

**PENERAPAN METODE DEEPLARNING UNTUK KLASIFIKASI  
TINGKAT ROASTING BIJI KOPI BERDASARKAN WARNA  
UNTUK EDUKASI BARISTA PEMULA**

**SKRIPSI**

**DISUSUN OLEH**

**JANUAR YUDA SAPUTRA**

**2009020079**



**UMSU**

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

**MEDAN**

**2024**

**PENERAPAN METODE DEEPLARNING UNTUK  
KLASIFIKASI TINGKAT ROASTING BIJI KOPI  
BERDASARKAN WARNA  
UNTUK EDUKASI BARISTA PEMULA**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana  
Komputer (S.Kom) dalam Program Studi Teknologi Informasi pada  
Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas  
Muhammadiyah Sumatera Utara**

**JANUAR YUDA SAPUTRA  
NPM. 2009020079**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA  
MEDAN  
2024**

**LEMBARAN PENGESAHAN**

Judul Skripsi : Penerapan Metode Deep Learning Untuk Klasifikasi  
Tingkat Roasting Biji Kopi Berdasarkan Warna Untuk  
Edukasi Barista Pemula  
Nama Mahasiswa : Januar Yuda Saputra  
NPM : 2009020079  
Program Studi : TEKNOLOGI INFORMASI

Menyetujui  
Komisi Pembimbing



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)  
NIDN. 0127099201

**Ketua Program Studi**



(Fatma Sari Hutagalung, S.Kom., M.Kom.)  
NIDN. 0117019301

**Dekan**



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)  
NIDN. 0127099201

PERNYATAAN ORISINALITAS

PENERAPAN METODE DEEPLARNING UNTUK KLASIFIKASI  
TINGKAT ROASTING BIJI KOPI BERDASARKAN WARNA  
UNTUK EDUKASI BARISTA PEMULA

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 28 Agustus 2024

Yang membuat pernyataan



Januar Yuda Saputra

NPM. 2009020079

**PERNYATAAN PERSETUJUAN PUBLIKASI**  
**KARYA ILMIAH UNTUK KEPENTINGAN**  
**AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Januar Yuda Saputra  
NPM : 2009020079  
Program Studi : Teknologi Informasi  
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul: Penerapan Metode Deep Learning Untuk Klasifikasi Tingkat Roasting Biji Kopi Berdasarkan Warna Untuk Edukasi Barista Pemula

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 28 Agustus 2024

Yang membuat pernyataan



Januar Yuda Saputra

NPM. 2009020079

## KATA PENGANTAR



Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Fatma Sari Hutagalung, S.Kom, M.Kom, Ketua Program Studi Teknologi Informasi
4. Bapak Mhd. Basri, S.Si, M.Kom Sekretaris Program Studi Teknologi Informasi
5. Pembimbing Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom
6. Ayahanda Suwardi Serta Ibunda Misnani atas doa dan kasih sayangnya yang tulus dan tak terhingga kepada penulis.
7. Kepada kakak tercinta Dwi Rahma Angreyani S. Keb dan Indah Junita Sari S. Pd terima kasih atas dukungan yang telah diberikan kepada penulis yang selalu jadi pendengar yang baik serta doa yang penuh untuk penulis dan salah satu alasan penulis untuk menjadi sarjana seperti beliau.
8. Sahabat Persaudaraan serta Teman-teman Seperjuangan terkhususnya Teman KKN BARUS yang telah memberikan motivasi dan perhatiannya.
9. Semua Pihak yang terlibat baik langsung maupun tidak langsung dalam pengerjaan Skripsi ini yang tidak penulis sebutkan satu persatu diucapkan terima kasih.

## ABSTRAK

Deep learning adalah cabang dari kecerdasan buatan yang menggunakan Convolutional Neural Network (CNN) untuk mempelajari representasi data secara bertingkat, terinspirasi dari struktur dan fungsi otak manusia. Teknologi ini memungkinkan komputer belajar langsung dari data tanpa pemrograman eksplisit, dan telah terbukti efektif dalam pengenalan pola kompleks. Penelitian ini mengembangkan metode deep learning untuk mengklasifikasikan tingkat roasting biji kopi berdasarkan warna dari gambar digital. Fokus utama adalah pada penerapan teknik deep learning untuk membantu barista pemula dalam menentukan tingkat roasting dengan akurasi tinggi. Penelitian dilakukan di coffee shop Medan, menggunakan deep learning untuk klasifikasi warna biji kopi, sementara aspek lain seperti aroma tidak menjadi fokus. Hasil penelitian menunjukkan bahwa model CNN yang diterapkan berhasil mencapai akurasi tinggi dan penurunan loss yang signifikan. Penggunaan fungsi aktivasi ReLU dan softmax serta optimizer Adam terbukti efektif. Model ini diharapkan menjadi alat bantu edukatif yang berguna bagi barista pemula dalam menentukan tingkat roasting, serta membantu produsen kopi dalam mengoptimalkan proses roasting.

**Kata Kunci** : Deep Learning, Convolutional Neural Network, Klasifikasi Roasting, Biji Kopi, Akurasi.

## ABSTRACT

*Deep learning is a branch of artificial intelligence that uses a Convolutional Neural Network (CNN) to learn multilevel representations of data, inspired by the structure and function of the human brain. This technology allows computers to learn directly from data without explicit programming, and has proven effective in complex pattern recognition. This research develops a deep learning method to classify the roasting level of coffee beans based on the color of digital images. The main focus is on the application of deep learning techniques to assist novice baristas in determining the roasting level with high accuracy. The research was conducted in Medan coffee shops, using deep learning for coffee bean color classification, while other aspects such as aroma were not the focus. The results showed that the applied CNN model successfully achieved high accuracy and significant loss reduction. The use of ReLU and softmax activation functions as well as Adam's optimizer proved to be effective. This model is expected to be a useful educational tool for novice baristas in determining roasting levels, as well as assisting coffee producers in optimizing the roasting process.*

**Keywords :** *Deep Learning, Convolutional Neural Network, Roasting Classification, Coffee Beans, Accuracy.*



## DAFTAR ISI

<b>LEMBARAN PENGESAHAN.....</b>	<b>iii</b>
<b>PERNYATAAN ORISINALITAS.....</b>	<b>iii</b>
<b>PERNYATAAN PERSETUJUAN PUBLIKASI.....</b>	<b>iv</b>
<b>KARYA ILMIAH UNTUK KEPENTINGAN .....</b>	<b>iv</b>
<b>AKADEMIS.....</b>	<b>iv</b>
<b>KATA PENGANTAR.....</b>	<b>v</b>
<b>ABSTRAK .....</b>	<b>vi</b>
<b>ABSTRACT.....</b>	<b>vii</b>
<b>DAFTAR ISI.....</b>	<b>viii</b>
<b>DAFTAR TABEL.....</b>	<b>xi</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah .....	4
1.4 Tujuan Penelitian .....	5
1.5 Manfaat Penelitian .....	5
<b>BAB II LANDASAN TEORI.....</b>	<b>6</b>
2.1 Pemrosesan Citra Digital .....	6
2.2 Biji Kopi .....	6
2.3 Deep Learning .....	7
2.4 Convolutional Neural Networks (CNNs) dalam Arsitektur Deep Learning...7	
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>10</b>
3.1 Metodologi penelitian dan pengembangan produk.....	10
3.2 DataSet.....	11
3.2.1 Pengumpulan Data .....	12

3.2.2	Data Preprocessing .....	<b>DAFTAR ISI</b> .....	13
3.2.3	Pelatihan Model Convolutional Neural Network.....		13
3.2.4	Testing dan Evaluasi Model .....		14
3.2.5	Implementasi Model dan Pengujian Performa Model pada Aplikasi.....		15
3.3	Kebutuhan Hardware & Software .....		16
3.3.1	Kebutuhan Hardware .....		16
3.3.2	Kebutuhan Software .....		17
	<b>BAB IV HASIL DAN PEMBAHASAN.....</b>		<b>18</b>
4.1	Data Training dan Validation .....		18
4.2	Pembangunan Model CNN.....		19
4.2.1	layer atau neuron .....		19
4.2.2	Fungsi Aktivasi .....		20
4.2.3	Loss Function.....		22
4.2.4	Optimizer.....		24
4.2.5	Metrics.....		25
4.3	Training Model.....		27
4.3.1	Akurasi .....		30
4.3.2	Loss .....		31
4.4	Perhitungan Model .....		32
4.4.1	Perhitungan Augmentasi Data.....		32
4.4.2	Perhitungan Preprocessing Data.....		33
4.4.3	Perhitungan Model .....		35
4.4.4	Compile Model.....		39
4.5	Pengujian Model.....		39
	<b>BAB V PENUTUP.....</b>		<b>41</b>
5.1	Kesimpulan.....		41
5.2	Saran .....		41

## **DAFTAR TABEL**

Tabel 3.1 Kebutuhan Hardware .....	16
Tabel 4.1 Jumlah Data Training Setiap Kelas .....	18
Tabel 4.2 Jumlah Data Validasi Setiap Kelas .....	18

## DAFTAR GAMBAR

Gambar 2.1 Proses Konvolusi.....	8
Gambar 3.1 Proses klasifikasi dataset.....	11
Gambar 3.2 Contoh dataset.....	13
Gambar 3.3 Hasil Preprocessing.....	13
Gambar 4.1 Gambar Hasil Training Model.....	30
Gambar 4.2 Akurasi.....	30

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Deep learning, cabang dari kecerdasan buatan atau Machine Learning, memanfaatkan Convolutional Neural Network (CNN) dalam pembelajaran data secara bertingkat. Metode ini terinspirasi oleh struktur dan fungsi otak manusia, di mana neuron-neuron dalam jaringan saraf otak saling berhubungan dalam pola kompleks—pola serupa diterapkan pula dalam CNN (Santoso & Ariyanto, 2018). Melalui deep learning, komputer mampu mempelajari pola data secara langsung tanpa program eksplisit. Model deep learning dapat mengenali pola-pola kompleks pada data, seperti gambar, teks, dan suara, sehingga memberikan wawasan serta prediksi yang cukup akurat (Iswahyudi et al., 2023). Struktur deep learning terdiri atas tiga lapisan utama: input, tersembunyi (hidden), dan output. Node-node pada lapisan input menyimpan nilai masukan, bersifat tetap selama proses pelatihan, dan hanya mengalami perubahan saat diberikan input baru. Python menjadi pilihan utama dalam pemrograman deep learning berkat dukungan pustaka TensorFlow berbasis Python (Sanjaya Chudhary, 2019).

Beberapa lapisan pada hidden layer memungkinkan eksplorasi kombinasi algoritma yang optimal, yang bertujuan meminimalkan kesalahan hasil. Sementara itu, lapisan output menampilkan hasil perhitungan berdasarkan fungsi aktivasi hidden layer sesuai input. Perkembangan industri kopi global sangat didorong oleh komunitas pecinta kopi, yang membangun ekosistem agar kopi tetap menjadi komoditas unik. Salah satu aspek penting dalam pengelolaan kopi mencakup penentuan grade kopi, yang telah mengikuti standar nasional sejak tahun 1984 melalui sistem penilaian cacat biji kopi (Ardi Gede, 2024). Sebagai minuman yang mendominasi pasar global, kopi

semakin diminati oleh konsumen di seluruh dunia. Proses pemanggangan biji kopi berperan penting dalam menghasilkan cita rasa yang diharapkan oleh penggemar kopi. Pada tahap ini, penentuan tingkat pemanggangan dapat dilakukan melalui identifikasi perubahan warna yang terjadi pada biji kopi selama proses pemanggangan (Yulia Eva, 2024).

Tingkat pemanggangan yang sesuai kadang sulit dikenali oleh barista pemula hanya dari warna biji kopi. Minimnya pengalaman sering kali mengakibatkan ketidakakuratan dalam menentukan tingkat yang diharapkan, yang berdampak pada kualitas rasa kopi. Penerapan metode Deep Learning memberikan solusi potensial bagi tantangan ini. Dalam berbagai aplikasi, metode ini terbukti efektif dalam pengenalan gambar serta klasifikasi data (Yoshua Bengio, 2019). Teknologi kian berperan penting dalam industri kopi demi efisiensi dan peningkatan kualitas. Beberapa penelitian mutakhir menguraikan kontribusi teknologi dalam membantu praktisi kopi, mulai dari algoritma hingga analisis data, yang berfungsi mengoptimalkan proses pemanggangan (Lee, C., & Kim, S., 2023). Penggunaan Deep Learning juga terbukti sangat membantu barista dalam memahami serta mengendalikan proses pemanggangan dengan lebih baik. Riset terdahulu menunjukkan bahwa metode Deep Learning memberikan hasil memuaskan dalam klasifikasi data visual (Smith, J., & Johnson, L., 2023). Kemajuan dalam penelitian ilmiah terkait warna biji kopi dan dampaknya pada tingkat pemanggangan menunjukkan perkembangan signifikan, sehingga mempermudah interpretasi perubahan warna biji kopi selama proses pemanggangan untuk menghasilkan rasa kopi yang diinginkan (Brown, J., & He, K., 2020). Kehadiran berbagai sumber daya digital juga memfasilitasi aksesibilitas informasi mengenai kopi serta proses pemanggangan bagi barista pemula (Maulana Hendra, 2023). Panduan daring dan aplikasi seluler yang menyediakan tutorial terkait pemanggangan kopi kini

menjadi sumber belajar yang berharga untuk mendalami seni pemanggangan (Peñuela-Martínez et al., 2023). Dengan demikian, pemanfaatan metode Deep Learning dalam mengklasifikasikan tingkat pemanggangan biji kopi berdasarkan warna memberikan pendekatan yang menjanjikan guna membantu barista pemula memahami proses ini dengan lebih baik.

## **1.2 Rumusan Masalah**

Berdasarkan permasalahan latar belakang diatas, maka penelitian ini mengembangkan metode menggunakan teknik deep learning untuk mengklasifikasikan tingkat roasting biji kopi berdasarkan warna yang diperoleh dari gambar digital. Tujuannya adalah untuk membekali barista pemula dengan alat yang andal untuk menentukan tingkat roasting, memungkinkan inovasi dalam profil rasa, dan memastikan keakuratan dalam memilih biji kopi yang memenuhi preferensi pelanggan?.

## **1.3 Batasan Masalah**

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini dilakukan disekitar coffeshop Medan.
2. Metode klasifikasi hanya menggunakan deeplearning.
3. Penelitian ini akan lebih berfokus pada aspek teknis klasifikasi tingkat roasting, dan mungkin tidak mencakup semua aspek edukasi barista pemula, seperti teknik penyajian kopi atau aspek sensorik lainnya.
4. Penelitian ini terbatas pada pengembangan sistem klasifikasi tingkat roasting biji kopi yang menggunakan informasi warna sebagai variabel utama. Aspek-aspek lain seperti aroma atau rasa yang mungkin terkait dengan tingkat roasting tidak akan menjadi fokus utama.

#### **1.4 Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Memberikan pemahaman barista pemula tentang tingkat roasting biji kopi berdasarkan ketentuan klasifikasi yang ditentukan pada deep learning sehingga dapat memengaruhi profil rasa kopi pada barista untuk menyajikan pada Pelanggan.
2. Mengidentifikasi hubungan antara kompleksitas proses roasting, termasuk variabel suhu dan waktu, dengan perubahan warna biji kopi menggunakan Deep Learning.
3. Untuk membantu barista pemula dalam memilih dan mengetahui roasting biji kopi berdasarkan preferensi customer berdasarkan dataset sehingga membangun model yang akurat.

#### **1.5 Manfaat Penelitian**

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Metode Deep Learning dapat memberikan tingkat presisi yang tinggi dalam klasifikasi tingkat roasting biji kopi berdasarkan warna. Metode Deep Learning dapat memberikan tingkat presisi yang tinggi dalam klasifikasi tingkat roasting biji kopi berdasarkan warna.
2. Dengan memahami tingkat roasting secara lebih akurat, barista pemula dapat lebih presisi dalam memilih biji kopi untuk menyajikan kopi sesuai dengan preferensi pelanggan.
3. Dengan klasifikasi otomatis tingkat roasting, penelitian ini dapat membantu produsen kopi dalam mengoptimalkan proses roasting mereka.



## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pemrosesan Citra Digital**

Pemrosesan citra digital adalah proses manipulasi dan analisis gambar menggunakan teknik komputasi. Hal ini meliputi berbagai tahapan, termasuk pra-pemrosesan, segmentasi, ekstraksi fitur, dan analisis. Tujuan utama dari pemrosesan citra adalah untuk mendapatkan informasi yang berguna atau memperbaiki kualitas gambar untuk aplikasi tertentu (Nucci et al., 2014).

Tahapan pra-pemrosesan melibatkan pengolahan gambar untuk menghilangkan noise, meningkatkan kontras, dan memperbaiki kecerahan, sehingga memastikan bahwa data citra siap untuk analisis lebih lanjut. Segmentasi mencakup proses memisahkan gambar menjadi bagian-bagian yang berbeda atau objek yang berbeda, sementara ekstraksi fitur melibatkan identifikasi fitur-fitur penting dari gambar, seperti tepi, tekstur, atau pola yang mungkin relevan untuk analisis yang sedang dilakukan (Turap et al., n.d.).

Dalam konteks klasifikasi tingkat pemanggangan biji kopi, pemrosesan citra digital dapat digunakan untuk mempersiapkan data citra biji kopi dengan memastikan bahwa citra-citra tersebut bersih dari noise dan tersegmentasi dengan baik, sehingga memungkinkan model Deep Learning untuk mempelajari pola-pola yang relevan dengan lebih baik.

#### **2.2 Biji Kopi**

Kopi adalah salah satu bahan minuman yang paling banyak dikonsumsi oleh masyarakat Indonesia. Kopi diyakini memiliki nilai ekonomi yang tinggi untuk meningkatkan ekonomi masyarakat dan sebagai sumber devisa negara.

Menurut data Badan Pusat Statistik (BPS) pada tahun 2021, Indonesia memproduksi biji kopi mencapai 774,6 ribu ton yang dimana pada tahun sebelumnya produksi kopi hanya sebesar 753,9 ribu ton, hal ini menunjukkan adanya peningkatan produksi kopi di Indonesia sebesar 2,75%. Selain berperan sebagai produk komoditas, kopi juga memberi pengaruh besar terhadap perkembangan usaha kedai kopi (coffee shop) di Indonesia. Dimana jumlah kedai kopi di Indonesia meningkat tiga kali lipat dari 1083 pada tahun 2016 menjadi 2937 pada tahun 2019 (Undip's Digitalization in Dealings, 2022).

### **2.3 Deep Learning**

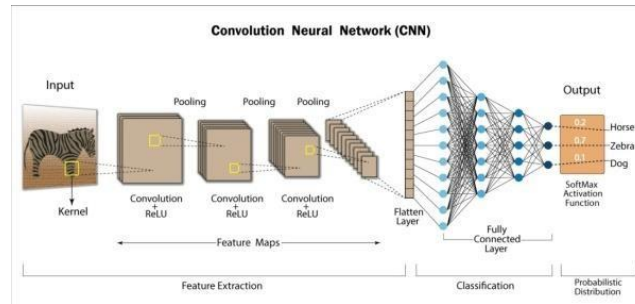
Deep learning adalah sebuah ilmu cabang dari kecerdasan buatan atau *Machine Learning* yang menggunakan *Convolutional Neural Network* untuk mempelajari representasi data secara bertingkat. Ilmu ini terinspirasi bagaimana struktur dan fungsi otak manusia dapat bekerja secara umum, di mana neuron-neuron jaringan saraf otak manusia saling terhubung dalam jaringan yang kompleks dimanapola ini digunakan juga dalam *Convolutional Neural Network* (Santoso & Ariyanto,2018).

Deep learning memungkinkan komputer untuk belajar dari data secara langsung tanpa diprogram secara eksplisit. Model deep learning dapat mengenali pola kompleks dalam gambar, teks, suara, dan data lain untuk menghasilkan wawasan dan prediksi yang cukup akurat (Iswahyudi et al., 2023).

### **2.4 Convolutional Neural Networks (CNNs) dalam Arsitektur Deep Learning**

Convolutional Neural Network (CNN) merupakan salah satu jenis neural network yang biasanya digunakan dalam pengolahan data image (Santoso et al.,

n.d.). Proses konvolusi yang dilakukan dalam metode CNN dapat dilihat pada gambar 1.



**Gambar 2.1** Proses Konvolusi

Gambar 1 di atas menjelaskan alur proses konvolusi dalam Convolutional Neural Network, penjelasan lebih lanjut akan diberikan sebagai berikut:

1. Feature Extraction: Pada tahap ini, dilakukan ekstraksi fitur menggunakan operasi konvolusi, pooling, dan fungsi aktivasi ReLU. Tahap ini melibatkan input data dan kernel (filter).
2. Pooling-Convolution + ReLU: Pertama, input data akan melewati operasi konvolusi dengan filter yang sesuai. Kemudian, hasil konvolusi akan diproses dengan fungsi aktivasi ReLU untuk menghasilkan feature maps. Proses ini dilakukan beberapa kali dengan filter yang berbeda, sehingga menghasilkan beberapa feature maps.
3. Flatten Layer: Setelah tahap ekstraksi fitur, feature maps yang dihasilkan akan diubah menjadi vektor satu dimensi melalui flatten layer. Ini berarti setiap elemen dalam setiap feature map akan disusun menjadi urutan linear dalam vektor.
4. Classification (Fully Connected Layer): Vektor fitur yang telah diflatten akan diteruskan ke fully connected layer. Pada tahap ini, setiap neuron dalam fully connected layer terhubung ke semua neuron dalam

lapisan sebelumnya. Neuron-neuron ini memiliki bobot yang akan diatur selama pelatihan. Aktivasi juga dapat diterapkan pada setiap neuron dalam lapisan ini.

5. Probabilistic Distribution (Output): Akhirnya, output dari fully connected layer akan diolah untuk menghasilkan distribusi probabilistik. Ini dilakukan dengan menggunakan fungsi aktivasi tertentu, seperti softmax, yang menghasilkan probabilitas kelas yang berbeda. Probabilitas ini mewakili prediksi model untuk setiap kelas yang mungkin dalam masukan.

Dengan demikian, proses konvolusi dalam CNN melibatkan ekstraksi fitur menggunakan konvolusi, pooling, dan fungsi aktivasi ReLU. Kemudian fitur-fitur yang dihasilkan diflatten, diolah melalui lapisan fully connected, dan output akhirnya menghasilkan distribusi probabilitas untuk klasifikasi.

Meskipun CNN memiliki kualitas yang menarik dan arsitektur lokalnya relatif efisien, menerapkan CNN dalam skala besar dengan gambar beresolusi tinggi masih membutuhkan biaya yang tinggi. Namun, saat ini GPU telah dikembangkan dengan implementasi konvolusi 2D yang sangat dioptimalkan, sehingga cukup kuat untuk melatih CNN dalam skala besar. Terdapat juga kumpulan data terbaru seperti ImageNet yang menyediakan contoh-contoh berlabel yang cukup untuk melatih model CNN tanpa mengalami overfitting yang parah (Nucci et al., 2014).

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metodologi penelitian dan pengembangan produk**

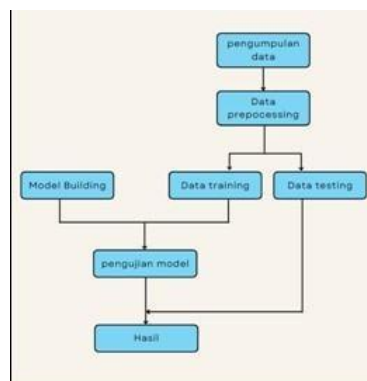
Metode penelitian adalah bentuk pembuktian ilmiah guna memperoleh informasi agar tercapainya tujuan dan manfaat yang diinginkan. Biasanya dalam penelitian diperlukan upaya yang harus didasari dengan bagian ciri dari keilmuan seperti, masuk akal, berdasarkan pengalaman atau ada bukti empiris, dan terstruktur. Penelitian yang dilakukan haruslah dapat diterima oleh akal sehat manusia atau rasional, jadi apa pun langkah-langkah yang diambil dalam penelitian tersebut dapat diterima oleh penalaran seseorang.

Pembuktian akan upaya penelitian ini dapat dilihat dan dipahami dengan penggunaan prosedur tertentu dimana penelitian harus terstruktur dan juga sistematis sehingga penelitian ini memiliki sifat logis atau masuk akal. Machine learning dilakukan pengembangan yang cukup berbeda dengan penelitian dimana pada pengembangan tersebut kita akan menggunakan metode Agile, dimana peneliti memungkinkan merespons perubahan kebutuhan dengan cepat dan fleksibel, membangun aplikasi dan model secara interaktif untuk meningkatkan kualitasnya.

Dari beberapa upaya pada jenis penelitian yang umum digunakan, untuk pendekatan penelitian ini maka dalam penelitian ini menggunakan metode pendekatan pada jenis penelitian R&D ((Research and Development) atau disebut penelitian pengembangan. Metode penelitian yang digunakan ini untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut dan dalam pengembangan produk peneliti akan membangun dalam konsep Agile.

### 3.2 DataSet

Dalam penelitian ini digunakan data citra image biji kopi dengan keseluruhan jumlah data sebanyak 1600 buah. Data tersebut terbagi dalam 4 class yang terdiri dari green (mentah), light, medium, dan dark. Data kemudian akan dibagi menjadi data training dan data testing. Data training berjumlah 1200 buah citra image dan data testing berjumlah 400 buah citra image dengan masing masing dataset memiliki ukuran 255p x 255p. Proses klasifikasi terdiri dari beberapa tahap, tahapan dapat dilihat pada Gambar 2.



**Gambar 3.1** Proses klasifikasi dataset

Proses klasifikasi dataset dijabarkan sebagai berikut:

1. Pengumpulan Data: Pada tahap ini, dilakukan pengumpulan 1600 citra biji kopi dengan ukuran 255p x 255p. Data ini terbagi menjadi 4 kelas yaitu green (mentah), light, medium, dan dark.
2. Data Preprocessing: Tahap ini melibatkan persiapan data sebelum digunakan dalam pembangunan model. Hal ini dapat mencakup langkah-langkah seperti resizing gambar menjadi ukuran yang seragam, normalisasi nilai piksel, dan pengaturan label kelas.
3. Pembangunan Model: Pada tahap ini, dilakukan pembangunan model menggunakan metode Convolutional Neural Network (CNN). Data

training sebanyak 1200 citra biji kopi digunakan untuk melatih model. Model ini akan mempelajari pola-pola dalam data training dan menghasilkan parameter-parameter yang optimal. Data testing sebanyak 400 citra biji kopi digunakan untuk menguji performa model yang telah dilatih.

4. Pengujian Model: Setelah model selesai dilatih, dilakukan pengujian menggunakan data testing. Model akan menerapkan klasifikasi pada citra-citra dalam data testing dan menghasilkan prediksi untuk setiap citra. Akurasi model diukur berdasarkan sejauh mana prediksi model sesuai dengan label kelas yang sebenarnya pada data testing.
5. Hasil: Tahap terakhir adalah presentasi hasil penelitian. Hasil dapat berupa akurasi model, matriks kebingungan yang menunjukkan distribusi prediksi kelas, dan metrik evaluasi lainnya. Hasil ini memberikan gambaran tentang performa model dalam melakukan klasifikasi tingkat roasting biji kopi berdasarkan citra.

Dengan mengikuti alur tahapan yang tertera pada Gambar 2, penelitian ini menggunakan metode Convolutional Neural Network untuk klasifikasi tingkat roasting biji kopi. Tahapan-tahapan tersebut memungkinkan pengumpulan data yang sesuai, pra-pemrosesan data, pembangunan model yang optimal, pengujian performa model, dan presentasi hasil secara terstruktur.

### **3.2.1 Pengumpulan Data**

Proses pengumpulan data dilakukan menggunakan aplikasi machine learning berbasis python. Pada proses pengumpulan data dilakukan pemanggilan data image dari dalam folder penyimpanan desktop ke dalam

notebooks python. Library python yang digunakan dalam proses ini adalah tensorflow, os, pathlib, matplotlib, numpy, dan pandas. Penggunaan library tensorflow sangat dibutuhkan untuk proses machine learning dikarenakan tensorflow memiliki library keras, yaitu library jaringan saraf tiruan tingkat tinggi yang ditulis dalam basis python (Rucci & Casile, 2005).

Contoh dataset yang berhasil dipanggil dapat dilihat pada Gambar 3.2



**Gambar 3.2** Contoh dataset

### 3.2.2 Data Preprocessing

Proses pengolahan citra yang dilakukan selanjutnya adalah data preprocessing. Pada tahap ini dilakukan rescale pada dataset dan mengubah ukuran image menjadi 224p x 224p untuk setiap dataset. Hasil data preprocessing dapat dilihat pada Gambar 3.3.



**Gambar 3.3** Hasil Preprocessing

### 3.2.3 Pelatihan Model Convolutional Neural Network

Dalam pelatihan model untuk klasifikasi roasting kopi menggunakan Convolutional Neural Networks (CNNs), langkah-langkahnya melibatkan



persiapan data, pembuatan model, pelatihan, validasi, optimisasi, evaluasi akhir, dan penyimpanan model. Pertama, data gambar biji kopi atau kopi yang disangrai dipersiapkan. Ini mencakup normalisasi intensitas piksel dan mungkin penerapan augmentasi data untuk meningkatkan variasi dataset.

Kemudian, arsitektur CNN dipilih dan dibangun, termasuk layer konvolusi, pooling, dan fully connected, serta fungsi aktivasi seperti ReLU. Bobot dan bias model diinisialisasi, dan model dilatih dengan meneruskan gambar-gambar melalui jaringan, menghitung loss antara prediksi dan label sebenarnya, dan menyesuaikan parameter dengan algoritma backpropagation.

Selama pelatihan, model dievaluasi menggunakan data validasi untuk memantau kinerja dan mencegah overfitting. Hyperparameter model seperti laju pembelajaran bisa disesuaikan untuk meningkatkan kinerja. Setelah pelatihan selesai, model dievaluasi pada data pengujian untuk mengukur kinerja secara akhir. Jika hasilnya memuaskan, model disimpan untuk penggunaan di masa depan, seperti prediksi pada data baru atau penggunaan dalam tahap produksi.

#### **3.2.4 Testing dan Evaluasi Model**

Dalam pelatihan model untuk klasifikasi roasting kopi menggunakan Convolutional Neural Networks (CNNs), langkah-langkahnya melibatkan persiapan data, pembuatan model, pelatihan, validasi, optimisasi, evaluasi akhir, dan penyimpanan model. Pertama, data gambar biji kopi atau kopi yang disangrai dipersiapkan. Ini mencakup normalisasi intensitas piksel dan mungkin penerapan augmentasi data untuk meningkatkan variasi dataset.

Kemudian, arsitektur CNN dipilih dan dibangun, termasuk layer konvolusi, pooling, dan fully connected, serta fungsi aktivasi seperti ReLU. Bobot dan bias

model diinisialisasi, dan model dilatih dengan meneruskan gambar-gambar melalui jaringan, menghitung loss antara prediksi dan label sebenarnya, dan menyesuaikan parameter dengan algoritma backpropagation. Selama pelatihan, model dievaluasi menggunakan data validasi untuk memantau kinerja dan mencegah overfitting. Hyperparameter model seperti laju pembelajaran bisa disesuaikan untuk meningkatkan kinerja.

Setelah pelatihan selesai, model dievaluasi pada data pengujian untuk mengukur kinerja secara akhir. Jika hasilnya memuaskan, model disimpan untuk penggunaan di masa depan, seperti prediksi pada data baru atau penggunaan dalam tahap produksi.

### **3.2.5 Implementasi Model dan Pengujian Performa Model pada Aplikasi**

Implementasi model dan pengujian performa model dalam klasifikasi roastingan kopi melibatkan langkah-langkah yang terstruktur. Pertama, setelah model CNN dilatih, langkah selanjutnya adalah mengimplementasikan model tersebut dalam lingkungan produksi. Ini melibatkan pembangunan sistem atau aplikasi yang menggunakan model untuk memprediksi roastingan kopi pada gambar-gambar yang diberikan. Kemudian, performa model diuji menggunakan data pengujian yang terpisah. Data ini harus mencakup gambar-gambar yang tidak pernah dilihat oleh model selama pelatihan. Data uji diproses dan model dijalankan pada data tersebut.

Hasil prediksi kemudian dievaluasi menggunakan metrik kinerja seperti akurasi, presisi, recall, dan F1-score. Hasil evaluasi digunakan untuk menganalisis kinerja model secara keseluruhan, dan jika diperlukan, langkah-langkah iteratif diambil untuk meningkatkan kinerja model.

Penting juga untuk memvalidasi model di luar lingkungan pengembangan dengan menggunakan dataset eksternal yang tidak pernah dilihat sebelumnya. Ini memastikan bahwa model dapat diandalkan dalam penggunaan dunia nyata. Melalui implementasi model dan pengujian performa, proses pengembangan model klasifikasi roastingan kopi menjadi lebih lengkap dan dapat diandalkan, memastikan bahwa model dapat digunakan secara efektif dalam aplikasi praktis.

### 3.3 Kebutuhan Hardware & Software

#### 3.3.1 Kebutuhan Hardware

Dalam pengimplementasian penelitian ini , diperlukan beberapa perangkat keras yang menggunakan arsitektur x64 dengan spesifikasi minimal tertentu. Spesifikasi ini menjadi kebutuhan kritis untuk menjalankan model dan memberikan landasan evaluasi kinerja kecepatan model dalam konteks penelitian.

**Tabel 3.1** Kebutuhan Hardware

No	Hardware	Spesifikasi
1	<i>Processor</i>	Intel Core i7-7700HQ
2	<i>RAM</i>	16 GB
3	<i>GPU</i>	Geforce GTX 1050 Ram 16Gb
4	Penyimpanan	1 <i>Terabyte</i>
5	Sistem Operasi	<i>Windows 10 / Windows 11</i>

Dalam konteks pelaksanaan penelitian, dapat dipilih untuk menggunakan sistem yang memiliki spesifikasi seperti yang telah dijelaskan sebelumnya atau memanfaatkan komputasi awan melalui platform seperti Google Collab atau layanan cloud lainnya untuk menjalankan model.

### 3.3.2 Kebutuhan Software

Pada perancangan sistem ini dibutuhkan beberapa perangkat lunak (*Software*) sebagai pendukung untuk melakukan penelitian ini. Ada dua *software* yang akan digunakan diantaranya yaitu *Android Studio*, dan Google Colaboratory. Pada perancangan ini peneliti menggunakan *Android Studio* sebagai software *IDE* untuk merancang *Mobile App* yang akan dikembangkan, Dan peneliti menggunakan Google Colaboratory sebagai tempat untuk merancang logika dari *Machine Learning* yang akan digunakan peneliti untuk membuat Sistem Klasifikasi Roastingan Kopi.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Data Training dan Validation

Untuk penelitian ini, data yang digunakan berasal dari citra biji kopi yang telah dikategorikan berdasarkan tingkat roasting, yaitu green (mentah), light, medium, dan dark. Data tersebut dibagi menjadi dua set: data training dan data validasi.

**Tabel 4.1** Jumlah Data Training Setiap Kelas

Kelas	Jumlah Data Training
Green	300
Light	300
Medium	300
Dark	300

Tabel 4.1 menunjukkan jumlah data training yang digunakan untuk setiap kelas tingkat roasting biji kopi. Masing-masing kelas memiliki jumlah data yang sama, yaitu 300 data gambar biji kopi. Distribusi yang seimbang ini membantu model dalam belajar secara efektif dari setiap kelas.

**Tabel 4.2** Jumlah Data Validasi Setiap Kelas

Kelas	Jumlah Data Validasi
Green	100
Light	100
Medium	100
Dark	100

Tabel 4.2 menunjukkan jumlah data validasi yang digunakan untuk setiap kelas tingkat roasting biji kopi. Sama seperti data training, masing-masing kelas memiliki jumlah data validasi yang sama, yaitu 100 data gambar biji kopi. Data validasi ini digunakan untuk menguji dan memantau kinerja model selama pelatihan, memastikan model dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

## **4.2 Pembangunan Model CNN**

Dalam pembangunan model CNN, ada beberapa komponen penting yang perlu diperhatikan untuk memastikan model dapat melakukan klasifikasi dengan akurat.

### **4.2.1 layer atau neuron**

Model CNN yang digunakan dalam penelitian ini terdiri dari beberapa lapisan utama yang berfungsi untuk mengekstraksi fitur, mengurangi dimensi data, dan melakukan klasifikasi. Struktur umum dari model CNN dimulai dengan Input Layer yang menerima gambar berukuran 224 x 224 piksel. Lapisan pertama adalah Convolutional Layer 1 dengan 32 filter berukuran 3x3, yang bertugas mengekstraksi fitur dasar dari gambar, seperti tepi dan tekstur awal. Hasil dari lapisan ini kemudian diproses oleh Pooling Layer 1 yang menggunakan max pooling berukuran 2x2 untuk mengurangi dimensi data dan kompleksitas komputasi. Selanjutnya, terdapat Convolutional Layer 2 dengan 64 filter berukuran 3x3 yang mengekstraksi fitur lebih kompleks dari gambar. Hasil dari lapisan ini kembali diproses oleh Pooling Layer 2 yang menggunakan max pooling berukuran 2x2 untuk lebih lanjut mengurangi dimensi data sambil mempertahankan fitur-fitur penting.

Setelah melalui dua lapisan konvolusi dan pooling, data kemudian masuk ke Fully Connected Layer yang terdiri dari 128 neuron. Lapisan fully connected ini berfungsi untuk menggabungkan semua fitur yang telah diekstraksi dan menginterpretasikan fitur-fitur tersebut dalam konteks klasifikasi. Akhirnya, Output Layer menggunakan fungsi aktivasi softmax untuk melakukan klasifikasi ke dalam 4 kelas, yaitu green, light, medium, dan dark. Fungsi softmax mengubah output dari lapisan fully connected menjadi probabilitas untuk masing-masing kelas, memungkinkan model untuk menentukan kelas mana yang paling sesuai dengan gambar yang diberikan.

#### **4.2.2 Fungsi Aktivasi**

Fungsi aktivasi yang digunakan dalam model CNN ini adalah ReLU (Rectified Linear Unit) dan softmax. Fungsi ReLU digunakan pada lapisan konvolusi dan fully connected untuk memperkenalkan non-linearitas ke dalam model. Fungsi ReLU sangat efektif karena dapat mempercepat proses pelatihan dengan cara mengaktifkan hanya neuron-neuron tertentu, sehingga mengurangi masalah vanishing gradient yang sering terjadi pada jaringan neural yang dalam. Fungsi ReLU bekerja dengan cara memetakan semua nilai negatif menjadi nol dan membiarkan nilai positif tetap sama, yang memungkinkan model untuk belajar pola-pola kompleks dari data gambar biji kopi.

Pada lapisan output, digunakan fungsi aktivasi softmax. Fungsi ini cocok untuk masalah klasifikasi multi-kelas karena mengubah output dari lapisan fully connected menjadi probabilitas yang menjumlahkan hingga 1 untuk setiap kelas. Dengan menggunakan softmax, model dapat memberikan probabilitas untuk masing-masing kelas tingkat roasting biji kopi (green, light, medium,

dark), sehingga memungkinkan identifikasi kelas mana yang paling mungkin sesuai dengan gambar yang diberikan. Probabilitas ini membantu dalam menentukan keputusan akhir model untuk klasifikasi tingkat roasting biji kopi berdasarkan warna.

Berikut adalah fungsi ReLU dan softmax yang digunakan dalam model:

```
def relu(X):  
    return np.maximum(0, X)
```

Fungsi ini memetakan semua nilai negatif pada input menjadi nol dan membiarkan nilai positif tetap sama. Ini adalah salah satu fungsi aktivasi yang paling umum digunakan dalam jaringan neural karena kesederhanaannya dan efektivitasnya dalam mengatasi masalah vanishing gradient.

Berikut adalah bagian dari kode di mana fungsi ReLU digunakan

```
def forward(X):  
    Z1 = conv2d(X, W1, b1)  
    A1 = relu(Z1)  
    P1 = max_pool(A1)  
  
    Z2 = conv2d(P1, W2, b2)  
    A2 = relu(Z2)  
    P2 = max_pool(A2)  
  
    F = flatten(P2)  
  
    Z3 = fully_connected(F, W3, b3)  
    A3 = relu(Z3)  
  
    Z4 = fully_connected(A3, W4, b4)  
    A4 = softmax(Z4)  
  
    return A4
```



Dengan menggunakan fungsi ReLU pada lapisan-lapisan konvolusi dan fully connected, model CNN ini mampu menangkap pola-pola non-linear yang kompleks dalam data gambar biji kopi, dan fungsi softmax pada lapisan output memungkinkan klasifikasi multi-kelas yang efektif.

### 4.2.3 Loss Function

Berikut adalah bagian kode yang menunjukkan definisi dan penggunaan fungsi loss.

```
# Loss Function
def compute_loss(Y, Y_hat):
    m = Y.shape[0]
    loss = -np.sum(Y * np.log(Y_hat)) / m
    return loss
```

Fungsi `compute_loss` menghitung categorical cross-entropy loss, yang umum digunakan dalam masalah klasifikasi multi-kelas. Berikut adalah penjelasan langkah demi langkah untuk fungsi ini:

1. Input:

- Y: Matriks one-hot encoded yang mewakili label sebenarnya dari data.
- Y\_hat: Matriks probabilitas prediksi yang dihasilkan oleh model (output dari softmax).

2. Jumlah Sampel (m) :

- `m = Y.shape[0]`: Menghitung jumlah sampel dalam batch.

### 3. Perhitungan Loss:

- $loss = -np.sum(Y * np.log(Y\_hat)) / m$ : Menghitung categorical cross-entropy loss.
- Untuk setiap sampel, loss dihitung dengan mengalikan label sebenarnya dengan log dari prediksi probabilitas, menjumlahkan semua nilai tersebut, dan mengubah tanda menjadi negatif. Ini kemudian dirata-rata dengan membagi jumlah sampel.

### 4. Output:

- Fungsi mengembalikan nilai loss rata-rata untuk batch.

Fungsi loss ini digunakan dalam training loop untuk mengukur kinerja model selama pelatihan:

```
# Training Loop
learning_rate = 0.01
num_epochs = 1
batch_size = 16

for epoch in range(num_epochs):
    indices = np.arange(X_train.shape[0])
    np.random.shuffle(indices)
    X_train = X_train[indices]
    y_train = y_train[indices]

    epoch_loss = 0
    epoch_accuracy = 0
    num_batches = 0

    for i in range(0, X_train.shape[0], batch_size):
        X_batch = X_train[i:i+batch_size]
        y_batch = y_train[i:i+batch_size]

        Y_hat = forward(X_batch)

        loss = compute_loss(y_batch, Y_hat)
        epoch_loss += loss
```

```
Y_pred = np.argmax(Y_hat, axis=1)
Y_true = np.argmax(y_batch, axis=1)
accuracy = compute_accuracy(Y_true, Y_pred)
epoch_accuracy += accuracy
num_batches += 1

epoch_loss /= num_batches
epoch_accuracy /= num_batches
print(f'Epoch {epoch + 1}/{num_epochs}, Loss: {epoch_loss:.4f},
Accuracy: {epoch_accuracy * 100:.2f}%')
```

Dalam training loop, `compute_loss` digunakan untuk menghitung loss untuk setiap batch data yang diproses oleh model selama setiap epoch. Loss ini kemudian dirata-rata untuk seluruh batch dalam satu epoch untuk memberikan gambaran tentang kinerja model selama pelatihan.

#### 4.2.4 Optimizer

Optimizer yang digunakan dalam model ini adalah algoritma stochastic gradient descent (SGD) dengan penurunan gradien mini-batch. Optimizer bertugas untuk meminimalkan fungsi loss dengan cara memperbarui bobot-bobot dalam model berdasarkan gradien dari fungsi loss. Dalam konteks model ini, optimizer SGD memperbarui bobot dengan menggunakan gradien dari fungsi loss yang dihitung berdasarkan prediksi model dan label sebenarnya. Setiap iterasi pelatihan menggunakan mini-batch data, yang membantu dalam mempercepat proses pelatihan dan stabilitas.

Proses optimasi dimulai dengan forward pass, di mana model menghasilkan prediksi berdasarkan input mini-batch. Kemudian, fungsi loss dihitung untuk mengukur seberapa jauh prediksi model dari nilai sebenarnya. Selanjutnya, gradien dari fungsi loss terhadap setiap parameter model dihitung. Gradien ini

menunjukkan arah dan besaran perubahan yang diperlukan untuk mengurangi loss. Akhirnya, bobot dan bias diperbarui menggunakan gradien yang telah dihitung dan learning rate yang telah ditentukan. Pembaruan ini dilakukan dengan mengurangi produk dari learning rate dan gradien dari bobot dan bias yang ada. Dengan menerapkan optimizer SGD, model secara iteratif memperbarui bobotnya untuk meminimalkan loss dan meningkatkan akurasi klasifikasi tingkat roasting biji kopi berdasarkan warna.

#### 4.2.5 Metrics

Metrik yang digunakan untuk mengevaluasi performa model dalam penelitian ini adalah akurasi. Akurasi adalah metrik yang paling umum digunakan dalam masalah klasifikasi, termasuk klasifikasi tingkat roasting biji kopi berdasarkan warna. Akurasi mengukur persentase prediksi yang benar dari total prediksi yang dilakukan oleh model.

Akurasi dihitung dengan membandingkan prediksi model dengan label sebenarnya dan menghitung rasio prediksi yang benar. Ini memberikan gambaran langsung tentang seberapa baik model dapat mengklasifikasikan data ke dalam kelas yang benar. Dalam penelitian ini, akurasi dihitung selama proses pelatihan dan pengujian untuk memantau kinerja model dan memastikan bahwa model tidak overfitting atau underfitting.

Dalam kode yang diberikan, akurasi dihitung menggunakan fungsi `compute_accuracy` yang membandingkan prediksi model dengan label sebenarnya:

```
def compute_accuracy(Y_true, Y_pred):  
    return np.mean(Y_true == Y_pred)
```

Selama proses pelatihan, akurasi dihitung untuk setiap batch data dan dirata-rata untuk seluruh batch dalam satu epoch. Ini memberikan gambaran tentang kinerja model pada data pelatihan. Selain itu, akurasi juga dihitung pada data pengujian untuk mengevaluasi generalisasi model:

```

for epoch in range(num_epochs):
    # ... (kode untuk pembaruan bobot dan bias)

    for i in range(0, X_train.shape[0], batch_size):
        # ... (kode untuk forward pass dan perhitungan loss)

        # Compute accuracy
        Y_pred = np.argmax(Y_hat, axis=1)
        Y_true = np.argmax(y_batch, axis=1)
        accuracy = compute_accuracy(Y_true, Y_pred)
        epoch_accuracy += accuracy
        num_batches += 1

    epoch_accuracy /= num_batches
    print(f'Epoch {epoch + 1}/{num_epochs},    Loss: {epoch_loss:.4f},
Accuracy: {epoch_accuracy * 100:.2f}%')

# Compute Test Accuracy
def predict(X):
    Y_hat = forward(X)
    return np.argmax(Y_hat, axis=1)

Y_pred = []
for i in range(0, X_test.shape[0], batch_size):
    X_batch = X_test[i:i+batch_size]
    Y_pred_batch = predict(X_batch)
    Y_pred.extend(Y_pred_batch)

Y_pred = np.array(Y_pred)
Y_true = np.argmax(y_test, axis=1)

test_accuracy = compute_accuracy(Y_true, Y_pred)
print(f'Test accuracy: {test_accuracy * 100:.2f}%')

```

Dengan menggunakan metrik akurasi, peneliti dapat memantau seberapa baik model mempelajari data pelatihan dan seberapa baik model dapat menggeneralisasi ke data yang belum pernah dilihat sebelumnya. Metrik akurasi memberikan informasi penting untuk menilai keberhasilan model dalam klasifikasi tingkat roasting biji kopi.

### 4.3 Training Model

Pada bagian ini, proses pelatihan model dijelaskan secara rinci. Model CNN dilatih menggunakan dataset gambar biji kopi yang telah diproses sebelumnya. Dataset dibagi menjadi dua bagian yaitu data pelatihan dan data validasi. Proses pelatihan melibatkan beberapa langkah, termasuk augmentasi data, kompilasi model, dan pelatihan model.

Augmentasi data dilakukan menggunakan ImageDataGenerator untuk meningkatkan variasi dataset dan mencegah overfitting. Model CNN dibangun dengan beberapa lapisan konvolusi, pooling, dan fully connected, serta menggunakan fungsi aktivasi ReLU untuk memperkenalkan non-linearitas dan fungsi softmax untuk klasifikasi multi-kelas. Model dikompilasi dengan loss function categorical\_crossentropy dan optimizer adam. Pelatihan model dilakukan selama 16 epoch dengan batch size 32, dan performa model dipantau melalui metrik akurasi dan loss pada data pelatihan dan validasi.

```
import os
import zipfile
import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
from google.colab import files, drive
```

```
drive.mount('/content/drive/')

training_path = '/content/drive/MyDrive/Kopi/Coffee_beans/train'
testing_path = '/content/drive/MyDrive/Kopi/Coffee_beans/test'

width = 160
height = 160

training_datagen = tf.keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255,
    horizontal_flip=True,
    vertical_flip=True,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.2,
    zoom_range=0.2
)

validation_datagen = tf.keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255
)

training_generator = training_datagen.flow_from_directory(
    training_path,
    target_size=(width, height),
    shuffle=True,
    class_mode='categorical',
    batch_size=32
)

validation_generator = validation_datagen.flow_from_directory(
    testing_path,
    target_size=(width, height),
    shuffle=True,
    class_mode='categorical',
    batch_size=32
)

model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(16, (3, 3), activation='relu',
input_shape=(width, height, 3)),
```

```
tf.keras.layers.MaxPooling2D(2, 2),
tf.keras.layers.Conv2D(8, (3, 3), activation='relu'),
tf.keras.layers.MaxPooling2D(2, 2),
tf.keras.layers.Conv2D(8, (3, 3), activation='relu'),
tf.keras.layers.MaxPooling2D(2, 2),
tf.keras.layers.Flatten(),
tf.keras.layers.Dense(64, activation='relu'),
tf.keras.layers.Dropout(0.3),
tf.keras.layers.Dense(16, activation='relu'),
tf.keras.layers.Dense(training_generator.num_classes,
activation='softmax')
])

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

history = model.fit(
    training_generator,
    validation_data = validation_generator,
    batch_size=32,
    verbose=True,
    epochs=16
)
```



```

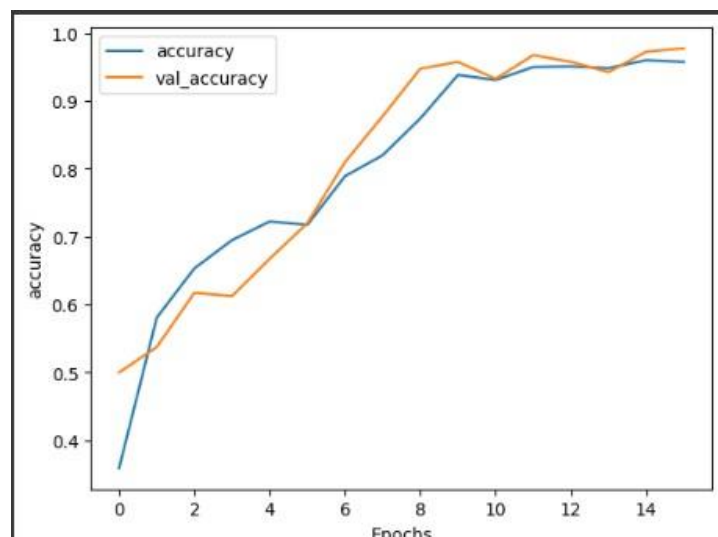
Epoch 1/16
Code cell output actions 60s 1s/step - accuracy: 0.2983 - loss: 1.3656 - val_accuracy: 0.5000 - val_loss: 0.8773
Epoch 2/16
38/38 65s 1s/step - accuracy: 0.5544 - loss: 0.8547 - val_accuracy: 0.5375 - val_loss: 0.8287
Epoch 3/16
38/38 40s 952ms/step - accuracy: 0.6526 - loss: 0.7010 - val_accuracy: 0.6175 - val_loss: 0.6618
Epoch 4/16
38/38 42s 955ms/step - accuracy: 0.6781 - loss: 0.6475 - val_accuracy: 0.6125 - val_loss: 0.7196
Epoch 5/16
38/38 40s 984ms/step - accuracy: 0.7214 - loss: 0.5729 - val_accuracy: 0.6675 - val_loss: 0.6695
Epoch 6/16
38/38 40s 955ms/step - accuracy: 0.7261 - loss: 0.5747 - val_accuracy: 0.7200 - val_loss: 0.6093
Epoch 7/16
38/38 43s 973ms/step - accuracy: 0.7853 - loss: 0.4985 - val_accuracy: 0.8100 - val_loss: 0.4345
Epoch 8/16
38/38 46s 1s/step - accuracy: 0.8124 - loss: 0.4699 - val_accuracy: 0.8775 - val_loss: 0.3662
Epoch 9/16
38/38 46s 1s/step - accuracy: 0.8686 - loss: 0.3272 - val_accuracy: 0.9475 - val_loss: 0.1644
Epoch 10/16
38/38 40s 966ms/step - accuracy: 0.9334 - loss: 0.2205 - val_accuracy: 0.9575 - val_loss: 0.1100
Epoch 11/16
38/38 41s 960ms/step - accuracy: 0.9162 - loss: 0.2372 - val_accuracy: 0.9325 - val_loss: 0.1760
Epoch 12/16
38/38 40s 954ms/step - accuracy: 0.9635 - loss: 0.1307 - val_accuracy: 0.9675 - val_loss: 0.0915
Epoch 13/16
38/38 47s 1s/step - accuracy: 0.9315 - loss: 0.1060 - val_accuracy: 0.9575 - val_loss: 0.1158
Epoch 14/16
38/38 40s 976ms/step - accuracy: 0.9570 - loss: 0.1224 - val_accuracy: 0.9425 - val_loss: 0.1394
Epoch 15/16
38/38 40s 959ms/step - accuracy: 0.9533 - loss: 0.1527 - val_accuracy: 0.9725 - val_loss: 0.0755
Epoch 16/16
38/38 39s 930ms/step - accuracy: 0.9526 - loss: 0.1237 - val_accuracy: 0.9775 - val_loss: 0.0683

```

**Gambar 4.1** Gambar Hasil Training Model

Gambar tersebut menunjukkan hasil dari proses pelatihan model CNN selama 16 epoch. Berikut adalah beberapa poin penting dari hasil Secara keseluruhan, hasil ini menunjukkan bahwa model CNN berhasil dilatih dengan baik, dengan peningkatan yang konsisten dalam akurasi dan penurunan loss baik pada data pelatihan maupun data validasi selama 16 epoch.

#### 4.3.1 Akurasi



**Gambar 4.2** Akurasi

Grafik yang ditampilkan menunjukkan tren akurasi dan loss selama proses pelatihan model CNN selama 16 epoch. Grafik akurasi menunjukkan

bagaimana akurasi pada data pelatihan dan data validasi meningkat seiring waktu. Garis biru mewakili akurasi pelatihan, sedangkan garis oranye mewakili akurasi validasi.

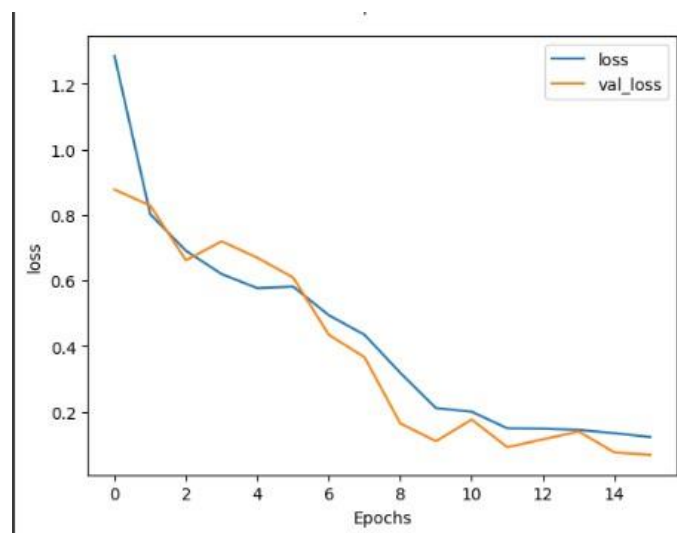
#### 4.3.1.1 Akurasi Pelatihan (accuracy)

Akurasi model pada data pelatihan meningkat dari 0.2093 pada epoch pertama hingga 0.9526 pada epoch ke-16. Ini menunjukkan bahwa model semakin baik dalam memprediksi kelas yang benar pada data pelatihan seiring berjalannya waktu.

#### 4.3.1.2 Akurasi Validasi (val\_accuracy)

Akurasi model pada data validasi juga menunjukkan peningkatan yang signifikan, dari 0.5000 pada epoch pertama hingga 0.9775 pada epoch ke-16. Ini menunjukkan bahwa model tidak hanya belajar dari data pelatihan tetapi juga menggeneralisasi dengan baik pada data yang belum dilihat sebelumnya.

#### 4.3.2 Loss



**Gambar 4.1** Loss

Grafik loss menunjukkan bagaimana nilai loss pada data pelatihan dan data validasi berkurang seiring waktu. Garis biru mewakili loss pada data pelatihan, sedangkan garis oranye mewakili loss pada data validasi.

#### 4.3.2.1 Loss Pelatihan (loss)

Nilai loss pada data pelatihan berkurang dari 1.3656 pada epoch pertama menjadi 0.1237 pada epoch ke-16. Penurunan ini menunjukkan bahwa model semakin baik dalam menyesuaikan prediksi dengan label sebenarnya.

#### 4.3.2.2 Loss Validasi (val\_loss)

Nilai loss pada data validasi berkurang dari 0.8773 pada epoch pertama menjadi 0.0863 pada epoch ke-16, yang mengindikasikan peningkatan kinerja model pada data validasi.

### 4.4 Perhitungan Model

#### 4.4.1 Perhitungan Augmentasi Data

Data augmentation adalah proses memperbesar jumlah dan keragaman data pelatihan dengan membuat variasi data dari set pelatihan asli. Di sini, kita menggunakan beberapa teknik augmentasi:

##### 4.4.1.1 Rescale

Setiap nilai piksel gambar dibagi dengan 255 untuk menormalkan nilai piksel ke rentang  $[0,1][0, 1][0,1]$ . Jika  $I$  adalah nilai piksel asli, maka nilai setelah rescale adalah:

$$Pixel_{normalize} = \frac{Pixel_{original}}{255}$$

##### 4.4.1.2 Horizontal Flip & Vertical Flip

Membalikkan gambar secara horizontal dan vertikal. Tidak ada rumus matematis spesifik, hanya transformasi citra.

$$(i, j) \rightarrow (i, W - j - 1)$$

$$(i, j) \rightarrow (H - i - 1, j)$$

##### 4.4.1.3 Width & Height Shift Range

Menggeser gambar secara horizontal dan vertikal. Misalkan, gambar di-

shift sebanyak 10% dari lebar atau tinggi aslinya.

$$\max width\ shift = W \times width\_shift\_range$$

$$\max height\ shift = H \times height\_shift\_range$$

#### 4.4.1.4 Shear Range

Shear range pergeseran bentuk digunakan dalam augmentasi gambar untuk mengubah sudut pandang gambar, seolah-olah gambar tersebut diambil dari sudut yang berbeda. Dalam konteks *image processing*, operasi ini dapat dianggap sebagai transformasi geometris yang menggeser satu bagian gambar dengan cara yang berbeda dari bagian lainnya.

$$Shear\ Matrix = \begin{bmatrix} 1 & \tan(\theta) \\ 0 & 1 \end{bmatrix}$$

#### 4.4.1.5 Zoom Range

Zoom range atau operasi yang mengubah ukuran gambar, baik dengan memperbesar (memperbesar objek dalam gambar) atau memperkecil (memperkecil objek dalam gambar). Saat melakukan zoom, bagian dari gambar dipilih, diperbesar atau diperkecil, dan dipetakan kembali ke ukuran asli.

$$(H^1, W^1) = (z \cdot H, z \cdot W)$$

#### 4.4.2 Perhitungan Preprocessing Data

Flow\_from\_directory adalah metode yang digunakan untuk memuat data gambar dari direktori, melakukan augmentasi, dan menghasilkan batch data yang dapat digunakan oleh model deep learning. Metode ini secara otomatis mengatur label berdasarkan struktur direktori. Kelas-kelas ditentukan oleh sub-direktori di dalam training\_path atau testing\_path.

#### 4.4.2.1 Mengambil Data Dari Direktori

Gambar diambil dari sub-direktori di dalam `training_path` dan `testing_path`. Label gambar dihasilkan berdasarkan nama sub-direktori.

- Misalkan ada  $C$  kelas dan total  $N$  gambar.
- Gambar akan dibagi kedalam  $C$  sub-kumpulan berdasarkan kelas, misalnya  $N_1, N_2, \dots, N_C$ .

#### 4.4.2.2 Resizing Gambar

Setiap gambar diubah ukurannya menjadi (width, height):

$$\text{Resized Image Size} = (H, W) \rightarrow (h, w)$$

Di mana:

- $H, W$  adalah tinggi dan lebar asli gambar.
- $h, w$  adalah tinggi dan lebar target yang ditentukan.

#### 4.4.2.3 OneHot Encoding Label

Jika ada  $X$  kelas, maka setiap label diubah menjadi vektor one-hot berukuran  $X$ :

$$\text{Label One - Hot} = \begin{matrix} (1,0,0, \dots, 0) \text{ jika kelas} = 1 \\ (0,1,0, \dots, 0) \text{ jika kelas} = 2 \\ \vdots \\ \mathbf{1} (0,0,0, \dots, 1) \text{ jika kelas} = C \end{matrix}$$

#### 4.4.2.4 Batch Generation

Gambar diambil dalam batch sebesar `batch_size` yang ditentukan (misalnya, 32). Setiap batch berisi gambar dan label yang diacak. Proses ini terus berlanjut sampai semua gambar telah digunakan, dan kemudian proses pengambilan gambar dimulai lagi dengan urutan acak yang baru.

- Ukuran Batch:  $B = \text{batch\_size}$

- Jumlah Batch per Epoch:  $\frac{N}{B}$

Di mana:

- N adalah jumlah total gambar dalam set pelatihan atau validasi.
- B adalah ukuran batch.

#### 4.4.3 Perhitungan Model

##### 4.4.3.1 Conv2D Layer

Conv2D Layer: Menggunakan filter konvolusi untuk mengekstrak fitur dari gambar.

Jumlah Filter: 16

Ukuran Filter: (3, 3)

Input Shape: (width, height, 3)

$$\text{Output Height} = \frac{(\text{Input Height} - \text{Filter Height} + 2 \times \text{Padding})}{\text{Stride}} + 1$$

$$\text{Output Width} = \frac{(\text{Input Width} - \text{Filter Width} + 2 \times \text{Padding})}{\text{Stride}} + 1$$

Dengan padding default = 0, stride = 1:

$$\text{Output Height} = \frac{(H - 3 + 2 \times 0)}{1} + 1 = H - 2$$

$$\text{Output Width} = \frac{(W - 3 + 2 \times 0)}{1} + 1 = W - 2$$

Output Shape setelah Conv2D:

$(H - 2, W - 2, 16)$

#### 4.4.3.2 MaxPooling2D Layer

MaxPooling2D Layer: Mengurangi dimensi gambar dengan operasi pooling.

Pool Size: (2, 2)

Stride: 2 (default)

$$\text{Output Height} = \frac{\text{Input Height}}{\text{Pool Size}} = \frac{(H-2)}{2}$$

$$\text{Output Width} = \frac{\text{Input Width}}{\text{Pool Size}} = \frac{(W-2)}{2}$$

Output Shape setelah MaxPooling2D:

$$\left(\frac{H-2}{2}, \frac{W-2}{2}, 16\right)$$

#### 4.4.3.3 Conv2D Layer (Second Occurrence)

Dengan input dari MaxPooling2D layer:

$$\text{Output Height} = \frac{\left(\frac{H-2}{2} - 3 + 2 \times 0\right)}{1} + 1 = \frac{H-8}{2}$$

$$\text{Output Width} = \frac{\left(\frac{W-2}{2} - 3 + 2 \times 0\right)}{1} + 1 = \frac{W-8}{2}$$

Output shape:

$$\left(\frac{H-8}{2}, \frac{W-8}{2}, 8\right)$$

#### 4.4.3.4 MaxPooling2D Layer (Second Occurrence)

$$\text{Output Height} = \frac{\left(\frac{H-8}{2}\right)}{2} = \frac{H-8}{4}$$

$$\text{Output Width} = \frac{\left(\frac{W-8}{2}\right)}{2} = \frac{W-8}{4}$$

Output shape:

$$\left(\frac{H-8}{4}, \frac{W-8}{4}, 8\right)$$

#### 4.4.3.5 Conv2D Layer (Third Occurrence)

Dengan input dari MaxPooling2D layer kedua:

$$\text{Output Height} = \frac{\left(\frac{H-8}{4} - 3 + 2 \times 0\right)}{1} + 1 = \frac{H-17}{4}$$

$$\text{Output Width} = \frac{\left(\frac{W-8}{4} - 3 + 2 \times 0\right)}{1} + 1 = \frac{W-17}{4}$$

Output shape:

$$\left(\frac{H-17}{4}, \frac{W-17}{4}, 8\right)$$

#### 4.4.3.6 MaxPooling2D Layer (Third Occurrence)

$$\text{Output Height} = \frac{\left(\frac{H-17}{4}\right)}{2} = \frac{H-17}{8}$$

$$\text{Output Width} = \frac{\left(\frac{W-17}{4}\right)}{2} = \frac{W-17}{8}$$

Output shape:

$$\left(\frac{H-17}{8}, \frac{W-17}{8}, 8\right)$$

#### 4.4.3.7 Flatten Layer

Flatten layer mengubah tensor 3D menjadi tensor 1D.

$$\text{Flattened Size} = \text{Height} \times \text{Width} \times \text{Channels}$$

$$\text{Flattened Size} = \left(\frac{H-17}{8}\right) \times \left(\frac{W-17}{8}\right) \times 8$$

#### 4.4.3.8 Dense Layer (1)

Untuk Dense layer, parameter dihitung sebagai:

$$\text{Jumlah Parameter} = (\text{Jumlah Neuron pada Layer Sebelumnya} + 1) \times$$

Jumlah Neuron di Layer Ini

$$\text{Jumlah Parameter} = \left(\frac{H-17}{8} \times \frac{W-17}{8} \times 8 + 1\right) \times 64$$

#### 4.4.3.8.1 Matrix Multiplication

Misalkan output dari layer sebelumnya adalah vektor  $x$  dengan dimensi  $n$ .



Matriks bobot  $W$  untuk dense layer memiliki dimensi  $m \times n$  ( $m$  adalah jumlah neuron di dense layer, dan  $n$  adalah dimensi dari vektor input).

Operasi ini menghasilkan vektor  $z$  dari dimensi  $m$ :

$$Z = Wx$$

#### 4.4.3.8.2 Bias Addition

Bias  $b$  adalah vektor dengan dimensi  $m$ , yang ditambahkan ke setiap elemen dari  $z$ . Hasilnya adalah vektor  $y$ .

$$y = z + b$$

#### 4.4.3.8.3 Activation Function

Fungsi aktivasi  $f(x)$  diterapkan ke setiap elemen dari  $y$ . Misalnya, jika kita menggunakan ReLU sebagai fungsi aktivasi, maka:

$$y_{\text{activated}} = \text{ReLU}(y)$$

#### 4.4.3.9 Dropout Layer

Dropout mengubah cara neuron digunakan selama pelatihan untuk mencegah overfitting dengan menonaktifkan sebagian neuron secara acak dan menyesuaikan output selama inferensi. 30% neuron dinonaktifkan selama pelatihan.

$$O_{\text{inferensi}} = O_{\text{pelatihan}} \times (1 - 0.3) = O_{\text{pelatihan}} \times 0.7$$

#### 4.4.3.10 Dense Layer (2)

Input ke layer ini adalah output dari Dense Layer 1 setelah melewati Dropout layer.

$$\text{Jumlah Parameter} = 65 \times 16 = 1040$$

#### 4.4.3.11 Dense Layer (3)

Input ke layer ini adalah output dari Dense Layer 2.

$$\text{Jumlah Parameter} = 17 \times 10 = 170$$

#### 4.4.4 Compile Model

##### 4.4.4.1 Loss Function

Fungsi loss adalah ukuran seberapa baik model Anda membuat prediksi.

$$Loss = - \sum_{i=1}^C y_i \log(p_i)$$

Di mana  $C$  adalah jumlah kelas,  $y_i$  adalah label target (1 jika benar, 0 jika tidak), dan  $p_i$  adalah probabilitas prediksi untuk kelas  $i$ .

##### 4.4.4.2 Optimizer

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\theta} L$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_{\theta} L)^2$$

$$\hat{m} = \frac{m_t}{1 - \beta_1^t}$$

$$\theta_{t+1} = \theta - \frac{\alpha \hat{m}}{\sqrt{v_t} + \epsilon}$$

Di mana  $m_t$  adalah momen pertama,  $v_t$  adalah momen kedua,  $\alpha$  adalah learning rate, dan  $\epsilon$  adalah nilai kecil untuk stabilitas numerik.

##### 4.4.4.3 Metrics

$$Accuracy = \frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Total Prediksi}}$$

Di mana "Jumlah Prediksi Benar" adalah jumlah kasus di mana prediksi model sesuai dengan label sebenarnya, dan "Jumlah Total Prediksi" adalah jumlah seluruh prediksi yang dilakukan.

#### 4.5 Pengujian Model

Setelah model dilatih, pengujian dilakukan untuk mengevaluasi performa model pada data yang belum pernah dilihat sebelumnya (data testing). Proses ini melibatkan pengujian model pada gambar baru untuk memprediksi kelas roasting biji kopi. Gambar diunggah dan diproses menggunakan model yang

telah dilatih, dan hasil prediksi ditampilkan. Pengujian ini memastikan bahwa model dapat menggeneralisasi dengan baik dan memberikan prediksi yang akurat pada data baru.

```
# Uploading and predicting a new image
image_data = files.upload()
image_file = list(image_data.keys())[0]
image = tf.keras.utils.load_img(image_file, target_size=(width, height))
image_array = tf.keras.utils.img_to_array(image)
image_array = image_array / 255.
image_array = np.array([image_array])
result_predict = model.predict(image_array)
predicted_class_index = np.argmax(result_predict, axis=1)[0]
class_labels = list(training_generator.class_indices.keys())
predicted_class_label = class_labels[predicted_class_index]

print(f'Hasil prediksi adalah: {predicted_class_label}')
```



**Gambar 4.2** Hasil Pengujian

Gambar tersebut menunjukkan hasil prediksi dari model CNN setelah melakukan klasifikasi tingkat roasting biji kopi. Proses ini melibatkan pengunggahan sebuah gambar biji kopi yang dipanggang, di mana gambar tersebut kemudian diprediksi oleh model yang telah dilatih.

Model berhasil mengklasifikasikan gambar tersebut sebagai "Dark", yang menunjukkan bahwa model dapat mengenali tingkat roasting biji kopi dari gambar yang diberikan dengan baik.

Hasil ini menunjukkan bahwa model mampu mengklasifikasikan tingkat roasting biji kopi dengan akurasi yang tinggi, sesuai dengan hasil pelatihan dan validasi yang telah dilakukan sebelumnya.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Dari perancangan yang telah dilakukan dengan dapat disimpulkan bahwa :

1. Penelitian ini berhasil mengimplementasikan metode Deep Learning dengan Convolutional Neural Network (CNN) untuk mengklasifikasikan tingkat roasting biji kopi berdasarkan warna, mencapai akurasi tinggi dan penurunan loss yang signifikan selama pelatihan dan validasi.
2. Penggunaan fungsi aktivasi ReLU dan softmax, serta optimizer Adam, efektif dalam mengoptimalkan model, menghasilkan prediksi yang akurat untuk gambar biji kopi baru.
3. Model ini berpotensi menjadi alat bantu edukatif yang berguna bagi barista pemula dalam memahami dan menentukan tingkat roasting biji kopi berdasarkan warna.

#### **5.2 Saran**

Adapun saran yang dapat diberikan oleh penulis untuk pengembang dan penelitian selanjutnya adalah sebagai berikut:

1. Mengumpulkan lebih banyak data dari berbagai sumber dan kondisi pencahayaan untuk meningkatkan generalisasi model.
2. Menerapkan teknik augmentasi data yang lebih canggih dan arsitektur CNN yang lebih kompleks untuk meningkatkan kinerja model.
3. Mengintegrasikan model ke dalam aplikasi mobile atau web yang dapat diakses oleh barista pemula, dengan panduan visual dan edukatif tentang tingkat roasting.

Dengan adanya saran-saran ini, diharapkan penelitian di bidang klasifikasi tingkat roasting biji kopi berbasis deep learning dapat terus berkembang dan memberikan kontribusi yang lebih besar dalam industri kopi, khususnya dalam mendukung edukasi barista pemula.

## DAFTAR PUSTAKA

- Alfiantama, Ilham, dkk. 2024. *Klasifikasi Tingkat Roasting Biji Kopi Dengan Metode CNN*. Jurnal STAINS. Vol. 3 No. 1: 285-290.
- A. Santoso and G. Ariyanto. 2018. *Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah*. Jurnal Emit. J. Tek. Elektro, vol. 18, no. 01 : 15–21.
- Iswahyudi & Santoso, H. 2023. *PyTorch Deep Learning for Food Image Classification with Food Dataset*. 8(4), 2651–2661.
- Nucci et al., 2014. *Digital Image Processing Second Edition Education*.
- Nugroho, M. Agung dan Sebatubun, Maria Mediatrix. 2020. *Klasifikasi Varietas Kopi Berdasarkan Green Bean Coffee Menggunakan Metode Machine Learning*. Jurnal of Information System Management. Vol. 1., No. 2: 1-5.
- Yoshua Bengio. 2019. *Deep Learning*. Nature, 521(7553), 436–444.
- Raharjo, Budi. 2022. *Deep Learning Dengan Phyton*. Semarang: YPAT.
- Rucci, M., & Casile, A. 2005. *Fixational instability and natural image statistics: Implications for early visual representations*. In Network: Computation in Neural Systems (Vol. 16, pp. 121–138).
- Setiadi, A Farhan, dkk. 2022. *Implementasi Metode CNN dan Deep Learning untuk Menentukan Tingkat Roasting Biji Kopi*. Information Technology Journal. Vol. 4 No. 2: 48-54.
- Sya'bani, Dino Rahman, dkk. 2022. *Klasifikasi Buah Segar Dan Busuk Menggunakan Algoritma Convolutional Neural Network Dengan Tflite Sebagai Media Penerapan Model Machine Learning*. Jurnal Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST). Vol. 1 No. 2: 7-16.
- Perdana, Ayu Lestari, dkk. 2023. *Pengenalan Ekspresi Wajah Pengunjung Deal Coffee Menggunakan Metode Convolutional Neural Network (CNN)*. Journal JESSI. Vol. 04 No. 2: 132-139.
- Pratiwi, Hanissa Anggraini, dkk. 2021. *Implementasi Deep Learning Flower Scanner Menggunakan Metode Convolutional Neural Network*. Sebatik. Vol. 25 No. 1 : 124-130.
- Pratama, Gege Ardi, dkk.. 2024. *Convolutional Neural Network Dan Faster Region*

- Convolutional Neural Network Untuk Klasifikasi Kualitas Biji Kopi Arabika.*  
Jurnal Informatika. Vol. 12, No. 3: 2776-2785.
- Peñuela-Martínez et al., 2023. *Characterization of Fermentations with Controlled Temperature with Three Varieties of Coffee (Coffea arabica L.).* Journal Fermentation. Vol. 9 Issue 11: 1-13.
- Turap et al., n.d. 2007. *Digital Image Processing 4th ed.* Los Altos. California: A John Wiley & Sons, Inc.
- Undip's Digitalization in Dealings. 2022. *Fenomena coffee shop di indonesia, bisnis kekinian di Indonesia.*
- V. Garchar and S. Chudhary. 2019. *Image Classification using Advanced CNN Based on Tensorflow.* Vol. 5 Issue 2: 1329–1335.