

DOKUMENTASI PROGRAM

Jupyter Notebook

1. Pembagian data split menjadi 80% Training dan 20% Testing

```
import splitfolders

dataset = 'Data\\dataset'

splitfolders.ratio(dataset, output='Data\\tmp\\kopi2', seed=1337, ratio=(.8, .2), group_prefix=None)
```

2. Inisialisasi untuk data augmentation

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    horizontal_flip=True,
    shear_range = 0.2,
    fill_mode = 'nearest'
)

test_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range = 20,
    horizontal_flip = True,
    shear_range = 0.2,
    fill_mode = 'nearest'
)
```

3. Pembuatan data augmentation berdasarkan dari directory testing dan training yang telah ditentukan

```
train_generator = train_datagen.flow_from_directory(  
    train_dir,  
    target_size=(150, 150),  
    batch_size = 6,  
    class_mode = 'binary'  
)  
  
validation_generator = test_datagen.flow_from_directory(  
    val_dir,  
    target_size = (150, 150),  
    batch_size = 6,  
    class_mode = 'binary'  
)
```

4. Inisialisasi arsitektur model CNN

```
model = tf.keras.Sequential([  
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(150, 150, 3)),  
    tf.keras.layers.MaxPool2D(2, 2),  
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),  
    tf.keras.layers.MaxPooling2D(2,2),  
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),  
    tf.keras.layers.MaxPooling2D(2,2),  
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),  
    tf.keras.layers.MaxPooling2D(2,2),  
    tf.keras.layers.Flatten(),  
    tf.keras.layers.Dense(512, activation='relu'),  
    tf.keras.layers.Dense(1, activation='sigmoid')  
])
```

5. Mengcompile arsitektur model ml

```
model.compile(loss='binary_crossentropy',  
              optimizer=tf.optimizers.Adam(),  
              metrics=['accuracy', 'AUC'])
```

6. Proses melakukan training model

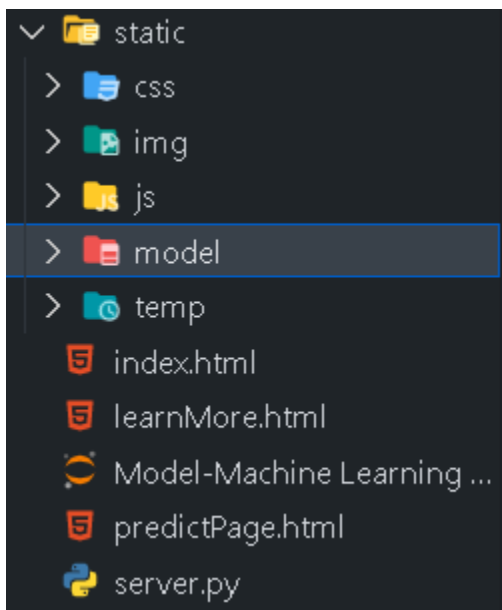
```
history = model.fit(  
    train_generator,  
  
    steps_per_epoch = 25,  
    epochs = 50,  
    validation_data = validation_generator,  
    validation_steps = 5,  
    verbose = 2)
```

7. Eksport model

```
model.save("Data")
```

Pembuatan Web Server

Web server dibangun dengan menggunakan web server flask dengan bantuan library dari bahasa pemrograman python dengan struktur project file sebagai berikut:



Web ini saat ini masih dijalankan pada remote server dengan menggunakan bantuan dari google cloud akan tetapi perlu dilakukan pengecekan setiap 2 hari sekali untuk memastikan server tunnelling pada remote windows server masih berjalan dengan bantuan tunnelling [ngrok.io](https://d1de-34-66-228-62.ngrok.io) dan saat ini sedang berjalan pada <https://d1de-34-66-228-62.ngrok.io> (Sewaktu – waktu masih bisa berubah).

Untuk *path* dari web kami menggunakan beberapa *path endpoint* sebagai berikut:

1. “/”, *method* = *get*. Path awal ini merupakan homepage dari halaman website kami dengan source code sebagai berikut:

```
def homeworld():  
    return render_template('index.html')
```

2. “/”, *method* = *post*. Path ini digunakan memasukkan data gambar yang akan digunakan untuk memprediksi daun dari tumbuhan kopi, dengan source code sebagai berikut:

```
def ImageInput():  
    file = request.files['img']  
    fileName = secure_filename(file.filename)  
    print(fileName)  
    file.save(os.path.join(app.config['UPLOAD_FOLDER'], fileName))  
    flag = prediction(os.path.join(app.config['UPLOAD_FOLDER'], fileName))  
    return render_template('predictPage.html', imgName=fileName, flag=flag)
```

Pada source code tersebut disebutkan adanya function *prediction* yang dimana digunakan sebagai tahapan preprocessing dari gambar sebelum di prediksi menggunakan model yang telah dibuat, dengan source codenya sebagai berikut:

```
def prediction (imgPath):  
    img = image.load_img(imgPath, target_size=(150,150))  
    x = image.img_to_array(img)  
    x = np.expand_dims(x, axis=0)  
    x/=255  
    value = np.vstack([x])  
    classes = model.predict(value, batch_size=10)  
    if classes >.5 :  
        return True  
    return False
```

Pada line 3 terakhir dijelaskan bahwa jika nilai prediksi diatas 0.5 maka disebutkan bahwa tumbuhan tersebut berpenyakit dan begitu sebaliknya.

3. “/learnMore”. Path endpoint ini digunakan untuk memberikan informasi tambahan kepada user mengenai penyakit yang menjangkit tumbuhan kopinya yaitu *coffe rust* dengan source code dari endpoint ini adalah:

```
def learnMore():  
    return render_template('learnMore.html')
```