# UNIVERSITAS GUNADARMA

# PRAKTIKUM KECERDASAN ARTIFICIAL

**MANUAL BOOKS**

**“Flowers Recognition CNN”**


Nama : Harbangan Panjaitan

NPM : 50423573

Kelas : 3IA18

Fakultas : Teknologi Industri

Jurusan : Teknik Informatika


Ditulis Guna Melengkapi Sebagian Syarat

Praktikum Kecerdasan Artificial

Universitas Gunadarma

**2025**

# DAFTAR ISI

Contents

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang

Pengolahan citra digital menjadi salah satu bidang penting dalam dunia Artificial Intelligence (AI), terutama dalam mengembangkan kemampuan komputer untuk memahami objek visual. Di era digital seperti sekarang, kebutuhan akan sistem yang dapat mengenali objek secara otomatis semakin meningkat, baik untuk keperluan industri, penelitian, maupun aplikasi sehari-hari. Salah satu aplikasi yang kini banyak dipelajari adalah sistem pengenalan bunga, di mana komputer dilatih untuk membedakan setiap jenis bunga berdasarkan karakteristik visualnya seperti bentuk kelopak, warna, tekstur, dan pola-pola unik lainnya.

Dengan munculnya metode deep learning seperti Convolutional Neural Network (CNN), proses identifikasi objek visual dapat dilakukan secara otomatis tanpa memerlukan ekstraksi fitur manual yang biasanya memakan waktu cukup lama dan membutuhkan keahlian khusus dalam bidang computer vision. Berbeda dengan metode tradisional yang mengharuskan kita untuk mendefinisikan fitur-fitur secara eksplisit, CNN mampu belajar sendiri fitur-fitur penting dari data gambar yang diberikan selama proses training. Hal ini membuat CNN sangat ideal untuk tugas klasifikasi bunga yang memiliki variasi visual yang cukup kompleks, seperti perbedaan bentuk, ukuran, sudut pengambilan gambar, kondisi pencahayaan, dan background yang berbeda-beda.

Notebook yang digunakan dalam proyek ini memuat seluruh proses, mulai dari pemuatan dataset bunga, preprocessing untuk menyiapkan data agar siap diproses, pembangunan model CNN sederhana dengan beberapa layer yang dirancang khusus untuk tugas klasifikasi gambar, pelatihan model menggunakan data training dan validasi, hingga pengujian terhadap gambar baru untuk melihat seberapa baik model dapat memprediksi jenis bunga yang belum pernah dilihat sebelumnya. Manual book ini disusun sebagai panduan yang menjelaskan seluruh proses tersebut secara runtut dan mudah dipahami, sehingga pembaca yang masih dalam tahap belajar pun dapat mengikuti alur kerja sistem dengan baik.

## 1.2 Tujuan

Manual book ini bertujuan memberikan penjelasan yang jelas dan detail mengenai bagaimana sistem pengenalan bunga dibangun menggunakan CNN dari awal hingga akhir. Segala proses mulai dari penjelasan konsep dasar AI dan pentingnya dalam kehidupan modern, teori dasar CNN beserta komponen-komponennya seperti convolutional layer, pooling layer, dan fully connected layer, cara pemrosesan dataset mulai dari loading hingga

augmentasi data, hingga bagaimana model dilatih dengan parameter-parameter tertentu dan diuji untuk mengukur performanya dijelaskan secara runtut berdasarkan isi file notebook yang digunakan.

Selain itu, manual book ini juga bertujuan untuk memberikan pemahaman praktis kepada pembaca tentang implementasi teknik deep learning dalam menyelesaikan permasalahan nyata di bidang computer vision. Dengan demikian, pembaca dapat memahami alur kerja sistem secara menyeluruh, mulai dari konsep teoritis hingga implementasi praktisnya, dan dapat menerapkannya kembali dengan mudah baik untuk proyek serupa maupun untuk pengembangan sistem yang lebih kompleks di masa depan.

# BAB II

# PEMBAHASAN

## 2.1 Artificial Intelligence

Artificial Intelligence (AI) adalah sebuah bidang yang berfokus pada usaha membuat mesin mampu melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia, seperti berpikir, belajar, mengambil keputusan, dan mengenali pola. Dalam perkembangannya, AI telah mengalami berbagai tahapan evolusi mulai dari sistem berbasis aturan (rule-based systems) hingga machine learning yang dapat belajar dari data tanpa diprogram secara eksplisit untuk setiap kemungkinan yang ada. Dalam konteks pengenalan gambar, AI dipadukan dengan teknik-teknik deep learning agar komputer mampu belajar langsung dari data visual yang diberikan. Berbeda dengan pendekatan konvensional yang memerlukan feature engineering secara manual, pendekatan deep learning memungkinkan model untuk secara otomatis menemukan representasi fitur yang optimal dari data mentah. Melalui proses pembelajaran tersebut, komputer dapat memahami pola dan ciri tertentu yang mungkin tidak terlihat jelas oleh mata manusia, sehingga mampu mengenali berbagai objek di dalam gambar dengan tingkat akurasi yang tinggi, termasuk bunga dengan berbagai variasinya.

AI dalam pengenalan gambar juga terus berkembang dengan berbagai inovasi seperti transfer learning, data augmentation, dan arsitektur neural network yang semakin canggih. Hal ini membuat sistem pengenalan objek semakin akurat dan dapat diterapkan pada berbagai domain aplikasi seperti kesehatan, pertanian, keamanan, hingga e-commerce.

## 2.2 Convolutional Neural Network (CNN)

CNN adalah model deep learning yang secara khusus dirancang untuk menganalisis data yang memiliki struktur dua dimensi seperti gambar, meskipun juga dapat digunakan untuk data dengan dimensi lain. Arsitektur CNN terinspirasi dari cara kerja visual cortex pada otak manusia yang memproses informasi visual secara hierarkis, dimulai dari deteksi fitur sederhana hingga fitur yang lebih kompleks.

CNN bekerja dengan cara mengekstraksi pola visual mulai dari tepi (edges), tekstur, hingga bentuk objek yang lebih kompleks melalui lapisan konvolusi dan pooling. Lapisan konvolusi bertugas untuk mendeteksi fitur-fitur lokal dengan menggunakan filter atau kernel yang bergerak melintasi gambar, sementara lapisan pooling berfungsi untuk mengurangi dimensi spasial dari representasi fitur sehingga komputasi menjadi lebih efisien dan model lebih tahan terhadap variasi kecil pada posisi objek.

Seiring bertambahnya kedalaman jaringan, CNN mampu memahami pola visual yang semakin kompleks dan abstrak. Layer-layer awal biasanya menangkap fitur-fitur sederhana seperti garis dan sudut, sedangkan layer-layer yang lebih dalam dapat mengenali bentuk-bentuk yang lebih kompleks seperti kelopak bunga, batang, atau bagian-bagian spesifik lainnya. Dalam proyek ini, CNN digunakan untuk membedakan lima jenis bunga berdasarkan karakteristik visual yang dimiliki masing-masing jenis bunga.

Keunggulan CNN dibandingkan dengan neural network biasa adalah kemampuannya dalam mempertahankan informasi spasial dari gambar, sehingga posisi dan hubungan antar piksel tetap terjaga selama proses pembelajaran. Selain itu, CNN juga menggunakan parameter sharing yang membuat jumlah parameter yang perlu dipelajari menjadi lebih sedikit dibandingkan dengan fully connected network, sehingga training menjadi lebih cepat dan risiko overfitting berkurang.

## 2.3 Dataset

Dataset bunga yang digunakan disimpan dalam folder flowers/ dan terdiri atas lima jenis bunga yang berbeda. Dataset ini menjadi komponen fundamental dalam proses pembelajaran mesin karena kualitas dan kuantitas data akan sangat mempengaruhi performa model yang dihasilkan. Setiap jenis bunga dalam dataset memiliki karakteristik visual yang unik, mulai dari bentuk kelopak, warna dominan, tekstur permukaan, hingga struktur keseluruhan bunga.

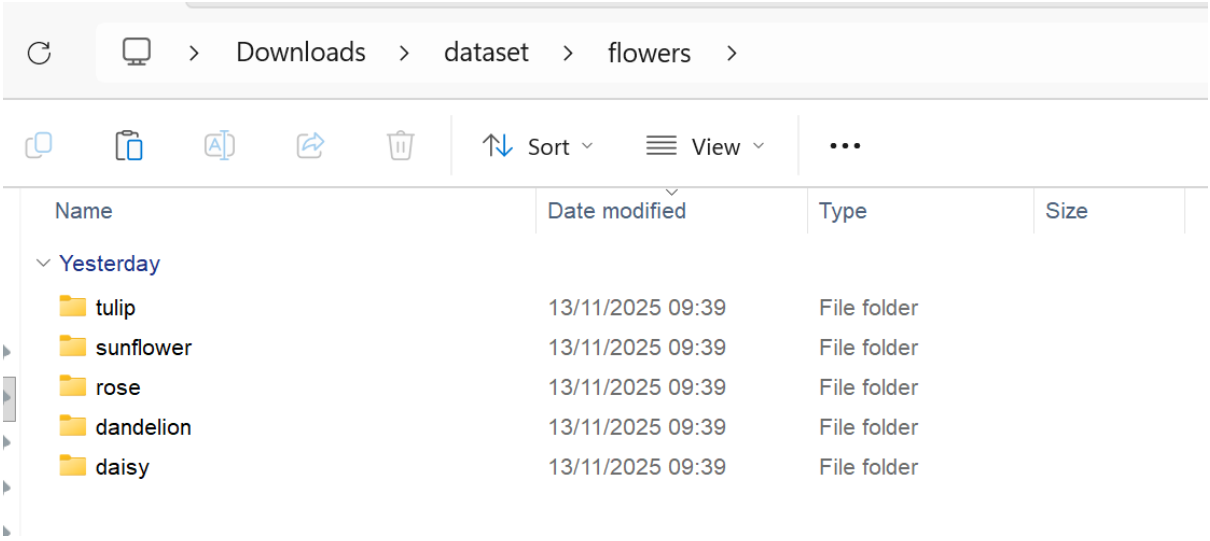
Setiap gambar pada dataset ini diubah ukurannya menjadi 150×150 piksel sebelum diproses oleh model untuk memastikan konsistensi input dan efisiensi komputasi. Proses resizing ini penting karena CNN memerlukan input dengan dimensi yang tetap, dan ukuran 150×150 dipilih sebagai kompromi antara detail informasi yang tersimpan dan kecepatan pemrosesan. Setelah itu, nilai piksel juga dinormalisasi ke dalam rentang 0 hingga 1 agar proses training menjadi lebih stabil dan konvergen lebih cepat.

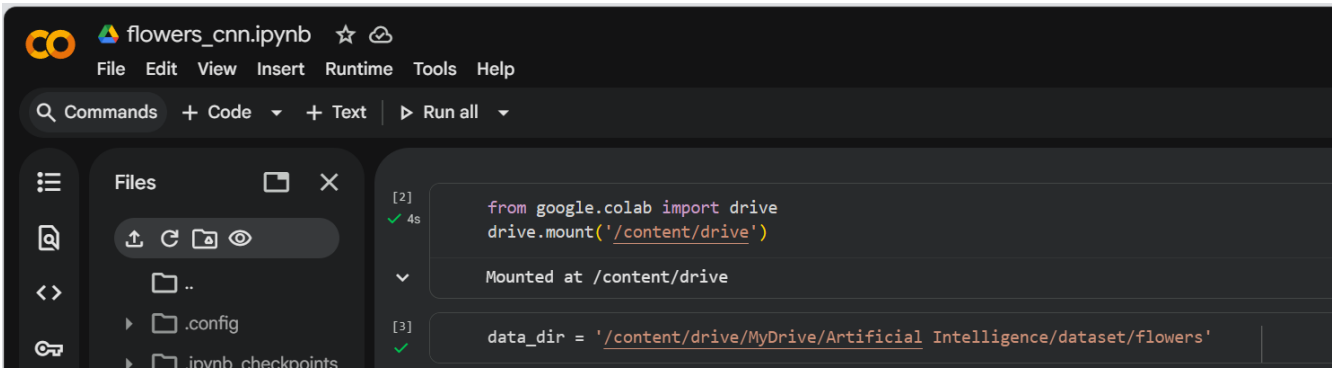
Dataset dibagi ke dalam tiga bagian, yaitu data latih (training set) yang digunakan untuk melatih model, data validasi (validation set) yang digunakan untuk memonitor performa model selama training dan melakukan hyperparameter tuning, dan data uji (test set) yang digunakan untuk evaluasi akhir performa model pada data yang benar-benar belum pernah dilihat sebelumnya. Pembagian ini diperlukan agar proses pelatihan lebih terkontrol dan performa model dapat dievaluasi dengan akurat tanpa bias.

Biasanya proporsi pembagian yang umum digunakan adalah 70% untuk training, 15% untuk validasi, dan 15% untuk testing, meskipun proporsi ini dapat disesuaikan tergantung pada jumlah total data yang tersedia. Dengan pembagian yang tepat, kita dapat memastikan bahwa model tidak hanya menghafal data training (memorization) tetapi benar-benar belajar pola umum yang dapat digeneralisasi ke data baru (generalization).

# BAB III
# ANALISA DAN PERANCANGAN

Proses pembangunan sistem pengenalan bunga dimulai dengan tahapan analisis kebutuhan, di mana perangkat pendukung seperti Python sebagai bahasa pemrograman utama, TensorFlow/Keras sebagai framework deep learning yang powerful dan mudah digunakan, Google Colab sebagai platform cloud computing yang menyediakan GPU gratis untuk mempercepat training, serta dataset gambar bunga yang sudah dikumpulkan dan disiapkan terlebih dahulu. Pada tahap ini juga dilakukan identifikasi terhadap spesifikasi hardware yang dibutuhkan, library-library tambahan yang perlu diinstall, dan struktur project secara keseluruhan.

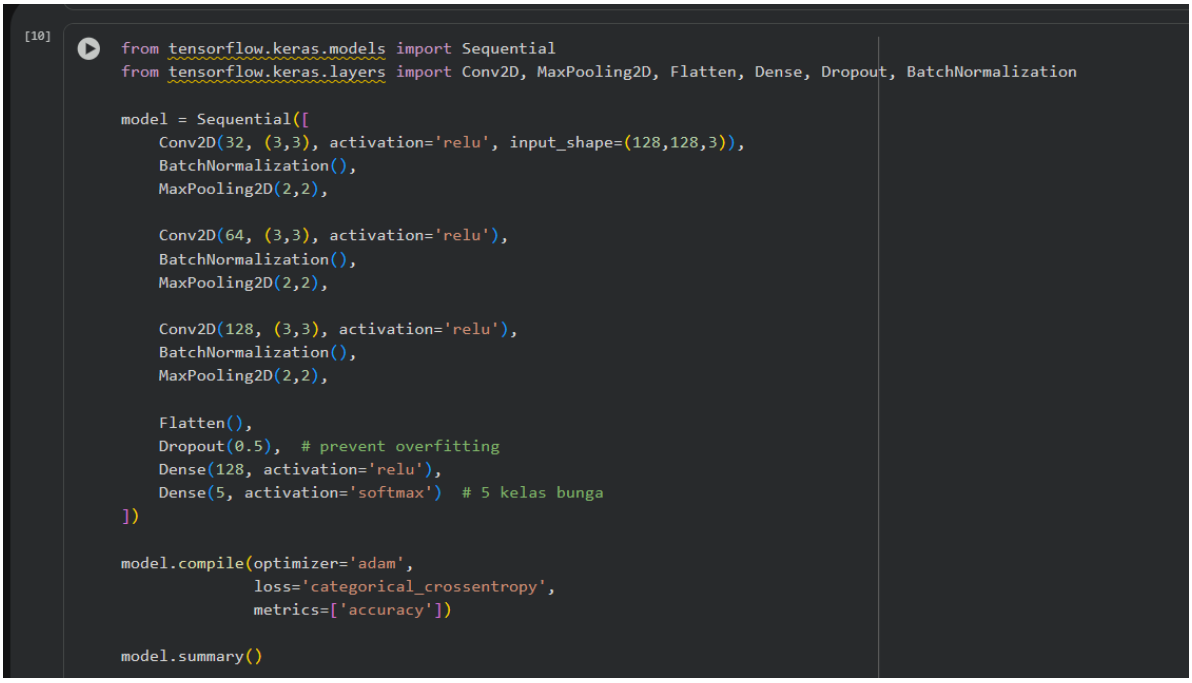


Setelah semua kebutuhan terpenuhi, dataset yang telah dikumpulkan diproses dengan cara mengubah resolusi gambar menjadi 150×150 piksel dan melakukan normalisasi pixel sehingga seluruh nilai berada pada rentang 0-1 yang membuat data siap digunakan model. Normalisasi ini dilakukan dengan membagi setiap nilai piksel (yang awalnya berkisar 0-255) dengan 255. Proses preprocessing ini sangat penting karena neural network bekerja lebih baik ketika input data berada pada skala yang seragam dan tidak terlalu besar.

```
[9]
✓ 0s
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Image Augmentation untuk akurasi tinggi
train_datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2,  # 20% data untuk validasi
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True
)

test_datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)

# Size gambar standar 128x128 px, batch 32
img_size = (128, 128)
batch_size = 32

train_generator = train_datagen.flow_from_directory(
    data_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
    subset='training',
    shuffle=True,
    seed=42
)

val_generator = test_datagen.flow_from_directory(
    data_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
    subset='validation',
    shuffle=True,
    seed=42
)

Found 3457 images belonging to 5 classes.
Found 860 images belonging to 5 classes.
```

Pada tahapan ini, seluruh gambar juga dikelompokkan ke dalam folder sesuai kelasnya (misalnya folder 'roses' untuk bunga mawar, folder 'tulips' untuk bunga tulip, dan seterusnya) sehingga mempermudah proses pelatihan menggunakan image generator dari Keras. Image generator ini sangat membantu karena dapat melakukan loading data secara batch, sehingga tidak semua gambar perlu dimuat ke memory sekaligus yang bisa menyebabkan out of memory error terutama jika dataset berukuran besar.



```
[10]
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization

model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(128,128,3)),
    BatchNormalization(),
    MaxPooling2D(2,2),

    Conv2D(64, (3,3), activation='relu'),
    BatchNormalization(),
    MaxPooling2D(2,2),

    Conv2D(128, (3,3), activation='relu'),
    BatchNormalization(),
    MaxPooling2D(2,2),

    Flatten(),
    Dropout(0.5),  # prevent overfitting
    Dense(128, activation='relu'),
    Dense(5, activation='softmax')  # 5 kelas bunga
])

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

model.summary()
```

Model CNN yang digunakan dalam notebook dibangun dengan arsitektur sederhana namun efektif untuk tugas klasifikasi gambar bunga. Model dimulai dengan lapisan konvolusi pertama yang menggunakan sejumlah filter (misalnya 32 atau 64 filter) dengan ukuran kernel tertentu (biasanya 3x3) yang berfungsi mengekstraksi fitur visual dasar dari gambar seperti tepi horizontal, vertikal, dan diagonal. Setiap filter akan belajar mendeteksi pola tertentu yang berguna untuk membedakan antar kelas.

```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/convolutional/base_conv.py:113: UserWarning: Do not pass an `input_shape`/`input_dim` argument to
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Model: "sequential"
```

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d (Conv2D) | (None, 126, 126, 32) | 896 |
| batch_normalization (BatchNormalization) | (None, 126, 126, 32) | 128 |
| max_pooling2d (MaxPooling2D) | (None, 63, 63, 32) | 0 |
| conv2d_1 (Conv2D) | (None, 61, 61, 64) | 18,496 |
| batch_normalization_1 (BatchNormalization) | (None, 61, 61, 64) | 256 |
| max_pooling2d_1 (MaxPooling2D) | (None, 30, 30, 64) | 0 |
| conv2d_2 (Conv2D) | (None, 28, 28, 128) | 73,856 |
| batch_normalization_2 (BatchNormalization) | (None, 28, 28, 128) | 512 |
| max_pooling2d_2 (MaxPooling2D) | (None, 14, 14, 128) | 0 |
| flatten (Flatten) | (None, 25088) | 0 |
| dropout (Dropout) | (None, 25088) | 0 |
| dense (Dense) | (None, 128) | 3,211,392 |
| dense_1 (Dense) | (None, 5) | 645 |

```
Total params: 3,306,181 (12.61 MB)
Trainable params: 3,305,733 (12.61 MB)
Non-trainable params: 448 (1.75 KB)
```

Setelah lapisan konvolusi, dilanjutkan dengan lapisan pooling (biasanya MaxPooling2D) untuk mereduksi dimensi spasial dari feature maps yang dihasilkan oleh lapisan konvolusi. Pooling tidak hanya mengurangi jumlah parameter sehingga komputasi lebih cepat, tetapi juga membuat model lebih robust terhadap pergeseran atau distorsi kecil pada gambar input. Proses ini diulang pada lapisan berikutnya dengan jumlah filter yang biasanya ditingkatkan (misalnya dari 32 menjadi 64, lalu menjadi 128) sehingga model mampu memahami pola visual yang lebih kompleks dan hierarkis.

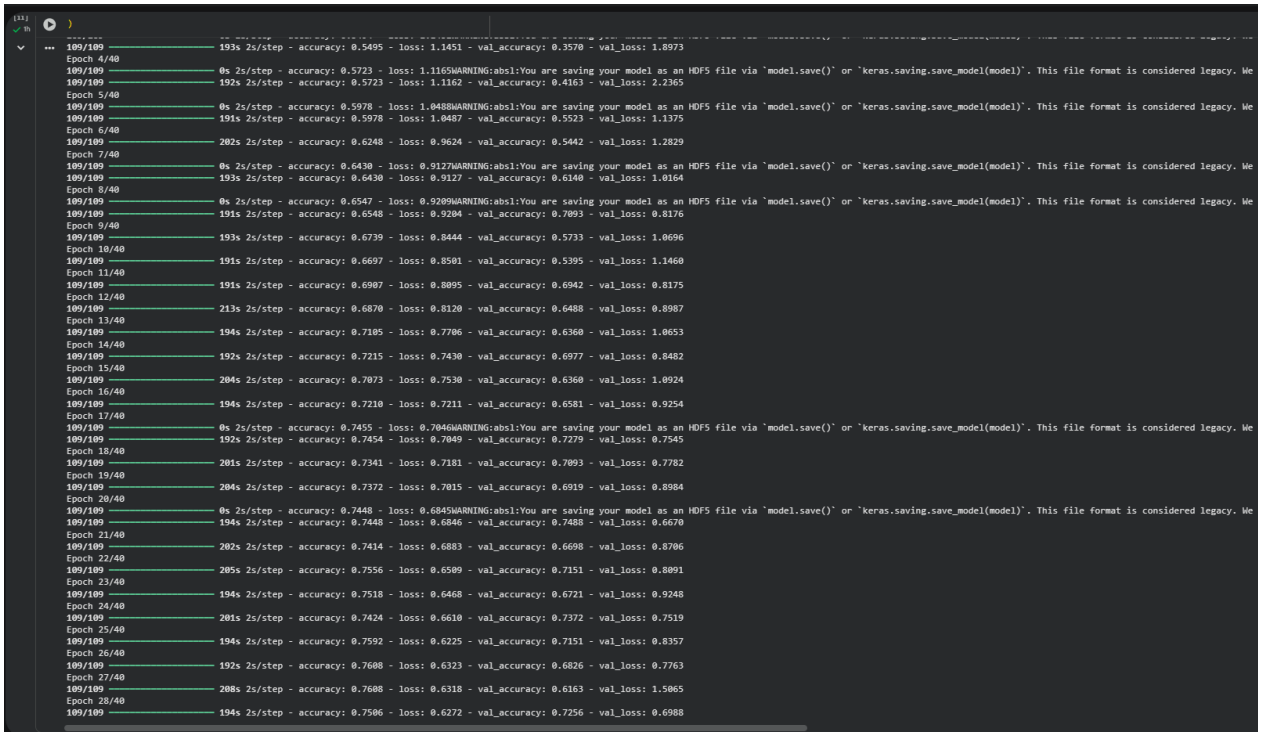
Setelah beberapa blok konvolusi-pooling, fitur-fitur yang dihasilkan diubah menjadi bentuk satu dimensi melalui proses flattening. Layer Flatten ini mengkonversi output dari layer konvolusi yang masih berbentuk matriks 3D menjadi vector 1D yang dapat diproses oleh fully connected layer. Model kemudian melanjutkan proses pengenalan pola menggunakan lapisan dense (fully connected layer) yang terdiri dari beberapa neuron.

```
[11]
✓ 1h
```

```python
# Early stopping mencegah overfitting dan save model terbaik
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint

early_stop = EarlyStopping(monitor='val_loss', patience=8, restore_best_weights=True)
checkpoint = ModelCheckpoint('best_model_flowers.h5', monitor='val_accuracy', save_best_only=True)

history = model.fit(
    train_generator,
    epochs=40,
    validation_data=val_generator,
    callbacks=[early_stop, checkpoint]
)
```

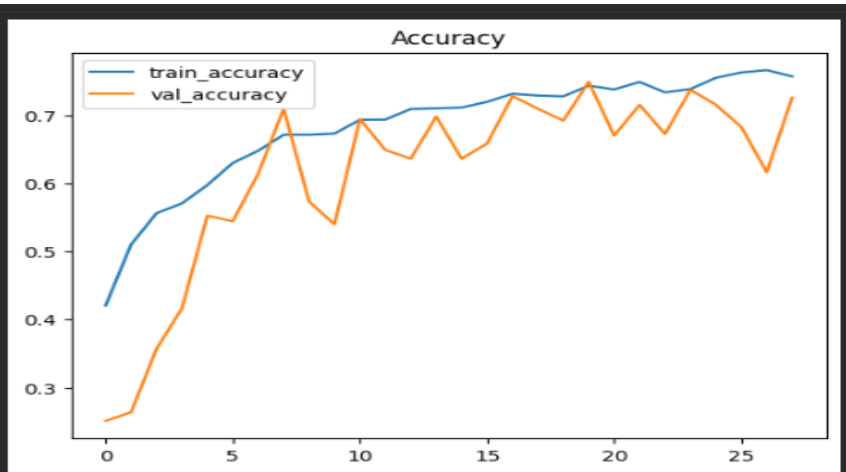
Biasanya ditambahkan juga teknik regularisasi seperti Dropout untuk mencegah overfitting dengan cara secara random "mematikan" sebagian neuron selama training. Layer dense terakhir menggunakan activation function softmax yang menghasilkan probabilitas untuk setiap kelas, sehingga model dapat memberikan prediksi berupa salah satu dari lima kelas bunga beserta tingkat kepercayaan (confidence score) dari prediksi tersebut.

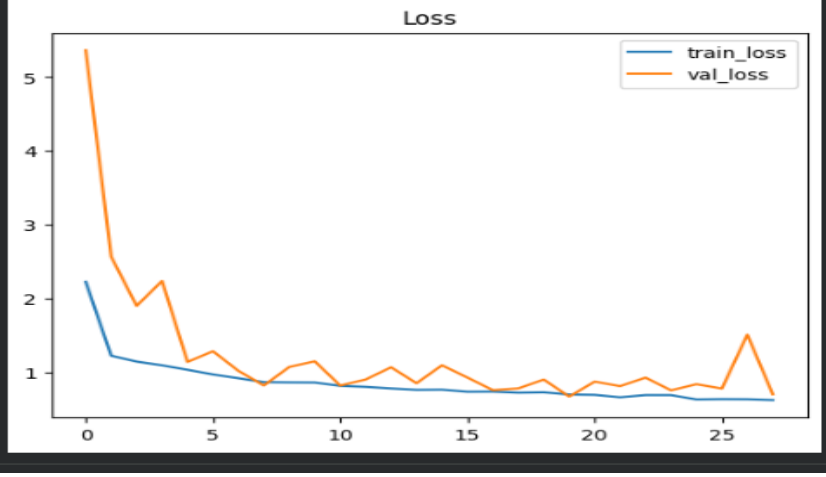
Pelatihan dilakukan dengan optimizer Adam yang merupakan salah satu optimizer paling populer karena menggabungkan kelebihan dari momentum dan RMSprop, sehingga dapat melakukan adaptasi learning rate secara otomatis untuk setiap parameter. Fungsi loss yang digunakan adalah categorical crossentropy yang cocok untuk masalah klasifikasi multi-class. Model dilatih selama beberapa epoch (misalnya 20-50 epoch tergantung pada kompleksitas data dan convergence dari model), dan setiap epoch dilakukan evaluasi terhadap data validasi untuk melihat apakah model belajar dengan baik atau mengalami overfitting yang ditandai dengan meningkatnya validation loss meskipun



[12]
✓ 1s

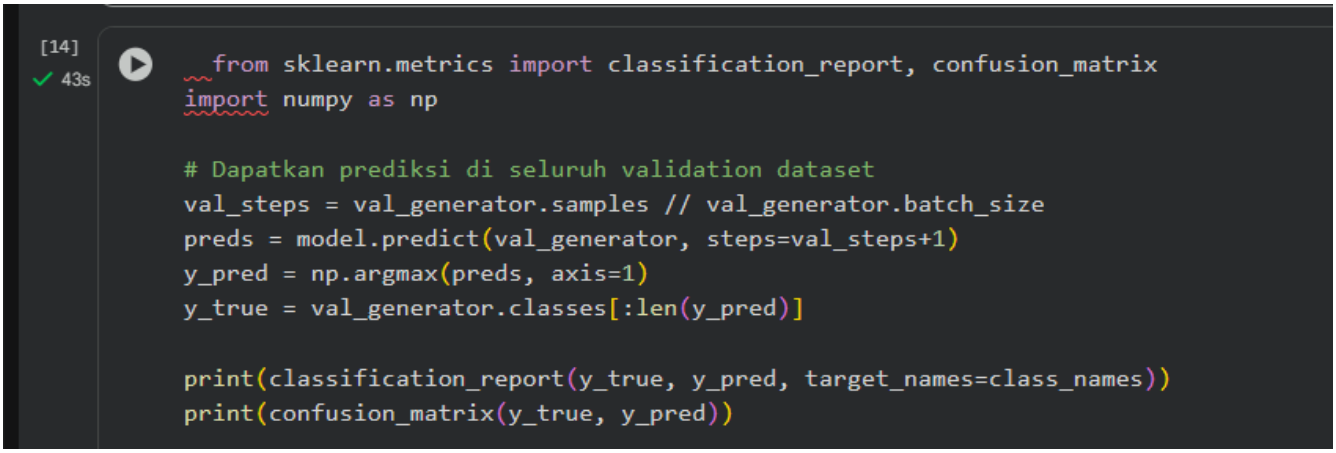
```
import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'], label='train_accuracy')
plt.plot(history.history['val_accuracy'], label='val_accuracy')
plt.legend()
plt.title('Accuracy')
plt.show()

plt.plot(history.history['loss'], label='train_loss')
plt.plot(history.history['val_loss'], label='val_loss')
plt.legend()
plt.title('Loss')
plt.show()
```



Selama proses training, perlu dimonitor beberapa metrik penting seperti training accuracy, validation accuracy, training loss, dan validation loss. Grafik yang menunjukkan perubahan metrik-metrik ini dari epoch ke epoch sangat berguna untuk memahami bagaimana model belajar. Jika terlihat gap yang terlalu besar antara training dan validation metrics, maka kemungkinan terjadi overfitting dan perlu dilakukan adjustment seperti menambahkan regularisasi, mengurangi kompleksitas model, atau menambah jumlah data training.

Setelah pelatihan selesai dan didapatkan model yang cukup baik berdasarkan validation metrics, performa model dievaluasi lebih lanjut menggunakan data uji yang benar-benar belum pernah dilihat oleh model selama proses training maupun validasi. Evaluasi pada test set ini memberikan gambaran objektif tentang seberapa baik model dapat digeneralisasi ke data baru di dunia nyata.

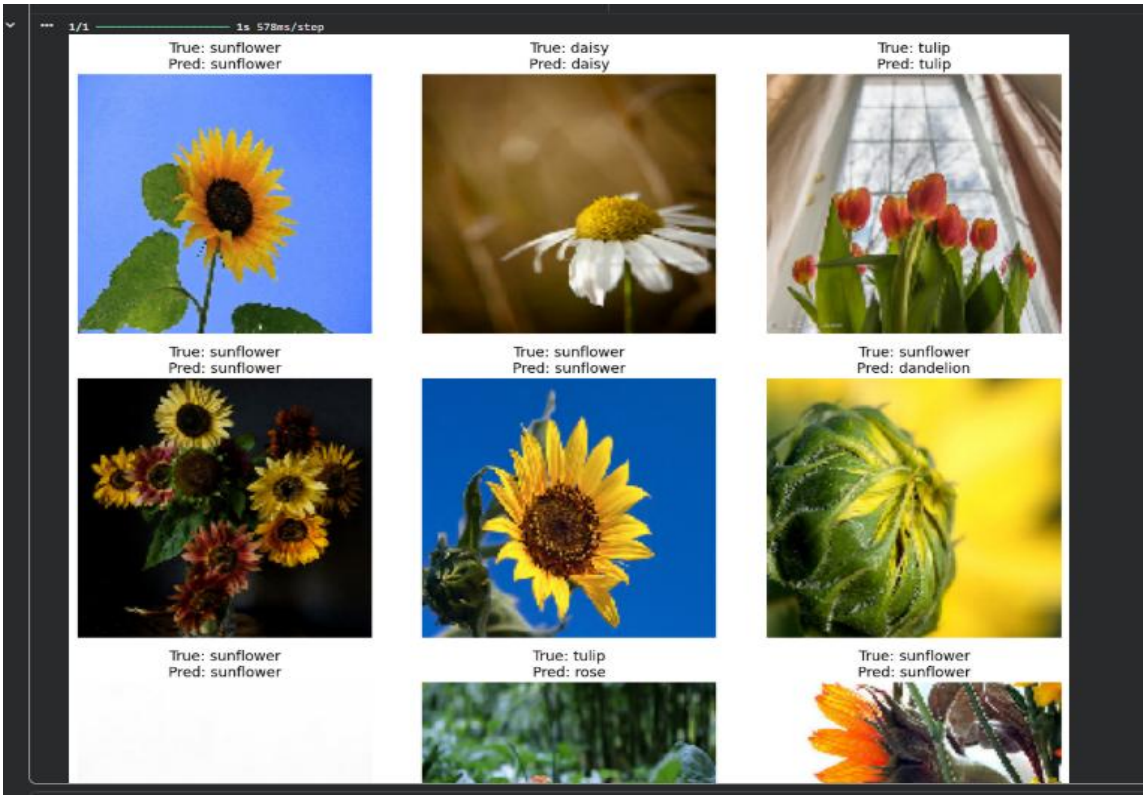


```
27/27 ━━━━━━━━━━━━━━━━━━━━ 22s 728ms/step
                precision    recall  f1-score   support

         daisy       0.15      0.16      0.16       152
     dandelion       0.28      0.25      0.26       210
          rose       0.16      0.17      0.17       156
     sunflower       0.16      0.21      0.18       146
         tulip       0.19      0.14      0.16       196

      accuracy                           0.19       860
     macro avg       0.19      0.19      0.19       860
  weighted avg       0.19      0.19      0.19       860

[[24 39 24 41 24]
 [34 53 48 43 32]
 [26 35 27 31 37]
 [26 29 36 31 24]
 [47 36 35 50 28]]
```

Confusion matrix juga dapat dibuat untuk melihat detail prediksi model, seperti kelas mana yang sering salah diprediksi sebagai kelas lain. Informasi ini sangat berguna untuk memahami kelemahan model dan melakukan improvement, misalnya dengan menambah data untuk kelas yang sering salah diprediksi atau melakukan feature engineering tambahan.

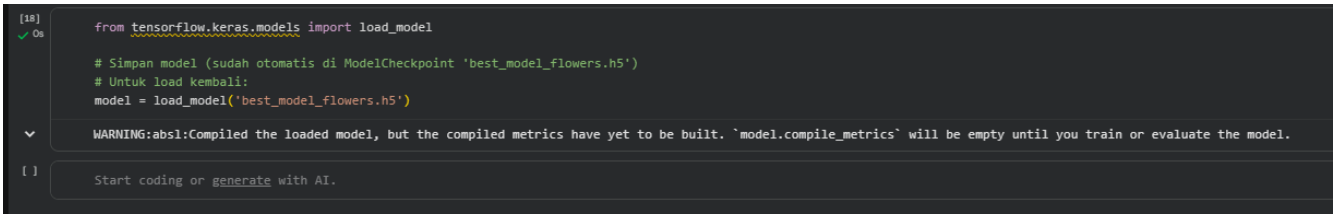
```
[13]
✓ 4s
```

```python
import numpy as np

images, labels = next(val_generator)  # satu batch
preds = model.predict(images)
pred_labels = np.argmax(preds, axis=1)
true_labels = np.argmax(labels, axis=1)

class_names = list(val_generator.class_indices.keys())

plt.figure(figsize=(12, 12))
for i in range(9):
    plt.subplot(3,3, i+1)
    plt.imshow(images[i])
    plt.title(f"True: {class_names[true_labels[i]]}\nPred: {class_names[pred_labels[i]]}")
    plt.axis('off')
plt.tight_layout()
plt.show()
```




Selanjutnya, model diuji kembali untuk memprediksi satu gambar baru dengan cara membaca gambar dari file, memprosesnya menjadi ukuran 150×150 piksel sama seperti data training, melakukan normalisasi nilai piksel, dan mengubah shape gambar agar sesuai dengan input yang diharapkan model (biasanya perlu ditambahkan batch dimension). Setelah itu, model akan memberikan hasil prediksi berdasarkan probabilitas kelas tertinggi, dan hasilnya dapat ditampilkan kepada user beserta confidence score-nya.

```
[16]
✓ 0s
    from tensorflow.keras.preprocessing import image

    img_path = '/content/drive/MyDrive/Artificial Intelligence/dataset/flowers/daisy/506348009_9ecff8b6ef.jpg'
    img = image.load_img(img_path, target_size=img_size)
    img_array = image.img_to_array(img) / 255.0
    img_array = np.expand_dims(img_array, axis=0)

    pred = model.predict(img_array)
    pred_class = class_names[np.argmax(pred)]

    print("Predicted class:", pred_class)

    1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 376ms/step
    Predicted class: daisy
```




Proses prediksi ini menunjukkan bahwa model yang telah dilatih dapat digunakan untuk aplikasi real-time, di mana user dapat mengupload gambar bunga dan sistem akan secara otomatis memberikan informasi tentang jenis bunga tersebut. Ke depannya, sistem ini dapat dikembangkan lebih lanjut dengan menambahkan fitur-fitur seperti web interface yang user-friendly, mobile application untuk memudahkan akses, atau bahkan integrasi dengan database yang lebih lengkap untuk memberikan informasi tambahan tentang karakteristik, cara perawatan, dan fakta menarik tentang setiap jenis bunga.

# BAB IV
# PENUTUP

## 4.1 Kesimpulan

Melalui proses yang dilakukan dalam file notebook, dapat disimpulkan bahwa CNN sederhana mampu memberikan hasil prediksi yang baik untuk klasifikasi gambar bunga. Meskipun arsitektur yang digunakan tidak terlalu kompleks dibandingkan dengan state-of-the-art models seperti ResNet atau EfficientNet, namun dengan preprocessing yang tepat, arsitektur model yang sesuai dengan kompleksitas permasalahan, serta pembagian dataset yang benar dan proporsional, model dapat mengenali pola visual dengan cukup akurat.

Sistem ini tidak hanya bermanfaat sebagai sarana pembelajaran untuk memahami konsep deep learning dan computer vision, tetapi juga dapat dikembangkan lebih lanjut untuk aplikasi skala lebih besar. Beberapa improvement yang dapat dilakukan di masa depan antara lain adalah menambah jumlah dan variasi dataset untuk meningkatkan generalisasi model, menggunakan teknik data augmentation untuk memperkaya variasi data training, mengimplementasikan transfer learning dengan memanfaatkan pre-trained model seperti VGG16 atau ResNet yang sudah dilatih pada dataset besar seperti ImageNet, serta melakukan hyperparameter tuning yang lebih optimal untuk mendapatkan performa terbaik.

Selain itu, sistem ini juga dapat diintegrasikan dengan aplikasi mobile atau web sehingga dapat digunakan oleh masyarakat umum untuk mengidentifikasi jenis bunga dengan mudah hanya melalui foto. Pengembangan lebih lanjut juga dapat mencakup penambahan jumlah kelas bunga yang dapat dikenali, implementasi confidence threshold untuk menolak prediksi yang tidak yakin, serta penambahan fitur-fitur tambahan seperti informasi detail tentang bunga dan rekomendasi perawatan.

Secara keseluruhan, proyek ini memberikan pemahaman yang baik tentang end-to-end process dalam membangun sistem computer vision menggunakan deep learning, mulai dari persiapan data, building model, training, evaluation, hingga deployment untuk prediksi data baru. Pengalaman ini menjadi fondasi yang kuat untuk mengerjakan proyek-proyek machine learning yang lebih kompleks di masa depan.

# DAFTAR PUSTAKA


Ian Goodfellow, Yoshua Bengio, Aaron Courville. *Deep Learning*. MIT Press.

Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.

TensorFlow Documentation. https://www.tensorflow.org/

Dataset Flowers Image Classification (open datasets).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Keras Documentation. https://keras.io/