Proyek Klasifikasi Gambar: CATS and DOGS

- **Nama:** FAIZAL RIZA
- **Email:** akuisal@gmail.com
- **ID Dicoding:** akuisal@gmail.com

Import Semua Packages/Library yang Digunakan

```
# Install TensorFlow
!pip install tensorflow
!pip install tensorflow-text

# Install package lain yang diperlukan
!pip install opencv-python
!pip install matplotlib
!pip install pandas
!pip install numpy
!pip install scikit-learn
!pip install tensorflowjs

# Install kaggle package untuk mengunduh dataset
!pip install kaggle

# Import library
import os
import cv2
import shutil
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pathlib import Path
import tensorflow as tf
from tensorflow.keras import layers, models
import tensorflowjs as tfjs
from sklearn.model_selection import train_test_split


# Check TensorFlow version
print("TensorFlow Version:", tf.__version__)

# Check if GPU is available
print("GPU Available:", tf.test.is_gpu_available())
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.11/dist-packages (2.18.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (25.2.10)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.6
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from tensorflow) (24.2)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.3 in /usr/local/lib/py
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from tensorflow) (75.2.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (4.13.2)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.2)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.71.0)
Requirement already satisfied: tensorboard<2.19,>=2.18 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.18.0)
Requirement already satisfied: keras>=3.5.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.8.0)
Requirement already satisfied: numpy<2.1.0,>=1.26.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.0.2)
Requirement already satisfied: h5py>=3.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.13.0)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0->tensorflow) (0
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (0.0.9)
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (0.15.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensor
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow) (3.10
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1->tensorboard<2.19,
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0->tensorflow
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0->tensorfl
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich->keras>=3.5.0
Requirement already satisfied: tensorflow-text in /usr/local/lib/python3.11/dist-packages (2.18.1)
Requirement already satisfied: tensorflow<2.19,>=2.18.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow-text) (2.18.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensorflow
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensorfl
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensc
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensor
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensorflo
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensorfl
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensorflow-text
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.3 in /usr/local/lib/py
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensor
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensorflow-text
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensorflow-te
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensorflo
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->t
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensorflow-t
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensor
Requirement already satisfied: tensorboard<2.19,>=2.18 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->te
Requirement already satisfied: keras>=3.5.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensorflow-te
Requirement already satisfied: numpy<2.1.0,>=1.26.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensc
Requirement already satisfied: h5py>=3.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->tensorflow-te
Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19,>=2.18.0->te
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow<2.19
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0->tensorflow<2.1
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow<2.19,>=2.18.0->tensorf
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow<2.19,>=2.18.0->tensor
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow<2.19,>=2.18.0->tensc
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensor
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow<2.19,>=
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow<2
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow<2
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow<
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow<
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1->tensorboard<2.19,
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0->tensorflow
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0->tensorfl
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich->keras>=3.5.0
Requirement already satisfied: opencv-python in /usr/local/lib/python3.11/dist-packages (4.11.0.86)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.11/dist-packages (from opencv-python) (2.0.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (2.0.2)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (2.0.2)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.15.2)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
Collecting tensorflowjs
  Downloading tensorflowjs-4.22.0-py3-none-any.whl.metadata (3.2 kB)
Requirement already satisfied: flax>=0.7.2 in /usr/local/lib/python3.11/dist-packages (from tensorflowjs) (0.10.6)
Requirement already satisfied: importlib_resources>=5.9.0 in /usr/local/lib/python3.11/dist-packages (from tensorflowjs) (6.5.2)
Requirement already satisfied: jax>=0.4.13 in /usr/local/lib/python3.11/dist-packages (from tensorflowjs) (0.5.2)
Requirement already satisfied: jaxlib>=0.4.13 in /usr/local/lib/python3.11/dist-packages (from tensorflowjs) (0.5.1)
Requirement already satisfied: tensorflow<3,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from tensorflowjs) (2.18.0)
Requirement already satisfied: tf-keras>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from tensorflowjs) (2.18.0)
Requirement already satisfied: tensorflow-decision-forests>=1.5.0 in /usr/local/lib/python3.11/dist-packages (from tensorflowjs) (1
Requirement already satisfied: six<2,>=1.16.0 in /usr/local/lib/python3.11/dist-packages (from tensorflowjs) (1.17.0)
Requirement already satisfied: tensorflow-hub>=0.16.1 in /usr/local/lib/python3.11/dist-packages (from tensorflowjs) (0.16.1)
Collecting packaging~=23.1 (from tensorflowjs)
  Downloading packaging-23.2-py3-none-any.whl.metadata (3.2 kB)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from flax>=0.7.2->tensorflowjs) (2.0.2)
Requirement already satisfied: msgpack in /usr/local/lib/python3.11/dist-packages (from flax>=0.7.2->tensorflowjs) (1.1.0)
Requirement already satisfied: optax in /usr/local/lib/python3.11/dist-packages (from flax>=0.7.2->tensorflowjs) (0.2.4)
Requirement already satisfied: orbax-checkpoint in /usr/local/lib/python3.11/dist-packages (from flax>=0.7.2->tensorflowjs) (0.11.13
Requirement already satisfied: tensorstore in /usr/local/lib/python3.11/dist-packages (from flax>=0.7.2->tensorflowjs) (0.1.74)
Requirement already satisfied: rich>=11.1 in /usr/local/lib/python3.11/dist-packages (from flax>=0.7.2->tensorflowjs) (13.9.4)
Requirement already satisfied: typing_extensions>=4.2 in /usr/local/lib/python3.11/dist-packages (from flax>=0.7.2->tensorflowjs) (4
Requirement already satisfied: PyYAML>=5.4.1 in /usr/local/lib/python3.11/dist-packages (from flax>=0.7.2->tensorflowjs) (6.0.2)
Requirement already satisfied: treescope>=0.1.7 in /usr/local/lib/python3.11/dist-packages (from flax>=0.7.2->tensorflowjs) (0.1.9)
Requirement already satisfied: ml_dtypes>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from jax>=0.4.13->tensorflowjs) (0.4.1)
Requirement already satisfied: opt_einsum in /usr/local/lib/python3.11/dist-packages (from jax>=0.4.13->tensorflowjs) (3.4.0)
Requirement already satisfied: scipy>=1.11.1 in /usr/local/lib/python3.11/dist-packages (from jax>=0.4.13->tensorflowjs) (1.15.2)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2.13.0->tensorflowjs)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2.13.0->tensorflow
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2.13.0->tensorfl
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2.13.0->tensorflo
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2.13.0->tensorflowjs
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.3 in /usr/local/lib/py
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2.13.0->tensorflo
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2.13.0->tensorflowjs) (75
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2.13.0->tensorflowjs
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2.13.0->tensorflowjs)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2.13.0->tensorflo
Requirement already satisfied: tensorboard<2.19,>=2.18 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2.13.0->tenso
Requirement already satisfied: keras>=3.5.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2.13.0->tensorflowjs) (3
Requirement already satisfied: h5py>=3.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=2.13.0->tensorflowjs) (3
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow<3,>=
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from tensorflow-decision-forests>=1.5.0->tensorflo
Requirement already satisfied: wheel in /usr/local/lib/python3.11/dist-packages (from tensorflow-decision-forests>=1.5.0->tensorflow
Requirement already satisfied: wurlitzer in /usr/local/lib/python3.11/dist-packages (from tensorflow-decision-forests>=1.5.0->tensor
Requirement already satisfied: ydf in /usr/local/lib/python3.11/dist-packages (from tensorflow-decision-forests>=1.5.0->tensorflowjs
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow<3,>=2.13.0->tensorflo
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow<3,>=2.13.0->tensorfl
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensor
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow<3,>=2.1
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow<3
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow<3
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich>=11.1->flax>=0.7.2->tenso
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich>=11.1->flax>=0.7.2->ten
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow<
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow<
Requirement already satisfied: chex>=0.1.87 in /usr/local/lib/python3.11/dist-packages (from optax->flax>=0.7.2->tensorflowjs) (0.1
Requirement already satisfied: etils[epy] in /usr/local/lib/python3.11/dist-packages (from optax->flax>=0.7.2->tensorflowjs) (1.12.2
Requirement already satisfied: nest_asyncio in /usr/local/lib/python3.11/dist-packages (from orbax-checkpoint->flax>=0.7.2->tensorfl
Requirement already satisfied: humanize in /usr/local/lib/python3.11/dist-packages (from orbax-checkpoint->flax>=0.7.2->tensorflowjs
Requirement already satisfied: simplejson>=3.16.0 in /usr/local/lib/python3.11/dist-packages (from orbax-checkpoint->flax>=0.7.2->te
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas->tensorflow-decision-f
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas->tensorflow-decision-forests>=1
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas->tensorflow-decision-forests>=
Requirement already satisfied: toolz>=0.9.0 in /usr/local/lib/python3.11/dist-packages (from chex>=0.1.87->optax->flax>=0.7.2->tenso
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich>=11.1->flax>=
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1->tensorboard<2.19,
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (from etils[epath,epy]->orbax-checkpoint->flax>=0.7
Requirement already satisfied: zipp in /usr/local/lib/python3.11/dist-packages (from etils[epath,epy]->orbax-checkpoint->flax>=0.7.2
Downloading tensorflowjs-4.22.0-py3-none-any.whl (89 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 89.1/89.1 kB 4.2 MB/s eta 0:00:00
Downloading packaging-23.2-py3-none-any.whl (53 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 53.0/53.0 kB 5.7 MB/s eta 0:00:00
Installing collected packages: packaging, tensorflowjs
  Attempting uninstall: packaging
    Found existing installation: packaging 24.2
    Uninstalling packaging-24.2:
      Successfully uninstalled packaging-24.2
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sou
```

```
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sou
google-cloud-bigquery 3.31.0 requires packaging>=24.2.0, but you have packaging 23.2 which is incompatible.
Successfully installed packaging-23.2 tensorflowjs-4.22.0
Requirement already satisfied: kaggle in /usr/local/lib/python3.11/dist-packages (1.7.4.2)
Requirement already satisfied: bleach in /usr/local/lib/python3.11/dist-packages (from kaggle) (6.2.0)
Requirement already satisfied: certifi>=14.05.14 in /usr/local/lib/python3.11/dist-packages (from kaggle) (2025.4.26)
Requirement already satisfied: charset-normalizer in /usr/local/lib/python3.11/dist-packages (from kaggle) (3.4.1)
Requirement already satisfied: idna in /usr/local/lib/python3.11/dist-packages (from kaggle) (3.10)
Requirement already satisfied: protobuf in /usr/local/lib/python3.11/dist-packages (from kaggle) (5.29.4)
Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/python3.11/dist-packages (from kaggle) (2.9.0.post0)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.11/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from kaggle) (2.32.3)
Requirement already satisfied: setuptools>=21.0.0 in /usr/local/lib/python3.11/dist-packages (from kaggle) (75.2.0)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.11/dist-packages (from kaggle) (1.17.0)
Requirement already satisfied: text-unidecode in /usr/local/lib/python3.11/dist-packages (from kaggle) (1.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from kaggle) (4.67.1)
Requirement already satisfied: urllib3>=1.15.1 in /usr/local/lib/python3.11/dist-packages (from kaggle) (2.4.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.11/dist-packages (from kaggle) (0.5.1)
```

🌲 Try YDF, the successor of TensorFlow Decision Forests using the same algorithms but with more features and faster training!

Old code	New code

`import tensorflow_decision_forests as tfdf`
 	
`tf_ds = tfdf.keras.pd_dataframe_to_tf_dataset(ds, label="l")`
`model = tfdf.keras.RandomForestModel(label="l")`
`model.fit(tf_ds)`
	`import ydf`
	
	`model = ydf.RandomForestLearner(label="l").train(ds)`

(Learn more in the migration guide)

```
WARNING:tensorflow:From <ipython-input-1-3890cf556073>:35: is_gpu_available (from tensorflow.python.framework.test_util) is deprecat
Instructions for updating:
Use `tf.config.list_physical_devices('GPU')` instead.
TensorFlow Version: 2.18.0
GPU Available: True
```

Data Preparation

Data Loading

```
# Setup Kaggle API (Anda perlu mengunggah kaggle.json)
from google.colab import files
files.upload()  # Unggah file kaggle.json Anda
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json

# Unduh dataset dari Kaggle
!kaggle datasets download -d shaunthesheep/microsoft-catsvsdogs-dataset
!unzip -q microsoft-catsvsdogs-dataset.zip -d dataset

# Definisikan path dataset
data_dir = '/content/dataset/PetImages'
categories = ['Cat', 'Dog']
```

```
Choose Files kaggle.json
• kaggle.json(application/json) - 65 bytes, last modified: 4/2/2025 - 100% done
Saving kaggle.json to kaggle.json
Dataset URL: https://www.kaggle.com/datasets/shaunthesheep/microsoft-catsvsdogs-dataset
License(s): other
```

Data Preprocessing

```
# Fungsi untuk membuat daftar path gambar dan label
def create_image_dataframe(data_dir, categories):
    image_paths = []
    labels = []

    for category in categories:
        path = os.path.join(data_dir, category)
        class_num = categories.index(category)  # 0 untuk Cat, 1 untuk Dog

        for img_name in os.listdir(path):
            img_path = os.path.join(path, img_name)
            # Validasi apakah file adalah gambar
            if img_name.lower().endswith(('.png', '.jpg', '.jpeg')):
                image_paths.append(img_path)
                labels.append(class_num)
            else:
                print(f"Skipped non-image file: {img_name}")

    # Buat DataFrame
    df = pd.DataFrame({
        'image_path': image_paths,
        'label': labels
    })
    df['label_name'] = df['label'].map({0: 'Cat', 1: 'Dog'})
    return df

# Buat DataFrame
df = create_image_dataframe(data_dir, categories)
```

```
Skipped non-image file: Thumbs.db
Skipped non-image file: Thumbs.db
```

Daftar Nama Kelas

```
# Daftar nama kelas
class_names = categories
print("Class Names:", class_names)
print("Total Images:", len(df))
print("DataFrame Head:")
print(df.head())
print("Label Distribution:")
print(df['label_name'].value_counts())
```

```
Class Names: ['Cat', 'Dog']
Total Images: 25000
DataFrame Head:
                                image_path  label label_name
```

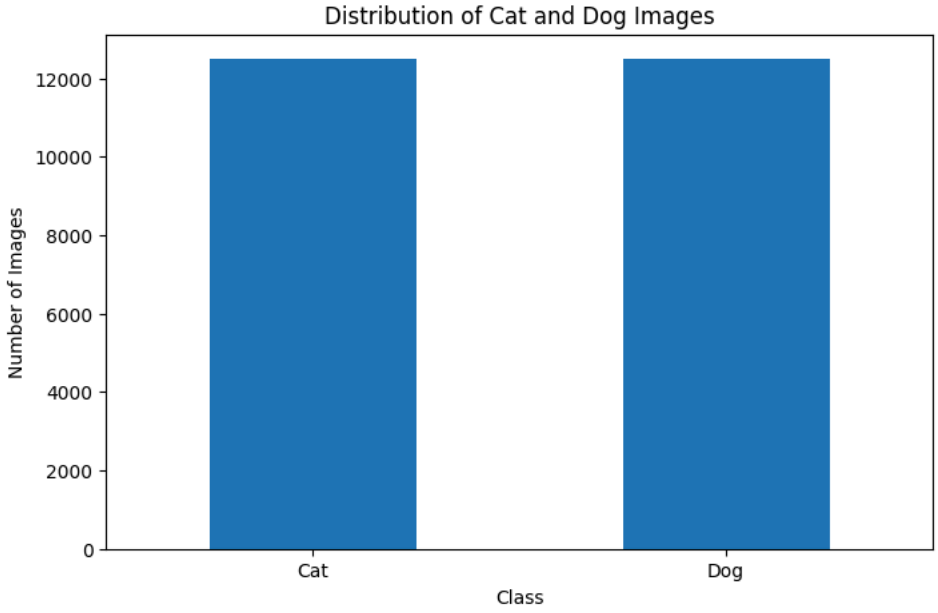
```
0  /content/dataset/PetImages/Cat/2860.jpg      0        Cat
1  /content/dataset/PetImages/Cat/5304.jpg      0        Cat
2  /content/dataset/PetImages/Cat/4699.jpg      0        Cat
3  /content/dataset/PetImages/Cat/6642.jpg      0        Cat
4   /content/dataset/PetImages/Cat/217.jpg      0        Cat
Label Distribution:
label_name
Cat    12500
Dog    12500
Name: count, dtype: int64
```

Plot Sampel Gambar

```
# Fungsi untuk memuat dan menampilkan sampel gambar
def plot_sample_images(df, class_names, num_samples=5, img_size=128):
    plt.figure(figsize=(15, 5))
    sample_df = df.sample(num_samples)

    for i, row in enumerate(sample_df.itertuples()):
        img = cv2.imread(row.image_path)
        if img is None:
            continue
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = cv2.resize(img, (img_size, img_size))

        plt.subplot(1, num_samples, i+1)
        plt.imshow(img)
        plt.title(class_names[row.label])
        plt.axis('off')
    plt.show()

# Tampilkan sampel gambar
plot_sample_images(df, class_names)
```


```
# Building Distribution Data
print("\n#Building Distribution Data")
label_counts = df['label_name'].value_counts()
print("Label Distribution:")
print(label_counts)

# Visualisasi distribusi data
plt.figure(figsize=(8, 5))
label_counts.plot(kind='bar')
plt.title('Distribution of Cat and Dog Images')
plt.xlabel('Class')
plt.ylabel('Number of Images')
plt.xticks(rotation=0)
plt.show()
```

```
#Building Distribution Data
Label Distribution:
label_name
Cat    12500
Dog    12500
Name: count, dtype: int64
```



Split Dataset

Dataset Dibagi Menjadi Train Set, Test Set dan Validation Set

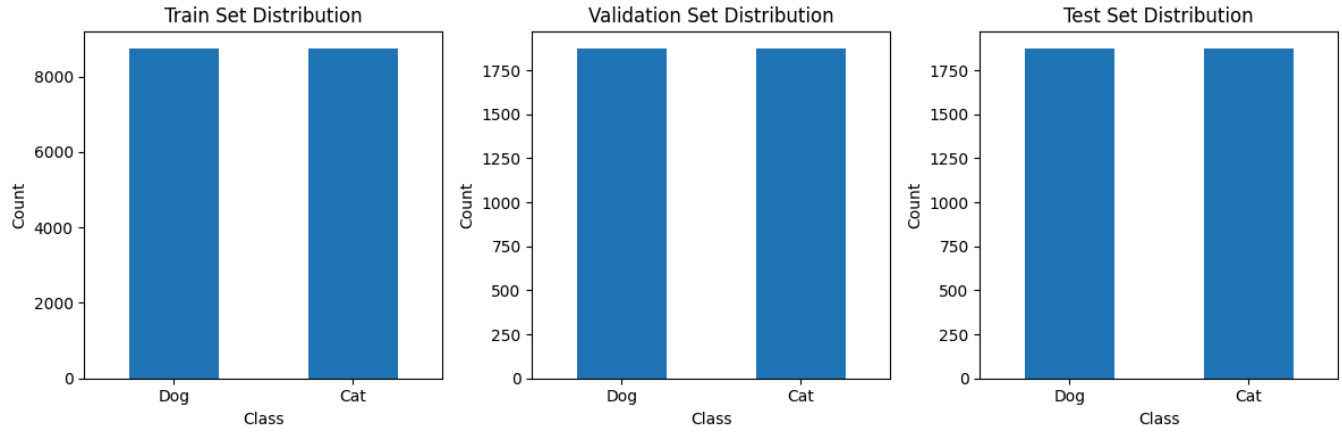
```
# Import library yang diperlukan
from sklearn.model_selection import train_test_split

# Split dataset menjadi train, validation, dan test set
def split_dataset(df, train_size=0.7, val_size=0.15, test_size=0.15):
    # Pastikan proporsi total = 1
    assert train_size + val_size + test_size == 1, "Proporsi split harus berjumlah 1"

    # Split train dan temp (val + test)
    df_train, df_temp = train_test_split(
        df,
        train_size=train_size,
        stratify=df['label'],
        random_state=42
    )

    # Split temp menjadi validation dan test
    val_relative_size = val_size / (val_size + test_size)
    df_val, df_test = train_test_split(
        df_temp,
        train_size=val_relative_size,
        stratify=df_temp['label'],
        random_state=42
    )

    return df_train, df_val, df_test

# Lakukan splitting
df_train, df_val, df_test = split_dataset(df)

# Tampilkan informasi hasil split
print("Train Set Size:", len(df_train))
print("Validation Set Size:", len(df_val))
print("Test Set Size:", len(df_test))
```

```
Train Set Size: 17500
Validation Set Size: 3750
Test Set Size: 3750
```

Count data image

```
# Tampilkan distribusi kelas di setiap set
print("\nTrain Set Label Distribution:")
print(df_train['label_name'].value_counts())
print("\nValidation Set Label Distribution:")
print(df_val['label_name'].value_counts())
print("\nTest Set Label Distribution:")
print(df_test['label_name'].value_counts())

# Visualisasi distribusi kelas
import matplotlib.pyplot as plt

def plot_split_distribution(df_train, df_val, df_test):
    plt.figure(figsize=(12, 4))

    # Train set
    plt.subplot(1, 3, 1)
    df_train['label_name'].value_counts().plot(kind='bar')
    plt.title('Train Set Distribution')
    plt.xlabel('Class')
    plt.ylabel('Count')
    plt.xticks(rotation=0)

    # Validation set
    plt.subplot(1, 3, 2)
    df_val['label_name'].value_counts().plot(kind='bar')
    plt.title('Validation Set Distribution')
    plt.xlabel('Class')
    plt.ylabel('Count')
    plt.xticks(rotation=0)

    # Test set
    plt.subplot(1, 3, 3)
    df_test['label_name'].value_counts().plot(kind='bar')
    plt.title('Test Set Distribution')
    plt.xlabel('Class')
    plt.ylabel('Count')
    plt.xticks(rotation=0)

    plt.tight_layout()
    plt.show()

# Tampilkan visualisasi distribusi
plot_split_distribution(df_train, df_val, df_test)
```

```
Train Set Label Distribution:
label_name
Dog    8750
Cat    8750
Name: count, dtype: int64

Validation Set Label Distribution:
label_name
Dog    1875
Cat    1875
Name: count, dtype: int64

Test Set Label Distribution:
label_name
Dog    1875
Cat    1875
Name: count, dtype: int64
```



Prepare Dataset

```
# Fungsi untuk mempersiapkan DataFrame
def prepare_dataframe(df, set_name):
    # Pastikan kolom label_name adalah string (diperlukan untuk ImageDataGenerator)
    df['label_name'] = df['label_name'].astype(str)

    # Tampilkan informasi DataFrame
    print(f"\n{set_name} DataFrame Info:")
    print("Shape:", df.shape)
    print("Columns:", df.columns.tolist())
    print("Sample Data:")
    print(df.head())
    print("Label Distribution:")
    print(df['label_name'].value_counts())

    return df

# Persiapkan DataFrame untuk train, validation, dan test
df_train = prepare_dataframe(df_train, "Train")
df_val = prepare_dataframe(df_val, "Validation")
df_test = prepare_dataframe(df_test, "Test")

# Simpan DataFrame ke file CSV (opsional, untuk debugging atau referensi)
df_train.to_csv('train_dataset.csv', index=False)
df_val.to_csv('validation_dataset.csv', index=False)
df_test.to_csv('test_dataset.csv', index=False)
print("\nDataFrames telah disimpan sebagai CSV: train_dataset.csv, validation_dataset.csv, test_dataset.csv")
```

```
Train DataFrame Info:
Shape: (17500, 3)
Columns: ['image_path', 'label', 'label_name']
Sample Data:
                                    image_path  label label_name
24393  /content/dataset/PetImages/Dog/4935.jpg      1        Dog
10884  /content/dataset/PetImages/Cat/5087.jpg      0        Cat
16419  /content/dataset/PetImages/Dog/4899.jpg      1        Dog
6596   /content/dataset/PetImages/Cat/6495.jpg      0        Cat
12718  /content/dataset/PetImages/Dog/9335.jpg      1        Dog
Label Distribution:
label_name
Dog    8750
Cat    8750
Name: count, dtype: int64

Validation DataFrame Info:
Shape: (3750, 3)
Columns: ['image_path', 'label', 'label_name']
Sample Data:
                                     image_path  label label_name
13100   /content/dataset/PetImages/Dog/8755.jpg      1        Dog
9908    /content/dataset/PetImages/Cat/2006.jpg      0        Cat
10301  /content/dataset/PetImages/Cat/11286.jpg      0        Cat
3310    /content/dataset/PetImages/Cat/6202.jpg      0        Cat
19000   /content/dataset/PetImages/Dog/1546.jpg      1        Dog
Label Distribution:
label_name
Dog    1875
Cat    1875
Name: count, dtype: int64

Test DataFrame Info:
Shape: (3750, 3)
Columns: ['image_path', 'label', 'label_name']
Sample Data:
                                    image_path  label label_name
23485  /content/dataset/PetImages/Dog/12288.jpg      1        Dog
9923   /content/dataset/PetImages/Cat/11236.jpg      0        Cat
24765   /content/dataset/PetImages/Dog/2794.jpg      1        Dog
2265    /content/dataset/PetImages/Cat/5469.jpg      0        Cat
19246   /content/dataset/PetImages/Dog/7558.jpg      1        Dog
Label Distribution:
label_name
Dog    1875
Cat    1875
Name: count, dtype: int64

DataFrames telah disimpan sebagai CSV: train_dataset.csv, validation_dataset.csv, test_dataset.csv
```

Image Data Generator

```
# Import library yang diperlukan
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Definisikan parameter
IMG_SIZE = 128  # Ukuran gambar
BATCH_SIZE = 32  # Ukuran batch untuk efisiensi memori

# Buat ImageDataGenerator untuk train dengan augmentasi
train_datagen = ImageDataGenerator(
    rescale=1./255,  # Normalisasi gambar
    rotation_range=20,  # Rotasi acak hingga 20 derajat
    width_shift_range=0.2,  # Pergeseran lebar
    height_shift_range=0.2,  # Pergeseran tinggi
    shear_range=0.2,  # Transformasi shear
    zoom_range=0.2,  # Zoom acak
    horizontal_flip=True,  # Flip horizontal
    fill_mode='nearest'  # Cara mengisi piksel kosong
)

# Buat ImageDataGenerator untuk validation dan test (tanpa augmentasi, hanya normalisasi)
val_test_datagen = ImageDataGenerator(
    rescale=1./255  # Normalisasi gambar
)

# Buat generator untuk train set
train_generator = train_datagen.flow_from_dataframe(
    dataframe=df_train,
    x_col='image_path',
    y_col='label_name',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',  # Karena klasifikasi biner (Cat vs Dog)
    shuffle=True
)

# Buat generator untuk validation set
validation_generator = val_test_datagen.flow_from_dataframe(
    dataframe=df_val,
    x_col='image_path',
    y_col='label_name',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    shuffle=False
)

# Buat generator untuk test set
test_generator = val_test_datagen.flow_from_dataframe(
    dataframe=df_test,
    x_col='image_path',
    y_col='label_name',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    shuffle=False
)

# Tampilkan informasi generator
print("Train Generator Classes:", train_generator.class_indices)
print("Validation Generator Classes:", validation_generator.class_indices)
print("Test Generator Classes:", test_generator.class_indices)
print(f"Train Samples: {train_generator.samples}")
print(f"Validation Samples: {validation_generator.samples}")
print(f"Test Samples: {test_generator.samples}")

# Visualisasi beberapa gambar dari train generator (dengan augmentasi)
import matplotlib.pyplot as plt

def plot_augmented_images(generator, num_samples=5):
    # Ambil satu batch gambar
    images, labels = next(generator)
    class_names = list(generator.class_indices.keys())

    plt.figure(figsize=(15, 5))
    for i in range(min(num_samples, len(images))):
        plt.subplot(1, num_samples, i+1)
        plt.imshow(images[i])
        plt.title(class_names[int(labels[i])])
        plt.axis('off')
    plt.show()

# Tampilkan sampel gambar yang telah diaugmentasi dari train generator
```

```
print("\nSample Augmented Images from Train Generator:")
plot_augmented_images(train_generator)
```

```
Found 17500 validated image filenames belonging to 2 classes.
Found 3750 validated image filenames belonging to 2 classes.
Found 3750 validated image filenames belonging to 2 classes.
Train Generator Classes: {'Cat': 0, 'Dog': 1}
Validation Generator Classes: {'Cat': 0, 'Dog': 1}
Test Generator Classes: {'Cat': 0, 'Dog': 1}
Train Samples: 17500
Validation Samples: 3750
Test Samples: 3750

Sample Augmented Images from Train Generator:
```

Modelling

```
# Definisikan parameter
IMG_SIZE = 128  # Sesuai dengan tahap 5
NUM_CLASSES = 1  # Karena klasifikasi biner (output sigmoid)

# Buat arsitektur CNN
model = models.Sequential([
    # Layer 1: Konvolusi + Pooling
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(IMG_SIZE, IMG_SIZE, 3)),
    layers.MaxPooling2D((2, 2)),

    # Layer 2: Konvolusi + Pooling
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),

    # Layer 3: Konvolusi + Pooling
    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),

    # Flatten untuk mengubah ke vektor
    layers.Flatten(),

    # Fully connected layers
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),  # Dropout untuk mencegah overfitting
    layers.Dense(NUM_CLASSES, activation='sigmoid')  # Output biner
])

# Kompilasi model
model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy']
)

# Tampilkan ringkasan model
model.summary()
```

```
Model: "sequential_2"
```

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d_6 (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_7 (Conv2D)	(None, 61, 61, 64)	18,496
max_pooling2d_7 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_8 (Conv2D)	(None, 28, 28, 128)	73,856
max_pooling2d_8 (MaxPooling2D)	(None, 14, 14, 128)	0
flatten_2 (Flatten)	(None, 25088)	0
dense_4 (Dense)	(None, 128)	3,211,392
dropout_2 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 1)	129

```
Total params: 3,304,769 (12.61 MB)
Trainable params: 3,304,769 (12.61 MB)
```

Implementasi Callback

```
# Import library yang diperlukan
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau

# Definisikan direktori untuk menyimpan model sementara
checkpoint_dir = 'checkpoints'

# Definisikan callback
callbacks = [
    # EarlyStopping: Hentikan training jika val_accuracy tidak membaik
    EarlyStopping(
        monitor='val_accuracy',
        patience=5,
        verbose=1,
        restore_best_weights=True
    ),
    # ModelCheckpoint: Simpan model terbaik dalam format .keras
    ModelCheckpoint(
        filepath=os.path.join(checkpoint_dir, 'checkpoint.keras'),
        monitor='val_accuracy',
        save_best_only=True,
        verbose=1
    ),
    # ReduceLROnPlateau: Kurangi learning rate jika val_accuracy stagnan
    ReduceLROnPlateau(
        monitor='val_accuracy',
        factor=0.2,
        patience=3,
        min_lr=1e-6,
        verbose=1
    )
]

# Buat direktori checkpoint jika belum ada
os.makedirs(checkpoint_dir, exist_ok=True)

# Tampilkan informasi callback
print("Callbacks yang akan digunakan:")
for callback in callbacks:
    print(f"- {callback.__class__.__name__}")
print(f"\nModel sementara akan disimpan di '{checkpoint_dir}/checkpoint.keras'.")
print("Model akhir akan disimpan dalam format SavedModel di direktori 'saved_model' setelah pelatihan.")
```

```
Callbacks yang akan digunakan:
- EarlyStopping
- ModelCheckpoint
- ReduceLROnPlateau

Model sementara akan disimpan di 'checkpoints/checkpoint.keras'.
Model akhir akan disimpan dalam format SavedModel di direktori 'saved_model' setelah pelatihan.
```

Training Model

```
# Import library yang diperlukan
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from PIL import Image

# Fungsi untuk memvalidasi file gambar
def is_valid_image(file_path):
    try:
        img = Image.open(file_path)
        img.verify()  # Verifikasi integritas file
        img.close()
        return True
    except Exception as e:
        print(f"Invalid image file: {file_path} - {str(e)}")
        return False

# Filter DataFrame untuk hanya menyertakan file gambar yang valid
def filter_valid_images(df):
    valid_rows = []
    invalid_files = []

    for idx, row in df.iterrows():
        if is_valid_image(row['image_path']):
            valid_rows.append(row)
        else:
            invalid_files.append(row['image_path'])

    valid_df = pd.DataFrame(valid_rows, columns=df.columns)
    print(f"Total invalid files skipped: {len(invalid_files)}")
    if invalid_files:
        print("Sample invalid files:", invalid_files[:5])
    return valid_df

# Terapkan filter ke DataFrame (train, validation, test)
print("Filtering invalid images...")
df_train = filter_valid_images(df_train)
df_val = filter_valid_images(df_val)
df_test = filter_valid_images(df_test)

# Perbarui generator dengan DataFrame yang sudah difilter
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

val_test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_dataframe(
    dataframe=df_train,
    x_col='image_path',
    y_col='label_name',
    target_size=(128, 128),
    batch_size=32,
    class_mode='binary',
    shuffle=True
)

validation_generator = val_test_datagen.flow_from_dataframe(
    dataframe=df_val,
    x_col='image_path',
    y_col='label_name',
    target_size=(128, 128),
    batch_size=32,
    class_mode='binary',
    shuffle=False
)
```

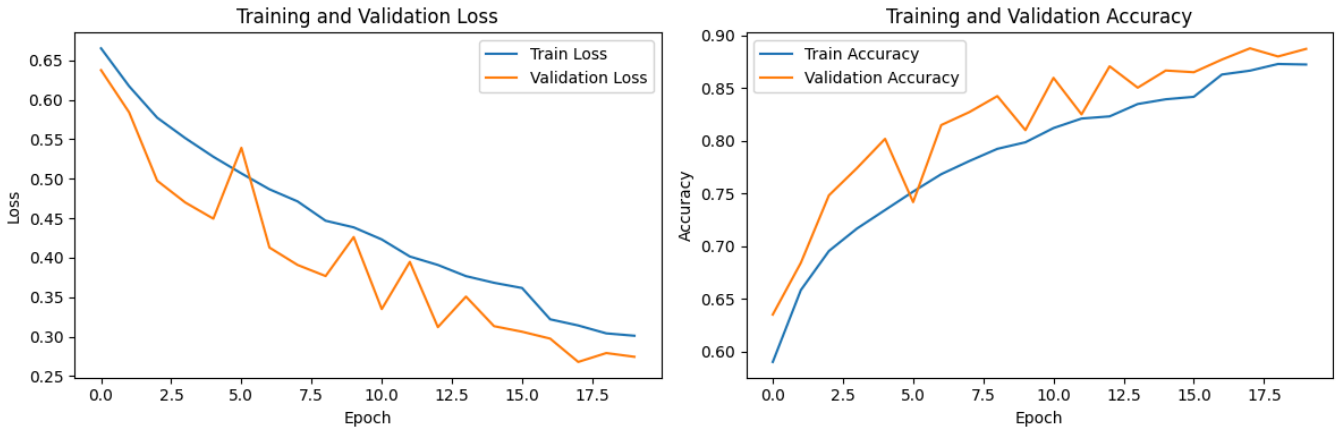
```
Filtering invalid images...
/usr/local/lib/python3.11/dist-packages/PIL/TiffImagePlugin.py:950: UserWarning: Truncated File Read
  warnings.warn(str(msg))
Invalid image file: /content/dataset/PetImages/Dog/11702.jpg - cannot identify image file '/content/dataset/PetImages/Dog/11702.jpg
Total invalid files skipped: 1
Sample invalid files: ['/content/dataset/PetImages/Dog/11702.jpg']
Invalid image file: /content/dataset/PetImages/Cat/666.jpg - cannot identify image file '/content/dataset/PetImages/Cat/666.jpg'
Total invalid files skipped: 1
Sample invalid files: ['/content/dataset/PetImages/Cat/666.jpg']
Total invalid files skipped: 0
Found 17499 validated image filenames belonging to 2 classes.
```

```
Found 3749 validated image filenames belonging to 2 classes.
```

```
# Definisikan parameter training
EPOCHS = 20

# Latih model
history = model.fit(
    train_generator,
    epochs=EPOCHS,
    validation_data=validation_generator,
    callbacks=callbacks,
    verbose=1
)

# Simpan model dalam format SavedModel setelah pelatihan
saved_model_dir = 'saved_model'
try:
    model.save(saved_model_dir, save_format='tf')
    print(f"\nModel telah disimpan dalam format SavedModel di '{saved_model_dir}'.")
    # Verifikasi isi direktori
    print(f"Isi {saved_model_dir}:")
    for item in os.listdir(saved_model_dir):
        print(f"- {item}")
except Exception as e:
    print(f"Error saat menyimpan SavedModel: {e}")

# Tampilkan ringkasan hasil training
print("\nTraining selesai!")
print("Akurasi terbaik pada validation set:", max(history.history['val_accuracy']))

# Visualisasi loss dan akurasi
def plot_training_history(history):
    plt.figure(figsize=(12, 4))

    # Plot loss
    plt.subplot(1, 2, 1)
    plt.plot(history.history['loss'], label='Train Loss')
    plt.plot(history.history['val_loss'], label='Validation Loss')
    plt.title('Training and Validation Loss')
    plt.xlabel('Epoch')
    plt.ylabel('Loss')
    plt.legend()

    # Plot akurasi
    plt.subplot(1, 2, 2)
    plt.plot(history.history['accuracy'], label='Train Accuracy')
    plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
    plt.title('Training and Validation Accuracy')
    plt.xlabel('Epoch')
    plt.ylabel('Accuracy')
    plt.legend()

    plt.tight_layout()
    plt.show()

# Tampilkan plot hasil training
plot_training_history(history)
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` cl
  self._warn_if_super_not_called()
Epoch 1/20
519/547 ━━━━━━━━━━━━━━━━━━━━ 4s 175ms/step - accuracy: 0.5621 - loss: 0.6781/usr/local/lib/python3.11/dist-packages/PIL/TiffImagePlu
  warnings.warn(str(msg))
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 177ms/step - accuracy: 0.5635 - loss: 0.6774
Epoch 1: val_accuracy improved from -inf to 0.63510, saving model to checkpoints/checkpoint.keras
547/547 ━━━━━━━━━━━━━━━━━━━━ 112s 193ms/step - accuracy: 0.5636 - loss: 0.6774 - val_accuracy: 0.6351 - val_loss: 0.6376 - learning_
Epoch 2/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 186ms/step - accuracy: 0.6525 - loss: 0.6248
Epoch 2: val_accuracy improved from 0.63510 to 0.68418, saving model to checkpoints/checkpoint.keras
547/547 ━━━━━━━━━━━━━━━━━━━━ 141s 198ms/step - accuracy: 0.6525 - loss: 0.6248 - val_accuracy: 0.6842 - val_loss: 0.5842 - learning_
Epoch 3/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 178ms/step - accuracy: 0.6788 - loss: 0.5880
Epoch 3: val_accuracy improved from 0.68418 to 0.74820, saving model to checkpoints/checkpoint.keras
547/547 ━━━━━━━━━━━━━━━━━━━━ 103s 188ms/step - accuracy: 0.6788 - loss: 0.5879 - val_accuracy: 0.7482 - val_loss: 0.4975 - learning_
Epoch 4/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 174ms/step - accuracy: 0.7109 - loss: 0.5579
Epoch 4: val_accuracy improved from 0.74820 to 0.77407, saving model to checkpoints/checkpoint.keras
547/547 ━━━━━━━━━━━━━━━━━━━━ 106s 193ms/step - accuracy: 0.7109 - loss: 0.5579 - val_accuracy: 0.7741 - val_loss: 0.4700 - learning_
Epoch 5/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 172ms/step - accuracy: 0.7268 - loss: 0.5350
Epoch 5: val_accuracy improved from 0.77407 to 0.80181, saving model to checkpoints/checkpoint.keras
547/547 ━━━━━━━━━━━━━━━━━━━━ 100s 183ms/step - accuracy: 0.7268 - loss: 0.5349 - val_accuracy: 0.8018 - val_loss: 0.4494 - learning_
Epoch 6/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 169ms/step - accuracy: 0.7464 - loss: 0.5081
Epoch 6: val_accuracy did not improve from 0.80181
547/547 ━━━━━━━━━━━━━━━━━━━━ 98s 180ms/step - accuracy: 0.7464 - loss: 0.5081 - val_accuracy: 0.7418 - val_loss: 0.5392 - learning_r
Epoch 7/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 168ms/step - accuracy: 0.7720 - loss: 0.4872
Epoch 7: val_accuracy improved from 0.80181 to 0.81488, saving model to checkpoints/checkpoint.keras
547/547 ━━━━━━━━━━━━━━━━━━━━ 98s 179ms/step - accuracy: 0.7720 - loss: 0.4872 - val_accuracy: 0.8149 - val_loss: 0.4128 - learning_r
Epoch 8/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 171ms/step - accuracy: 0.7777 - loss: 0.4720
Epoch 8: val_accuracy improved from 0.81488 to 0.82715, saving model to checkpoints/checkpoint.keras
547/547 ━━━━━━━━━━━━━━━━━━━━ 100s 182ms/step - accuracy: 0.7777 - loss: 0.4720 - val_accuracy: 0.8272 - val_loss: 0.3908 - learning_
Epoch 9/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 168ms/step - accuracy: 0.7938 - loss: 0.4451
Epoch 9: val_accuracy improved from 0.82715 to 0.84236, saving model to checkpoints/checkpoint.keras
547/547 ━━━━━━━━━━━━━━━━━━━━ 98s 179ms/step - accuracy: 0.7938 - loss: 0.4451 - val_accuracy: 0.8424 - val_loss: 0.3767 - learning_r
Epoch 10/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 168ms/step - accuracy: 0.8006 - loss: 0.4415
Epoch 10: val_accuracy did not improve from 0.84236
547/547 ━━━━━━━━━━━━━━━━━━━━ 98s 179ms/step - accuracy: 0.8006 - loss: 0.4414 - val_accuracy: 0.8101 - val_loss: 0.4261 - learning_r
Epoch 11/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 170ms/step - accuracy: 0.8123 - loss: 0.4204
Epoch 11: val_accuracy improved from 0.84236 to 0.85970, saving model to checkpoints/checkpoint.keras
547/547 ━━━━━━━━━━━━━━━━━━━━ 99s 181ms/step - accuracy: 0.8123 - loss: 0.4204 - val_accuracy: 0.8597 - val_loss: 0.3349 - learning_r
Epoch 12/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 170ms/step - accuracy: 0.8187 - loss: 0.4066
Epoch 12: val_accuracy did not improve from 0.85970
547/547 ━━━━━━━━━━━━━━━━━━━━ 99s 181ms/step - accuracy: 0.8187 - loss: 0.4066 - val_accuracy: 0.8250 - val_loss: 0.3947 - learning_r
Epoch 13/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 170ms/step - accuracy: 0.8246 - loss: 0.3875
Epoch 13: val_accuracy improved from 0.85970 to 0.87063, saving model to checkpoints/checkpoint.keras
547/547 ━━━━━━━━━━━━━━━━━━━━ 99s 182ms/step - accuracy: 0.8246 - loss: 0.3876 - val_accuracy: 0.8706 - val_loss: 0.3120 - learning_r
Epoch 14/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 171ms/step - accuracy: 0.8352 - loss: 0.3755
Epoch 14: val_accuracy did not improve from 0.87063
547/547 ━━━━━━━━━━━━━━━━━━━━ 99s 181ms/step - accuracy: 0.8352 - loss: 0.3755 - val_accuracy: 0.8504 - val_loss: 0.3507 - learning_r
Epoch 15/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 168ms/step - accuracy: 0.8381 - loss: 0.3710
Epoch 15: val_accuracy did not improve from 0.87063
547/547 ━━━━━━━━━━━━━━━━━━━━ 98s 179ms/step - accuracy: 0.8381 - loss: 0.3710 - val_accuracy: 0.8666 - val_loss: 0.3132 - learning_r
Epoch 16/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 167ms/step - accuracy: 0.8417 - loss: 0.3610
Epoch 16: val_accuracy did not improve from 0.87063

Epoch 16: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
547/547 ━━━━━━━━━━━━━━━━━━━━ 96s 176ms/step - accuracy: 0.8417 - loss: 0.3610 - val_accuracy: 0.8650 - val_loss: 0.3062 - learning_r
Epoch 17/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 170ms/step - accuracy: 0.8584 - loss: 0.3343
Epoch 17: val_accuracy improved from 0.87063 to 0.87703, saving model to checkpoints/checkpoint.keras
547/547 ━━━━━━━━━━━━━━━━━━━━ 99s 180ms/step - accuracy: 0.8584 - loss: 0.3343 - val_accuracy: 0.8770 - val_loss: 0.2974 - learning_r
Epoch 18/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 169ms/step - accuracy: 0.8659 - loss: 0.3152
Epoch 18: val_accuracy improved from 0.87703 to 0.88770, saving model to checkpoints/checkpoint.keras
547/547 ━━━━━━━━━━━━━━━━━━━━ 98s 179ms/step - accuracy: 0.8659 - loss: 0.3152 - val_accuracy: 0.8877 - val_loss: 0.2679 - learning_r
Epoch 19/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 170ms/step - accuracy: 0.8719 - loss: 0.3053
Epoch 19: val_accuracy did not improve from 0.88770
547/547 ━━━━━━━━━━━━━━━━━━━━ 98s 179ms/step - accuracy: 0.8719 - loss: 0.3053 - val_accuracy: 0.8800 - val_loss: 0.2792 - learning_r
Epoch 20/20
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 171ms/step - accuracy: 0.8689 - loss: 0.3042
Epoch 20: val_accuracy did not improve from 0.88770
547/547 ━━━━━━━━━━━━━━━━━━━━ 104s 190ms/step - accuracy: 0.8689 - loss: 0.3042 - val_accuracy: 0.8872 - val_loss: 0.2744 - learning_
Restoring model weights from the end of the best epoch: 18.
Error saat menyimpan SavedModel: The `save_format` argument is deprecated in Keras 3. Please remove this argument and pass a file pa

Training selesai!
```

```
Akurasi terbaik pada validation set: 0.8877033591270447
```




```
# Simpan model dalam format .keras (cadangan)
keras_model_path = 'model.keras'
try:
    model.save(keras_model_path)
    print(f"\nModel telah disimpan dalam format .keras di '{keras_model_path}'.")
except Exception as e:
    print(f"Error saat menyimpan model .keras: {e}")

# Simpan model dalam format SavedModel
saved_model_dir = 'saved_model'
try:
    tf.saved_model.save(model, saved_model_dir)
    print(f"Model telah disimpan dalam format SavedModel di '{saved_model_dir}'.")
    # Verifikasi isi direktori
    print(f"Isi {saved_model_dir}:")
    for item in os.listdir(saved_model_dir):
        print(f"- {item}")
except Exception as e:
    print(f"Error saat menyimpan SavedModel: {e}")
```

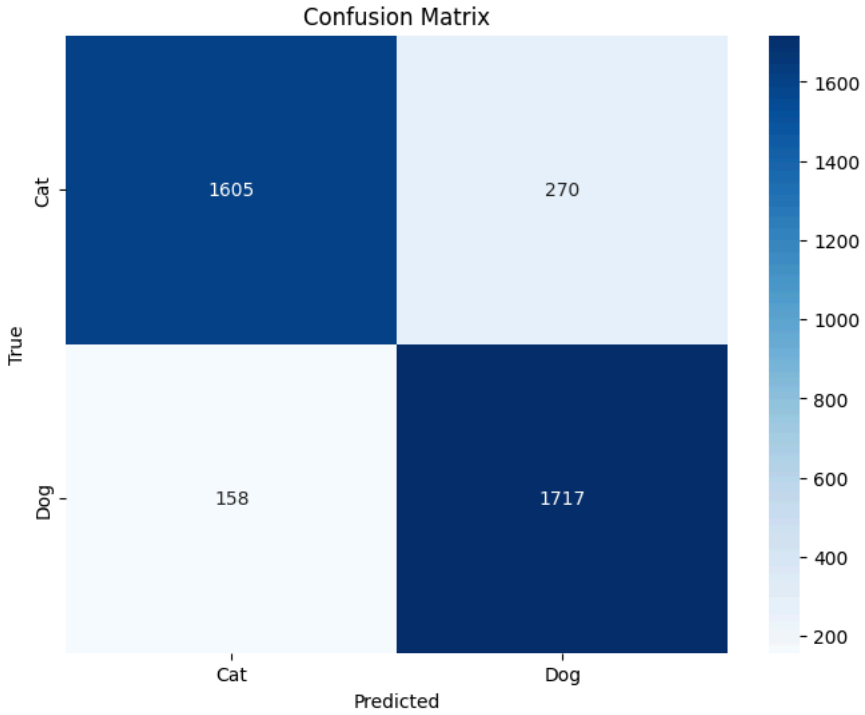
```
Model telah disimpan dalam format .keras di 'model.keras'.
Model telah disimpan dalam format SavedModel di 'saved_model'.
Isi saved_model:
- saved_model.pb
- variables
- assets
- fingerprint.pb
```

Evaluasi dan Visualisasi

```
# Import library evaluasi
from sklearn.metrics import confusion_matrix, classification_report

# Evaluasi model pada test set
test_loss, test_accuracy = model.evaluate(test_generator, verbose=1)
print(f"\nTest Loss: {test_loss:.4f}")
print(f"Test Accuracy: {test_accuracy:.4f}")

# Prediksi pada test set
test_generator.reset()  # Reset generator untuk memastikan urutan benar
y_pred = model.predict(test_generator)
y_pred_classes = (y_pred > 0.5).astype(int).flatten()  # Konversi probabilitas ke kelas (0 atau 1)
y_true = test_generator.classes  # Label sebenarnya dari test_generator

# Buat confusion matrix
cm = confusion_matrix(y_true, y_pred_classes)
class_names = list(test_generator.class_indices.keys())  # ['Cat', 'Dog']

# Visualisasi confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=class_names, yticklabels=class_names)
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
```

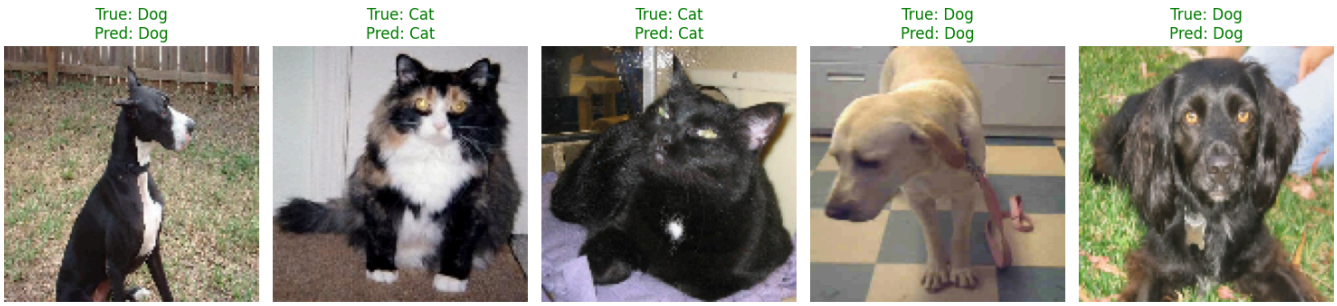
```
plt.ylabel('True')
plt.show()

# Classification report
print("\nClassification Report:")
print(classification_report(y_true, y_pred_classes, target_names=class_names))

# Visualisasi beberapa prediksi pada test set
def plot_sample_predictions(generator, y_true, y_pred_classes, class_names, num_samples=5):
    generator.reset()  # Reset generator
    images, labels = next(generator)  # Ambil batch pertama
    indices = np.random.choice(len(images), num_samples, replace=False)

    plt.figure(figsize=(15, 5))
    for i, idx in enumerate(indices):
        plt.subplot(1, num_samples, i+1)
        plt.imshow(images[idx])
        true_label = class_names[int(labels[idx])]
        pred_label = class_names[y_pred_classes[idx]]
        title = f"True: {true_label}\nPred: {pred_label}"
        plt.title(title, color='green' if true_label == pred_label else 'red')
        plt.axis('off')
    plt.tight_layout()
    plt.show()

# Tampilkan beberapa prediksi
print("\nSample Predictions on Test Set:")
plot_sample_predictions(test_generator, y_true, y_pred_classes, class_names)
```

```
118/118 ━━━━━━━━━━━━━━━━━━━━ 12s 98ms/step - accuracy: 0.8880 - loss: 0.2623

Test Loss: 0.2754
Test Accuracy: 0.8859
118/118 ━━━━━━━━━━━━━━━━━━━━ 6s 45ms/step
```




```
Classification Report:
              precision    recall  f1-score   support

         Cat       0.91      0.86      0.88      1875
         Dog       0.86      0.92      0.89      1875

    accuracy                           0.89      3750
   macro avg       0.89      0.89      0.89      3750
weighted avg       0.89      0.89      0.89      3750
```

Sample Predictions on Test Set:




```
# Definisikan path SavedModel
saved_model_dir = 'saved_model'

# Verifikasi keberadaan SavedModel
if not os.path.exists(saved_model_dir):
    raise FileNotFoundError(f"Direktori SavedModel '{saved_model_dir}' tidak ditemukan.")

# Konversi model ke format TFLite
try:
    converter = tf.lite.TFLiteConverter.from_saved_model(saved_model_dir)
    tflite_model = converter.convert()
    print("Model SavedModel berhasil dikonversi ke TFLite.")
except Exception as e:
    print(f"Error saat mengonversi model ke TFLite: {e}")
    raise

# Simpan model TFLite ke file
tflite_model_path = 'model.tflite'
with open(tflite_model_path, 'wb') as f:
    f.write(tflite_model)
```

```
# Tampilkan ukuran file TFLite
tflite_size = os.path.getsize(tflite_model_path) / (1024 * 1024)  # Ukuran dalam MB
print(f"\nModel TFLite telah disimpan sebagai '{tflite_model_path}'")
print(f"Ukuran file TFLite: {tflite_size:.2f} MB")

# Verifikasi model TFLite
interpreter = tf.lite.Interpreter(model_path=tflite_model_path)
interpreter.allocate_tensors()
print("Model TFLite berhasil diverifikasi.")
```

```
Model SavedModel berhasil dikonversi ke TFLite.

Model TFLite telah disimpan sebagai 'model.tflite'
Ukuran file TFLite: 0.01 MB
Model TFLite berhasil diverifikasi.
```

Konversi Model

TFLite

```
# Bagian 1: Verifikasi Model TFLite
tflite_model_path = 'model.tflite'
try:
    # Muat dan verifikasi model TFLite
    interpreter = tf.lite.Interpreter(model_path=tflite_model_path)
    interpreter.allocate_tensors()
    print("Model TFLite berhasil dimuat dan diverifikasi.")

    # Tampilkan detail input dan output
    input_details = interpreter.get_input_details()
    output_details = interpreter.get_output_details()
    print("Input Details:", input_details)
    print("Output Details:", output_details)
except Exception as e:
    print(f"Error saat memuat model TFLite: {e}")
```

```
Model TFLite berhasil dimuat dan diverifikasi.
Input Details: [{'name': 'serving_default_inputs:0', 'index': 0, 'shape': array([  1, 128, 128,   3], dtype=int32), 'shape_signature
Output Details: [{'name': 'StatefulPartitionedCall:0', 'index': 45, 'shape': array([1, 1], dtype=int32), 'shape_signature': array([
```

TFJS

```
# Bagian 2: Konversi Model ke TensorFlow.js
saved_model_dir = 'saved_model'
tfjs_model_dir = 'tfjs_model'

try:
    # Konversi langsung dari SavedModel ke TFJS
    tfjs.converters.convert_tf_saved_model(saved_model_dir, tfjs_model_dir)
    print(f"\nModel TensorFlow.js telah disimpan di direktori '{tfjs_model_dir}'")

    # Tampilkan isi direktori TFJS
    print("Isi direktori TFJS:")
    for file in os.listdir(tfjs_model_dir):
        print(f"- {file}")
except Exception as e:
    print(f"Error saat mengonversi model ke TFJS: {e}")
```

```
Model TensorFlow.js telah disimpan di direktori 'tfjs_model'
Isi direktori TFJS:
- group1-shard1of4.bin
- model.json
- group1-shard4of4.bin
- group1-shard2of4.bin
- group1-shard3of4.bin
```

ZIP Model TFJS

```
# Bagian 3: Kompresi folder tfjs_model menjadi ZIP
tfjs_zip_path = 'tfjs_model.zip'
try:
    shutil.make_archive('tfjs_model', 'zip', tfjs_model_dir)
    print(f"\nFolder '{tfjs_model_dir}' telah dikompresi menjadi '{tfjs_zip_path}'")
except Exception as e:
```

```
    print(f"Error saat mengompresi folder TFJS: {e}")

# Tampilkan ukuran file TFLite dan direktori TFJS
tflite_size = os.path.getsize(tflite_model_path) / (1024 * 1024)  # Ukuran dalam MB
print(f"\nUkuran file TFLite: {tflite_size:.2f} MB")

tfjs_size = sum(os.path.getsize(os.path.join(tfjs_model_dir, f)) for f in os.listdir(tfjs_model_dir)) / (1024 * 1024)
print(f"Ukuran total direktori TFJS: {tfjs_size:.2f} MB")

tfjs_zip_size = os.path.getsize(tfjs_zip_path) / (1024 * 1024)  # Ukuran dalam MB
print(f"Ukuran file ZIP TFJS: {tfjs_zip_size:.2f} MB")
```

```
Folder 'tfjs_model' telah dikompresi menjadi 'tfjs_model.zip'

Ukuran file TFLite: 0.01 MB
Ukuran total direktori TFJS: 12.62 MB
Ukuran file ZIP TFJS: 11.81 MB
```

Download Model

```
# Import library yang diperlukan
from google.colab import files
import zipfile

# Definisikan path file dan direktori
saved_model_dir = 'saved_model'
tfjs_model_dir = 'tfjs_model'
keras_model_path = 'model.keras'
tflite_model_path = 'model.tflite'
test_dataset_path = 'test_dataset.csv'
train_dataset_path = 'train_dataset.csv'
validation_dataset_path = 'validation_dataset.csv'
output_zip = 'faizalriza_savedModel.zip'

# Verifikasi keberadaan file dan direktori
print("Memeriksa keberadaan file dan direktori untuk di-zip:")
files_to_zip = [
    (saved_model_dir, "direktori"),
    (tfjs_model_dir, "direktori"),
    (keras_model_path, "file"),
    (tflite_model_path, "file"),
    (test_dataset_path, "file"),
    (train_dataset_path, "file"),
    (validation_dataset_path, "file")
]

all_files_exist = True
for path, type_ in files_to_zip:
    if os.path.exists(path):
        print(f"- {path} ({type_}) ditemukan.")
        if type_ == "direktori":
            print(f"  Isi {path}:")
            for item in os.listdir(path):
                print(f"    - {item}")
    else:
        print(f"- {path} ({type_}) tidak ditemukan!")
        all_files_exist = False

if not all_files_exist:
    print("\nPeringatan: Beberapa file/direktori tidak ditemukan. Proses kompresi tetap akan dilanjutkan dengan file yang ada.")

# Buat file ZIP
try:
    print(f"\nMengompresi file ke '{output_zip}'...")
    with zipfile.ZipFile(output_zip, 'w', zipfile.ZIP_DEFLATED) as zipf:
        # Tambahkan folder saved_model
        if os.path.exists(saved_model_dir):
            for root, _, files in os.walk(saved_model_dir):
                for file in files:
                    file_path = os.path.join(root, file)
                    arcname = os.path.relpath(file_path, start=os.path.dirname(saved_model_dir))
                    zipf.write(file_path, arcname)
            print(f"- {saved_model_dir} ditambahkan ke ZIP.")

        # Tambahkan folder tfjs_model
        if os.path.exists(tfjs_model_dir):
            for root, _, files in os.walk(tfjs_model_dir):
                for file in files:
                    file_path = os.path.join(root, file)
```

```
                    arcname = os.path.relpath(file_path, start=os.path.dirname(tfjs_model_dir))
                    zipf.write(file_path, arcname)
            print(f"- {tfjs_model_dir} ditambahkan ke ZIP.")

        # Tambahkan file individu
        for file_path in [keras_model_path, tflite_model_path, test_dataset_path, train_dataset_path, validation_dataset_path]:
            if os.path.exists(file_path):
                arcname = os.path.basename(file_path)
                zipf.write(file_path, arcname)
                print(f"- {file_path} ditambahkan ke ZIP.")

    # Tampilkan ukuran file ZIP
    zip_size = os.path.getsize(output_zip) / (1024 * 1024)  # Ukuran dalam MB
    print(f"\nFile '{output_zip}' berhasil dibuat.")
    print(f"Ukuran file ZIP: {zip_size:.2f} MB")

except Exception as e:
    print(f"Error saat mengompresi file: {e}")
    raise

# Unduh file ZIP
try:
    print(f"\nMengunduh '{output_zip}'...")
    files.download(output_zip)
except Exception as e:
    print(f"Error saat mengunduh {output_zip}: {e}")
```

```
Memeriksa keberadaan file dan direktori untuk di-zip:
- saved_model (direktori) ditemukan.
  Isi saved_model:
    - saved_model.pb
    - variables
    - assets
    - fingerprint.pb
- tfjs_model (direktori) ditemukan.
  Isi tfjs_model:
    - group1-shard1of4.bin
    - model.json
    - group1-shard4of4.bin
    - group1-shard2of4.bin
    - group1-shard3of4.bin
- model.keras (file) ditemukan.
- model.tflite (file) ditemukan.
- test_dataset.csv (file) ditemukan.
- train_dataset.csv (file) ditemukan.
- validation_dataset.csv (file) ditemukan.

Mengompresi file ke 'faizalriza_savedModel.zip'...
- saved_model ditambahkan ke ZIP.
- tfjs_model ditambahkan ke ZIP.
- model.keras ditambahkan ke ZIP.
- model.tflite ditambahkan ke ZIP.
- test_dataset.csv ditambahkan ke ZIP.
- train_dataset.csv ditambahkan ke ZIP.
- validation_dataset.csv ditambahkan ke ZIP.

File 'faizalriza_savedModel.zip' berhasil dibuat.
Ukuran file ZIP: 66.45 MB

Mengunduh 'faizalriza_savedModel.zip'...
Error saat mengunduh faizalriza_savedModel.zip: 'list' object has no attribute 'download'
```

Implementasi Model

```
#periksa signature savedModel
saved_model_dir = 'saved_model'
loaded = tf.saved_model.load(saved_model_dir)
print(list(loaded.signatures.keys()))
print(loaded.signatures['serving_default'].structured_input_signature)
print(loaded.signatures['serving_default'].structured_outputs)
```

```
['serving_default']
((), {'inputs': TensorSpec(shape=(None, 128, 128, 3), dtype=tf.float32, name='inputs')})
{'output_0': TensorSpec(shape=(None, 1), dtype=tf.float32, name='output_0')}
```

```
# Import library yang diperlukan
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from google.colab import files
import cv2
from PIL import Image
import os
```

```
# Definisikan path SavedModel
saved_model_dir = 'saved_model'

# Verifikasi keberadaan SavedModel
if not os.path.exists(saved_model_dir):
    raise FileNotFoundError(f"Direktori SavedModel '{saved_model_dir}' tidak ditemukan.")

# Periksa isi direktori SavedModel
print(f"Isi direktori {saved_model_dir}:")
for item in os.listdir(saved_model_dir):
    print(f"- {item}")

# Periksa signature SavedModel
try:
    loaded_model = tf.saved_model.load(saved_model_dir)
    signatures = list(loaded_model.signatures.keys())
    print("\nSignature yang tersedia:", signatures)
    if 'serving_default' in signatures:
        print("Input signature:", loaded_model.signatures['serving_default'].structured_input_signature)
        print("Output signature:", loaded_model.signatures['serving_default'].structured_outputs)
    else:
        print("Peringatan: Signature 'serving_default' tidak ditemukan!")
except Exception as e:
    print(f"Error saat memeriksa signature SavedModel: {e}")
    raise

# Muat model dari format SavedModel sebagai inference layer
try:
    inference_layer = tf.keras.layers.TFSMLayer(saved_model_dir, call_endpoint='serving_default')
    print("Model SavedModel berhasil dimuat sebagai inference layer.")
except Exception as e:
    print(f"Error saat memuat model SavedModel: {e}")
    raise

# Definisikan parameter
IMG_SIZE = 128  # Ukuran gambar sesuai pelatihan
class_names = ['Cat', 'Dog']  # Mapping kelas

# Fungsi untuk memproses gambar baru
def preprocess_image(image_path):
    try:
        # Baca gambar dengan OpenCV
        img = cv2.imread(image_path)
        if img is None:
            raise ValueError("Gagal membaca gambar.")

        # Konversi ke RGB (OpenCV menggunakan BGR)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # Ubah ukuran ke 128x128
        img = cv2.resize(img, (IMG_SIZE, IMG_SIZE))

        # Normalisasi (0-1)
        img = img / 255.0

        # Tambahkan dimensi batch (1, 128, 128, 3)
        img = np.expand_dims(img, axis=0).astype(np.float32)

        # Validasi bentuk input
        if img.shape != (1, IMG_SIZE, IMG_SIZE, 3):
            raise ValueError(f"Bentuk input tidak valid: {img.shape}. Harus (1, {IMG_SIZE}, {IMG_SIZE}, 3).")

        print(f"Bentuk input tensor: {img.shape}, dtype: {img.dtype}")
        return img
    except Exception as e:
        print(f"Error saat memproses gambar: {e}")
        return None

# Fungsi untuk memprediksi dan menampilkan hasil
```