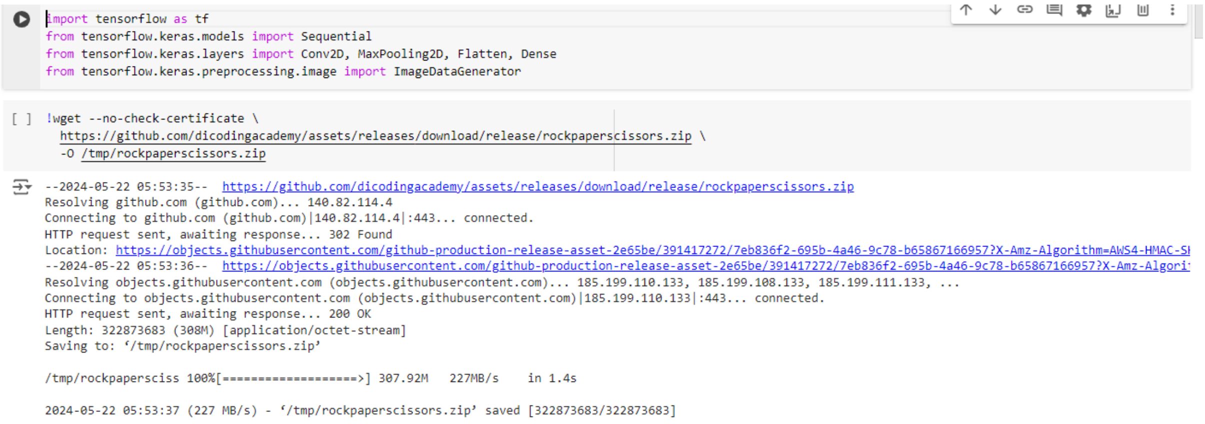
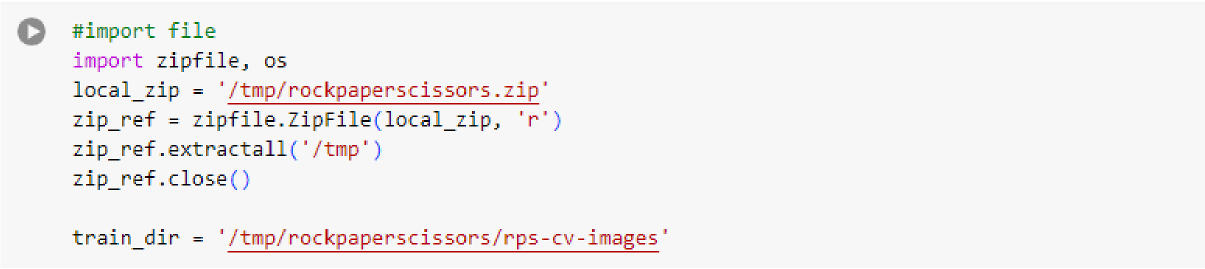


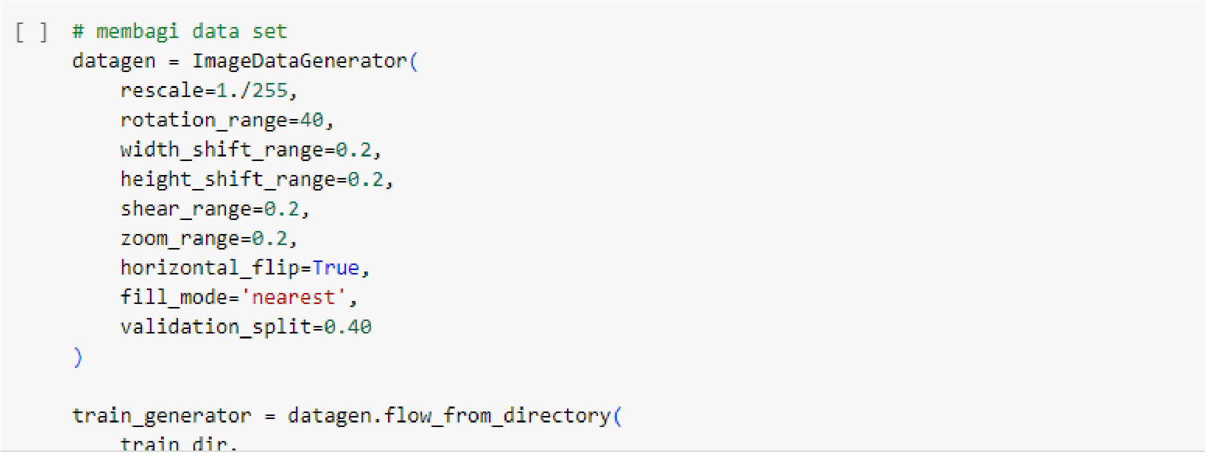
```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
!wget --no-check-certificate \
  https://github.com/dicodingacademy/assets/releases/download/release/rockpaperscissors.zip \
  -O /tmp/rockpaperscissors.zip
```

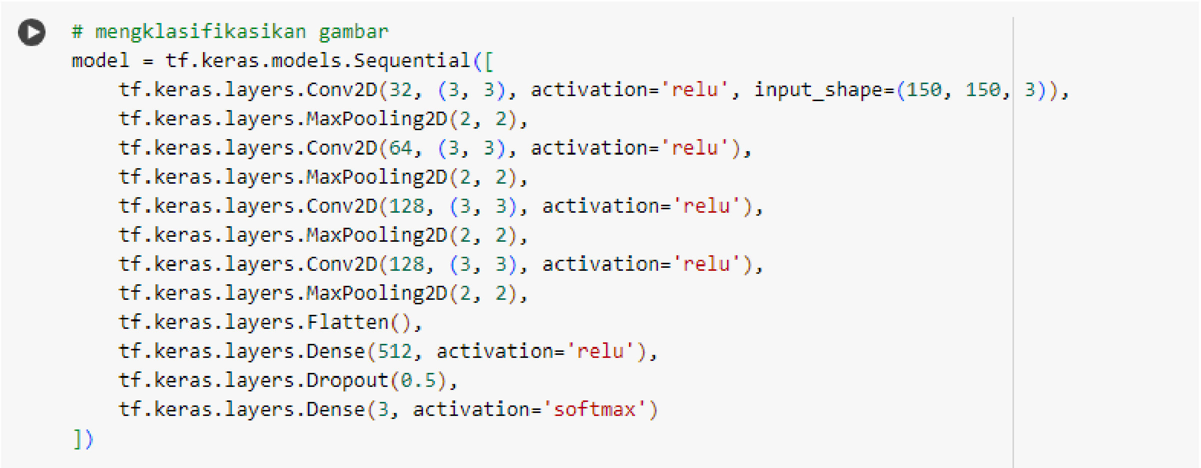
```
--2024-05-22 05:53:35--  https://github.com/dicodingacademy/assets/releases/download/release/rockpaperscissors.zip
Resolving github.com (github.com)... 140.82.114.4
Connecting to github.com (github.com)|140.82.114.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/391417272/7eb836f2-695b-4a46-9c78-b65867166957?X-Amz-Algorithm=AWS4-HMAC-SH
--2024-05-22 05:53:36--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/391417272/7eb836f2-695b-4a46-9c78-b65867166957?X-Amz-Algori
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.108.133, 185.199.111.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 322873683 (308M) [application/octet-stream]
Saving to: ‘/tmp/rockpaperscissors.zip’

/tmp/rockpapersciss 100%[===================>] 307.92M   227MB/s    in 1.4s

2024-05-22 05:53:37 (227 MB/s) - ‘/tmp/rockpaperscissors.zip’ saved [322873683/322873683]
```



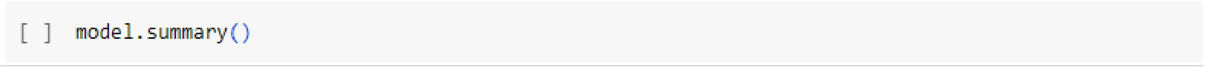
```
#import file
import zipfile, os
local_zip = '/tmp/rockpaperscissors.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('/tmp')
zip_ref.close()

train_dir = '/tmp/rockpaperscissors/rps-cv-images'
```



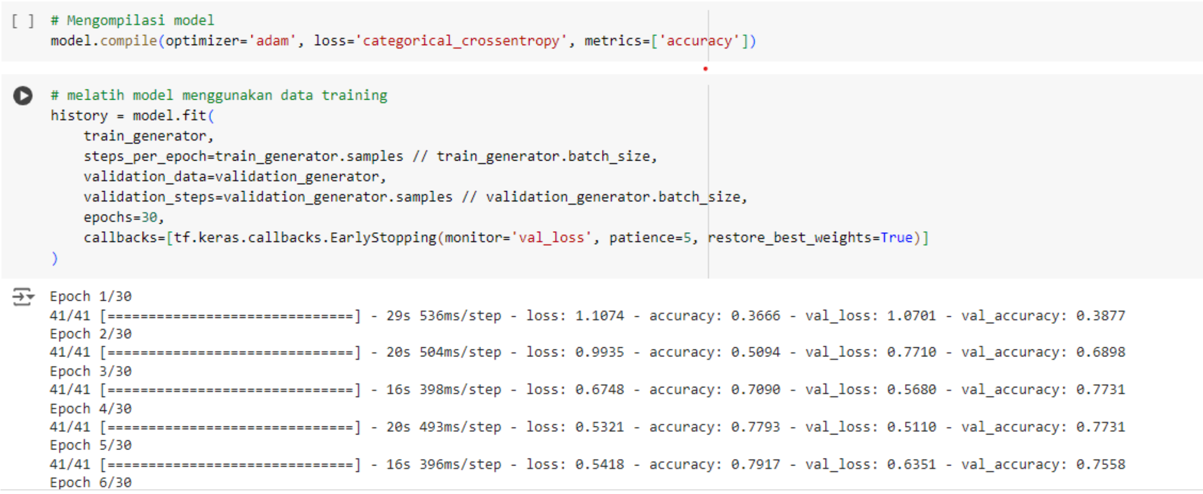
```
# membagi data set
datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest',
    validation_split=0.40
)

train_generator = datagen.flow_from_directory(
    train_dir,
```



```
# mengklasifikasikan gambar
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(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.Dropout(0.5),
    tf.keras.layers.Dense(3, activation='softmax')
])
```



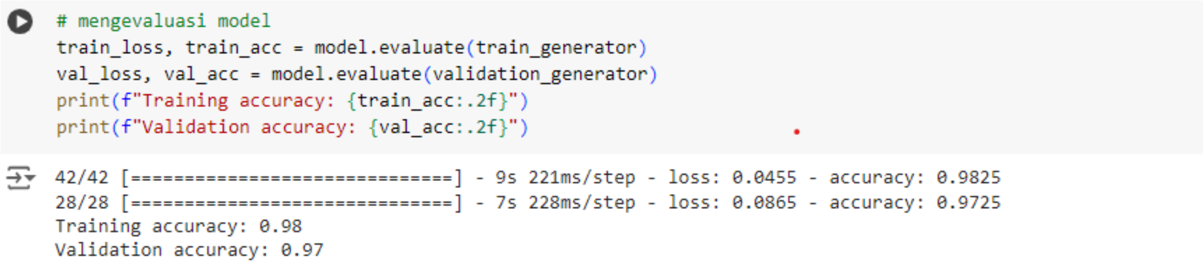
```
model.summary()
```



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

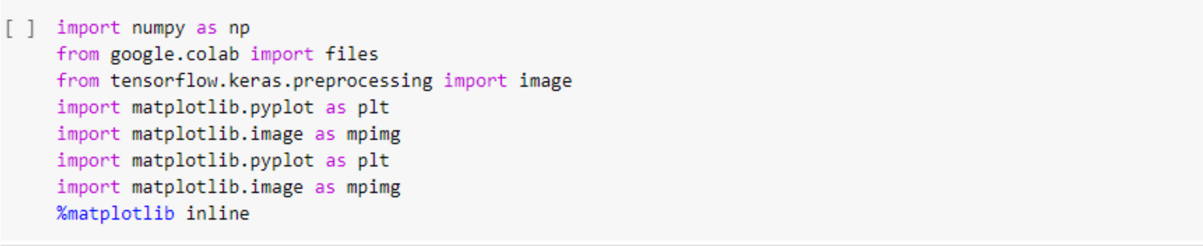
```
# melatih model menggunakan data training
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // validation_generator.batch_size,
    epochs=30,
    callbacks=[tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)]
)
```

```
Epoch 1/30
41/41 [==============================] - 29s 536ms/step - loss: 1.1074 - accuracy: 0.3666 - val_loss: 1.0701 - val_accuracy: 0.3877
Epoch 2/30
41/41 [==============================] - 20s 504ms/step - loss: 0.9935 - accuracy: 0.5094 - val_loss: 0.7710 - val_accuracy: 0.6898
Epoch 3/30
41/41 [==============================] - 16s 398ms/step - loss: 0.6748 - accuracy: 0.7090 - val_loss: 0.5680 - val_accuracy: 0.7731
Epoch 4/30
41/41 [==============================] - 20s 493ms/step - loss: 0.5321 - accuracy: 0.7793 - val_loss: 0.5110 - val_accuracy: 0.7731
Epoch 5/30
41/41 [==============================] - 16s 396ms/step - loss: 0.5418 - accuracy: 0.7917 - val_loss: 0.6351 - val_accuracy: 0.7558
Epoch 6/30
```

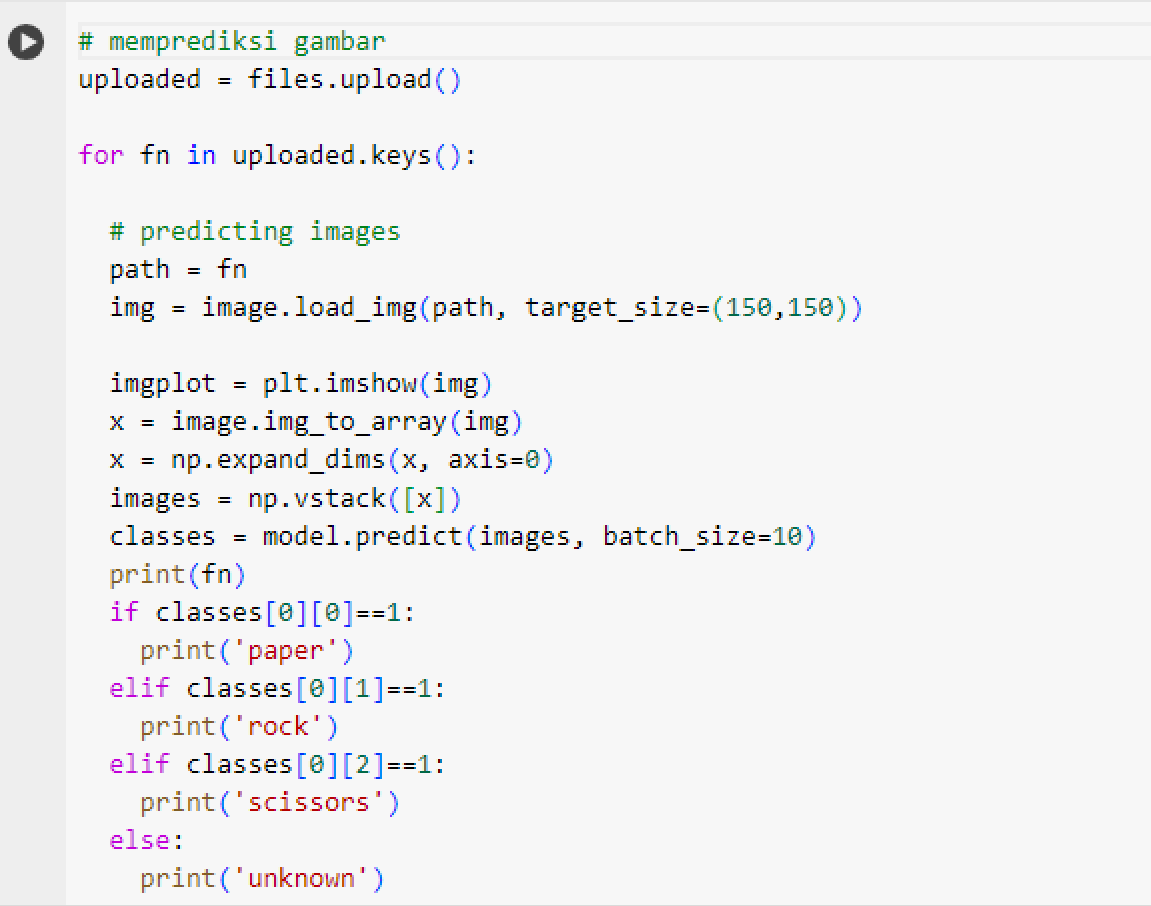


```
# mengevaluasi model
train_loss, train_acc = model.evaluate(train_generator)
val_loss, val_acc = model.evaluate(validation_generator)
print(f"Training accuracy: {train_acc:.2f}")
print(f"Validation accuracy: {val_acc:.2f}")
```

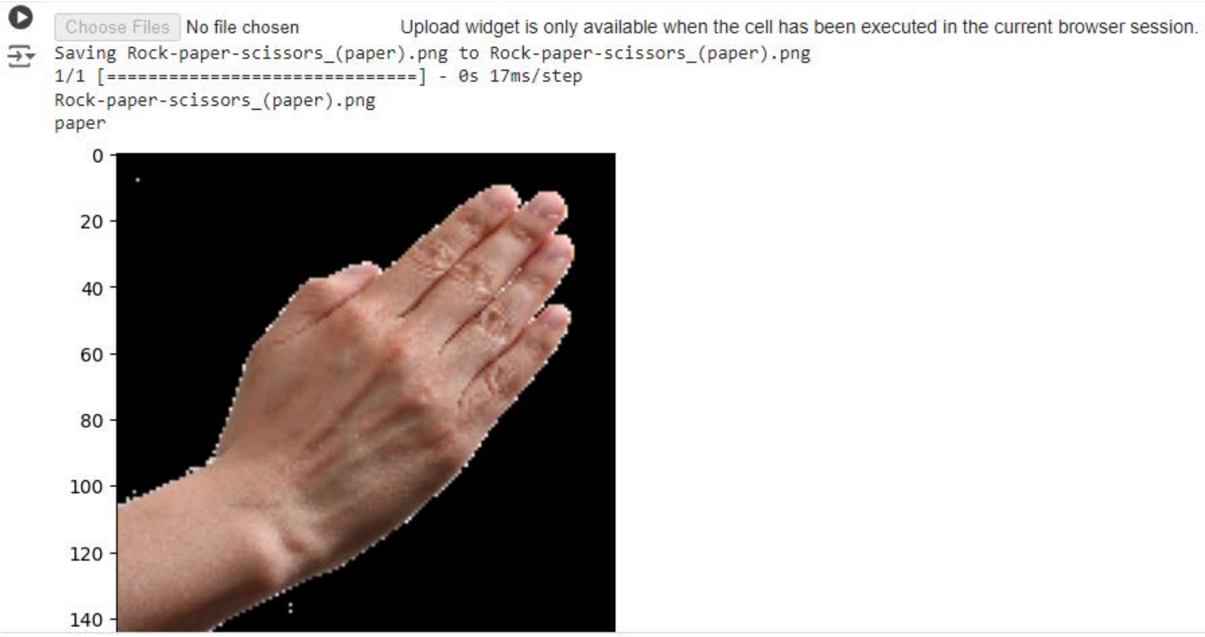
```
42/42 [==============================] - 9s 221ms/step - loss: 0.0455 - accuracy: 0.9825
28/28 [==============================] - 7s 228ms/step - loss: 0.0865 - accuracy: 0.9725
Training accuracy: 0.98
Validation accuracy: 0.97
```



```
[ ] import numpy as np
    from google.colab import files
    from tensorflow.keras.preprocessing import image
    import matplotlib.pyplot as plt
    import matplotlib.image as mpimg
    import matplotlib.pyplot as plt
    import matplotlib.image as mpimg
    %matplotlib inline
```



```
# memprediksi gambar
uploaded = files.upload()

for fn in uploaded.keys():

  # predicting images
  path = fn
  img = image.load_img(path, target_size=(150,150))

  imgplot = plt.imshow(img)
  x = image.img_to_array(img)
  x = np.expand_dims(x, axis=0)
  images = np.vstack([x])
  classes = model.predict(images, batch_size=10)
  print(fn)
  if classes[0][0]==1:
    print('paper')
  elif classes[0][1]==1:
    print('rock')
  elif classes[0][2]==1:
    print('scissors')
  else:
    print('unknown')
```



Upload widget is only available when the cell has been executed in the current browser session.

```
Saving Rock-paper-scissors_(paper).png to Rock-paper-scissors_(paper).png
1/1 [==============================] - 0s 17ms/step
Rock-paper-scissors_(paper).png
paper
```