Daftar Isi

[Bab 1 Pendahuluan 1](#_Toc182044348)

[1.1 Latar Belakang 1](#_Toc182044349)

[1.2 Rumusan Masalah 3](#_Toc182044350)

[1.3 Tujuan 4](#_Toc182044351)

[1.4 Manfaat 5](#_Toc182044352)

[Bab 2 Landasan Teori 6](#_Toc182044353)

[2.1 Deteksi Objek 6](#_Toc182044354)

[2.2 Segmentasi 8](#_Toc182044355)

[Bab 3 Metodologi Penelitian 10](#_Toc182044356)

[3.1 Desain Sistem 10](#_Toc182044357)

[3.1.1 Struktur Utama Kode 10](#_Toc182044358)

[3.1.2 Input dan Pra-Pemrosesan Gambar 10](#_Toc182044359)

[3.1.3 Pengembangan Model CNN 10](#_Toc182044360)

[3.1.4 Proses Pelatihan dan Evaluasi Model 11](#_Toc182044361)

[3.1.5 Prediksi dan Implementasi 11](#_Toc182044362)

[3.2 Dataset dan Pra-pemrosesan 12](#_Toc182044363)

[3.2.1 Dataset 12](#_Toc182044364)

[3.2.2 Pemuatan Data dan Pengaturan Batch 12](#_Toc182044365)

[3.2.3 Augmentasi Gambar 12](#_Toc182044366)

[3.2.4 Normalisasi Gambar 13](#_Toc182044367)

[3.3 Arsitektur Model 14](#_Toc182044368)

[3.3.1 Definisi Model Sequential 14](#_Toc182044369)

[3.3.2 Lapisan Konvolusi (Convolutional Layers) 14](#_Toc182044370)

[3.3.3 Lapisan Pooling (Pooling Layers) 14](#_Toc182044371)

[3.3.4 Lapisan Dropout (Dropout Layer) 15](#_Toc182044372)

[3.3.5 Lapisan Fully Connected (Dense Layer) 15](#_Toc182044373)

[3.3.6 Kompilasi Model 15](#_Toc182044374)

[3.4 Pelatihan dan Validasi 16](#_Toc182044375)

[3.4.1 Pembagian Dataset 16](#_Toc182044376)

[3.4.2 Proses Pelatihan 16](#_Toc182044377)

[3.4.3 Penggunaan Fungsi Loss dan Optimizer 16](#_Toc182044378)

[3.4.4 Pemantauan Metrik Evaluasi 17](#_Toc182044379)

[3.4.5 Evaluasi Akhir dengan Data Validasi 17](#_Toc182044380)

# Pendahuluan

## Latar Belakang

Kemajuan teknologi di bidang “deep learning” telah membuka peluang besar untuk berbagai aplikasi, termasuk di sektor pertanian. Salah satu bidang yang mendapat perhatian signifikan adalah pemrosesan citra untuk deteksi dan klasifikasi objek. Sistem otomatis berbasis “Convolutional Neural Network” (CNN) kini banyak digunakan untuk menggantikan metode manual dalam analisis data visual yang rentan terhadap kesalahan manusia. CNN memiliki kemampuan untuk mendeteksi karakteristik visual secara otomatis, termasuk pola-pola spesifik seperti warna, tekstur, dan bentuk yang terdapat pada permukaan objek (LeCun, Bengio, & Hinton, 2015). Hal ini memungkinkan sistem berbasis CNN untuk digunakan secara luas dalam berbagai aplikasi, termasuk deteksi objek dan pengenalan pola pada gambar buah.

Dalam konteks deteksi kematangan buah, CNN menjadi solusi yang efektif untuk klasifikasi tingkat kematangan buah dengan akurasi tinggi, terutama pada buah mangga yang memiliki variasi warna dan tekstur yang kompleks seiring dengan proses pematangannya (Patel, Shah, & Joshi, 2020). Penelitian oleh Sibomana et al. (2021) menunjukkan bahwa penerapan CNN pada deteksi kematangan buah meningkatkan akurasi klasifikasi hingga 25% dibandingkan metode konvensional, menjadikannya pilihan ideal untuk industri pertanian yang menginginkan proses penentuan kematangan secara efisien dan konsisten.

Selain itu, CNN memungkinkan deteksi objek pada citra buah mangga tanpa terganggu oleh elemen latar belakang, suatu keuntungan besar dibandingkan metode manual yang cenderung terpengaruh oleh faktor subjektivitas manusia (Redmon et al., 2016). Dengan menggunakan teknologi ini, proses deteksi objek dapat dilakukan dengan cepat dan konsisten, mengatasi keterbatasan sistem manual. Teknologi ini juga memfasilitasi penyimpanan data otomatis yang memudahkan evaluasi kualitas produk secara berkelanjutan dalam skala besar.

Oleh karena itu, tujuan utama dari penelitian ini adalah mengembangkan sistem deteksi kematangan buah berbasis CNN yang mampu melakukan klasifikasi kematangan dengan akurasi tinggi dan konsisten. Dengan adanya teknologi ini, diharapkan sistem yang dikembangkan dapat diimplementasikan pada rantai pasokan untuk memastikan kualitas produk yang terstandarisasi, sehingga meningkatkan nilai pasar produk pertanian secara keseluruhan (Jiang, Wang, & Yang, 2021).

## Rumusan Masalah

Bagaimana memastikan tingkat kematangan buah mangga secara akurat untuk memenuhi standar kualitas dengan metode yang konsisten?

Bagaimana mengatasi keterbatasan metode manual dalam mendeteksi kematangan buah mangga yang sering kali dipengaruhi oleh faktor subjektif dan tidak konsisten?

Bagaimana membangun sistem otomatis berbasis Convolutional Neural Network (CNN) yang mampu mengklasifikasikan tingkat kematangan buah mangga berdasarkan karakteristik visualnya, seperti warna, tekstur, dan bentuk?

Bagaimana mengukur akurasi dan kinerja model CNN yang dikembangkan untuk memastikan bahwa sistem dapat diandalkan dalam kondisi nyata?

## Tujuan

Mengembangkan sistem deteksi tingkat kematangan buah mangga berbasis Convolutional Neural Network (CNN) yang mampu mengklasifikasikan kematangan berdasarkan karakteristik visual seperti warna, tekstur, dan bentuk.

Memastikan akurasi dan konsistensi model CNN dalam mendeteksi tingkat kematangan buah mangga agar hasil klasifikasi dapat diandalkan.

Menguji performa model CNN yang dibangun melalui pengujian pada dataset gambar mangga dengan berbagai tingkat kematangan.

Menyediakan alternatif metode deteksi kematangan yang lebih efisien dan praktis dibandingkan metode manual konvensional.

## Manfaat

Memberikan solusi otomatis yang dapat membantu petani dan distributor untuk menentukan waktu panen dan distribusi buah mangga secara tepat berdasarkan tingkat kematangannya.

Meningkatkan efisiensi proses seleksi kematangan buah dalam skala besar dengan hasil yang konsisten dan objektif, tanpa dipengaruhi oleh faktor subjektivitas manusia.

Mendukung inovasi di bidang pertanian modern dengan memanfaatkan teknologi deep learning untuk meningkatkan kualitas dan daya saing produk buah mangga di pasar.

4. Menyediakan sistem yang dapat diterapkan secara luas dalam industri pertanian sebagai alat bantu untuk meningkatkan rantai pasokan dan pengelolaan hasil panen.

# Landasan Teori

## Deteksi Objek

Deteksi objek adalah salah satu bidang utama dalam pemrosesan citra dan “computer vision” yang berfokus pada identifikasi dan lokalisasi objek tertentu dalam suatu gambar atau video. Deteksi objek memungkinkan sistem untuk mengenali keberadaan objek, menentukan posisi, dan mengklasifikasikannya berdasarkan kategori tertentu (Redmon et al., 2016). Dalam konteks deteksi kematangan buah, deteksi objek berperan penting untuk memastikan bahwa sistem dapat mengenali dan memfokuskan analisis pada buah yang akan dievaluasi, mengabaikan elemen latar belakang yang tidak relevan.

Berbagai metode telah dikembangkan untuk deteksi objek, mulai dari metode konvensional seperti “Haar-like features” dan “Histogram of Oriented Gradients” (HOG) hingga metode berbasis pembelajaran mendalam (“deep learning”). Metode konvensional, meskipun efektif dalam kondisi sederhana, sering kali mengalami keterbatasan dalam menghadapi variasi kompleks pada data gambar, seperti perubahan pencahayaan, latar belakang yang beragam, dan bentuk objek yang dinamis (Dalal & Triggs, 2005). Untuk mengatasi keterbatasan ini, metode berbasis “deep learning”, khususnya “Convolutional Neural Network” (CNN), telah dikembangkan dengan keunggulan yang lebih baik dalam mengenali pola kompleks dalam gambar (LeCun et al., 2015).

Model “deep learning” seperti CNN menjadi dasar dalam berbagai arsitektur deteksi objek modern, termasuk “You Only Look Once” (YOLO) dan “Region-based Convolutional Neural Networks” (R-CNN). YOLO, yang dikembangkan oleh Redmon et al. (2016), mampu melakukan deteksi objek secara real-time dengan kecepatan tinggi tanpa mengorbankan akurasi. Dalam penerapannya pada deteksi kematangan buah, YOLO memungkinkan sistem untuk mengidentifikasi buah secara cepat dan efektif, membuatnya sangat cocok untuk lingkungan produksi yang membutuhkan waktu respons cepat. Menurut penelitian Lin et al. (2020), penggunaan YOLO pada deteksi buah telah meningkatkan efisiensi proses pengklasifikasian objek hingga 30% dibandingkan dengan metode deteksi objek berbasis R-CNN.

R-CNN dan turunannya, seperti Faster R-CNN, adalah arsitektur lain yang sering digunakan dalam deteksi objek karena kemampuannya dalam menghasilkan deteksi yang lebih presisi meskipun memiliki waktu pemrosesan yang lebih lama (Ren et al., 2015). Dalam penelitian oleh Ren et al. (2015), Faster R-CNN terbukti sangat andal dalam mendeteksi objek dengan berbagai ukuran dan latar belakang yang kompleks, sehingga cocok untuk aplikasi deteksi objek yang membutuhkan akurasi tinggi.

Di bidang pertanian, CNN juga telah banyak digunakan untuk deteksi buah-buahan, dengan hasil yang sangat memuaskan dalam mengidentifikasi tingkat kematangan buah berdasarkan fitur visual yang diekstraksi dari permukaan buah (Sibomana et al., 2021). CNN mampu mendeteksi perubahan warna, tekstur, dan bentuk pada buah yang menjadi indikator kematangan secara otomatis, mengurangi kesalahan yang mungkin terjadi pada pengamatan manual. Studi yang dilakukan oleh Sibomana et al. (2021) menunjukkan bahwa penerapan CNN dalam deteksi kematangan buah meningkatkan akurasi klasifikasi hingga 25% dibandingkan metode tradisional.

Dengan berbagai keunggulan yang ditawarkan oleh teknologi “deep learning”, khususnya CNN, deteksi objek pada buah mangga dapat dilakukan dengan akurasi dan kecepatan yang lebih baik, mendukung proses pengambilan keputusan yang optimal dalam industri pertanian. Hal ini menjadi dasar penting dalam penerapan CNN untuk proyek deteksi kematangan buah mangga yang efisien dan dapat diandalkan.

## Segmentasi

Segmentasi adalah proses penting dalam pemrosesan citra yang bertujuan untuk membagi suatu gambar menjadi beberapa bagian atau wilayah yang lebih kecil berdasarkan karakteristik tertentu, seperti warna, tekstur, dan bentuk. Proses ini memungkinkan pemisahan objek yang menjadi fokus dari latar belakang atau elemen lain dalam gambar, yang selanjutnya mempermudah analisis lanjutan, seperti deteksi objek, pengenalan pola, atau klasifikasi (Gonzalez & Woods, 2018). Dalam aplikasi deteksi kematangan buah, segmentasi membantu menyoroti area buah yang relevan, seperti permukaan yang memperlihatkan perubahan warna atau tekstur, yang menjadi indikasi tingkat kematangan.

Menurut penelitian oleh Patel et al. (2020), segmentasi yang tepat pada gambar buah dapat meningkatkan akurasi deteksi kematangan hingga 20% dibandingkan dengan citra yang tidak disegmentasi. Segmentasi berfungsi untuk mengisolasi fitur yang relevan dari latar belakang atau gangguan lainnya, sehingga sistem dapat lebih fokus menganalisis perubahan visual pada objek yang menjadi indikator penting dari kematangan.

Beberapa teknik segmentasi yang umum digunakan meliputi segmentasi berbasis tepi, berbasis wilayah, dan berbasis clustering. Segmentasi berbasis tepi bekerja dengan mendeteksi tepi-tepi dalam gambar berdasarkan perubahan intensitas atau warna. Di sisi lain, segmentasi berbasis wilayah memanfaatkan kesamaan karakteristik dalam area tertentu, sedangkan segmentasi berbasis clustering, seperti “K-means clustering”, mengelompokkan piksel dengan karakteristik serupa ke dalam satu kelompok atau klaster (Jiang et al., 2021).

Dalam pengembangan lebih lanjut, teknik “deep learning” telah membawa kemajuan signifikan dalam segmentasi citra. Metode seperti “Fully Convolutional Networks” (FCN) dan “U-Net” memiliki kemampuan untuk melakukan segmentasi secara otomatis dengan akurasi yang tinggi, terutama pada gambar yang kompleks (Ronneberger et al., 2015). Berdasarkan studi dari Ronneberger et al. (2015), U-Net mampu memberikan performa segmentasi yang optimal dalam pemrosesan citra medis, dan metode ini mulai diadaptasi pada bidang pertanian untuk pemrosesan citra buah, termasuk deteksi kematangan.

Dengan menggunakan teknik segmentasi berbasis deep learning seperti FCN dan U-Net, sistem dapat melakukan segmentasi yang lebih baik pada permukaan buah mangga, yang memungkinkan deteksi kematangan yang lebih akurat. Menurut Zhang et al. (2019), penerapan U-Net dalam segmentasi citra buah telah terbukti mampu memisahkan area penting dengan noise yang lebih sedikit, sehingga meningkatkan akurasi klasifikasi pada tahap selanjutnya.

Implementasi segmentasi yang tepat sangat krusial dalam sistem deteksi kematangan berbasis CNN. Dengan memisahkan area yang relevan pada gambar buah, sistem dapat mengurangi pengaruh dari objek atau latar belakang yang tidak relevan, sehingga proses analisis menjadi lebih fokus dan efisien.

# Metodologi Penelitian

## Desain Sistem

Sistem pendeteksi tingkat kematangan buah mangga ini dirancang untuk menerima gambar sebagai input, memprosesnya melalui model Convolutional Neural Network (CNN), dan menghasilkan output berupa tingkat kematangan buah. Berikut adalah penjelasan singkat setiap komponen utama yang terdapat pada kode:

### Struktur Utama Kode

Kode program dibangun dalam beberapa bagian sesuai dengan pipeline deep learning. Dimulai dari penyusunan dataset, pra-pemrosesan data, arsitektur model, hingga proses pelatihan dan validasi. Setiap bagian dipisahkan dalam sel-sel Google Collaboratory sehingga memungkinkan analisis bertahap terhadap data dan model.

### Input dan Pra-Pemrosesan Gambar

Pada bagian awal, gambar input dikumpulkan dan diproses sesuai spesifikasi model. Langkah-langkah pra-pemrosesan ini mencakup perubahan ukuran gambar ke resolusi standar, konversi ke format yang sesuai (RGB), dan normalisasi. Contohnya, pada bagian kode pra-pemrosesan, fungsi `ImageDataGenerator` dari Keras digunakan untuk melakukan augmentasi gambar yang menghasilkan variasi baru dalam dataset. Teknik augmentasi, seperti rotasi, zoom, dan flip, memastikan model belajar dari variasi gambar yang lebih luas, meningkatkan kemampuan generalisasi.

### Pengembangan Model CNN

CNN digunakan karena kemampuannya untuk mengekstraksi fitur visual dari gambar input secara efektif. Bagian ini didefinisikan dengan menambahkan lapisan-lapisan konvolusi dan pooling. Misalnya, `Conv2D` dan `MaxPooling2D` digunakan secara bergantian untuk mengekstraksi dan mengurangi dimensi fitur dari gambar. Kode yang mendefinisikan arsitektur ini termasuk blok `Sequential` dari Keras, di mana lapisan dense dan lapisan dropout juga ditambahkan untuk mencegah overfitting selama pelatihan.

### Proses Pelatihan dan Evaluasi Model

Bagian pelatihan menggunakan data latih dan validasi. Model dioptimalkan menggunakan algoritma `Adam` dan menggunakan fungsi loss `categorical\_crossentropy` untuk mengukur kesalahan dalam klasifikasi. Setelah pelatihan, performa model dinilai menggunakan data validasi dan berbagai metrik evaluasi seperti akurasi dan F1-score. Setiap iterasi atau epoch dalam proses pelatihan dicatat untuk menganalisis perkembangan akurasi dan loss model, yang ditampilkan dalam grafik untuk melihat trend overfitting atau underfitting.

### Prediksi dan Implementasi

Setelah model dilatih, model disimpan dan siap digunakan pada aplikasi berbasis gambar. Implementasi akhir dapat dikembangkan sebagai aplikasi berbasis web atau seluler yang memungkinkan pengguna mengunggah gambar dan memperoleh prediksi tingkat kematangan mangga secara instan. Dalam kode, model disimpan menggunakan `model.save()` untuk mempermudah integrasi dengan aplikasi front-end nantinya.

## Dataset dan Pra-pemrosesan

Pada bagian ini, sistem memanfaatkan dataset gambar buah mangga dengan berbagai tingkat kematangan, yang kemudian melalui serangkaian proses untuk menyiapkan data agar sesuai dengan persyaratan model deep learning. Tahap ini terdiri dari pengumpulan dan penyusunan dataset, kemudian diikuti dengan teknik pra-pemrosesan untuk memaksimalkan kinerja model.

### Dataset

Dataset yang digunakan dalam proyek ini terdiri dari gambar-gambar mangga yang telah diklasifikasikan ke dalam beberapa kategori tingkat kematangan, seperti matang, setengah matang, dan mentah. Setiap gambar diberi label sesuai kategori tersebut, yang nantinya akan berfungsi sebagai target dalam pelatihan model. Pada kode, dataset diatur menggunakan struktur direktori yang berisi subfolder untuk setiap label kematangan, sehingga memudahkan proses pemuatan data secara terorganisir dengan `ImageDataGenerator` dari Keras.

### Pemuatan Data dan Pengaturan Batch

Untuk memudahkan pelatihan, data diatur dalam batch dan diotomatisasi proses pemuatannya menggunakan `ImageDataGenerator`. Pada bagian kode ini, `train\_datagen` dan `val\_datagen` digunakan untuk memisahkan data pelatihan dan validasi. Setiap generator dikonfigurasi dengan ukuran batch yang sesuai dan pengacakan (shuffle) untuk memastikan model belajar dari variasi data yang acak. Pemuatan batch memungkinkan pelatihan pada data yang besar tanpa kehabisan memori, yang merupakan aspek penting pada deep learning.

### Augmentasi Gambar

Agar model mampu mengenali pola visual yang bervariasi, teknik augmentasi gambar diterapkan. Dalam kode, augmentasi diterapkan melalui parameter `ImageDataGenerator` dengan beberapa opsi, seperti:

- Rotasi: Memutar gambar dalam rentang tertentu untuk membuat model lebih robust terhadap orientasi buah mangga.

- Zoom: Memperbesar atau memperkecil gambar, memungkinkan model mempelajari fitur dari skala yang berbeda.

- Flip: Membalik gambar secara horizontal, memberikan model lebih banyak contoh dari berbagai sudut pandang.

- Shear: Menggeser piksel secara proporsional, menambah variasi dalam posisi fitur gambar.

Augmentasi ini meningkatkan jumlah data efektif dan membantu model untuk belajar dari variasi tambahan yang mungkin muncul dalam penggunaan nyata.

### Normalisasi Gambar

Gambar yang diproses juga dinormalisasi dalam rentang nilai 0-1 untuk mempercepat pelatihan dan mencegah masalah numerik yang dapat timbul dari nilai piksel yang besar. Normalisasi dilakukan dengan membagi nilai piksel setiap gambar dengan 255, sehingga setiap piksel memiliki nilai dalam rentang standar ini. Langkah ini dilakukan langsung di `ImageDataGenerator` dengan mengatur parameter `rescale=1./255` pada bagian pra-pemrosesan data.

Pra-pemrosesan ini bertujuan untuk memastikan bahwa data input sudah siap untuk digunakan dalam pelatihan model CNN. Dengan langkah-langkah ini, model akan menerima data gambar yang lebih konsisten dan bervariasi, yang sangat penting untuk meningkatkan akurasi dan generalisasi saat memprediksi tingkat kematangan mangga dari gambar input baru.

## Arsitektur Model

Model yang digunakan dalam sistem pendeteksi kematangan buah mangga ini dibangun dengan arsitektur Convolutional Neural Network (CNN) untuk memanfaatkan kemampuannya dalam mengenali pola visual dari gambar. CNN dipilih karena kemampuannya yang unggul dalam mendeteksi fitur spasial yang ada pada gambar, seperti tekstur, warna, dan bentuk, yang penting dalam membedakan tingkat kematangan buah mangga. Berikut adalah penjelasan setiap komponen arsitektur model.

### Definisi Model Sequential

Model CNN didefinisikan menggunakan pendekatan `Sequential` dari Keras, yang memungkinkan penyusunan lapisan secara berurutan. Pendekatan ini memudahkan pengaturan struktur layer dalam model secara berurutan dan intuitif. Struktur model terdiri dari beberapa lapisan konvolusi yang diikuti dengan lapisan pooling dan diakhiri dengan lapisan fully connected yang berfungsi untuk klasifikasi tingkat kematangan.

### Lapisan Konvolusi (Convolutional Layers)

Lapisan konvolusi adalah komponen inti dari arsitektur CNN, yang berfungsi untuk mengekstraksi fitur-fitur penting dari gambar input. Setiap lapisan konvolusi memiliki sejumlah filter atau kernel yang berperan untuk mendeteksi pola visual tertentu. Pada bagian kode, lapisan-lapisan ini ditambahkan menggunakan `Conv2D` dengan fungsi aktivasi ReLU (`activation='relu'`) untuk memberikan non-linearitas, serta ukuran kernel yang disesuaikan (misalnya, `(3, 3)` atau `(5, 5)`). Semakin dalam model, jumlah filter meningkat untuk menangkap pola-pola visual yang semakin kompleks.

### Lapisan Pooling (Pooling Layers)

Setelah setiap lapisan konvolusi, lapisan pooling ditambahkan untuk mereduksi dimensi spasial dari fitur yang diekstraksi. Pengurangan ini dilakukan tanpa menghilangkan informasi penting, yang membantu mengurangi jumlah parameter dan mencegah overfitting. Kode menggunakan `MaxPooling2D`, yang memilih nilai maksimum dari setiap patch fitur, sehingga model fokus pada fitur dominan. Ukuran pooling biasanya `(2, 2)` untuk setiap lapisan pooling, yang mempercepat proses komputasi dan memperkuat fitur utama dalam data.

### Lapisan Dropout (Dropout Layer)

Untuk mengatasi overfitting, lapisan dropout digunakan setelah lapisan konvolusi dan lapisan dense. `Dropout` dengan nilai seperti `0.5` memastikan bahwa sejumlah node dalam lapisan tersebut akan dinonaktifkan secara acak selama pelatihan, membantu model belajar lebih banyak dari fitur data secara umum dan tidak hanya dari fitur spesifik. Kode ini menggunakan dropout terutama di lapisan-lapisan fully connected untuk menambah regularisasi.

### Lapisan Fully Connected (Dense Layer)

Di akhir model, lapisan fully connected (dense) bertindak sebagai klasifikasi. Lapisan ini menerima input dari lapisan sebelumnya yang telah diratakan (flatten) untuk menghasilkan output prediksi. Pada kode, lapisan terakhir adalah `Dense` dengan jumlah unit sesuai jumlah kelas kematangan, dilengkapi dengan fungsi aktivasi softmax (`activation='softmax'`) untuk menghasilkan probabilitas setiap kelas. Lapisan ini memastikan bahwa model menghasilkan output berupa probabilitas setiap kelas, seperti matang, setengah matang, atau mentah, sesuai dengan kebutuhan klasifikasi.

### Kompilasi Model

Setelah arsitektur terbentuk, model dikompilasi menggunakan algoritma optimasi Adam (`optimizer='adam'`) dan fungsi loss categorical cross-entropy (`loss='categorical\_crossentropy'`) yang sesuai untuk masalah klasifikasi multi-kelas. Selain itu, metrik akurasi ditambahkan untuk melacak kinerja model selama pelatihan. Kompilasi ini mempersiapkan model untuk proses pelatihan dengan parameter yang diatur.

Arsitektur model CNN yang terstruktur ini memungkinkan sistem untuk mengekstraksi fitur visual dari gambar buah mangga dengan efektif. Dengan konfigurasi yang tepat, model ini mampu melakukan klasifikasi tingkat kematangan berdasarkan pola visual yang terdapat dalam gambar.

## Pelatihan dan Validasi

Setelah arsitektur model dirancang, tahap selanjutnya adalah proses pelatihan untuk mengajarkan model dalam mengenali dan mengklasifikasikan tingkat kematangan buah mangga berdasarkan data gambar yang diberikan. Pelatihan model ini mencakup pembagian dataset, penggunaan fungsi loss dan optimizer, serta evaluasi performa model melalui data validasi.

### Pembagian Dataset

Dataset gambar dibagi menjadi dua bagian utama: data pelatihan dan data validasi. Data pelatihan digunakan untuk mengajarkan model mengenali pola dalam gambar dan mengaitkannya dengan tingkat kematangan yang benar. Sedangkan, data validasi digunakan untuk menilai performa model terhadap data yang tidak dilihat model selama pelatihan, sehingga dapat memberikan gambaran akurasi model di luar data pelatihan. Pada kode, pembagian ini dilakukan dengan menggunakan `ImageDataGenerator` yang dikonfigurasi untuk membuat dua generator berbeda bagi data pelatihan dan validasi.

### Proses Pelatihan

Model dilatih dengan menggunakan algoritma optimasi Adam, yang dikenal efektif untuk mempercepat konvergensi model pada masalah klasifikasi. Proses pelatihan diatur dalam beberapa epoch, yaitu sejumlah iterasi di mana model memproses seluruh data latih. Dalam kode, fungsi `fit` atau `fit\_generator` digunakan untuk melatih model pada data yang dihasilkan oleh generator, dan setiap epoch menghasilkan metrik seperti akurasi dan loss untuk menilai peningkatan kinerja model dari waktu ke waktu.

### Penggunaan Fungsi Loss dan Optimizer

Fungsi loss yang digunakan adalah categorical cross-entropy, yang cocok untuk masalah klasifikasi multi-kelas. Fungsi ini mengukur seberapa jauh prediksi model dari label sebenarnya, sehingga model dapat meminimalkan kesalahan ini. Optimizer Adam (`optimizer='adam'`) dipilih karena efisiensi dan kestabilannya dalam memperbarui bobot model, bahkan pada dataset yang besar dan kompleks.

### Pemantauan Metrik Evaluasi

Selama proses pelatihan, metrik akurasi dipantau untuk melihat seberapa baik model mengenali kelas tingkat kematangan dengan benar. Setelah setiap epoch, model menghitung akurasi pada data pelatihan dan validasi untuk mendeteksi tanda-tanda overfitting. Jika akurasi pelatihan meningkat tetapi akurasi validasi menurun, ini menunjukkan bahwa model mungkin mengalami overfitting. Dalam kode, grafik akurasi dan loss biasanya ditampilkan untuk setiap epoch, memungkinkan visualisasi proses pelatihan.

### Evaluasi Akhir dengan Data Validasi

Setelah pelatihan selesai, model diuji secara keseluruhan pada data validasi untuk menilai performanya dalam kondisi yang mirip dengan penggunaan nyata. Metrik seperti akurasi, precision, recall, dan F1-score dapat digunakan untuk analisis yang lebih mendalam tentang seberapa baik model menangani setiap kelas kematangan. Nilai-nilai ini menunjukkan kemampuan model untuk mengklasifikasikan gambar mangga dengan benar sesuai tingkat kematangannya.

Tahap pelatihan dan validasi ini memungkinkan model untuk belajar dari pola dalam dataset dan menilai kemampuannya dalam menggeneralisasi pola tersebut ke gambar baru. Dengan pengaturan epoch yang tepat, pemilihan fungsi loss, dan evaluasi yang teliti, model dapat mencapai akurasi yang tinggi dalam pendeteksian tingkat kematangan buah mangga.