yakni model CNN, Model CNN-transfer learning VGG16 dan MobileNetV2 didapatkan hasil bahwa akurasi yang paling tinggi didapatkan ketika melakukan klasifikasi citra biji kopi dengan menggunakan CNN-transfer learning model MobileNetV2 yakni sebesar 96%.	Melakukan uji coba lebih banyak lagi pada parameter yang ada pada model CNN-transfer learning sehingga didapatkan akurasi klasifikasi yang lebih tinggi	Penelitian yang akan dilakukan menggunakan arsitektur InceptionV3 sedangkan penelitian terdahulu VGG16.

## 2.3. Landasan Teori

### 2.3.1. Klasifikasi

Klasifikasi adalah teknik analisis data yang membangun model untuk mengkarakterisasi kelas yang sudah ada dari data. Oleh karena itu, kelas-kelas yang ditunjukkan dalam data akan dikompilasi menggunakan klasifikasi ini (Wijaya et al., 2021). Klasifikasi adalah metode yang digunakan untuk membangun fungsi atau model yang menjelaskan kelas pada data atau untuk meramalkan kelas suatu objek yang labelnya belum ditetapkan (Wijaya et al., 2021).

### 2.3.2. Biji Kopi Sangrai

Biji kopi sangrai merupakan kopi yang sudah melalui proses penghapusan air dari kopi, pengembangan biji, pengurangan berat hingga 20%, dan transit gula menjadi CO<sub>2</sub> yang memberi aroma pada kopi. Biji kopi sangrai biasanya digunakan untuk membuat kopi. Prosedur penyangraian berdampak pada rasa, aroma, dan komposisi kimia bioaktif kopi, yang juga memengaruhi aktivitas antioksidan (Mangiwa & Maryuni, 2019). Biji kopi berwarna *light* hingga *dark* dihasilkan selama proses pengsangraian yang biasanya berlangsung pada suhu 200–240 C (Mangiwa & Maryuni, 2019).

### 2.3.3. Convolutional Neural Network (CNN)

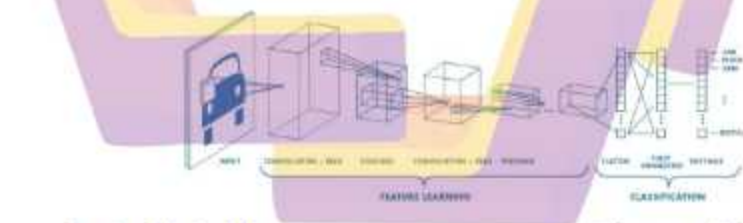
Salah satu jenis *neural network* yang sering digunakan dalam pengolahan data gambar adalah *Convolutional Neural Network* (CNN) (Saputra et al., 2020). *Convolutional Neural Network* (CNN) merupakan hasil pengembangan dari *Multilayer Perceptron* (MLP) yang dirancang khusus untuk memproses data dua dimensi. Karena strukturnya yang dalam dan sering digunakan dalam pemrosesan

gambar, CNN termasuk dalam kategori *Deep Neural Network*. CNN pertama kali dikembangkan oleh peneliti Jepang bernama Kunihiko Fukushima pada tahun 1980 dan dinamai sebagai NeoCognitron Fukushima.

*Convolutional Neural Network* (CNN) digunakan untuk menganalisis gambar visual dengan tujuan mendeteksi dan mengenali objek dalam gambar. Ini melibatkan penggunaan vektor berdimensi tinggi yang memerlukan banyak parameter untuk menggambarkan jaringan. Secara umum, CNN tidak terlalu berbeda dengan jaringan saraf biasa, karena keduanya terdiri dari neuron yang memiliki bobot, bias, dan fungsi aktivasi (Nugroho et al., 2020).

#### 2.3.4. Gambaran Arsitektur *Convolutional Neural Network*

*Convolutional Neural Network* (CNN) memiliki banyak lapisan dengan sejumlah *neuron* pada setiap lapisan (Saputra et al., 2020). CNN terdiri dari 2 lapisan arsitektur, yaitu *feature learning* dan *classification layer* seperti yang terlihat pada gambar 2.1.



Gambar 2.1. Arsitektur *Convolutional Neural Network* (Saputra et al., 2020)

Berdasarkan gambar diatas, pada bagian *feature learning* terdapat lapisan yang bermanfaat untuk menerima input berupa gambar, kemudian melakukan proses hingga menghasilkan output data. Lapisan yang ada pada proses ini terdiri dari lapisan konvolusi dan lapisan pooling, dimana setiap proses lapisan tersebut

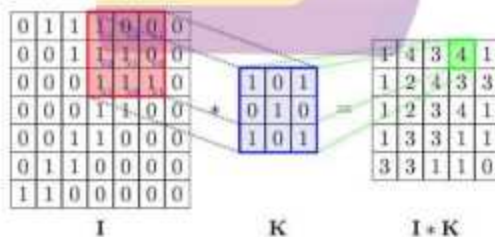
akan menghasilkan *feature maps* berupa angka-angka yang merepresentasikan gambar untuk kemudian diteruskan pada bagian lapisan *classification*. Selanjutnya pada lapisan *classification*, terdiri dari beberapa lapisan yang berisi *neuron* yang terkoneksi penuh (*fully connected*) dengan lapisan lain. Lapisan ini menerima input dari output layer bagian *feature learning* yang kemudian diproses pada *flatten* dengan tambahan beberapa *hidden layer* pada *fully connected* hingga menghasilkan output berupa akurasi klasifikasi dari setiap kelas (Azmi et al., 2023).

### 2.3.5. Layer Convolutional Neural Network

Arsitektur *Convolutional Neural Network* (CNN) terdiri dari sejumlah lapisan atau biasa disebut *multi-building blocks* (Alzubaidi et al., 2021). Adapun setiap lapisan dalam arsitektur CNN termasuk fungsinya akan dijelaskan secara rinci di bawah ini.

#### 3.3.5.1 Convolutional Layer

*Convolutional layer* terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan tinggi dan panjang (pixels). Tiap filter memiliki nilai dan nilai dari filter inilah yang menjadi parameter yang akan di-update dalam proses *learning* (Charli et al., 2020).

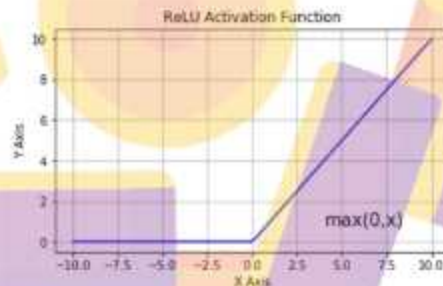


Gambar 2.2. Convolutional Layer (Azmi et al., 2023)

Proses konvolusi menggunakan *kernel* dan *stride*, proses konvolusi ini adalah proses kombinasi antara dua buah matriks yang berbeda untuk menghasilkan suatu nilai matriks yang baru (Azmi et al., 2023). Lapisan ini berfungsi sebagai lapisan pondasi dan berguna untuk mengekstraksi fitur selama entri data (Sabrina, 2022).

### 3.3.5.2 Active Function Rectified Linear Unit (ReLU)

Setelah model melakukan proses konvolusi dalam *convolution layer* maka layer berikutnya adalah ReLU layer. Fungsi aktivasi ReLU bertujuan untuk membatasi keluaran yang dihasilkan dari lapisan pooling dengan cara memetakan nilai negatif menjadi nol, sehingga dapat meningkatkan kecepatan pelatihan, sementara nilai positif disimpan dalam fungsi ReLU (Nihayatul Husna et al., 2022).



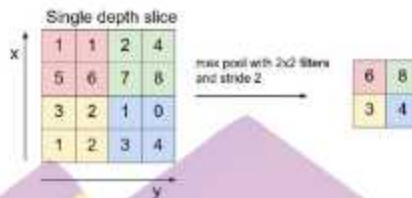
Gambar 2.3. ReLU Function (Nihayatul Husna et al., 2022)

### 3.3.5.3 Pooling Layer

*Pooling layer* adalah lapisan dimana proses pengurangan ukuran citra terjadi setelah terjadinya kontaminasi lapisan. Pada layer *max pooling*, pooling membagi keluaran jaringan konvolusional menjadi beberapa grid kecil, dimana



nilai maksimum tiap grid akan dimasukkan ke dalam matriks citra yang telah direduksi oleh *pooling layer* (Nihayatul Husna et al., 2022).



Gambar 2.4. *Pooling Layer* (Nihayatul Husna et al., 2022)

Kotak yang berwarna merah, hijau, kuning dan biru pada sisi kiri merupakan kelompok kotak yang akan dipilih nilai maksimumnya. Sehingga hasil dari proses tersebut dapat dilihat pada kumpulan kotak disebelah kanannya.

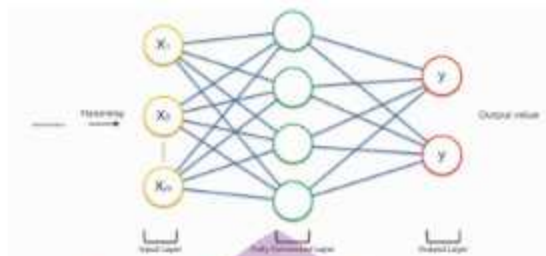
#### 3.3.5.4 Flatten Layer

Lapisan yang disebut *flatten layer* digunakan untuk mengubah menjadi tensor satu dimensi. Hal ini dilakukan agar fitur vektor dapat dibuat dengan menyelaraskan struktur dan memanfaatkan output dari lapisan konvolusi sebelumnya saat menerapkan operasi pembentukan ulang kerangka *tensorflow*. Untuk kategorisasi, layer yang terhubung hanya dapat menggunakan satu layer (Ode & Sagala, 2022).

#### 3.3.5.5 Fully Connected Layer

Lapisan yang mirip dengan jaringan saraf tiruan, menghubungkan setiap aktivitas neuron dari lapisan sebelumnya ke setiap neuron di lapisan berikutnya (Azmi et al., 2023).





Gambar 2.5. *Fully Connected Layer* (Azmi et al., 2023)

### 3.3.5.6 *Softmax Classifier*

Sebuah fungsi di *Softmax* dapat digunakan untuk menentukan kemungkinan setiap kelas target potensial dan akan membantu inputnya (Nihayatul Husna et al., 2022). Keuntungan dari *softmax* adalah rentang probabilitas *output* dengan 0 hingga 1, dan jumlah semua probabilitas akan sama dengan satu. Model multiklasifikasi yang menggunakan fungsi *softmax* akan mengembalikan probabilitas setiap kelas, dengan probabilitas yang tinggi untuk kelas target.

### 2.3.6. *Hyperparameter*

*Hyperparameter* merupakan suatu parameter yang dapat digunakan untuk mengatur dan mengontrol sebuah metode pemodelan sehingga dapat menghasilkan performa prediksi yang lebih baik. *Hyperparameter* penting digunakan dan didefinisikan karena dapat mempengaruhi keandalan dan performa dari suatu teknik pemodelan (Aristyanto & Kurniawan, 2021). Alasan untuk melakukan optimisasi *hyperparameter* adalah untuk memastikan bahwa model dapat belajar dengan sebaik mungkin dari data yang tersedia, dan memberikan prediksi yang akurat pada data yang belum pernah dilihat sebelumnya dan juga dengan optimisasi *hyperparameter*, dapat menemukan keseimbangan yang tepat untuk menghindari

*overfitting* (model terlalu menyesuaikan diri dengan data pelatihan) dan *underfitting* (model tidak cukup mempelajari data).

Berikut adalah penjelasan dari masing-masing *hyperparameter* yang digunakan.

#### 1. *Batch Size*

*Batch size* adalah jumlah sampel data yang digunakan dalam satu *batch* di antara pembaruan bobot model. Pemilihan ukuran *batch* mungkin dipengaruhi oleh banyaknya jumlah kumpulan data yang digunakan selama fase pelatihan. Waktu pelatihan dan performa model dapat dipengaruhi secara signifikan oleh ukuran *batch* (Hariz et al., 2022).

#### 2. *Learning Rate*

*Learning rate* merupakan parameter yang mengontrol seberapa cepat atau lambat model mengambil informasi baru selama pelatihan. Hal ini berdampak signifikan terhadap skor akurasi yang diperoleh dari pengeboran model (Kurnia & Wibowo, 2021).

#### 3. *Epoch*

*Epoch* merupakan gambaran numerik dari satu siklus penuh proses pelatihan di seluruh kumpulan data. Skor akurasi meningkat dengan meningkatnya nilai *epoch* dan menurun dengan menurunnya nilai kerugian. Jika nilai akurasi dan kerugian tidak berubah, proses dihentikan berubah secara dramatis (Kurnia & Wibowo, 2021).

#### 4. *Optimizer*

Optimasi model dalam penelitian ini dilakukan menggunakan Adam (*Adaptive Moment Estimation*), yang merupakan salah satu optimizers yang efektif dalam pelatihan jaringan saraf tiruan (*neural networks*). Adam menggabungkan keuntungan dari algoritma optimasi RMSProp dan momentum stokastik.

Mekanisme Adam menggunakan *adaptive learning rates*, yang berarti setiap parameter memiliki *learning rate* yang berbeda. Hal ini memungkinkan optimasi yang lebih efisien karena dapat menyesuaikan *learning rate* berdasarkan parameter yang berbeda.

### 2.3.7. MobileNetV2

Sebagai model pengembangan MobileNetV1, MobileNetV2 adalah arsitektur CNN untuk deteksi gambar yang memanfaatkan konvolusi yang dalam dan terarah. Dengan parameter yang lebih sedikit, MobileNetV2 mengungguli MobileNetV1 dalam hal akurasi. MobileNetV2 menambahkan dua fitur baru yaitu *linear bottlenecks* dan *shortcut connections between bottlenecks* (Hariz et al., 2022). Perbedaan utama antara arsitektur CNN dan desain MobileNet sering kali terletak pada penggunaan lapisan konvolusi dengan filter ketebalan yang sesuai dengan ketebalan gambar inputan. Tabel 2.2 menunjukkan arsitektur MobileNetV2.

Tabel 2.2. Arsitektur MobileNetV2

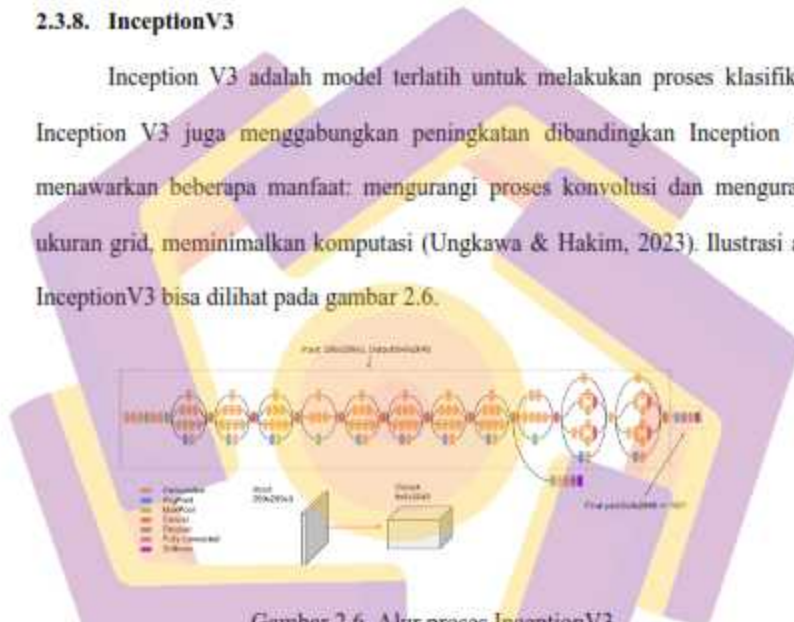
Input	Operator	<i>t</i>	<i>c</i>	<i>n</i>	<i>s</i>
224 <sup>2</sup> x 3	Conv2d	-	32	1	2
112 <sup>2</sup> x 32	Bottleneck	1	16	1	1
112 <sup>2</sup> x 16	Bottleneck	6	24	2	2
56 <sup>2</sup> x 24	Bottleneck	6	32	3	2
28 <sup>2</sup> x 32	Bottleneck	6	64	4	2
14 <sup>2</sup> x 64	Bottleneck	6	96	3	1

Tabel 2.2. Arsitektur MobileNetV2 (Lanjutan)

$14^2 \times 96$	Bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

### 2.3.8. InceptionV3

Inception V3 adalah model terlatih untuk melakukan proses klasifikasi. Inception V3 juga menggabungkan peningkatan dibandingkan Inception V2, menawarkan beberapa manfaat: mengurangi proses konvolusi dan mengurangi ukuran grid, meminimalkan komputasi (Ungkawa & Hakim, 2023). Ilustrasi alur InceptionV3 bisa dilihat pada gambar 2.6.



Gambar 2.6. Alur proses InceptionV3

### 2.3.9. Metode Evaluasi

Metode evaluasi yang digunakan dalam penelitian ini adalah *Confusion Matrix*. *Confusion matrix* merupakan suatu alat ukur berbentuk matrik  $2 \times 2$  yang digunakan untuk memperoleh jumlah ketepatan klasifikasi dataset terhadap satu kelas dengan kelas lain. *Confusion matrix* juga mempermudah dalam analisis performa algoritma dikarenakan *confusion matrix* dapat menampilkan informasi berbentuk angka, sehingga bias dilakukan perhitungan rasio keberhasilan

klasifikasi (Fanani, 2020). Metode evaluasi *confusion matrix* dapat dilihat pada tabel 2.3.

Tabel 2.3. Tabel Pengujian *Confusion Matrix*

		Kelas Prediksi	
		Positif	Negatif
Kelas Sebenarnya	Positif	TP	FN
	Negatif	FP	TN

Berdasarkan table 2.3 hasil evaluasi dari model confusion matrix dapat dijabarkan yakni TP (True Positif) adalah data label positif dan diprediksi dengan benar sebagai data positif. FP (False Positif) merupakan data label negatif, namun diprediksi sebagai data positif. FN (False Negatif) merupakan data label positif, namun diprediksi sebagai data negatif. TN (True Negatif) adalah data label negatif dan diprediksi dengan benar sebagai data negatif.

Keefektifan model klasifikasi yang dihasilkan dapat dievaluasi menggunakan sejumlah parameter dalam metode matriks konfusi ini (Yudianto et al., 2020).

#### Presisi

Besarnya informasi antara nilai sebenarnya dengan hasil prediksi model, sebagaimana ditentukan oleh rumus pada persamaan (1) di bawah ini, merupakan parameter penilaian yang menghitung nilai presisi rata-rata dari hasil klasifikasi.

$$Presisi = \frac{\sum_i^n \frac{TP_i}{TP_i + FP_i}}{n} \quad (1)$$

#### Recall

Kriteria evaluasi diperoleh dari kuantitas data sebenarnya berapa banyak data yang terungkap dalam hasil klasifikasi dengan rumus persamaan (2)

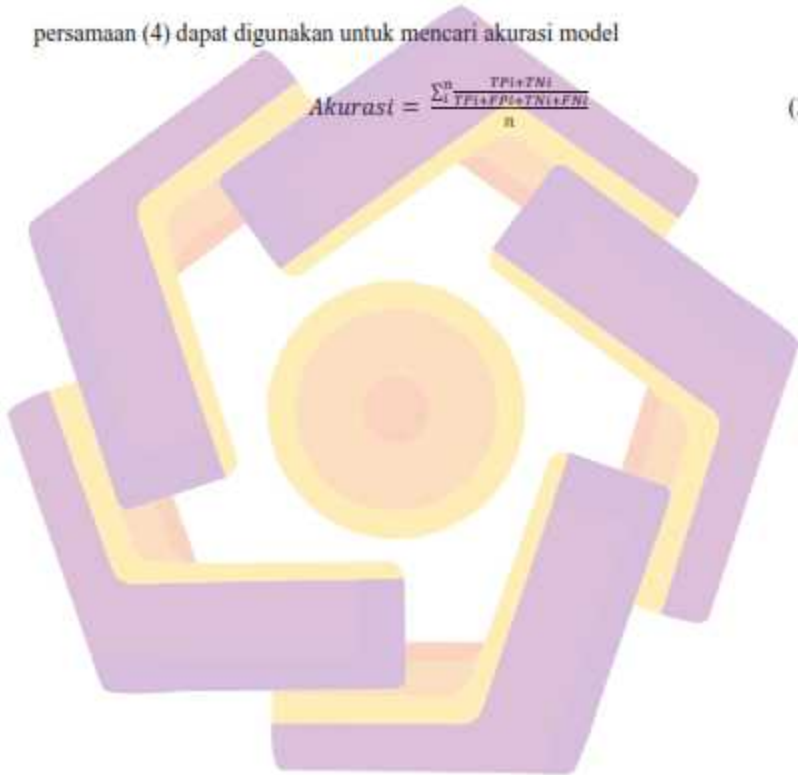


$$Recall = \frac{\sum_i^n \frac{TP_i}{TP_i + FN_i}}{n} \quad (2)$$

Akurasi

Kriteria penilaian untuk menilai presisi model dan tingkat kepercayaan dalam menerapkan kategorisasi pada data baru. Rumus yang ditunjukkan pada persamaan (4) dapat digunakan untuk mencari akurasi model

$$Akurasi = \frac{\sum_i^n \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}}{n} \quad (3)$$



## BAB III

### METODE PENELITIAN

#### 3.1. Jenis, Sifat, dan Pendekatan Penelitian

Penelitian ini menggunakan Penelitian Eksperimen. Penelitian eksperimen adalah metodologi penelitian yang melibatkan variabel yang diteliti dengan cara tertentu. Sedangkan sifat penelitian ini adalah penelitian Eksperimental Komputasi. Penelitian ini menggunakan pendekatan dengan melakukan suatu eksperimen untuk klasifikasi biji kopi sangrai menggunakan metode *Convolutional Neural Network* (CNN).

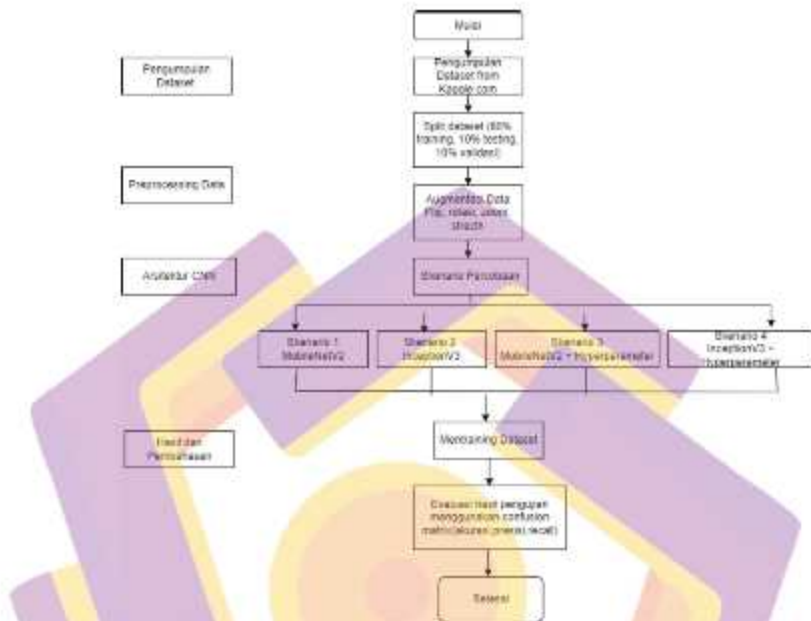
#### 3.2. Metode Pengumpulan Data

Pengumpulan data dilakukan untuk mendapatkan data gambar berupa gambar biji kopi sangrai. Metode pengumpulan data dilakukan dengan cara mengambil data gambar dari *Kaggle.com* (<https://www.kaggle.com/datasets/gpiosenka/coffee-bean-dataset-resized-224-x-224>).

#### 3.3. Metode Analisis Data

Metode analisis yang digunakan dalam penelitian ini adalah analisis kuantitatif menggunakan algoritma *Convolutional Neural Network* (CNN). Setelah data-data terkumpul, kemudian dilakukan *preprocessing* data yaitu split data dan augmentasi data lalu selanjutnya dimasukkan ke dalam model arsitektur MobileNetV2 dan InceptionV3.

### 3.4. Alur Penelitian



Gambar 3.1. Alur Penelitian

Pada penelitian ini terdiri dari tahapan tahapan seperti pada gambar 3.1 dengan rincian alur sebagai berikut.

#### 1. Pengumpulan dataset

Setelah membaca beberapa referensi yang relevan, tahap selanjutnya adalah proses pengumpulan data berupa gambar biji kopi dari sumber *Kaggle.com* (<https://www.kaggle.com/datasets/gpiosenka/coffee-bean-dataset-resized-224-x-224>). Dalam penelitian ini jenis biji kopi yang digunakan adalah Arabika coffee dengan tingkatan berwarna hijau dengan nama Green Laos Typica Bolaven, tingkatan terang dengan nama Light Laos Typica Bolaven, kemudian tingkatan

medium dengan nama Doi Chaang, dan yang terakhir tingkatan dark dengan nama Brazil Cerrado.

## 2. *Preprocessing data*

Pada tahap ini, dataset yang didapatkan akan disiapkan untuk diproses oleh sistem. Tahapan *preprocessing* ini mensplit dataset sebanyak 80% untuk data training, 10% untuk data testing dan 10% untuk data validasi, kemudian melakukan augmentasi data.



Gambar 3.2: Tahapan *Preprocessing*

### a. Split dataset

Seperti yang terlihat pada Gambar 3.2 proses split dataset dalam penelitian ini dilakukan dengan membagi dataset berupa data *training* dengan rasio 80%, data *testing* 10% data validasi 10%. Berikut di bawah ini tabel pembagian dataset.

Tabel 3.1. Detail Pembagian Dataset

Jenis Biji Kopi	Pembagian Dataset		
	Training (80%)	Testing (10%)	Validasi (10%)
Green Laos Typica Bolaven	300	50	50
Light Laos Typica Bolaven	300	50	50
Doi Chaang	300	50	50
Brazil Cerrado	300	50	50
Total	1200	200	200

#### b. Augmentasi data

Tahap selanjutnya adalah augmentasi, teknik ini digunakan dalam pengolahan data untuk membuat variasi baru dari data latihan yang ada. Penerapan augmentasi dapat membantu meningkatkan keberagaman data, mengurangi *overfitting*, dan membuat model pembelajaran mesin lebih umum dan mampu menangani data baru yang belum pernah dilihat sebelumnya. Ada 4 jenis augmentasi yang digunakan dalam penelitian ini, diantaranya *flipping*, *rotasi*, *zoom*, dan *stretch*.

*Flipping* adalah teknik augmentasi di mana gambar atau data lainnya diputar secara *horizontal* atau *vertikal* tergantung pada sumbu tertentu. Pemutaran *horizontal* mengubah gambar dari kiri ke kanan atau sebaliknya, sementara pemutaran *vertikal* mengubah gambar dari atas ke bawah atau sebaliknya.

*Rotasi* merupakan teknik augmentasi di mana gambar atau data lainnya diputar sekitar titik pusatnya dalam berbagai sudut.

*Zoom* berfungsi untuk memperbesar atau memperkecil untuk membuat variasi dari skala gambar. Dengan *zoom in*, bagian dari gambar diperbesar



sehingga detailnya lebih jelas, sedangkan dengan *zoom out*, gambar menjadi lebih kecil.

*Stretch* pada gambar yang telah diubah ukuran atau skalanya secara tidak proporsional, yaitu dengan mengubah lebar atau tinggi gambar dengan faktor tertentu tanpa menjaga aspek rasio aslinya. Dengan *stretch*, dapat menambah variasi pada dataset gambar, yang dapat membantu model pembelajaran mesin untuk lebih mengenali berbagai bentuk dan proporsi dari objek yang ada dalam gambar. *Stretch* merupakan salah satu teknik augmentasi data yang dapat membantu meningkatkan kinerja model dengan memperkaya dataset tanpa perlu menambah gambar baru secara manual.

### 3. Arsitektur *Convolutional Neural Network* (CNN)

Karena jenis penelitian ini bersifat eksperimen maka alur penelitian ini disediakan langkah skenario, sebelum peneliti memulai tahap skenario ada beberapa tahap sebelumnya yang dilakukan. Salah satu tahapnya adalah menyiapkan data gambar dengan preprocessing sebagai data *training* dan evaluasi. Untuk dapat mengetahui performa dari peningkatan kualitas gambar, maka dilakukan 4 skenario menggunakan algoritma *Convolutional Neural Network* (CNN) yaitu MobileNetV2 dan InceptionV3.

### 4. *Training* dataset

Setelah menentukan skenario yang ada yaitu *transfer learning* menggunakan MobileNetV2 dan InceptionV3, kita mulai melakukan percobaan menggunakan dataset gambar biji kopi yang telah dipersiapkan sebelumnya. *Training* data

bertujuan untuk melatih skenario yang ada sehingga saat tahap evaluasi dapat menghasilkan nilai akurasi yang tinggi.

#### 5. Evaluasi data

Tahap ini merupakan tahap untuk melakukan evaluasi (penilaian) menggunakan *confusion matrix* berdasarkan tahap pelatihan menggunakan MobileNetV2 dan InceptionV3. Ada 4 evaluasi yang dilakukan, yaitu mengetahui performa penerapan arsitektur MobileNetV2 dan InceptionV3 dan performa penerapan arsitektur MobileNetV2 dan InceptionV3 dengan *adjusting hyperparameter*, selanjutnya membandingkan akurasi dari arsitektur MobileNetV2 dan InceptionV3 pada peningkatan kualitas gambar.

Untuk mengevaluasi kinerja model, peneliti memantau *loss value* selama proses pelatihan dan validasi pada setiap *epoch*. *Loss value* pada data *training* dan data validasi peneliti catat untuk memonitor kemungkinan *overfitting* atau *underfitting*. Jika *loss value* pada data *training* terus menurun sementara *loss value* pada data validasi mulai meningkat, ini menandakan kemungkinan *overfitting*. Sebaliknya, jika *loss value* pada kedua data *training* dan data validasi tinggi, ini menandakan kemungkinan *underfitting*.

## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1. Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah gambar biji kopi yang terdiri dari empat jenis yaitu, tingkatan berwarna hijau dengan nama Green Laos Typica Bolaven, tingkatan berwarna terang dengan nama Light Laos Typica Bolaven, kemudian tingkatan medium dengan nama Doi Chaang, dan yang terakhir tingkatan dengan warna dark dengan nama Brazil Cerrado. Data biji kopi tersebut didapatkan dari sumber *kaggle.com* (<https://www.kaggle.com/datasets/gpiosenka/coffee-bean-dataset-resized-224-x-224>). Total data berjumlah 1600, dengan kelas tingkatan berwarna hijau dengan nama Green Laos Typica Bolaven berjumlah 400, tingkatan berwarna terang dengan nama Light Laos Typica Bolaven berjumlah 400, kemudian tingkatan berwarna medium dengan nama Doi Chaang berjumlah 400, dan yang terakhir dengan tingkatan berwarna dark dengan nama Brazil Cerrado berjumlah 400. Adapun ukuran gambar data biji kopi memiliki resolusi sebesar 224x224 piksel. Berikut tabel di bawah ini yang menyajikan beberapa sampel gambar dari biji kopi sesuai dengan kelasnya.

Tabel 4.1. Contoh Sampel Biji Kopi

Tingkatan Kelas	Gambar Biji Kopi
Hijau dengan nama Green Laos Typica Bolaven	
Terang dengan nama Light Laos Typica Bolaven	
Medium dengan nama Doi Chaang	
Dark dengan nama Brazil Cerrado	

Setelah gambar biji kopi dikumpulkan, selanjutnya gambar diupload ke dalam *drive* dan ditempatkan di dalam masing masing folder untuk selanjutnya ditampilkan ke dalam *google colab* dengan *library os* dan *drive*. Berikut di bawah ini gambar list kode dari *library* yang digunakan untuk menampilkan dataset.

```
from google.colab import drive
import os
drive.mount('/content/drive')
```

Gambar 4.1. *Library* import dataset

Baris pertama `from google.colab import drive` mengimpor modul *drive* dari pustaka *google.colab*. Baris kedua `import os` mengimpor modul *os*. Baris ketiga

`drive.mount('/content/drive')` memanggil fungsi `mount()` dari modul `drive`, yang digunakan untuk mengaitkan *Google Drive* dengan sesi *notebook Colab* saat ini. Saat menjalankan baris ini, akan diminta untuk memberikan kode otentikasi melalui tautan yang diberikan, dan setelah berhasil *Google Drive* akan terhubung dengan lokasi yang ditentukan (dalam hal ini, `/content/drive`).

#### 4.2. Preprocessing

Setelah data gambar biji kopi di *import*, tahap selanjutnya adalah tahap *preprocessing* dengan melakukan proses *split* data terlebih dahulu.

##### 4.2.1. Split Data

*Split* dataset digunakan untuk membagi dataset menjadi dua atau lebih subset yang berbeda. Dataset dibagi menjadi subset pelatihan (*train set*), subset pengujian (*test set*) dan subset validasi (*validation set*). Pada penelitian ini, peneliti men-split dataset yaitu sebanyak 80% untuk data *training*, 10% untuk data *testing*, dan 10% untuk data validasi. Adapun seperti pada gambar di bawah ini.

```
print(f'Train: {total_x_train}')
print(f'Validation: {total_x_val}')
print(f'Test: {total_x_test}')

Train: 1280
Validation: 160
Test: 160
```

Gambar 4.2. *Split* dataset

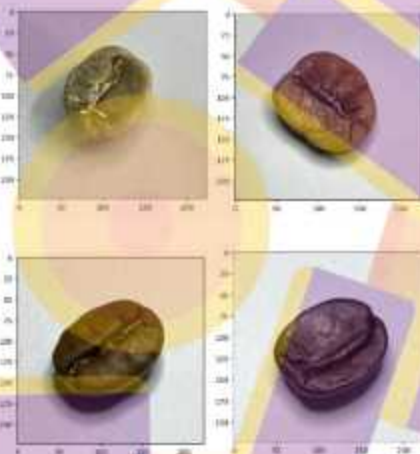
Kode program pada gambar 4.2 menggunakan f-string untuk mencetak informasi tentang jumlah data pada setiap bagian dari data yang disebutkan (*train*, *validation*, dan *test*). Kode `total_trainX`, `total_valX`, dan `total_testX` adalah variabel



yang mewakili jumlah data dalam setiap bagian dari dataset. Fungsi `print()` digunakan untuk mencetak pesan.

#### 4.2.2. Augmentasi

Setelah gambar dimuat, berbagai transformasi yang akan diterapkan pada gambar. Beberapa transformasi yang digunakan diantaranya pemutaran (*flips*), rotasi (*rotations*), memperbesar dan memperkecil gambar secara proporsional (*zoom*), dan stretch.



Gambar 4.3. Sampe Biji Kopi Sangrai

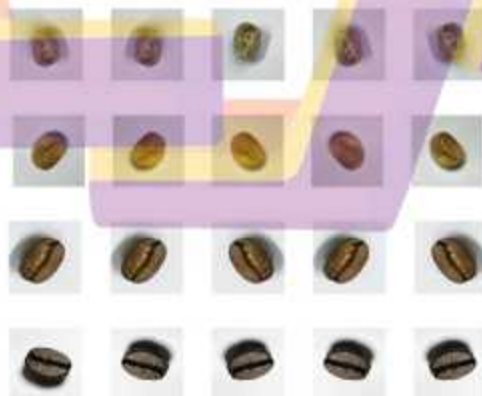
##### 1. Pemutaran (*Flips*)

```
datagen = default_datagen()
datagen.horizontal_flip = True
datagen.vertical_flip = True
plot_augmentation(datagen, data)
```

Gambar 4.4. List kode proses Augmentasi (*Flips*)

Terlihat pada Gambar 4.4 untuk proses pemutaran gambar. Pertama, `default_datagen()` digunakan untuk membuat objek datagen dengan pengaturan `default`. Datagen ini akan digunakan untuk menghasilkan data *augmented*.

Selanjutnya, `datagen.horizontal flip = True` digunakan untuk mengaktifkan augmentasi *flip horizontal*. Ini berarti setiap gambar dalam data akan diputar secara *horizontal* dengan membuat versi *flipped* dari gambar-gambar tersebut. Kemudian, `datagen.vertical flip=True` digunakan untuk mengaktifkan augmentasi *flip vertikal*. Dengan mengatur nilai ini menjadi *true*, setiap gambar dalam data akan diputar secara *vertikal* dengan membuat versi *flipped* dari gambar-gambar tersebut. Terakhir `plot_augmentation(datagen, data)` digunakan untuk memvisualisasikan hasil augmentasi data. Fungsi `plot augmentation` akan menerima objek datagen dan data asli (`data`), dan menghasilkan plot yang menunjukkan perbedaan antara gambar asli dan versi *augmented* dari gambar tersebut. Hasil dari pemutaran gambar bisa dilihat pada gambar 4.5 di bawah ini.



Gambar 4.5. Hasil Augmentasi Pemutaran (*Flips*)

## 2. Rotasi (*Rotations*)

```
datagen = default_datagen()  
datagen.rotation_range = 15  
plot_augmentation(datagen, data)
```

Gambar 4.6. List kode proses Rotasi (*Rotations*)

Pada proses rotasi digunakan `datagen.rotation_range=15`, mengatur rentang rotasi yang akan diterapkan pada data saat dilakukan augmentasi. Dalam hal ini, rentang rotasi adalah 15 derajat. Artinya, setiap gambar dalam data dapat dirotasi secara acak antara -15 derajat hingga +15 derajat. Setelah mengatur parameter augmentasi, kode tersebut kemungkinan besar akan memanggil fungsi `plot_augmentation()` untuk memvisualisasikan perubahan yang dihasilkan oleh augmentasi pada data, seperti pada Gambar 4.7.



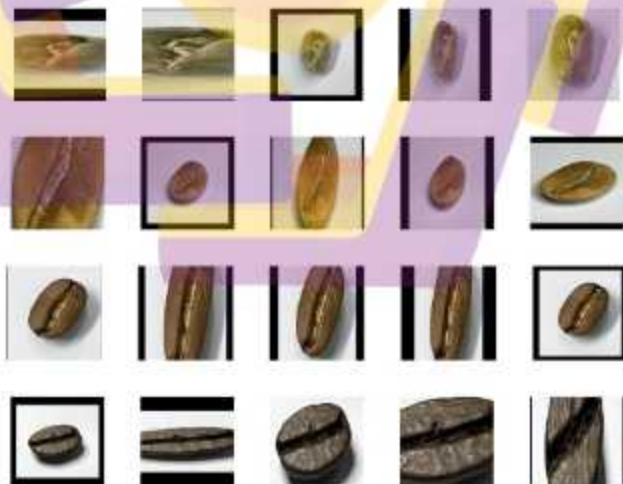
Gambar 4.7. Hasil Augmentasi Rotasi (*rotations*)

### 3. Zoom

```
datagen = default_datagen()
datagen.zoom_range = [0.15, 1.5]
plot_augmentation(datagen, data)
```

Gambar 4.8. List kode proses *Zoom*

Pada proses augmentasi selanjutnya adalah *zoom*, fungsi yang digunakan untuk menginisialisasi objek *datagen* dengan pengaturan *default* yang sesuai untuk augmentasi data adalah *datagen.zoom\_range*. Pada baris ini, rentang *zoom* untuk augmentasi data diatur pada objek *datagen*. Rentang *zoom* didefinisikan dalam bentuk daftar (*min\_zoom*, *max\_zoom*). Rentang *zoom* diatur dari 0.15 hingga 1.5, yang berarti gambar-gambar akan di-*zoom-in* atau di-*zoom-out* dalam rentang tersebut selama proses augmentasi. Terlihat pada gambar 4.9 di bawah ini.



Gambar 4.9. Hasil proses Augmentasi (*Zoom*)

#### 4. *Stretch*

```

datagen = ImageDataGenerator(
    horizontal_flip=True,
    vertical_flip=True
)

plot_augmentation(datagen, stretched_image)

```

Gambar 4.10. List kode proses *Stretch*

Pada proses augmentasi *stretch* terdapat `datagen = ImageDataGenerator` yang berfungsi untuk melakukan augmentasi gambar. Dan juga berguna saat melatih model pembelajaran mesin untuk memperluas dataset dengan membuat variasi dari gambar asli. `horizontal_flip=True` menghasilkan gambar secara acak dibalik secara *horizontal* selama augmentasi. Dan membantu meningkatkan variasi dalam dataset juga membuat model lebih *robust* terhadap perbedaan orientasi gambar. `vertical_flip=True` mengakibatkan gambar secara acak dibalik secara *vertikal* selama augmentasi. Sama seperti `horizontal_flip`, ini juga menambah variasi dan membantu model belajar fitur yang lebih umum. Selanjutnya `plot_augmentation(datagen, stretched_image)` untuk memanggil fungsi `plot_augmentation` dengan `datagen` (instansi `ImageDataGenerator` yang sudah diinisialisasi) dan `stretched_image` (gambar yang sudah di-stretch) sebagai argumen.





Gambar 4.11. Hasil proses Augmentasi *Stretch*

#### 4.3. Proses Klasifikasi menggunakan Convolutional Neural Network

Penelitian ini menggunakan percobaan terhadap 2 arsitektur yang telah teruji tingkat kinerja dari masing-masing arsitektur.

##### 4.3.1. MobileNetV2

Ukuran input shape dasar untuk arsitektur MobileNetV2 adalah [224, 224, 3]. `include_top = false`, `weights = imagenet`, `input_tensor = none`, `pooling = none`, `kelas = 4` dan klasifikasi kelas menggunakan softmax.

```
model = tf.keras.applications.MobileNetV2(
    input_shape=[224, 224, 3],
    include_top=False,
    weights="imagenet",
    input_tensor=None,
    pooling=None,
    classes=4,
    classifier_activation="softmax",
)
```

Gambar 4.12. List kode fungsi Arsitektur MobileNetV2

List kode fungsi arsitektur mobileNetV2 mempunyai penjelasan sebagai berikut. `Input_shape=[224, 224, 3]` menentukan bentuk input dari gambar yang akan diterima oleh model. `Include_top=False`, dalam hal ini nilai `False` menandakan bahwa lapisan penuh tidak akan disertakan seperti contohnya seperti lapisan `Dense`, yang berarti akan membangun bagian atas model sendiri di atas arsitektur dasar. `Weights="imagenet"`, menentukan bobot pra-pelatihan yang akan digunakan. `Input_tensor=None`, tensor opsional yang digunakan sebagai input ke model. Dalam hal ini, nilai default `None` digunakan, yang berarti model akan membuat tensor inputnya sendiri. `Pooling=None`, jenis operasi pooling yang digunakan setelah lapisan konvolusi terakhir. Dalam hal ini, nilai `None` berarti tidak ada operasi pooling yang diterapkan, sehingga lapisan konvolusi terakhir akan menghasilkan keluaran pada resolusi yang sama dengan keluaran dari lapisan konvolusi sebelumnya. `Classes=4`, jumlah kelas output dari model. Di sini, nilai 4 menunjukkan bahwa model output untuk 4 kelas. `Classifier_activation="softmax"` fungsi aktivasi yang akan digunakan pada lapisan klasifikasi. Dalam hal ini, fungsi aktivasi *softmax* akan diterapkan pada output klasifikasi, yang merupakan standar untuk masalah klasifikasi multikelas.

#### 4.3.2. InceptionV3

Selanjutnya gambar memasuki proses pelatihan model pada arsitektur InceptionV3. Dalam penelitian ini menggunakan input shape = [224, 224, 3], include top = True, input tensor = None, pooling = None, weights = imagenet, aktivasi kelas menggunakan softmax, dan kelas = 4.

```

model = tf.keras.applications.InceptionV3(
    include_top=True,
    weights="imagenet",
    input_tensor=None,
    input_shape=[224, 224, 3],
    pooling=None,
    classes=4,
    classifier_activation="softmax",
)

```

Gambar 4.13. List kode Fungsi Arsitektur InceptionV3

Dilihat dari gambar 4.13 adapun penjelasannya adalah sebagai berikut. `Include_top=True`, true berarti menyertakan lapisan atas (*fully connected layer*) dalam model. `Weights="imagenet"` bobot awal model ditetapkan ke bobot yang dilatih di *ImageNet*, sehingga model dapat dimulai dengan rangkaian bobot yang telah dioptimalkan sebelumnya, bukan bobot acak. `Input_tensor=None`, dalam penelitian ini, tidak ada tensor input khusus yang ditentukan sehingga menggunakan input *default* dari model. `Input_shape=[224, 224, 3]`, bentuk masukan model diatur ke [224, 224, 3], yang menunjukkan bahwa gambar masukan akan memiliki resolusi 224x224 piksel dan 3 saluran warna (RGB). `Pooling=None`, opsi pooling di sini menentukan jenis pooling yang dilakukan pada fitur terakhir. None berarti tidak ada pooling tambahan yang dilakukan. `Classes=4`, menentukan jumlah kelas yang akan diprediksi oleh model. `Classifier_activation="softmax"`, menentukan fungsi aktivasi yang akan digunakan di lapisan klasifikasi terakhir. *Softmax* digunakan untuk tugas klasifikasi multi-kelas di mana output merupakan probabilitas dari masing-masing kelas.

#### 4.4. Skenario Percobaan

Untuk menghasilkan hasil yang diharapkan, diperlukan beberapa skenario percobaan yang akan diterapkan. Dalam penelitian ini penulis menentukan 4 skenario percobaan yang dapat dilihat pada Tabel 4.2 berikut.

Tabel 4.2. Skenario Percobaan

No	Skenario	Arsitektur
1	Skenario 1	MobileNetV2
2	Skenario 2	InceptionV3
3	Skenario 3	MobileNetV2 + Hyperparameter (Epoch 50/100) (Batch Size 32/64) (Lr 0.1/0.01/0.001/0.0001/0.00001)
4	Skenario 4	InceptionV3 + Hyperparameter (Epoch 50/100) (Batch Size 32/64) (Lr 0.1/0.01/0.001/0.0001/0.00001)

Empat skenario yang ditunjukkan pada Tabel 4.2 akan diterapkan terhadap kedua arsitektur yang telah penulis sebutkan sebelumnya. Pada skenario percobaan ketiga dan keempat diterapkan *adjusting hyperparameter* yaitu *epoch*, *batch size*, dan *learning rate*. Hal ini ditinjau oleh penelitian (Rivalto et al., 2020) dalam penelitian tersebut hanya melakukan sekali pelatihan terhadap data yang cukup banyak, sehingga sering mengalami penurunan dan juga tidak mengalami peningkatan pada hasil akurasi.

Pada skenario pertama dan kedua ditetapkan penggunaan *epoch* 100, *batch size* 32 dan *learning rate* 0.001. Di penelitian sebelumnya terkait *hyperparameter* pada proses *training* yang dilakukan oleh (Afis Julianto et al., 2022), dengan penggunaan *epoch* 100, *batch size* 32 dan *learning rate* 0.001, mendapatkan nilai akurasi yang optimal. Setiap skenario percobaan di atas akan diujikan dalam pembelajaran data dengan rasio pembagian data *training* 80%, 10%

data *testing* dan 10% untuk data validasi. Dalam penelitian ini penulis ingin mengetahui nilai akurasi tertinggi dari arsitektur MobileNetV2 dan InceptionV3.

#### 4.4.1. Skenario Percobaan Pertama

Setelah tahap *preprocessing* selesai, data dibagi dengan rasio perbandingan 80% data *training*, 10% data *testing*, dan 10% data validasi, maka selanjutnya akan dilakukan proses *training* data dengan menggunakan arsitektur MobileNetV2.

Tabel 4.3. Percobaan skenario 1.

Skenario 1	Epoch	Batch Size	Learning Rate	Nilai Akurasi	Nilai Validasi	Nilai Loss
MobileNetV2	100	32	0.001	93.12%	97.50%	17.19%

Seperti yang terlihat pada Tabel 4.3, percobaan skenario pertama menggunakan arsitektur MobileNetV2 dengan *epoch* 100, *batch* 32, dan *learning rate* 0.001. Mendapatkan nilai akurasi sebesar 93.12% dan nilai validasi sebesar 97.50%.

#### 4.4.2. Skenario Percobaan Kedua

Pada skenario kedua menggunakan arsitektur InceptionV3, dengan penggunaan *epoch* 100, *batch size* 32, dan *learning rate* 0.001.

Tabel 4.4. Percobaan skenario 2

Skenario 2	Epoch	Batch Size	Learning Rate	Nilai Akurasi	Nilai Validasi	Nilai Loss
InceptionV3	100	32	0.001	96.25%	97.50%	23.37%

Seperti yang terlihat pada Tabel 4.4, pada percobaan skenario kedua mendapatkan nilai akurasi sebesar 96.25 dan nilai validasi sebesar 97.50%.



#### 4.4.3. Skenario Percobaan Ketiga

Pada skenario percobaan ketiga menggunakan arsitektur MobileNetV2 ditambah penyetelan *hyperparameter* pada proses pelatihan, seperti kode yang terlihat pada Gambar 4.14.

```
# Define hyperparameter
batch_size = 32/64
epochs = 50/100
optimizer = Adam(learning_rate=0.1
                 0.01
                 0.001
                 0.0001
                 0.00001)
```

Gambar 4.14. Kode penambahan *hyperparameter* pada skenario percobaan

Terlihat pada gambar 4.14 *Batch\_size* mengacu pada jumlah sampel data yang akan diproses sebelum memperbarui bobot model dengan nilai 32 atau 64. Ukuran *batch* 32 menyebabkan pembaruan model lebih sering, sedangkan ukuran *batch* 64 dapat memberikan stabilitas yang lebih baik tetapi dengan frekuensi pembaruan yang lebih sedikit. *Epoch* mengacu pada jumlah kali seluruh dataset yang akan dilalui selama pelatihan yaitu dengan nilai 50 atau 100. *Epoch* yang lebih banyak dapat menghasilkan akurasi yang lebih tinggi, namun juga meningkatkan resiko *overfitting* jika dilakukan secara berlebihan. *Optimizer* mengacu pada algoritma yang digunakan untuk memperbarui bobot model berdasarkan *gradien* yang dihitung dari fungsi loss. *Optimizer* yang digunakan adalah Adam. *Learning\_rate* merupakan parameter dalam pengoptimal yang menentukan besarnya pembaruan bobot pada setiap iterasi. Disini menggunakan 0.1, 0.01, 0.001, 0.0001, dan 0.00001. *Learning rate* yang tinggi (0,1) menghasilkan pembaruan bobot yang lebih besar tetapi dapat menyebabkan model tidak stabil.

*Learning rate* yang rendah (0,00001) menghasilkan pembaruan bobot yang lebih kecil dan dapat menyebabkan konvergensi lebih lambat namun model dapat lebih stabil.

Tabel 4.5. Percobaan skenario 3

Skenario 3	Epoch	Batch Size	Learning Rate	Nilai Akurasi	Nilai Validasi	Nilai Loss
MobileNetV2 + Hyperparameter	Epoch 50	Batch Size 32	Lr 0.1	97.50%	98.12%	19.23%
			Lr 0.01	93.12%	96.88%	38.52%
			Lr 0.001	95.00%	98.75%	3.97%
			Lr 0.0001	96.88%	98.12%	14.99%
			Lr 0.00001	95.62%	99.37%	4.14%
		Batch Size 64	Lr 0.1	95.00%	98.12%	3.42%
			Lr 0.01	97.50%	98.75%	14.49%
			<b>Lr 0.001</b>	<b>98.12%</b>	<b>98.75%</b>	<b>14.92%</b>
			Lr 0.0001	96.25%	98.75%	10.39%
			Lr 0.00001	95.62%	98.12%	29.12%
	Epoch 100	Batch Size 32	Lr 0.1	92.50%	96.88%	10.55%
			Lr 0.01	95.62%	98.12%	30.66%
			Lr 0.001	93.12%	97.50%	17.19%
			Lr 0.0001	96.25%	98.12%	41.01%
			Lr 0.00001	94.38%	98.75%	3.43%
		Batch Size 64	Lr 0.1	96.25%	98.12%	37.69%
			Lr 0.01	96.25%	99.37%	0.94%
			Lr 0.001	96.25%	98.75%	5.33%
			Lr 0.0001	97.50%	99.37%	5.40%
			Lr 0.00001	96.25%	98.75%	7.41%

Jumlah *epoch* yang digunakan dalam skenario ketiga adalah *epoch* 50 dan 100, *batch size* 32 dan 64 dengan *learning rate* dari rentan nilai maksimum 0.1 hingga nilai minimum 0.00001, 10 kali pelatihan menggunakan *epoch* 50/100, dan

10 kali menggunakan *batch size* 32/64. Pada skenario ketiga memperoleh nilai akurasi tertinggi menggunakan *epoch* 50, *batch size* 64 dengan *learning rate* 0.001 yaitu nilai akurasi sebesar 98.12%, dan nilai validasi sebesar 98.75%. Jumlah total proses pelatihan secara keseluruhan dapat dilihat pada Tabel 4.5.

#### 4.4.4. Skenario Percobaan Keempat

Tabel 4.6. Percobaan skenario 4

Skenario 3	Epoch	Batch Size	Learning Rate	Nilai Akurasi	Nilai Validasi	Nilai Loss
InceptionV3 + Hyperparameter	Epoch 50	Batch Size 32	Lr 0.1	93.75%	95.00%	24.43%
			Lr 0.01	96.25%	97.50%	18.23%
			Lr 0.001	66.88%	75.63%	69.52%
			Lr 0.0001	96.25%	96.25%	25.47%
			Lr 0.00001	96.25%	95.63%	25.64%
		Batch Size 64	Lr 0.1	94.38%	95.00%	29.50%
			Lr 0.01	93.12%	96.25%	17.56%
			Lr 0.001	96.25%	96.88%	13.60%
			Lr 0.0001	91.88%	98.75%	12.17%
			Lr 0.00001	93.75%	98.12%	13.74%
	Epoch 100	Batch Size 32	Lr 0.1	96.25%	97.50%	18.70%
			Lr 0.01	95.00%	97.50%	17.84%
			Lr 0.001	96.25%	97.50%	23.37%
			Lr 0.0001	94.38%	96.88%	20.15%
			Lr 0.00001	95.62%	95.63%	15.02%
		Batch Size 64	Lr 0.1	94.38%	94.38%	28.36%
			<b>Lr 0.01</b>	<b>96.88%</b>	<b>97.50%</b>	<b>13.23%</b>
			Lr 0.001	93.12%	95.63%	29.15%
			Lr 0.0001	95.62%	97.50%	12.93%
			Lr 0.00001	94.38%	98.12%	11.90%

Proses yang diterapkan pada skenario keempat yaitu menggunakan arsitektur InceptionV3 dengan penambahan *hyperparameter*, sama seperti pada

proses skenario ketiga, menggunakan *epoch* 50 dan 100, *batch size* 32 dan 64 serta penggunaan *learning rate* dari nilai maksimum 0.1 hingga nilai minimum 0.00001. Seperti yang terlihat pada Tabel 4.6 total proses *training* sebanyak 20 kali, 10 kali menggunakan *epoch* 50/100 dan 10 kali menggunakan *batch size* 32/64. Pada skenario keempat mendapatkan nilai tertinggi dari *epoch* 100, *batch size* 64 dan *learning rate* 0.01 sebesar 96.88% untuk nilai akurasi dan 97.50% untuk nilai validasi.

#### 4.5. Proses Pengujian

Penelitian ini menggunakan *confusion matrix* sebagai teknik atau metode dalam menguji kinerja dari model yang telah dibuat. Dalam metode *confusion matrix* ini dapat digunakan dalam mencari nilai akurasi, presisi, dan *recall* dari model klasifikasi yang telah dibuat.

##### 4.5.1. Pengujian Skenario Pertama

Pengujian skenario pertama menggunakan arsitektur MobileNetV2. Hasil pengujian skenario pertama terdapat pada Gambar 4.15 dalam menentukan nilai akurasi, presisi, dan *recall*.



Gambar 4.15. Hasil pengujian MobileNetV2

### 1) Akurasi

Untuk menentukan nilai akurasi model melalui gambar tabel *confusion matrix* dibutuhkan total nilai TP yang dibagi dengan jumlah data latih. Seperti pada gambar 4.11 jumlah nilai TP adalah  $47+41+32+29 = 149$  dan jumlah data latih yang digunakan adalah 160. Sehingga nilai akurasinya adalah  $149/160 = 0.93125$  atau 93.12%.

### 2) Presisi

Rumus yang digunakan untuk menentukan nilai presisi adalah  $TP/(TP+FP)$ . Pada kelas banyak diperlukan perhitungan presisi terhadap masing-masing kelas, kemudian hasil dari perhitungan tersebut dijumlahkan dan dibagi dengan jumlah kelas. Nilai posisinya bisa dilihat pada gambar 4.11.



- a. Nilai presisi kelas (Dark) Brazil Cerrado

$$P(\text{Dark Brazil Cerrado}) = 47/(47+0+0+1) \Rightarrow 47/48 \Rightarrow 0.98.$$

- b. Nilai presisi kelas (Green) Laos Typica Bolaven

$$P(\text{Green Laos Typica Bolaven}) = 41/(41+0+1+0) \Rightarrow 41/42 \Rightarrow 0.98.$$

- c. Nilai presisi kelas (Light) Laos Typica Bolaven

$$P(\text{Light Laos Typica Bolaven}) = 32/(32+0+0+0) \Rightarrow 32/32 \Rightarrow 1.00.$$

- d. Nilai presisi kelas (Medium) Doi Chaang

$$P(\text{Medium Doi Chaang}) = 29/(29+3+0+6) \Rightarrow 29/38 \Rightarrow 0.76.$$

Sehingga nilai presisi untuk skenario pertama adalah  $(0.98+0.98+1.00+0.76)/4 \Rightarrow 0.93$ .

### 3) Recall

Rumus yang digunakan dalam menentukan nilai *recall* adalah  $TP/(TP+FN)$ .

Pada penentuan *recall* ini dilakukan sama dengan penentuan presisi, yaitu diperlukan perhitungan *recall* terhadap masing-masing kelas dan dilakukan proses penjumlahan dan pembagian dengan jumlah kelasnya.

- a. Nilai recall kelas (Dark) Brazil Cerrado

$$R(\text{Dark Brazil Cerrado}) = 47/(47+0+0+3) \Rightarrow 47/50 \Rightarrow 0.94.$$

- b. Nilai recall kelas (Green) Laos Typica Bolaven

$$R(\text{Green Laos Typica Bolaven}) = 41/(41+0+0+0) \Rightarrow 41/41 \Rightarrow 1.00.$$

- c. Nilai recall kelas (Light) Laos Typica Bolaven

$$R(\text{Light Laos Typica Bolaven}) = 32/(32+0+1+6) \Rightarrow 32/39 \Rightarrow 0.82.$$

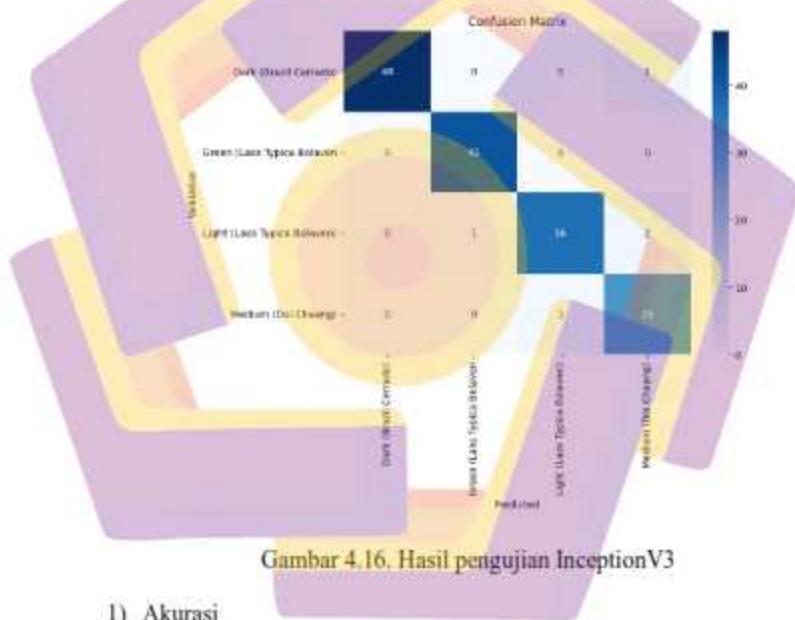
- d. Nilai recall kelas (Medium) Doi Chaang

$$R(\text{Medium Doi Chaang}) = 29/(29+1+0+0) \Rightarrow 29/30 \Rightarrow 0.97.$$

Sehingga nilai *recall* yang diperoleh untuk skenario pertama adalah  $(0.94+1.00+0.82+0.97)/4 \Rightarrow 0.93$ .

#### 4.5.2. Pengujian Skenario Kedua

Pada pengujian skenario kedua menggunakan arsitektur InceptionV3. Hasil pengujian skenario kedua menggunakan arsitektur InceptionV3 terlihat pada Gambar 4.16. Dalam menentukan nilai akurasi, presisi, dan *recall* adalah sebagai berikut.



##### 1) Akurasi

Untuk menentukan nilai akurasi model melalui gambar tabel *confusion matrix* dibutuhkan total nilai TP yang dibagi dengan jumlah data latih. Seperti pada gambar 4.12 jumlah nilai TP adalah  $48+41+36+29 = 154$  dan jumlah data latih yang digunakan adalah 160. Sehingga nilai akurasinya adalah  $154/160 = 0.9625$  atau 96.25%

## 2) Presisi

Rumus yang digunakan dalam menentukan nilai presisi adalah  $TP/(TP+FP)$ . Pada kelas banyak diperlukan perhitungan presisi terhadap masing-masing kelas, kemudian hasil dari perhitungan tersebut dijumlahkan dan dibagi dengan jumlah kelas.

- a. Nilai presisi kelas (Dark) Brazil Cerrado

$$P(\text{Dark Brazil Cerrado}) = 48/(48+0+0+0) \Rightarrow 48/48 \Rightarrow 1.00.$$

- b. Nilai presisi kelas (Green) Laos Typica Bolaven

$$P(\text{Green Laos Typica Bolaven}) = 41/(41+0+1+0) \Rightarrow 41/42 \Rightarrow 0.98.$$

- c. Nilai presisi kelas (Light) Laos Typica Bolaven

$$P(\text{Light Laos Typica Bolaven}) = 36/(36+0+0+1) \Rightarrow 36/37 \Rightarrow 0.97.$$

- d. Nilai presisi kelas (Medium) Doi Chaang

$$P(\text{Medium Doi Chaang}) = 29/(29+2+0+2) \Rightarrow 29/33 \Rightarrow 0.88.$$

Sehingga nilai presisi untuk skenario kedua adalah  $(1.00+0.98+0.97+0.88)/4 \Rightarrow 0.96$ .

## 3) Recall

Rumus yang digunakan dalam menentukan nilai *recall* adalah  $TP/(TP+FN)$ . Pada penentuan *recall* ini dilakukan sama dengan penentuan presisi, yaitu diperlukan perhitungan *recall* terhadap masing-masing kelas dan dilakukan proses penjumlahan dan pembagian dengan jumlah kelasnya.

- a. Nilai recall kelas (Dark) Brazil Cerrado

$$R(\text{Dark Brazil Cerrado}) = 48/(48+0+0+2) \Rightarrow 48/50 \Rightarrow 0.96.$$

- b. Nilai recall kelas (Green) Laos Typica Bolaven

$$R(\text{Green Laos Typica Bolaven}) = 41/(41+0+0) \Rightarrow 41/41 \Rightarrow 1.00.$$

- c. Nilai recall kelas (Light) Laos Typica Bolaven

$$R(\text{Light Laos Typica Bolaven}) = 36/(36+0+1+2) \Rightarrow 36/39 \Rightarrow 0.92.$$

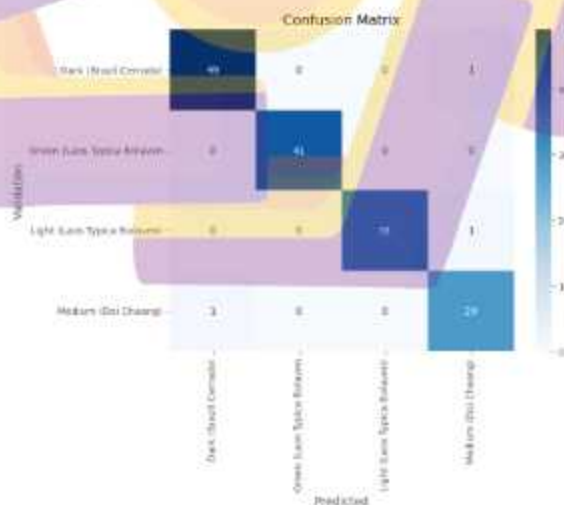
- d. Nilai recall kelas (Medium) Doi Chaang

$$R(\text{Medium Doi Chaang}) = 29/(29+0+0+1) \Rightarrow 29/30 \Rightarrow 0.97.$$

Sehingga nilai *recall* yang diperoleh untuk skenario kedua adalah  $(0.96+1.00+0.92+0.97)/4 \Rightarrow 0.96$ .

#### 4.5.3. Pengujian Skenario Ketiga

Pada pengujian skenario ketiga terdapat 20 kali proses pengujian namun hanya menampilkan hasil pengujian dari nilai presisi, *recall*, dan akurasi tertinggi. Hasil pengujian skenario ketiga menggunakan arsitektur MobileNetV2 yang ditambah *hyperparameter* terlihat pada Gambar 4.17.



Gambar 4.17. Hasil pengujian MobileNetV2+*hyperparameter*

Berikut hasil pengujian arsitektur MobileNetV2 dengan penambahan *hyperparameter*, *epoch*, *batch size*, dan *learning rate*. Dalam menentukan nilai akurasi, presisi, dan *recall* adalah sebagai berikut.

#### 1) Akurasi

Untuk menentukan nilai akurasi model melalui gambar tabel *confusion matrix* dibutuhkan total nilai TP yang dibagi dengan jumlah data latih. Seperti pada gambar 4.13 jumlah nilai TP adalah  $49 + 41 + 38 + 29 = 157$  dan jumlah data latih yang digunakan adalah 160. Sehingga nilai akurasinya adalah  $157/160 = 0.98125$  atau 98.12%.

#### 2) Presisi

Rumus yang digunakan untuk menentukan nilai presisi adalah  $TP/(TP+FP)$ . Pada kelas banyak diperlukan perhitungan presisi terhadap masing-masing kelas, kemudian hasil dari perhitungan tersebut dijumlahkan dan dibagi dengan jumlah kelas.

##### a. Nilai presisi kelas (Dark) Brazil Cerrado

$$P(\text{Dark Brazil Cerrado}) = 49/(49+0+0+1) \Rightarrow 49/50 \Rightarrow 0.98.$$

##### b. Nilai presisi kelas (Green) Laos Typica Bolaven

$$P(\text{Green Laos Typica Bolaven}) = 41/(41+0+0+0) \Rightarrow 41/41 \Rightarrow 1.00.$$

##### c. Nilai presisi kelas (Light) Laos Typica Bolaven

$$P(\text{Light Laos Typica Bolaven}) = 38/(38+0+0+0) \Rightarrow 38/38 \Rightarrow 1.00.$$

##### d. Nilai presisi kelas (Medium) Doi Chaang

$$P(\text{Medium Doi Chaang}) = 29/(29+1+0+1) \Rightarrow 29/31 \Rightarrow 0.93.$$



Sehingga nilai presisi untuk skenario ketiga adalah  $(0.98+1.00+1.00+0.93)/4 \Rightarrow 0.98$ .

### 3) Recall

Rumus yang digunakan dalam menentukan nilai *recall* adalah  $TP/(TP+FN)$ . Pada penentuan *recall* ini dilakukan sama dengan penentuan presisi, yaitu diperlukan perhitungan *recall* terhadap masing-masing kelas dan dilakukan proses penjumlahan dan pembagian dengan jumlah kelasnya.

- a. Nilai recall kelas (Dark) Brazil Cerrado

$$R(\text{Dark Brazil Cerrado}) = 49/(49+0+0+1) \Rightarrow 49/50 \Rightarrow 0.98$$

- b. Nilai recall kelas (Green) Laos Typica Bolaven

$$R(\text{Green Laos Typica Bolaven}) = 41/(41+0+0+0) \Rightarrow 41/41 \Rightarrow 1.00$$

- c. Nilai recall kelas (Light) Laos Typica Bolaven

$$R(\text{Light Laos Typica Bolaven}) = 38/(38+0+0+1) \Rightarrow 38/39 \Rightarrow 0.97$$

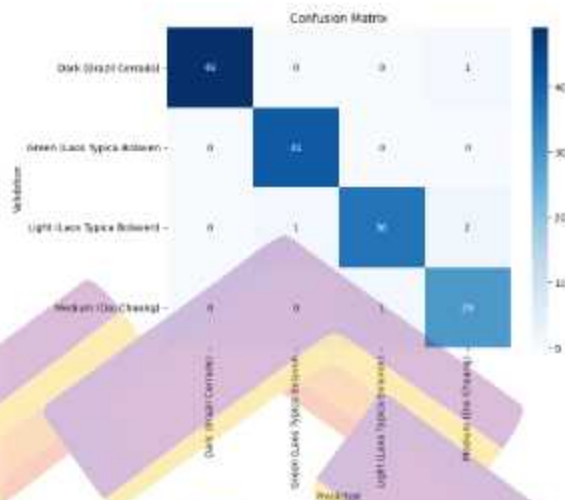
- d. Nilai recall kelas (Medium) Doi Chaang

$$R(\text{Medium Doi Chaang}) = 29/(29+1+0+0) \Rightarrow 29/30 \Rightarrow 0.97$$

Sehingga nilai *recall* yang diperoleh untuk skenario ketiga adalah  $(0.98+1.00+0.97+0.97)/4 \Rightarrow 0.98$ .

#### 4.5.4. Pengujian Skenario Keempat

Pada pengujian skenario keempat juga terdapat 20 kali proses pengujian, namun yang akan ditampilkan adalah hasil pengujian yang mendapatkan nilai akurasi, presisi, dan *recall* tertinggi. Berikut hasil pengujian InceptionV3 dengan penyetelan *hyperparameter* dalam menentukan nilai akurasi, presisi, dan *recall*.



Gambar 4.18. Hasil pengujian InceptionV3+*hyperparameter*

Seperti halnya pada skenario keempat, terjadi 20 kali proses pengujian namun pengujian ditampilkan adalah yang memperoleh nilai tertinggi. Seperti akurasi, presisi, dan *recall* sebagai berikut.

#### 1) Akurasi

Untuk menentukan nilai akurasi model melalui gambar tabel *confusion matrix* dibutuhkan total nilai TP yang dibagi dengan jumlah data latih. Seperti pada gambar 4.13 jumlah nilai TP adalah  $49+41+36+29 = 155$  dan jumlah data latih yang digunakan adalah 160. Sehingga nilai akurasinya adalah  $157/155 = 0.96875$  atau 96.88%.

#### 2) Presisi

Rumus yang digunakan untuk menentukan nilai presisi adalah  $TP/(TP+FP)$ . Pada kelas banyak diperlukan perhitungan presisi terhadap

masing-masing kelas, kemudian hasil dari perhitungan tersebut dijumlahkan dan dibagi dengan jumlah kelas.

- a. Nilai presisi kelas (Dark) Brazil Cerrado

$$P(\text{Dark Brazil Cerrado}) = 49/(49+0+0+0) \Rightarrow 49/49 \Rightarrow 1.00.$$

- b. Nilai presisi kelas (Green) Laos Typica Bolaven

$$P(\text{Green Laos Typica Bolaven}) = 41/(41+0+1+0) \Rightarrow 41/42 \Rightarrow 0.98$$

- c. Nilai presisi kelas (Light) Laos Typica Bolaven

$$P(\text{Light Laos Typica Bolaven}) = 36/(36+0+0+1) \Rightarrow 36/37 \Rightarrow 0.97$$

- d. Nilai presisi kelas (Medium) Doi Chaang

$$P(\text{Medium Doi Chaang}) = 29/(29+1+0+2) \Rightarrow 29/32 \Rightarrow 0.91$$

Sehingga nilai presisi untuk skenario keempat adalah  $(1.00+0.98+0.97+0.91)/4 \Rightarrow 0.96$ .

### 3) Recall

Rumus yang digunakan dalam menentukan nilai *recall* adalah  $TP/(TP+FN)$ . Pada penentuan *recall* ini dilakukan sama dengan penentuan presisi, yaitu diperlukan perhitungan *recall* terhadap masing-masing kelas dan dilakukan proses penjumlahan dan pembagian dengan jumlah kelasnya.

- a. Nilai recall kelas (Dark) Brazil Cerrado

$$R(\text{Dark Brazil Cerrado}) = 49/(49+0+0+1) \Rightarrow 49/50 \Rightarrow 0.98$$

- b. Nilai recall kelas (Green) Laos Typica Bolaven

$$R(\text{Green Laos Typica Bolaven}) = 41/(41+0+0+0) \Rightarrow 41/41 \Rightarrow 1.00$$

- c. Nilai recall kelas (Light) Laos Typica Bolaven

$$R(\text{Light Laos Typica Bolaven}) = 36/(36+0+1+2) \Rightarrow 36/39 \Rightarrow 0.92$$

d. Nilai recall kelas (Medium) Doi Chaang

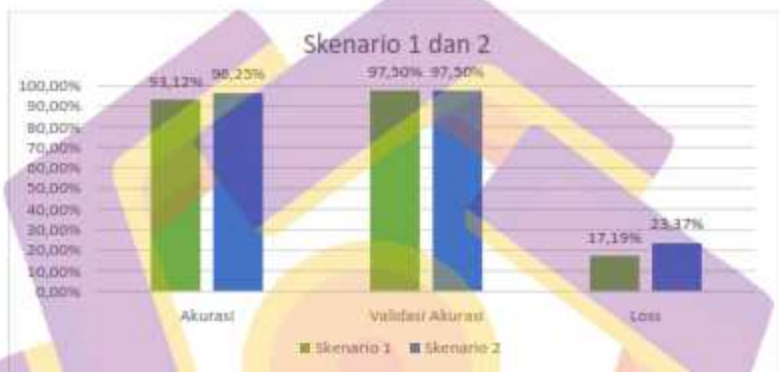
$$R(\text{Medium Doi Chaang}) = 29/(29+0+0+1) \Rightarrow 29/30 \Rightarrow 0.97$$

Sehingga nilai *recall* yang diperoleh untuk skenario keempat adalah  $(0.98+1.00+0.92+0.97)/4 \Rightarrow 0.97$ .

Berdasarkan tabel *confusion matrix*, hasil pengujian yang dilakukan terhadap keseluruhan skenario masih mengalami *error detection*, dimana pada skenario 1 jumlah dark brazil cerrado yang terdeteksi salah sebanyak 1, green laos typica bolaven sebanyak 1, tetapi light laos typica bolaven tidak mengalami *error detection*, dan medium doi chaang terdeteksi salah sebanyak 9. Pada pengujian skenario 2, jumlah dark brazil cerrado yang terdeteksi salah tidak ada, green laos typica bolaven sebanyak 1, light laos typica bolaven terdeteksi salah 1, dan medium doi chaang terdeteksi salah sebanyak 4. Selanjutnya, untuk pengujian skenario 3 jumlah dark brazil cerrado yang terdeteksi salah sebanyak 1, green laos typica bolaven tidak mengalami kesalahan deteksi, begitu juga pada light laos typica bolaven tidak mengalami *error detection* dan medium doi chaang terdeteksi salah sebanyak 2. Dan untuk pengujian yang terakhir yaitu pengujian skenario 4, jumlah dark brazil cerrado tidak ada yang terdeteksi salah, green laos typica bolaven mengalami kesalahan deteksi berjumlah 1, begitu juga pada light laos typica bolaven mengalami *error detection* sebanyak 1, dan medium doi chaang terdeteksi salah sebanyak 3. Jika ditinjau kembali tinjauan literatur, biji kopi sangrai brazil cerrado, laos typica bolaven, dan doi chaang memiliki kesamaan bentuk dan warna.

#### 4.6. Evaluasi Hasil Penelitian

Setelah dilakukan pengolahan data (*preprocessing*) dan proses pembelajaran model terhadap skenario percobaan, diperoleh beberapa model klasifikasi sebanyak 4 skenario yang digunakan. Hasil percobaan keseluruhan skenario disajikan dalam grafik persentase, berikut di bawah ini.



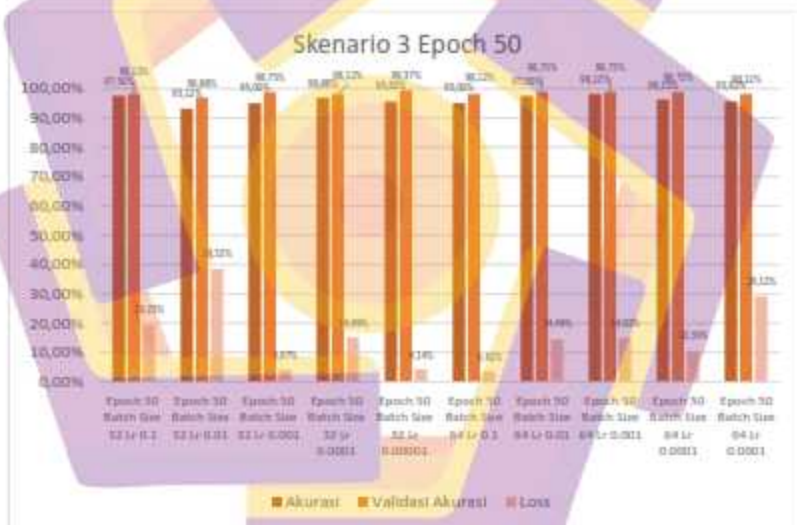
Gambar 4.19. Grafik persentase hasil percobaan skenario 1 dan 2

Dapat dilihat pada grafik persentase hasil percobaan skenario 1 dan 2 pada Gambar 4.19 dimana skenario 1 menggunakan MobileNetV2 dan skenario 2 menggunakan Inception V3. Penggabungan hasil percobaan skenario 1 dan 2 dalam grafik persentase dikarenakan pada percobaan skenario 1 dan 2 menggunakan *epoch*, *batch size* serta nilai *lr* yang sama, yaitu *epoch* 100, *batch size* 32 dan *lr* 0.001. Pada skenario 1 menggunakan MobileNetV2, memperoleh nilai akurasi sebesar 93.12%, sedangkan pada skenario 2 menggunakan InceptionV3 mengalami kenaikan akurasi yaitu 96.25% dan validasi akurasi sama-sama sebesar 87.50% namun pada nilai *loss*, skenario 2 mengalami peningkatan dari skenario 1 yaitu



sebesar 23.37%, sedangkan pada skenario 1 memperoleh nilai *loss* lebih kecil yaitu sebesar 17.19%.

Pada skenario percobaan 3 menggunakan MobileNetV2 dengan *adjusting hyperparameters* pada proses *training*, yaitu *epoch* 100/50, *batch size* 32/64, dan *lr* dengan rentan nilai dari maksimum 0.1 sampai dengan nilai minimum 0.00001. Terdapat 20 kali proses percobaan dimana 10 kali menggunakan *epoch* 100/50 dan 10 kali menggunakan *batch size* 32/64, sehingga dalam hal ini penyajian hasil persentase percobaan pada skenario 3 disajikan dalam 2 grafik yang berbeda.

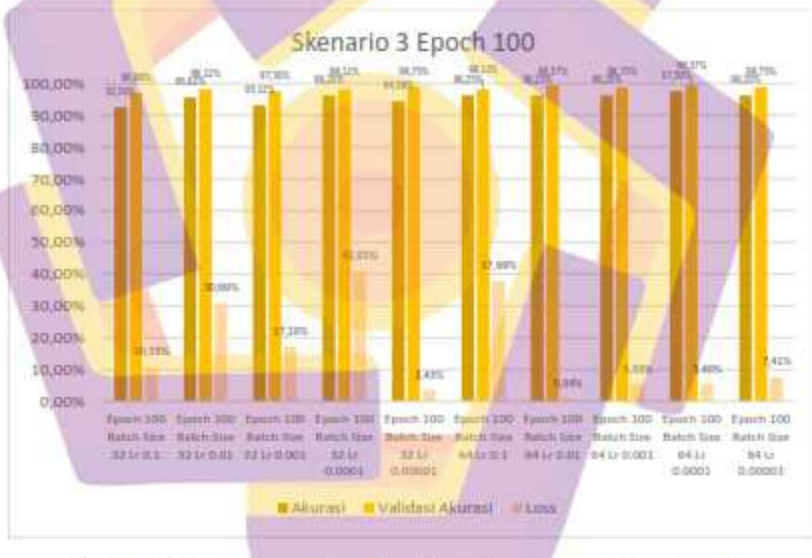


Gambar 4.20. Grafik persentase hasil percobaan skenario 3 *epoch* 50

Dapat dilihat pada Gambar 4.20 grafik persentase skenario percobaan 3 menggunakan MobileNetV2 + hyperparameter *epoch* 50 memperoleh nilai akurasi tertinggi pada penggunaan *batch size* 64 dan *lr* 0.001 yaitu sebesar 98.12% dengan nilai validasi 98.75% dan nilai *loss* sebesar 14.92%, sedangkan nilai akurasi

terendah diperoleh pada penggunaan *batch size* 32 lr 0.01 dengan nilai akurasi sebesar 93.12%, nilai validasi sebesar 96.88% dan nilai *loss* 38.52%.

Pada skenario 3 penggunaan *epoch* 100 seperti yang terlihat pada Gambar 4.21 memperoleh nilai akurasi tertinggi pada *batch size* 64 dan lr 0.0001 dengan akurasi sebesar 97.50%, nilai validasi sebesar 99.37% dan nilai *loss* sebesar 5.40%, sedangkan hasil akurasi terendah diperoleh pada penggunaan *batch size* 32 lr 0.1 memperoleh nilai akurasi 92.50%, validasi akurasi 96.88%, serta nilai *loss* sebesar 10.55%.



Gambar 4.21. Grafik persentase hasil percobaan skenario 3 *epoch* 100

Jika dilihat hasil keseluruhan percobaan pada skenario 3 menggunakan MobileNetV2 ditambah *hyperparameter*, nilai tertinggi diperoleh dengan penggunaan *epoch* 100 *batch size* 64 dan lr 0.0001.

Pada skenario percobaan 4 menggunakan InceptionV3 juga diterapkan *adjusting hyperparameters*, yaitu *epoch* 100/50, *batch size* 32/64, dan *lr* dengan rentan nilai dari maksimum 0.1 sampai dengan nilai minimum 0.00001. Terdapat 20 kali proses percobaan dimana 10 kali menggunakan *epoch* 100/50 dan 10 kali menggunakan *batch size* 32/64, sehingga dalam hal ini penyajian hasil persentase percobaan pada skenario 4 disajikan dalam 2 grafik yang berbeda.



Gambar 4.22. Grafik persentase hasil percobaan skenario 4 epoch 50

Dapat dilihat pada Gambar 4.22, Grafik persentase hasil percobaan skenario 4 epoch 50 mengalami nilai yang naik turun baik akurasi, validasi serta nilai loss. Dimana nilai tertinggi diperoleh pada penggunaan *batch size* 32 lr 0.01, 0.0001, dan lr 0.00001 yaitu dengan hasil akurasi sebesar 96.25%, nilai validasi pada lr 0.01 sebesar 97.50%, lr 0.0001 sebesar 96.25% dan pada lr 0.00001 sebesar 95.63%, nilai loss sebesar 18.23% pada lr 0.01, 25.47% pada lr 0.0001, dan 25.64% pada lr

0.00001. Serta pada *batch size* 64 yaitu dengan lr 0.001 mendapatkan nilai sebesar 96.25% untuk akurasi, nilai validasi 96.88%, dan nilai *loss* 13.60%. Sedangkan nilai terendah diperoleh dengan penggunaan *batch size* 32 lr 0.001 dengan nilai akurasi sebesar 66.88%, nilai validasi sebesar 75.63% dan nilai *loss* sebesar 69.52%.



Gambar 4.23. Grafik persentase hasil percobaan skenario 4 epoch 100

Pada Gambar 4.23 merupakan grafik hasil persentase nilai akurasi, validasi akurasi dan *loss* pada skenario 4 epoch 100, terlihat pada gambar grafik tersebut mengalami nilai yang naik turun seperti halnya pada skenario 4 epoch 50 sebelumnya, nilai *loss* terendah diperoleh pada penggunaan *batch size* 64 lr 0.00001, dan nilai *loss* meningkat pada penggunaan *batch size* 64 dan lr 0.001, sedangkan nilai akurasi tertinggi diperoleh pada *batch size* 64 lr 0.01, begitupun nilai validasi diperoleh sebesar 85.52%, namun mengalami penurunan hasil akurasi dan validasi pada penggunaan *batch size* 32 lr 0.1, *batch size* 64 lr 0.1 dan *batch size* 64 lr 0.00001 dengan mendapatkan hasil yang sama yaitu 49.59% untuk nilai akurasi dan 52.19% untuk nilai validasi akurasi. Jika dilihat dari grafik persentase



skenario *epoch* 50 dan 100, maka nilai tertinggi yang diperoleh adalah dari *epoch* 100, *batch size* 64 dan *lr* 0.001.

Dapat disimpulkan berdasarkan grafik persentase hasil percobaan terhadap 2 arsitektur bahwa penerapan *adjusting hyperparameter* terbukti memberikan hasil yang lebih signifikan jika dibandingkan dengan hasil percobaan tanpa *adjusting hyperparameter* memperoleh nilai akurasi sebesar 93.12% untuk MobileNetV2, dan 96.25% untuk InceptionV3, setelah diterapkan *adjusting hyperparameter* pada proses *training* seperti *epoch*, *batch size* dan *lr*, hasil akurasi InceptionV3 meningkat menjadi 96.88% dan hasil akurasi MobileNetV2 meningkat menjadi 98.12%. Ini membuktikan bahwa kinerja *hyperparameter* pada proses *training* dapat memberikan hasil akurasi yang lebih baik, namun perlu digaris bawahi bahwa dalam mendapatkan akurasi tertinggi pada *adjusting hyperparameter*, perlu dilakukan beberapa kali pelatihan dalam mencari kombinasi nilai *hyperparameter* yang terbaik.

Setelah didapatkan fitur pada gambar, kemudian dilakukan proses pengenalan dan pengidentifikasian pada layer *fully connected layer* yang akan menentukan kelas dari gambar biji kopi sangrai. Tahap selanjutnya model tersebut akan dilakukan pengujian menggunakan *confusion matrix* untuk mengetahui nilai nilai true positif, false positif, true negatif dan false negatif untuk selanjutnya digunakan untuk mengevaluasi perhitungan seperti akurasi, recall dan presisi. Berikut adalah hasil keseluruhan pengujian berdasarkan akurasi, presisi dan recall terhadap model klasifikasi dapat dilihat pada Tabel 4.7.



Tabel 4.7. Hasil Pengujian

<b>Skenario 1</b>				
Keterangan	Lr	Akurasi	Precision	Recall
MobileNetV2 Epoch 100 Batch Size 32	0.001	93.12%	0.93	0.93
<b>Skenario 2</b>				
Keterangan	Lr	Akurasi	Precision	Recall
InceptionV3 Epoch 100 Batch Size 32	0.001	96.25%	0.96	0.96
<b>Skenario 3</b>				
Keterangan	Lr	Akurasi	Precision	Recall
MobileNetV2 +Hyperparameter Epoch 50 Batch Size 32	0.1	97.50%	0.98	0.98
	0.01	93.12%	0.94	0.93
	0.001	95.00%	0.95	0.95
	0.0001	96.88%	0.97	0.97
	0.00001	95.62%	0.96	0.96
<b>Epoch 50 Batch Size 64</b>	0.1	95.00%	0.95	0.95
	0.01	97.50%	0.98	0.98
	<b>0.001</b>	<b>98.12%</b>	<b>0.98</b>	<b>0.98</b>
	0.0001	96.25%	0.97	0.96
Epoch 100 Batch Size 32	0.00001	95.62%	0.96	0.96
	0.1	92.50%	0.92	0.92
	0.01	95.62%	0.96	0.96
	0.001	93.12%	0.94	0.93
	0.0001	96.25%	0.94	0.93
Epoch 100 Batch Size 64	0.00001	94.38%	0.95	0.94
	0.1	96.25%	0.96	0.96
	0.01	96.25%	0.96	0.96
	0.001	96.25%	0.96	0.96
	0.0001	97.50%	0.98	0.98
0.00001	96.25%	0.96	0.96	0.96
<b>Skenario 4</b>				
Keterangan	Lr	Akurasi	Precision	Recall
InceptionV3 +Hyperparameter Epoch 50	0.1	93.75%	0.95	0.94
	0.01	96.25%	0.96	0.96
	0.001	66.88%	0.87	0.67

Tabel 4.7. Hasil Pengujian (Lanjutan)

Batch Size 32	0.0001	96.25%	0.96	0.96
	0.00001	96.25%	0.96	0.96
Epoch 50 Batch Size 64	0.1	94.38%	0.95	0.94
	0.01	93.12%	0.94	0.93
	Lr 0.001	96.25%	0.97	0.96
	Lr 0.0001	91.88%	0.93	0.92
	Lr 0.00001	93.75%	0.95	0.94
Epoch 100 Batch Size 32	Lr 0.1	96.25%	0.95	0.96
	Lr 0.01	95.00%	0.95	0.95
	Lr 0.001	96.25%	0.96	0.96
	Lr 0.0001	94.38%	0.95	0.94
	Lr 0.00001	95.62%	0.96	0.96
Epoch 100 Batch Size 64	Lr 0.1	94.38%	0.95	0.94
	<b>Lr 0.01</b>	<b>96.88%</b>	<b>0.96</b>	<b>0.97</b>
	Lr 0.001	93.12%	0.94	0.93
	Lr 0.0001	95.62%	0.96	0.96
	Lr 0.00001	94.38%	0.95	0.94

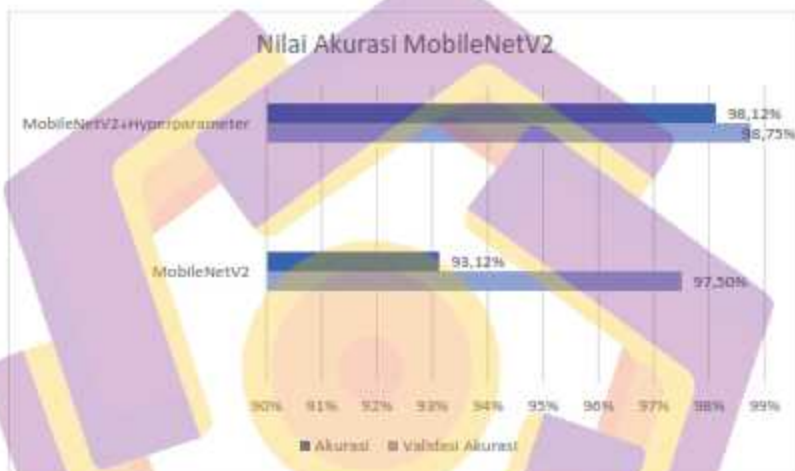
Pengujian diterapkan pada kedua arsitektur yang digunakan, jika dilihat pada Tabel 4.7 hasil pengujian dari keseluruhan skenario, nilai akurasi tertinggi diperoleh pada pengujian skenario ketiga dengan *adjusting hyperparameter* arsitektur MobileNetV2, dengan mendapatkan nilai akurasi sebesar 98,12%, presisi mendapatkan nilai sebesar 98, begitupun *recall* mendapatkan nilai sebesar 98.

Jika melihat hasil pengujian yang didapatkan dapat disimpulkan bahwa kinerja MobileNetV2 lebih baik daripada InceptionV3 jika dilihat dari nilai akurasi, presisi dan *recall* pada tabel pengujian. Dalam fleksibilitas *hyperparameter*, MobileNetV2 lebih sensitif terhadap penyesuaian *hyperparameter* tertentu, yang memungkinkan peningkatan akurasi yang lebih signifikan dibandingkan InceptionV3. Hal ini menunjukkan bahwa dengan pengaturan *hyperparameter* yang tepat, model yang lebih efisien dan ringan seperti MobileNetV2 dapat bersaing, atau

bahkan mengungguli, model yang lebih kompleks seperti InceptionV3 dalam hal akurasi.

#### 4.6.1. Hasil Akurasi MobileNetV2

Berikut adalah hasil perbandingan akurasi arsitektur MobileNetV2, dan InceptionV3 dengan penerapan *adjusting hyperparameter*.

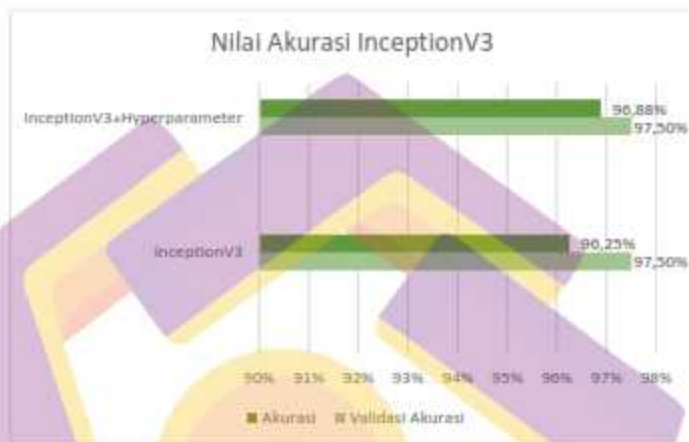


Gambar 4.24. Hasil Arsitektur MobileNetV2

Seperti yang terlihat pada Gambar 4.24 penggunaan MobileNetV2 dalam penelitian ini menghasilkan nilai akurasi 93.12% dan nilai validasi sebesar 97.50%, jika dibandingkan hasil akurasi MobileNetV2 dengan penerapan *hyperparameter* mengalami peningkatan, sebesar 5% dan nilai validasi mengalami peningkatan sebesar 1% yaitu 98.75%. Namun bisa dikatakan hasil akurasi maupun hasil validasi yang dihasilkan, telah mencapai kinerja terbaik pada penggunaan *adjusting hyperparameter*.

#### 4.6.2. Hasil Akurasi InceptionV3

Berikut sajian tabel perbandingan untuk hasil akurasi menggunakan arsitektur InceptionV3.



Gambar 4.25. Hasil Arsitektur InceptionV3

Gambar 4.25 merupakan perbandingan hasil akurasi InceptionV3 dengan InceptionV3 ditambah penyetelan *hyperparameter*. Dapat diperhatikan bahwa hasil akurasi InceptionV3 mengalami kenaikan yang tidak signifikan yaitu dari 96.25% meningkat pada penyetelan *hyperparameter* sebesar 96.88%. Dan pada nilai validasi tetap berada di nilai yang sama yaitu sebesar 97.50%.

#### 4.6.3. Perbandingan Hasil Akurasi Kedua Arsitektur

Pemilihan arsitektur MobileNetV2 dan InceptionV3 pada penelitian ini untuk mengetahui performa kedua model dalam mengklasifikasikan biji kopi sangrai yang hasilnya nanti agar dapat dipertimbangkan digunakan sebaik mungkin untuk penelitian selanjutnya. Parameter akurasi menjadi perbandingan utama dalam

penelitian ini. Nilai akurasi dapat menggambarkan bagus tidaknya suatu model klasifikasi akan digunakan untuk mengklasifikasi data biji kopi sangrai.

Jika dilihat pada gambar grafik hasil arsitektur MobileNetV2 dan InceptionV3 performa terbaik dalam penelitian ini adalah MobileNetV2 dengan penerapan *hyperparameter* pada proses *training* untuk mencari kombinasi terbaik dari penggunaan *epoch*, *batch size*, dan *learning rate*. Dimana penggunaan *epoch* 50, *batch size* 64 dan *learning rate* 0.001 yang menghasilkan nilai akurasi tertinggi, yaitu sebesar 98.12%.

Ditinjau pada penelitian sebelumnya yang dilakukan oleh (Ontoum et al., 2022) mendapatkan akurasi tertinggi pada penggunaan arsitektur MobileNet sebesar 82.00%. Dengan menggunakan *hyperparameter* *batch size* 32, *learning rate* 0.00001, dan jumlah *epoch* 10-40. *Batch size* yang seimbang, *learning rate* yang kecil untuk stabilitas, dan rentang *epoch* yang fleksibel membantu dalam mengembangkan model yang mampu generalisasi dengan baik terhadap data baru tanpa *overfitting*. Jika dibandingkan dengan penelitian saat ini yang dilakukan, *hyperparameter* yang ditetapkan dalam penelitian ini menggunakan *epoch* 50 dan 100, *batch size* 32 dan 64, serta *learning rate* 0.1 hingga 0.00001. Penelitian ini berhasil mendapatkan akurasi tertinggi sebesar 98.12% menggunakan arsitektur MobileNetV2 dengan penggunaan *adjusting hyperparameter* dalam mencari kombinasi terbaik terhadap hasil akurasi yang dihasilkan. Sehingga penggunaan arsitektur MobileNetV2 dapat dipertimbangkan untuk penelitian selanjutnya.

Masalah yang ada dalam penelitian sebelumnya telah diatasi berdasarkan temuan yang disajikan. Upaya telah dilakukan untuk meningkatkan kurangnya



pelatihan model yang sebelumnya tidak memiliki cukup kesempatan untuk mempelajari seluruh data pelatihan. Selain itu, tidak ada eksperimen yang dilakukan untuk menentukan kombinasi nilai *hyperparameter* yang digunakan dalam pelatihan. Akibatnya, model mengalami stagnasi, artinya tidak mengalami peningkatan atau penurunan dalam hasil akurasi. Visualisasi untuk menemukan *hyperparameter tuning* seharusnya menggunakan metode *grid search*, tetapi pada penelitian ini tidak dilakukan karena beberapa pertimbangan praktis dan ilmiah. *Grid search*, meskipun merupakan metode yang sistematis untuk mengeksplorasi ruang *hyperparameter*, memiliki keterbatasan dalam hal efisiensi dan skala, terutama ketika jumlah *hyperparameter* dan rentangnya sangat besar.



## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Berdasarkan analisa hasil pada pengujian yang telah dilakukan terhadap arsitektur MobileNetV2 dan arsitektur InceptionV3 dapat disimpulkan bahwa:

1. Nilai akurasi tertinggi pada penelitian ini terhadap klasifikasi biji kopi sangrai adalah arsitektur MobileNetV2 dengan penerapan *hyperparameter* pada proses *training*, mendapatkan nilai akurasi sebesar 98.12%. Sedangkan pada penggunaan InceptionV3 mendapatkan nilai akurasi sebesar 96.88%.
2. Faktor yang mempengaruhi tingkat akurasi yang dihasilkan dari kedua arsitektur MobileNetV2 dan InceptionV3
  - a. Jika ada banyak kombinasi yang diuji, proses pelatihan dapat menjadi sangat lambat dan membutuhkan waktu serta sumber daya komputasi yang signifikan.
  - b. Kompleksitas arsitektur yang lebih kompleks cenderung memiliki kemampuan untuk menangkap fitur yang lebih abstrak dan kompleks dari data gambar. Kinerja MobileNetV2 lebih baik dari Inception sehingga penggunaan MobileNetV2 dengan *adjusting hyperparameter* bisa dipertimbangkan untuk penelitian berikutnya.

- c. Menerapkan *hyperparameter* dalam proses pelatihan untuk menemukan kombinasi optimal dapat meningkatkan tingkat akurasi dalam proses klasifikasi.

## 5.2. Saran

Berikut ini adalah beberapa saran yang dapat dijadikan pedoman untuk melakukan pengembangan penelitian ini, diantaranya adalah:

1. Mengeksplorasi dan mencoba beberapa arsitektur *Convolutional Neural Network (CNN)* lainnya untuk mencapai hasil optimal.
2. Mengumpulkan data yang berkualitas tinggi penting untuk menghindari ambiguitas dan gangguan dalam pengolahan data.
3. Memperhitungkan penggunaan *hyperparameter* saat melatih model untuk menemukan kombinasi optimal.
4. Disarankan agar mempertimbangkan penggunaan metode *grid search* untuk mendapatkan *hyperparameter tuning*, terutama jika memiliki waktu dan sumber daya yang memadai. *Grid search* dapat memberikan pendekatan sistematis untuk mengeksplorasi ruang *hyperparameter* secara menyeluruh, yang dapat membantu dalam menemukan kombinasi parameter optimal dengan lebih tepat.

## DAFTAR PUSTAKA

### PUSTAKA ELEKTRONIK

- Afis Julianto, Andi Sunyoto, & Ferry Wahyu Wibowo. (2022). Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi. *TEKNIMEDIA: Teknologi Informasi Dan Multimedia*, 3(2), 98–105. <https://doi.org/10.46764/teknimedia.v3i2.77>
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaria, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. In *Journal of Big Data* (Vol. 8, Issue 1). Springer International Publishing. <https://doi.org/10.1186/s40537-021-00444-8>
- Amtate, G., & Teferi, D. (2022). Multiclass classification of Ethiopian coffee bean using deep learning. *SINET: Ethiopian Journal of Science*, 45(3), 309–321. <https://doi.org/10.4314/sinet.v45i3.6>
- Aristyanto, M. Y., & Kumiawan, R. (2021). Pengembangan Metode Neural Machine Translation Berdasarkan Hyperparameter Neural Network. *Seminar Nasional Official Statistics*, 2021(1), 935–946. <https://doi.org/10.34123/semnasoffstat.v2021i1.789>
- Azmi, K., Defit, S., & Sumijan, S. (2023). Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat. *Jurnal Unitek*, 16(1), 28–40. <https://doi.org/10.52072/unitek.v16i1.504>
- Cahyono, B. E., Nugroho, A. T., & Maulinida, I. W. (2023). Klasifikasi Jenis Biji Kopi dengan Menggunakan Metode Gray Level Co-occurrence Matrix (GLCM). *Teknotan*, 16(3), 191. <https://doi.org/10.24198/jt.voll6n3.9>
- Charli, F., Syaputra, H., Akbar, M., Sauda, S., & Panjaitan, F. (2020). Implementasi Metode Faster Region Convolutional Neural Network (Faster R-CNN) Untuk Pengenalan Jenis Burung Lovebird. *Journal of Information Technology Ampera*, 1(3), 185–197. <https://doi.org/10.51519/journalita.volume1.issue3.year2020.page185-197>
- Fanani, M. R. (2020). Algoritma Naïve Bayes Berbasis Forward Selection Untuk Prediksi Bimbingan Konseling Siswa. *Jurnal DISPROTEK*, 11(1), 13–22. <https://doi.org/10.34001/jdpt.v11i1.952>
- Hariz, F. A., Yulita, I. N., & Suryana, I. (2022). Human Activity Recognition Berdasarkan Tangkapan Webcam Menggunakan Metode Convolutional Neural Network (CNN) Dengan Arsitektur MobileNet. *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi*, 3(4), 103–115. <https://doi.org/10.30630/jitsi.3.4.97>



- Hendrawan, Y., Rohmatulloh, B., Ilmi, F. I., Fauzy, M. R., Damayanti, R., Al Riza, D. F., Hermanto, M. B., & Sandra. (2021). AlexNet convolutional neural network to classify the types of Indonesian coffee beans. *IOP Conference Series: Earth and Environmental Science*, 905(1). <https://doi.org/10.1088/1755-1315/905/1/012059>
- Janandi, R., & Cenggoro, T. W. (2020). An Implementation of Convolutional Neural Network for Coffee Beans Quality Classification in a Mobile Information System. *Proceedings of 2020 International Conference on Information Management and Technology, ICIMTech 2020, August*, 218–222. <https://doi.org/10.1109/ICIMTech50083.2020.9211257>
- Kurnia, D., & Wibowo, A. T. (2021). Klasifikasi Spesies Tanaman Kaktus Grafting Berdasarkan Citra Scion Menggunakan Metode Convolutional Neural Network (Cnn). *Jurnal Smart Teknologi*, 8(4), 4171.
- Mangiwa, S., & Maryuni, A. E. (2019). Skrining Fitokimia dan Uji Antioksidan Ekstrak Biji Kopi Sangrai Jenis Arabika (*Coffea arabica*) Asal Wamena dan Moanemani, Papua. *Jurnal Biologi Papua*, 11(2), 103–109. <https://doi.org/10.31957/jbp.925>
- Miftahuddin, Y., & Rais, R. (2024). Classification of Roasted Coffee Beans with Principal Component Analysis and Random Forest. *E3S Web of Conferences*, 484. <https://doi.org/10.1051/e3sconf/202448402009>
- Muammar Khadafi Ichsan. (2021). *Klasifikasi Kualitas Biji Kopi Menggunakan Pengolahan Citra Dan Algoritma Jaringan Syaraf Tiruan*. September 2017, 1–93.
- Nihayatul Husna, I., Ulum, M., Kurniawan Saputro, A., Tri Laksono, D., & Neipa Purnamasari, D. (2022). Rancang Bangun Sistem Deteksi Dan Perhitungan Jumlah Orang Menggunakan Metode Convolutional Neural Network (CNN). *Seminar Nasional Fortei Regional*, 7(Vol. 5 No. 1 (2022): SinarFe7-5 2022), 3–8.
- Nugroho, P. A., Fenriana, I., & Arijanto, R. (2020). Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia. *Algor*, 2(1), 12–21.
- Ode, L., & Sagala, A. S. (2022). *Klasifikasi Cats dan Dogs dengan Metode CNN dalam Fungsi Aktivasi relu , sigmoid , softmax , softplus , softsign , dan selu*. September.
- Ontoum, S., Khemanantakul, T., Sroison, P., Triyason, T., & Watanapa, B. (2022). *Coffee Roast Intelligence*. <http://arxiv.org/abs/2206.01841>
- Rivalto, A., Pranowo, & Santoso, A. J. (2020). Classification of Indonesian coffee types with deep learning. *AIP Conference Proceedings*, 2217(April). <https://doi.org/10.1063/5.0000678>
- Sabrina, A. (2022). *Klasifikasi Penyakit pada Tanaman Kopi Robusta Berdasarkan*



*Citra Daun Menggunakan Convolutional Neural Network*. 9(3), 1919–1927.

- Santoso, B., Zasyah, A. K., & Clarissa, D. A. (2022). Karakteristik produk penyangraian (roasting) biji kopi robusta menggunakan microwave heating. *Jurnal Teknik Kimia*, 28(2). <http://ejournal.ft.unsri.ac.id/index.php/JTK/article/view/926%0Ahttp://ejournal.ft.unsri.ac.id/index.php/JTK/article/download/926/736>
- Saputra, M., Kumiawan, M. P., & Informatika, M. T. (2020). Identifikasi Mutu Biji Kopi Arabika Berdasarkan Cacat. *Jurnal Teknologi Informasi Dan Komunikasi*, 10(1), 27–35.
- Teniro, A., & Zainudin, Z. (2022). Optimalisasi Pengolahan Biji Kopi Dalam Upaya Peningkatan Pendapatan Petani. *Jurnal Pengabdian Pada Masyarakat Indonesia*, 1(3), 24–28. <https://doi.org/10.55542/jppmi.v1i3.229>
- UNGKAWA, U., & HAKIM, G. AL. (2023). Klasifikasi Warna pada Kematangan Buah Kopi Kuning menggunakan Metode CNN Inception V3. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 11(3), 731. <https://doi.org/10.26760/elkomika.v11i3.731>
- Waliyansyah, R. R., & Umar Hafidz Asy'ari Hasbullah. (2021). Comparison of Tree Method, Support Vector Machine, Naïve Bayes, and Logistic Regression on Coffee Bean Image. *EMITTER International Journal of Engineering Technology*, 9(1), 126–136. <https://doi.org/10.24003/emitter.v9i1.536>
- Wijaya, Y. A., Bahtiar, A., Kaslani, & R. N. (2021). Analisa Klasifikasi menggunakan Algoritma Decision Tree pada Data Log Firewall. *Jurnal Sistem Informasi Dan Manajemen*, 9(3), 256–264. <https://ejournal.stmikgici.ac.id/>
- Yudianto, M. R. A., Kusriani, K., & Al Fatta, H. (2020). Analisis Pengaruh Tingkat Akurasi Klasifikasi Citra Wayang dengan Algoritma Convolutional Neural Network. *Jurnal Teknologi Informasi*, 4(2), 182–191. <https://doi.org/10.36294/jurti.v4i2.1319>
- Yunita, R. D. (2022). Implementasi Metode Linear Discriminan Analysis Untuk Klasifikasi Biji Kopi. *Jurnal Teknologi Informatika Dan Komputer*, 8(1), 27–39. <https://doi.org/10.37012/jtik.v8i1.664>