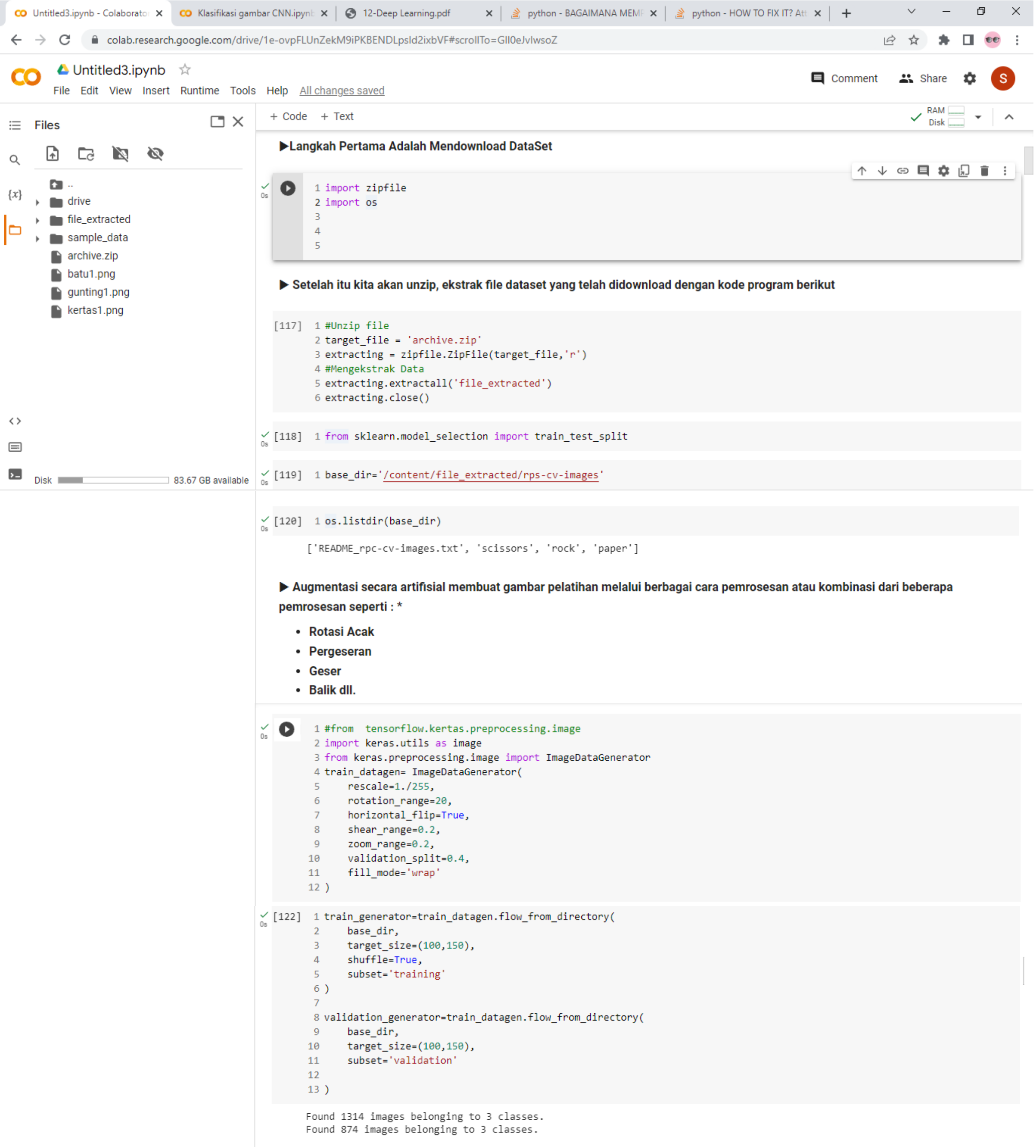
MODUL 12

Deep Learning : Pengenalan Gambar Batu, Gunting, Kertas dengan Convolutional Neural Network

Implementasi Dengan Python

Dalam proses pengenalan, gambar akan diubah menjadi matriks dan vektor kemudian akan di training. Setelah proses tersebut model akan bisa digunakan untuk pendeteksian.


Untitled3.ipynb - Colaborato
Klasifikasi gambar CNN.ipynb
12-Deep Learning.pdf
python - BAGAIMANA MEM
python - HOW TO FIX IT? Att
colab.research.google.com/drive/1e-ovpFLUnZekM9iPKBENDLpsld2ixbVF#scrollTo=GlI0eJvlwsoZ
Untitled3.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
S
+ Code + Text
RAM
Disk
Files
..
drive
file_extracted
sample_data
archive.zip
batu1.png
gunting1.png
kertas1.png
Disk
83.67 GB available
0s
[117]
[118]
[119]
[120]
[122]

▶Langkah Pertama Adalah Mendownload DataSet

```
import zipfile
import os



```

▶ Setelah itu kita akan unzip, ekstrak file dataset yang telah didownload dengan kode program berikut

```
#Unzip file
target_file = 'archive.zip'
extracting = zipfile.ZipFile(target_file,'r')
#Mengekstrak Data
extracting.extractall('file_extracted')
extracting.close()
```

```
from sklearn.model_selection import train_test_split
```

```
base_dir='/content/file_extracted/rps-cv-images'
```

```
os.listdir(base_dir)
```

```
['README_rpc-cv-images.txt', 'scissors', 'rock', 'paper']
```

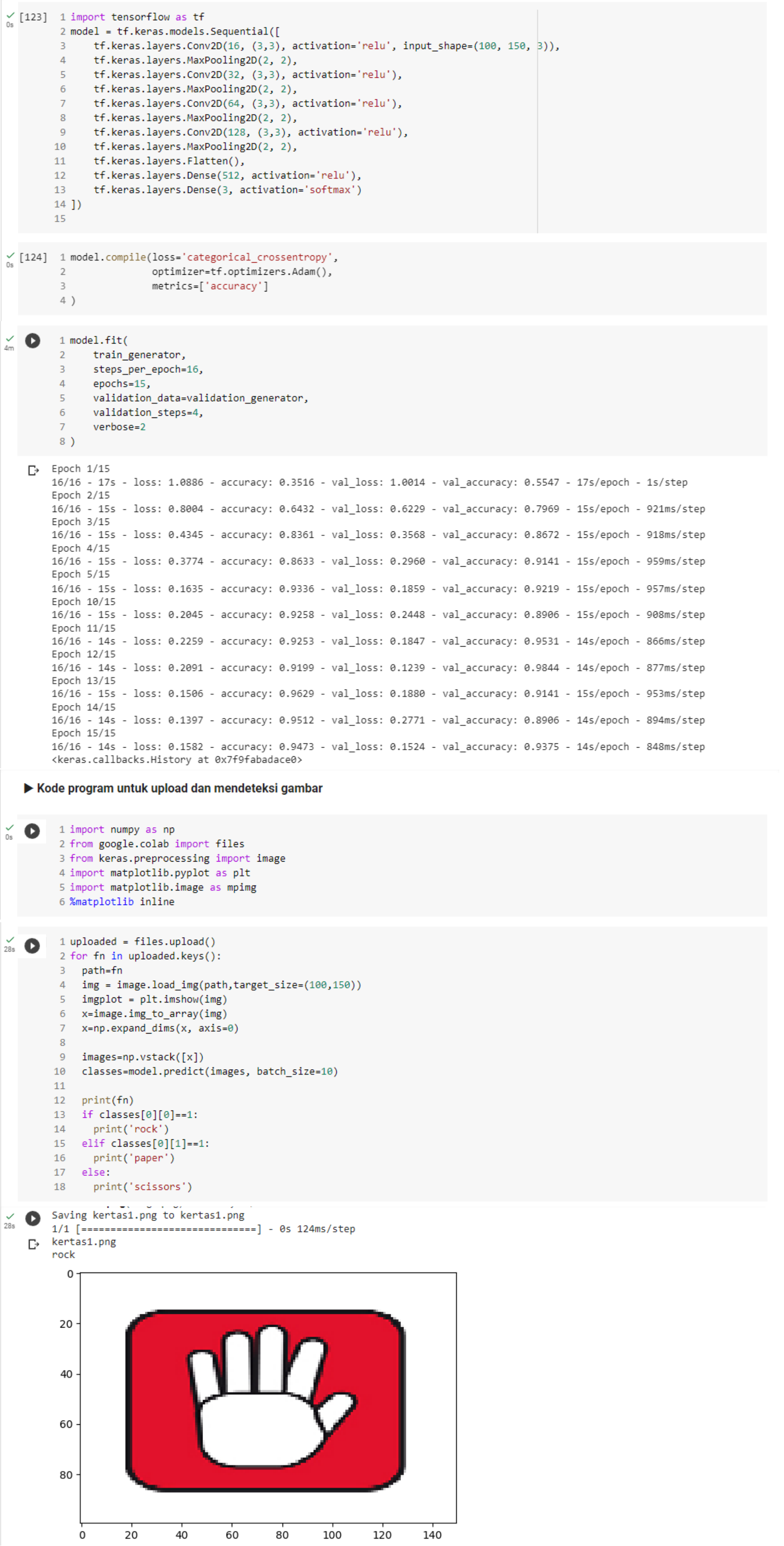
▶ Augmentasi secara artifisial membuat gambar pelatihan melalui berbagai cara pemrosesan atau kombinasi dari beberapa pemrosesan seperti : *

- Rotasi Acak
- Pergeseran
- Geser
- Balik dll.

```
#from  tensorflow.kertas.preprocessing.image
import keras.utils as image
from keras.preprocessing.image import ImageDataGenerator
train_datagen= ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    horizontal_flip=True,
    shear_range=0.2,
    zoom_range=0.2,
    validation_split=0.4,
    fill_mode='wrap'
)
```

```
train_generator=train_datagen.flow_from_directory(
    base_dir,
    target_size=(100,150),
    shuffle=True,
    subset='training'
)

validation_generator=train_datagen.flow_from_directory(
    base_dir,
    target_size=(100,150),
    subset='validation'

)
```

```
Found 1314 images belonging to 3 classes.
Found 874 images belonging to 3 classes.
```



[123]

```python
import tensorflow as tf
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(100, 150, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
    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.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(3, activation='softmax')
])

```

[124]

```python
model.compile(loss='categorical_crossentropy',
              optimizer=tf.keras.optimizers.Adam(),
              metrics=['accuracy']
)
```

```python
model.fit(
    train_generator,
    steps_per_epoch=16,
    epochs=15,
    validation_data=validation_generator,
    validation_steps=4,
    verbose=2
)
```

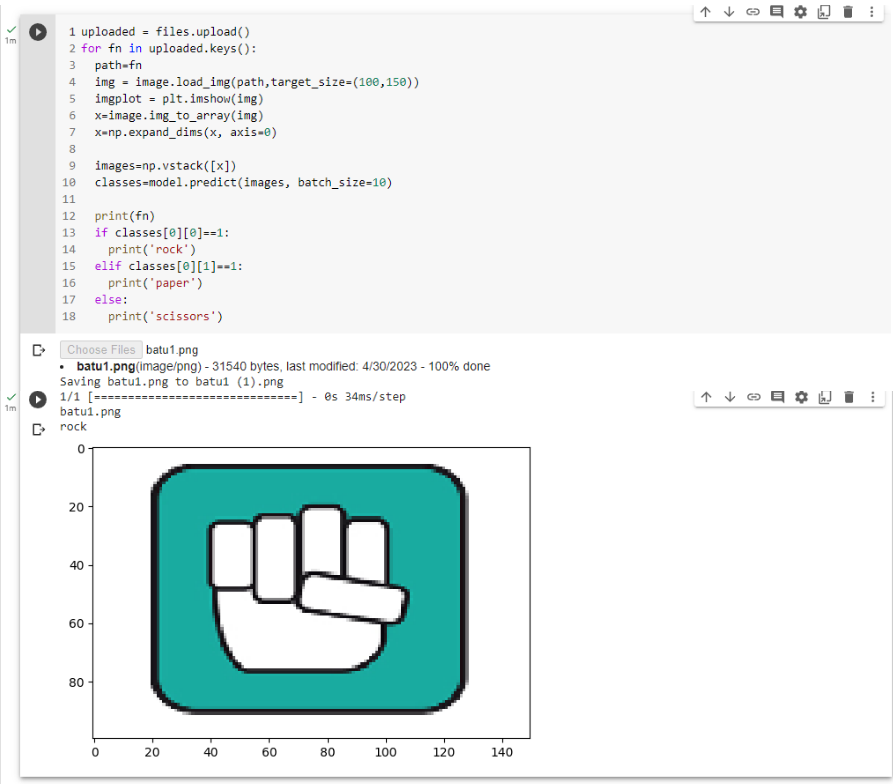
```
Epoch 1/15
16/16 - 17s - loss: 1.0886 - accuracy: 0.3516 - val_loss: 1.0014 - val_accuracy: 0.5547 - 17s/epoch - 1s/step
Epoch 2/15
16/16 - 15s - loss: 0.8004 - accuracy: 0.6432 - val_loss: 0.6229 - val_accuracy: 0.7969 - 15s/epoch - 921ms/step
Epoch 3/15
16/16 - 15s - loss: 0.4345 - accuracy: 0.8361 - val_loss: 0.3568 - val_accuracy: 0.8672 - 15s/epoch - 918ms/step
Epoch 4/15
16/16 - 15s - loss: 0.3774 - accuracy: 0.8633 - val_loss: 0.2960 - val_accuracy: 0.9141 - 15s/epoch - 959ms/step
Epoch 5/15
16/16 - 15s - loss: 0.1635 - accuracy: 0.9336 - val_loss: 0.1859 - val_accuracy: 0.9219 - 15s/epoch - 957ms/step
Epoch 10/15
16/16 - 15s - loss: 0.2045 - accuracy: 0.9258 - val_loss: 0.2448 - val_accuracy: 0.8906 - 15s/epoch - 908ms/step
Epoch 11/15
16/16 - 14s - loss: 0.2259 - accuracy: 0.9253 - val_loss: 0.1847 - val_accuracy: 0.9531 - 14s/epoch - 866ms/step
Epoch 12/15
16/16 - 14s - loss: 0.2091 - accuracy: 0.9199 - val_loss: 0.1239 - val_accuracy: 0.9844 - 14s/epoch - 877ms/step
Epoch 13/15
16/16 - 15s - loss: 0.1506 - accuracy: 0.9629 - val_loss: 0.1880 - val_accuracy: 0.9141 - 15s/epoch - 953ms/step
Epoch 14/15
16/16 - 14s - loss: 0.1397 - accuracy: 0.9512 - val_loss: 0.2771 - val_accuracy: 0.8906 - 14s/epoch - 894ms/step
Epoch 15/15
16/16 - 14s - loss: 0.1582 - accuracy: 0.9473 - val_loss: 0.1524 - val_accuracy: 0.9375 - 14s/epoch - 848ms/step
<keras.callbacks.History at 0x7f9fabadace0>
```

▶ **Kode program untuk upload dan mendeteksi gambar**

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

```python
uploaded = files.upload()
for fn in uploaded.keys():
  path=fn
  img = image.load_img(path,target_size=(100,150))
  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('rock')
  elif classes[0][1]==1:
    print('paper')
  else:
    print('scissors')
```

```
Saving kertas1.png to kertas1.png
1/1 [==============================] - 0s 124ms/step
kertas1.png
rock
```

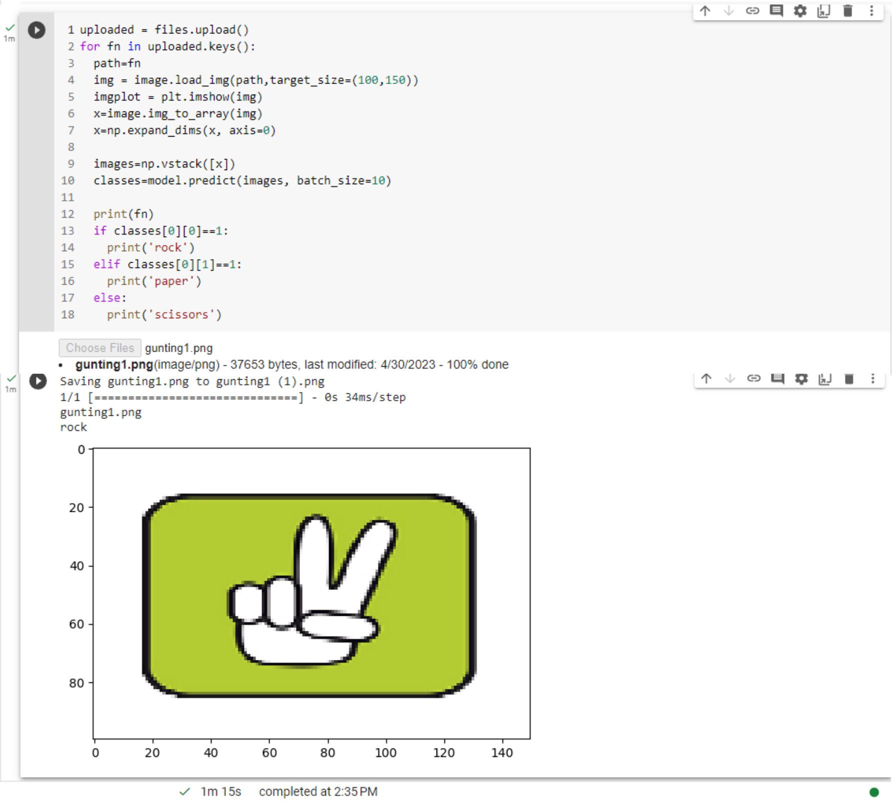


```python
uploaded = files.upload()
for fn in uploaded.keys():
  path=fn
  img = image.load_img(path,target_size=(100,150))
  imgplot = plt.imshow(img)
  x=image.img_to_array(img)
  x=np.expand_dims(x, axis=0)

  images=np.vstack([x])
  classes=model.predict(images, batch_size=10)

  print(fn)
  if classes[0][0]==1:
    print('rock')
  elif classes[0][1]==1:
    print('paper')
  else:
    print('scissors')
```

Choose Files batu1.png

- **batu1.png**(image/png) - 31540 bytes, last modified: 4/30/2023 - 100% done

```
Saving batu1.png to batu1 (1).png
1/1 [==============================] - 0s 34ms/step
batu1.png
rock
```



```python
uploaded = files.upload()
for fn in uploaded.keys():
  path=fn
  img = image.load_img(path,target_size=(100,150))
  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('rock')
  elif classes[0][1]==1:
    print('paper')
  else:
    print('scissors')
```

Choose Files gunting1.png

- **gunting1.png**(image/png) - 37653 bytes, last modified: 4/30/2023 - 100% done

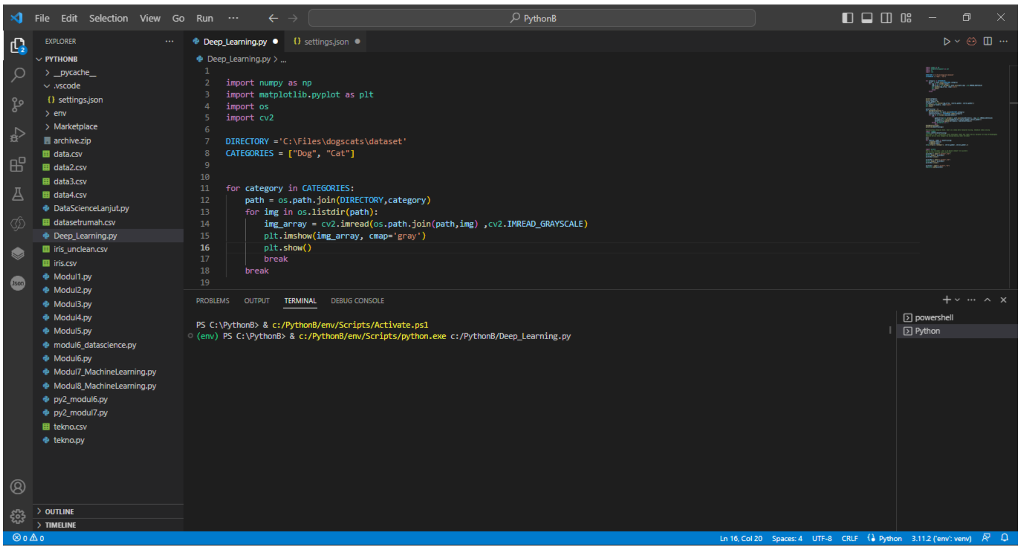
```
Saving gunting1.png to gunting1 (1).png
1/1 [==============================] - 0s 34ms/step
gunting1.png
rock
```

✓ 1m 15s completed at 2:35 PM

MODUL 13

Deep Learning : Pengenalan Anjing dan Kucing dengan TensorFlow dan Arsitektur Convolutional Neural Network

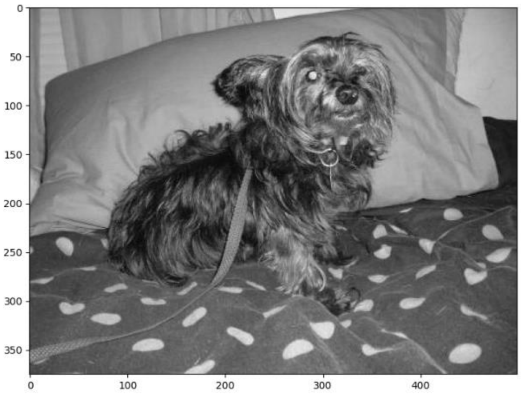
Implementasi dengan Python



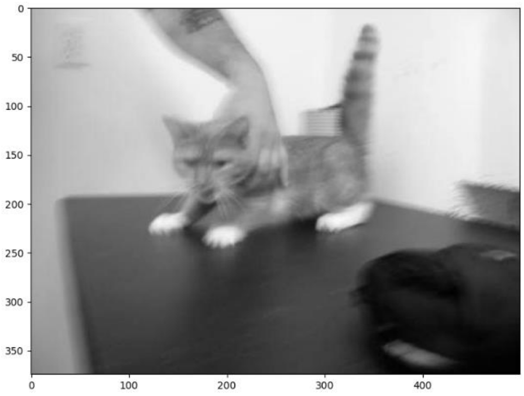
```python

import numpy as np
import matplotlib.pyplot as plt
import os
import cv2

DIRECTORY ='C:\Files\dogscats\dataset'
CATEGORIES = ["Dog", "Cat"]


for category in CATEGORIES:
    path = os.path.join(DIRECTORY,category)
    for img in os.listdir(path):
        img_array = cv2.imread(os.path.join(path,img) ,cv2.IMREAD_GRAYSCALE)
        plt.imshow(img_array, cmap='gray')
        plt.show()
        break
    break

```

```
PS C:\PythonB> & c:/PythonB/env/Scripts/Activate.ps1
(env) PS C:\PythonB> & c:/PythonB/env/Scripts/python.exe c:/PythonB/Deep_Learning.py
```

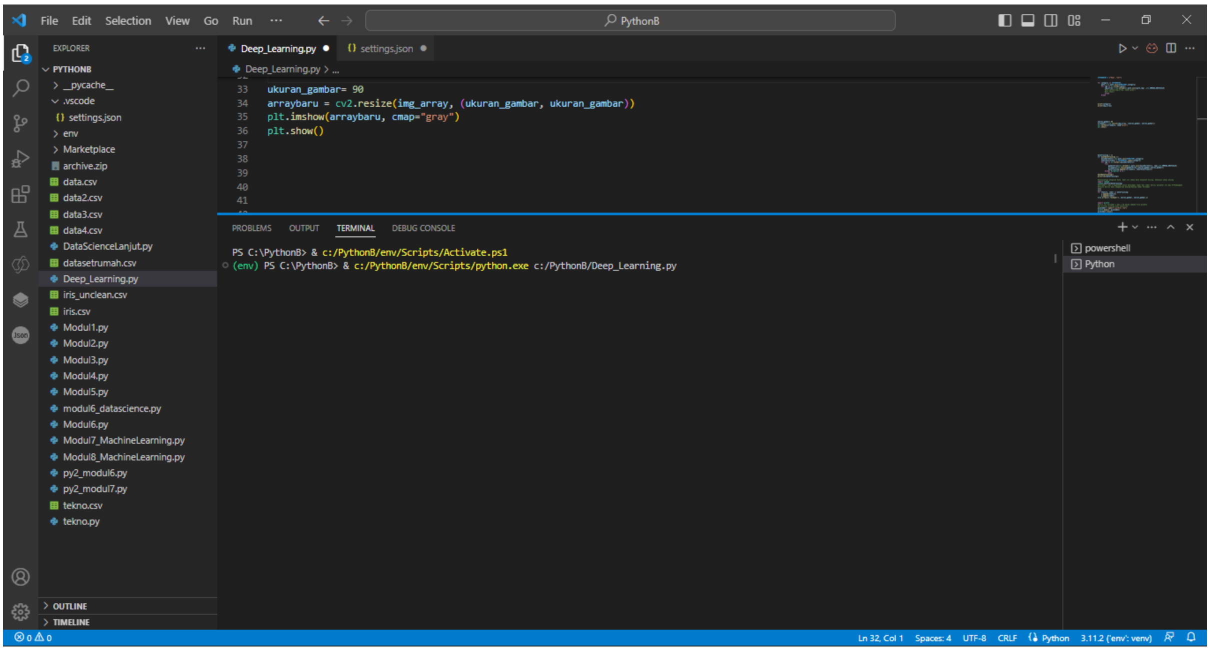
Kategori Anjing



Kategori Kucing



Resize gambar ke ukuran 90x90 pixel

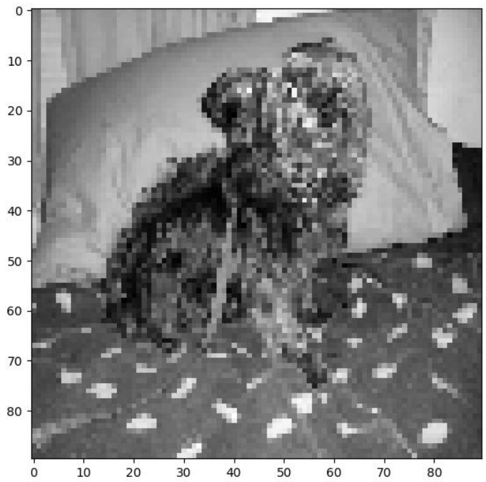


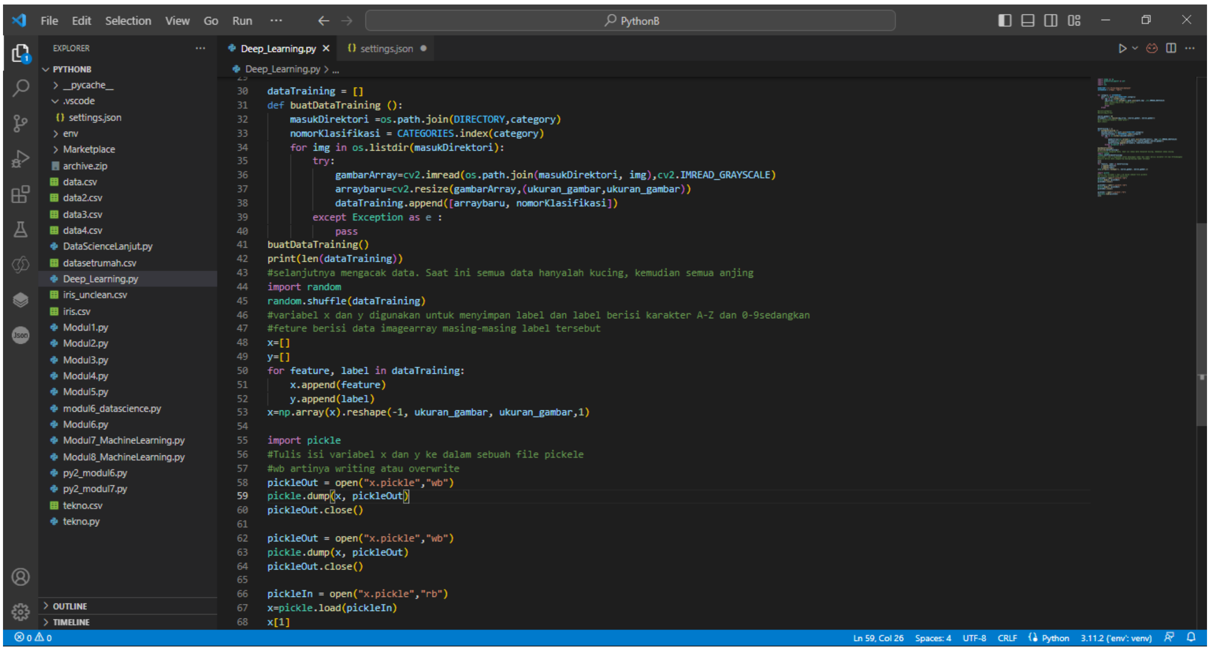
File Edit Selection View Go Run
PythonB
EXPLORER
PYTHONB
Deep_Learning.py
settings.json
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
Python
Ln 32, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.2 ('env': venv)

```python
ukuran_gambar= 90
arraybaru = cv2.resize(img_array, (ukuran_gambar, ukuran_gambar))
plt.imshow(arraybaru, cmap="gray")
plt.show()
```

```
PS C:\PythonB> & c:/PythonB/env/Scripts/Activate.ps1
(env) PS C:\PythonB> & c:/PythonB/env/Scripts/python.exe c:/PythonB/Deep_Learning.py
```

Hasil resize gambar anjing





```python
dataTraining = []
def buatDataTraining ():
    masukDirektori =os.path.join(DIRECTORY,category)
    nomorKlasifikasi = CATEGORIES.index(category)
    for img in os.listdir(masukDirektori):
        try:
            gambarArray=cv2.imread(os.path.join(masukDirektori, img),cv2.IMREAD_GRAYSCALE)
            arraybaru=cv2.resize(gambarArray,(ukuran_gambar,ukuran_gambar))
            dataTraining.append([arraybaru, nomorKlasifikasi])
        except Exception as e :
            pass
buatDataTraining()
print(len(dataTraining))
#selanjutnya mengacak data. Saat ini semua data hanyalah kucing, kemudian semua anjing
import random
random.shuffle(dataTraining)
#variabel x dan y digunakan untuk menyimpan label dan label berisi karakter A-Z dan 0-9sedangkan
#feture berisi data imagearray masing-masing label tersebut
x=[]
y=[]
for feature, label in dataTraining:
    x.append(feature)
    y.append(label)
x=np.array(x).reshape(-1, ukuran_gambar, ukuran_gambar,1)

import pickle
#Tulis isi variabel x dan y ke dalam sebuah file pickele
#wb artinya writing atau overwrite
pickleOut = open("x.pickle","wb")
pickle.dump(x, pickleOut)
pickleOut.close()

pickleOut = open("x.pickle","wb")
pickle.dump(x, pickleOut)
pickleOut.close()

pickleIn = open("x.pickle","rb")
x=pickle.load(pickleIn)
x[1]
```