



LAPORAN PRAKTIKUM
Praktikum Penambangan Data
Projek Ujian Akhir Semester

Pengampu :
Warnia Ningsih Sikumbang, S.Kom, M.Kom
Asmarini, S.Tr.Kom

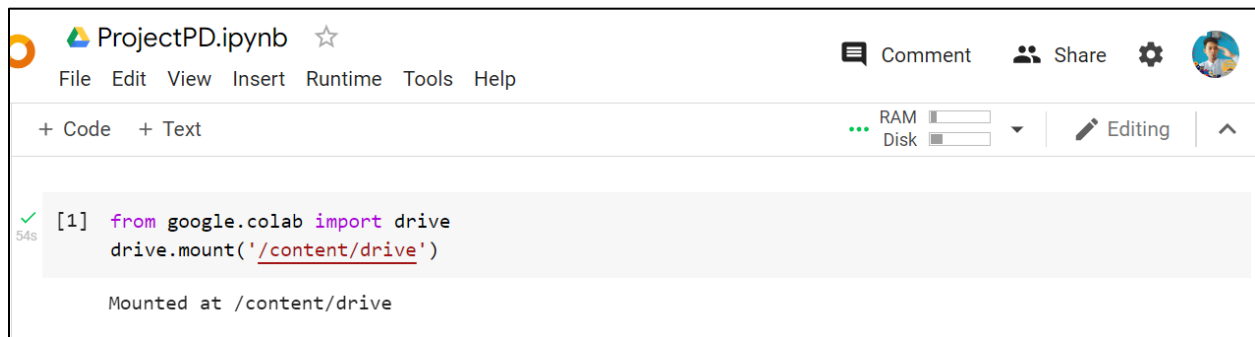
Kelompok 11
Nagita Tria Jeslina & Zulkarnain
2157301062 & 2157301097
Kelas : 2 SI A

Hari/Tanggal : 14 January 2023
TA 2023 / 2024

Pemodelan Data H5

H5 adalah salah satu Format Data Hierarkis (HDF) yang digunakan untuk menyimpan data dalam jumlah besar. Ini digunakan untuk menyimpan sejumlah besar data dalam bentuk array multidimensi. Format ini terutama digunakan untuk menyimpan data ilmiah yang tertata dengan baik untuk pengambilan dan analisis cepat. H5 diperkenalkan sebagai format file yang lebih disempurnakan ke H4. Awalnya dikembangkan oleh National Center for Supercomputing Applications, dan sekarang didukung oleh The HDF Group.

1.)



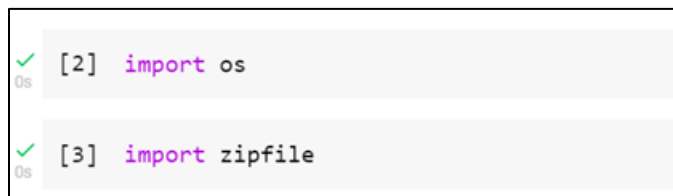
The screenshot shows a Google Colab notebook titled 'ProjectPD.ipynb'. The interface includes a top bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help' menus. Below the menu bar, there are tabs for '+ Code' and '+ Text'. On the right side, there are icons for 'Comment', 'Share', and a settings gear. A status bar at the bottom right shows 'RAM' and 'Disk' usage. The first code cell is executed, showing the output 'Mounted at /content/drive'.

```
[1] from google.colab import drive
    drive.mount('/content/drive')
```

Mounted at /content/drive

Analisa : Pada percobaan kali ini kita akan mengimport drive

2.)



The screenshot shows the second and third code cells of the notebook. The second cell imports the 'os' module, and the third cell imports the 'zipfile' module. Both cells are executed successfully.

```
[2] import os
```

```
[3] import zipfile
```

Analisa : Dan disini kita akan mengimport os yang nantinya ada hubungannya dengan sistem operasi dan disini ada import zipfile yang berguna untuk mengextract zip

3.)



The screenshot shows the fourth and fifth code cells of the notebook. The fourth cell imports 'ImageDataGenerator' from 'tensorflow.keras.preprocessing.image', and the fifth cell imports 'layers' from 'tensorflow.keras'. Both cells are executed successfully.

```
[4] from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
[5] from tensorflow.keras import layers
```

Analisa : Dan disini terdapat tensorflow.keras.preprocessing. image import ImageDataGenerator yang fungsinya nantinya untuk menggunakan framework machine learning tensorflow. Dan disini akan dibuatkan juga dari fungsi image nya yang bisa membuat program bisa mengenerate data dari image atau dari sebuah file yang sudah kita buat. Dan disini ada tensorflow.keras import layers yang berarti ia mengimport data dari tensorflow.keras dalam bentuk layers.

4.)

```
✓ [6] local_zip = '/content/drive/MyDrive/Colab Notebooks/Dataset Seragam TNI.zip'
✓ [7] zip_ref = zipfile.ZipFile(local_zip, 'r')

[8] zip_ref.extractall('/content/drive/MyDrive/Colab Notebooks')
[9] zip_ref.close()
```

Analisa : Dan disini terdapat local_zip = '/content/drive/MyDrive/Colab Notebooks/Dataset Seragam TNI.zip' merupakan file yang akan di extract dan akan diubah dalam bentuk zipfile. Dan seperti biasa kita masukkan kedalam google drive yang foldernya terdapat google colab. Dan disini juga ada zip_ref yang = zip.file.ZipFile(local_zip, 'r') yang berfungsi untuk mengextract file zip. Dan disini juga terdapat zip_ref.extractall('/content/drive/MyDrive/Colab Notebookd') yang berfungsi untuk meletakkan dimana file hasil dari extract zip file tadi yang sudah kita extract.

5.)

```
✓ [10] train_dir = os.path.join('/content/drive/MyDrive/Colab Notebooks/Dataset Seragam TNI')
```

Analisa : Dan disini berfungsi untuk membuat direktori

6.)

```
✓ [11] train_datagen = ImageDataGenerator(rescale=1./255,
0s rotation_range=20,
zoom_range=0.2,
shear_range=0.2,
fill_mode='nearest',
validation_split=0.1)
```

```
✓ [12] train_generator = train_datagen.flow_from_directory(
0s train_dir,
target_size=(150,150),
batch_size=8,
class_mode='categorical',
subset='training')
```

Found 277 images belonging to 3 classes.

Analisa : Dan disini kita akan diperintahkan untuk membuat objek di nama train_datagen. Dan disini juga ada train_dir yang merupakan direktori untuk data latih. Dan ada rescale yang berfungsi untuk menormalisasi gambar dengan pixel yang nilainya menjadi dari 0 – 1. Dan ada Rotation_range yang berfungsi untuk menentukan fix nya dari derajat rotasi. Dan ada shear_renge untuk memiringkan gambar berdasarkan sumbu. dan ada fill_mode yang fungsinya untuk memilih gambar mana yang telah dilakukan testingnya. Dan ada validation split yang fungsinya terbagi dengan 2 macam yang bersifat acak. Dan data trainingnya untuk sebagai data testing. Dan ada target size berfungsi untuk mengubah kualitas dari gambar. Dan batch size berfungsi untuk menentukan berapa jumlah sampel yang harus dikerjakan dan diperbarui menggunakan parameter. Dan ada class_mode merupakan pembagian kelas dikarenakan kita akan menentukan 3 jenis klasifikasi yaitu seragam TNI AD, seragam TNI AL, dan seragam TNI AU. Dan ada subset = 'training' yang fungsinya untuk training model. Dan disini terdapat 277 gambar yang terbaca dengan terbagi 3 kelas.

7.)

```
✓ [13] validation_generator = train_datagen.flow_from_directory(
0s train_dir,
target_size=(150,150),
batch_size=16,
class_mode='categorical',
subset='validation')
```

Found 29 images belonging to 3 classes.

Analisa : dan disini ada subset='validation' yang fungsinya untuk menandai

8.)

```
✓ [14] import tensorflow as tf
0s

✓ [15] from tensorflow.keras.layers import Input
0s

✓ [16] from tensorflow.keras.applications import ResNet50
0s

✓ [17] from tensorflow.keras.applications import ResNet152V2
0s

✓ [18] from tensorflow.python.ops.gen_array_ops import Shape
0s
```

Analisa : Terdapat import tensorflow as tf yang merupakan library untuk mengimport tensorflow. Dan ada import Input yang berfungsi untuk memproses inputan. Dan ada import ResNet50 untuk klasifikasi gambar. Dan ada import ResNet152V2 yang berfungsi untuk learning dari klasifikasi gambar. Dan ada import Shape yang berfungsi untuk mendapatkan berapa banyak elemen yang ada di setiap dimensi.

9.)

```
[19] import tensorflow as tf
      from tensorflow.keras.layers import Input
      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(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(64, (3,3), activation='relu'),
          tf.keras.layers.MaxPooling2D(2,2),
          tf.keras.layers.Dropout(0.4),
          tf.keras.layers.Flatten(),
          tf.keras.layers.Dense(512,activation='relu'),
          tf.keras.layers.Dropout(0.4),
          tf.keras.layers.Dense(256,activation='relu'),
          tf.keras.layers.Dropout(0.4),
          tf.keras.layers.Dense(3,activation='softmax')#for multiclass
      ])

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

Analisa : Disini terdapat perintah untuk membuat model dari CNN yang merupakan jaringan neural untuk penggunaan klasifikasi gambar dan sudah menghasilkan hasil dari beberapa klasifikasi gambar yang ada. Dan disini terdapat variabel model yang akan menampung tf.keras.model. Dan disini ada Conv2 yang fungsinya digunakan untuk melakukan 2D gambar dengan nilai filternya 32, dan disini diensi dari filternya adalah 3x3 dan ada input shape yang berfungsi untuk membentuk dari tipe induknya. Dan ada Maxpooling 2D yang memiliki fungsi isinya kernel yang nantinya akan mengambil nilai paling besar dari dimensi kernel. Dan ada Dropout yang fungsinya untuk mencegah overfitting yang nantinya akan mengurangi koneksi dari neuron yang berbentuk matriks 2x2. Dan ada Flatten yang fungsinya untuk membuat inputan yang dimensinya banyak berubah menjadi 1 dimensi. Dan dense merupakan layer pada model arsitektur yang mempunyai banyak neuron, dan untuk menambahkan layers fully connected yang labelnya berjumlah 3. Dan ada model.compile yang fungsinya untuk membaca dan menggabungkan source code dan diterjemahkan kedalam bahasa lain. Dan ada optimizers untuk mempercepat dari proses optimasi dengan algoritma adam. Dan ada loss yang nantinya akan digunakan untuk masalah klasifikasi biner. Dan ada metrics yang fungsinya untuk mengimplementasikan fungsi dari kerugian, score dan utilitas dalam mengukur kinerja dari klasifikasi gambar.

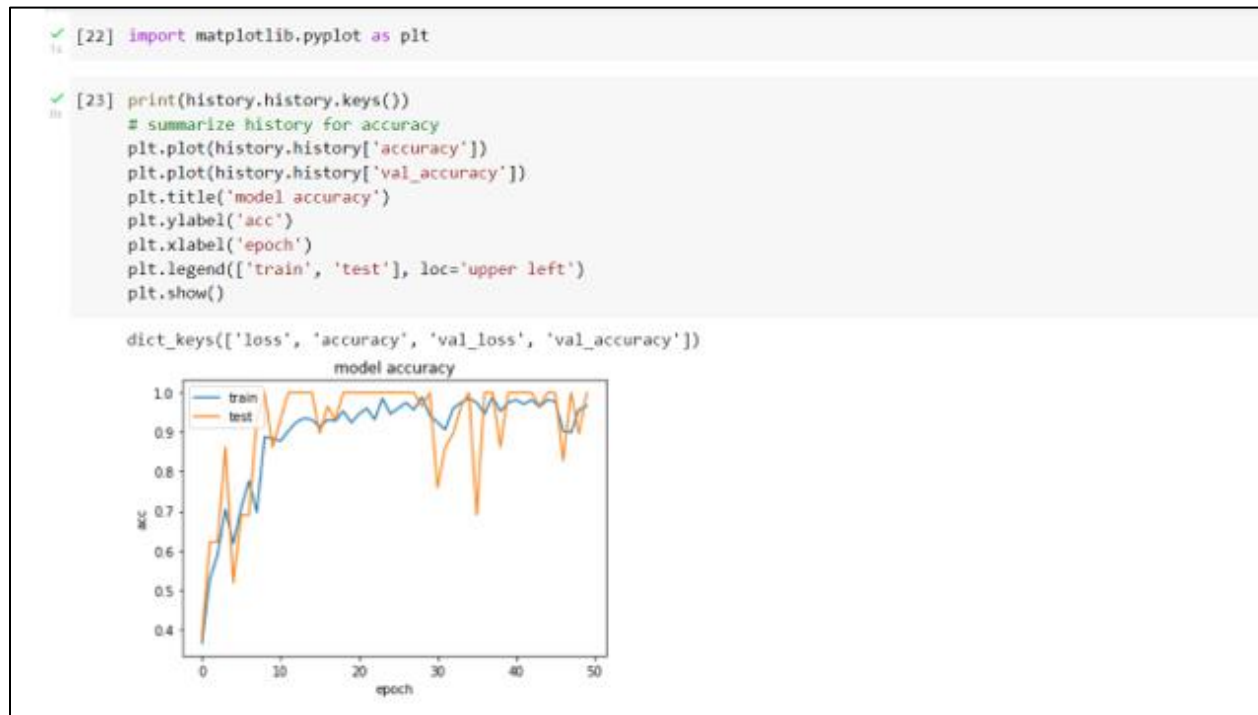
10.)

```
[21] history = model.fit(train_generator,
                        validation_data=validation_generator,
                        epochs=50,
                        verbose=2)

Epoch 1/50
35/35 - 14s - loss: 1.1093 - accuracy: 0.3646 - val_loss: 1.0957 - val_accuracy: 0.3793 - 14s/epoch - 395ms/step
Epoch 2/50
35/35 - 14s - loss: 0.9900 - accuracy: 0.5271 - val_loss: 0.7017 - val_accuracy: 0.6207 - 14s/epoch - 391ms/step
Epoch 3/50
35/35 - 11s - loss: 0.7506 - accuracy: 0.5884 - val_loss: 0.6819 - val_accuracy: 0.6207 - 11s/epoch - 322ms/step
Epoch 4/50
35/35 - 11s - loss: 0.6410 - accuracy: 0.7040 - val_loss: 0.3956 - val_accuracy: 0.8621 - 11s/epoch - 324ms/step
Epoch 5/50
35/35 - 11s - loss: 0.7266 - accuracy: 0.6173 - val_loss: 0.9988 - val_accuracy: 0.5172 - 11s/epoch - 323ms/step
Epoch 6/50
35/35 - 11s - loss: 0.6131 - accuracy: 0.7076 - val_loss: 0.5531 - val_accuracy: 0.6897 - 11s/epoch - 326ms/step
Epoch 7/50
35/35 - 11s - loss: 0.5058 - accuracy: 0.7762 - val_loss: 0.6474 - val_accuracy: 0.6897 - 11s/epoch - 325ms/step
Epoch 8/50
35/35 - 13s - loss: 0.7214 - accuracy: 0.6968 - val_loss: 0.3439 - val_accuracy: 0.9310 - 13s/epoch - 379ms/step
Epoch 9/50
35/35 - 11s - loss: 0.3786 - accuracy: 0.8881 - val_loss: 0.1097 - val_accuracy: 1.0000 - 11s/epoch - 321ms/step
Epoch 10/50
35/35 - 11s - loss: 0.3103 - accuracy: 0.8845 - val_loss: 0.4116 - val_accuracy: 0.8621 - 11s/epoch - 319ms/step
Epoch 11/50
35/35 - 11s - loss: 0.3223 - accuracy: 0.8773 - val_loss: 0.1726 - val_accuracy: 0.9310 - 11s/epoch - 321ms/step
Epoch 12/50
35/35 - 11s - loss: 0.2427 - accuracy: 0.9025 - val_loss: 0.0324 - val_accuracy: 1.0000 - 11s/epoch - 323ms/step
Epoch 13/50
35/35 - 12s - loss: 0.2660 - accuracy: 0.9242 - val_loss: 0.0945 - val_accuracy: 1.0000 - 12s/epoch - 333ms/step
Epoch 14/50
35/35 - 11s - loss: 0.1934 - accuracy: 0.9350 - val_loss: 0.0958 - val_accuracy: 1.0000 - 11s/epoch - 320ms/step
Epoch 15/50
35/35 - 11s - loss: 0.1728 - accuracy: 0.9314 - val_loss: 0.0306 - val_accuracy: 1.0000 - 11s/epoch - 321ms/step
Epoch 16/50
35/35 - 11s - loss: 0.3258 - accuracy: 0.9134 - val_loss: 0.2524 - val_accuracy: 0.8966 - 11s/epoch - 319ms/step
Epoch 17/50
35/35 - 13s - loss: 0.2062 - accuracy: 0.9314 - val_loss: 0.0886 - val_accuracy: 0.9655 - 13s/epoch - 381ms/step
Epoch 18/50
35/35 - 11s - loss: 0.2859 - accuracy: 0.9278 - val_loss: 0.1729 - val_accuracy: 0.9310 - 11s/epoch - 321ms/step
Epoch 19/50
35/35 - 11s - loss: 0.1500 - accuracy: 0.9531 - val_loss: 0.0417 - val_accuracy: 1.0000 - 11s/epoch - 322ms/step
Epoch 20/50
35/35 - 11s - loss: 0.2188 - accuracy: 0.9242 - val_loss: 0.0303 - val_accuracy: 1.0000 - 11s/epoch - 322ms/step
Epoch 21/50
35/35 - 11s - loss: 0.2630 - accuracy: 0.9458 - val_loss: 0.0774 - val_accuracy: 1.0000 - 11s/epoch - 320ms/step
Epoch 22/50
35/35 - 11s - loss: 0.1282 - accuracy: 0.9603 - val_loss: 0.0040 - val_accuracy: 1.0000 - 11s/epoch - 323ms/step
Epoch 23/50
35/35 - 11s - loss: 0.1761 - accuracy: 0.9314 - val_loss: 0.0358 - val_accuracy: 1.0000 - 11s/epoch - 318ms/step
Epoch 24/50
35/35 - 11s - loss: 0.0840 - accuracy: 0.9856 - val_loss: 0.0388 - val_accuracy: 1.0000 - 11s/epoch - 319ms/step
Epoch 25/50
35/35 - 11s - loss: 0.1434 - accuracy: 0.9458 - val_loss: 0.0396 - val_accuracy: 1.0000 - 11s/epoch - 318ms/step
Epoch 26/50
35/35 - 11s - loss: 0.1072 - accuracy: 0.9603 - val_loss: 0.0316 - val_accuracy: 1.0000 - 11s/epoch - 320ms/step
Epoch 27/50
35/35 - 11s - loss: 0.0698 - accuracy: 0.9747 - val_loss: 0.0127 - val_accuracy: 1.0000 - 11s/epoch - 319ms/step
Epoch 28/50
35/35 - 11s - loss: 0.1055 - accuracy: 0.9567 - val_loss: 0.0079 - val_accuracy: 1.0000 - 11s/epoch - 322ms/step
Epoch 29/50
35/35 - 11s - loss: 0.0387 - accuracy: 0.9892 - val_loss: 0.1590 - val_accuracy: 0.9655 - 11s/epoch - 320ms/step
Epoch 30/50
35/35 - 11s - loss: 0.2178 - accuracy: 0.9422 - val_loss: 0.0389 - val_accuracy: 1.0000 - 11s/epoch - 319ms/step
Epoch 31/50
35/35 - 11s - loss: 0.2434 - accuracy: 0.9242 - val_loss: 0.9712 - val_accuracy: 0.7586 - 11s/epoch - 321ms/step
Epoch 32/50
35/35 - 11s - loss: 0.2448 - accuracy: 0.9061 - val_loss: 0.2287 - val_accuracy: 0.8621 - 11s/epoch - 320ms/step
Epoch 33/50
35/35 - 11s - loss: 0.1227 - accuracy: 0.9603 - val_loss: 0.1330 - val_accuracy: 0.8966 - 11s/epoch - 321ms/step
Epoch 34/50
35/35 - 11s - loss: 0.0781 - accuracy: 0.9747 - val_loss: 0.0398 - val_accuracy: 0.9655 - 11s/epoch - 320ms/step
Epoch 35/50
35/35 - 11s - loss: 0.0471 - accuracy: 0.9856 - val_loss: 0.0055 - val_accuracy: 1.0000 - 11s/epoch - 320ms/step
```

Analisa : Dan disini kita diperintahkan untuk menampilkan history yang menampung model.fit yang berfungsi untuk mengukur nilai akurasi dataset dan memiliki parameter train_generator, validation_data yang isinya validation_generator. Dan ada epoch yang fungsinya untuk melakukan perulangan sebanyak 50 kali yang bertujuan untuk mendapatkan nilai akurasi. Dan ada verbose yang berfungsi untuk membuat tampilan progress menjadi 1/50 dan begitu sampai seterusnya.

11.)



Analisa : Dan disini terdapat import matplotlib.pyplot als plt merupakan library untuk membuat visualisasi hasil dari data yang sudah diproses dan hasilnya seperti plotgrafik,chart dan lain lain. Dan ada print history yang berfungsi untuk memanggil fungsi key. Dan ada accuracy untuk memanggil nilai akurasi. Dan ada val accuracy untuk memanggil nilai dari validation accuracy. Dan ada tittle yang berfungsi untuk membuat judul dari model accuracy. Dan ada ylabel yang fungsinya untuk membuat nama label y. Dan ada xlabel yang fungsinya untuk membuat nama label x. Dan ada legend untuk menampilkan label pada plot. Dan ada show yang fungsinya untuk menampilkan model.

Back End

Merupakan sisi server atau server side dari sebuah website atau aplikasi.

1.)

```
Python
%pip install flask-ngrok
%pip install gevent
%pip install flask
%pip install keras
%pip install tensorflow

[1] ✓ 11m 28.5s

... Requirement already satisfied: flask-ngrok in c:\users\zulka\appdata\local\programs\python\python38\lib\site-packages (0.0.25)
Requirement already satisfied: Flask>=0.8 in c:\users\zulka\appdata\local\programs\python\python38\lib\site-packages (from flask-ngrok) (2.2.2)
Requirement already satisfied: requests in c:\users\zulka\appdata\local\programs\python\python38\lib\site-packages (from flask-ngrok) (2.28.1)
Requirement already satisfied: Werkzeug>=2.2.2 in c:\users\zulka\appdata\local\programs\python\python38\lib\site-packages (from Flask>=0.8->flask-ngrok) (2.2.2)
Requirement already satisfied: click>=8.0 in c:\users\zulka\appdata\local\programs\python\python38\lib\site-packages (from Flask>=0.8->flask-ngrok) (8.1.3)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\zulka\appdata\local\programs\python\python38\lib\site-packages (from Flask>=0.8->flask-ngrok) (2.1.2)
Requirement already satisfied: Jinja2>=3.0 in c:\users\zulka\appdata\local\programs\python\python38\lib\site-packages (from Flask>=0.8->flask-ngrok) (3.1.2)
Requirement already satisfied: importlib-metadata>=3.6.0; python_version < "3.10" in c:\users\zulka\appdata\local\programs\python\python38\lib\site-packages (from Flask>=0.8->flask-ngrok) (5.2.0)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\zulka\appdata\local\programs\python\python38\lib\site-
```

Analisa : Disini kita akan melakukan penginstalan terhadap library yang akan kita gunakan. Disini ada instal flask-ngrok yang fungsinya untuk menjadi kerangka dari suatu web. Dan disini ada instal gevent yang fungsinya untuk kebutuhan dari perulangan data saat melakukan test. Dan disini ada keras yang merupakan pengembangan untuk sebuah pengaplikasian segala jenis operasi dari fungsi dee learning yang dilakukan secara umum. Dan disini ada tensorflow yang fungsinya untuk mengaplikasikan fungsi deep learning.

2.)

```
from __future__ import division, print_function
# coding=utf-8
import sys
import os
import glob
import re
import numpy as np

# Keras
from keras.applications.imagenet_utils import preprocess_input, decode_predictions
from keras.models import load_model
from keras.preprocessing import image

# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer
```

✓ 0.1s

Python

Analisa : Disini kita diperintahkan untuk mengimport library yang akan kita buat. Disini ada import sys yang fungsinya untuk mengakses konfigurasi yang berhubungan dengan sistem operasi. Dan ada os yang fungsinya untuk melakukan operasi dengan sistem operasi. Dan ada glob yang fungsinya untuk mengakses atau mengambil file yang nama pathnya sesuai dengan yang sudah ditentukan. Dan ada re yang merupakan pengurutan pencarian. Dan disini ada numpy yang berfungsi untuk memudahkan kita melakukan penghitungan.

3.)

```
%pip install pandas
```

✓

Python

^C

Note: you may need to restart the kernel to use updated packages.

Collecting pandas Downloading pandas-1.5.2-cp38-cp38-win_amd64.whl (11.0 MB) Requirement already satisfied: numpy>=1.20.3; python_version < "3.10" in c:\users\zulka\appdata\local\programs\python\python38\lib\site-packages (from pandas) (1.24.0) Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\zulka\appdata\roaming\python\python38\site-packages (from pandas) (2.8.2) Collecting pytz>=2020.1 Downloading pytz-2022.7-py2.py3-none-any.whl (499 kB) Requirement already satisfied: six>=1.5 in c:\users\zulka\appdata\roaming\python\python38\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0) Installing collected packages: pytz, pandas Successfully installed pandas-1.5.2 pytz-2022.7

WARNING: You are using pip version 20.1.1; however, version 22.3.1 is available.

You should consider upgrading via the 'c:\Users\zulka\AppData\Local\Programs\Python\Python38\python.exe -m pip install --upgrade pip' command.

Analisa : Disini kita akan melakukan penginstalan pandas untuk melakukan dataset

4.)

```
import flask
from flask import request
import pandas as pd
import tensorflow as tf
import keras
import numpy as np
import random
import os
from os.path import join, dirname, realpath
from tensorflow.keras.preprocessing.image import img_to_array, load_img
from flask_ngrok import run_with_ngrok
from werkzeug.utils import secure_filename
```

✓ 5.6s

Python

Analisa : Disini terdapat import flask yang berfungsi untuk menjadi kerangka kerja tampilan dari web agar lebih rapi dan terstruktur. Dan disini ada import request yang berfungsi untuk menngirim berbagai request HTTP. Dan disini ada import pandas yang gunanya nanti akan mengelola dataset. Dan ada import tenserflow yang berfungsi untuk mendeskripsikan properti skala dan vektor. Dan ada import keras yang fungsinya untuk menyederhanakan pengimplementasian deep learning. Dan ada import numpy untuk mengelola pendataan array. Dan ada import random untuk menghasilkan nilai acak. Dan ada import os yang fungsinya berkaitan dengan sistem operasi dan pembuatan direktori,membuat file,menghapus direktori dan menggabungkan direktori menjadi satu. Dan ada import join yang fungsinya untuk mengurutkan elemen yang telah dipisah oleh separator. Dan ada img_to_array yang fungsinya untuk konversi image ke array dan load_img untuk load data gambar. Dan ada import run_with_ngrok yang fungsinya menjalankan data dengan ngrok. Dan disini ada import secure_filename yang fungsinya untuk menjaga gambar yang hendak kita kelola.

5.)

```
app = Flask(__name__)

model = tf.keras.models.load_model('PrediksiSeragamTNI.h5')
run_with_ngrok(app)
```

✓ 1.9s

Python

Analisa : Disini terdapat variabel app yang tampungannya ada flask(name). Dan disini ada variabel model yang tampungannya tf.keras.model untuk pemodelan file H5 yang sudah kita buat. Dan disini ada run_with_ngrok yang memiliki parameter app

6.)

```
%pip install pillow
✓ 13.4s Python

Collecting pillowNote: you may need to restart the kernel to use updated packages.

WARNING: You are using pip version 20.1.1; however, version 22.3.1 is available.
You should consider upgrading via the 'c:\Users\zulka\AppData\Local\Programs\Python\Python38\python.exe -m pip install --upgrade pip' command.

Downloading Pillow-9.3.0-cp38-cp38-win_amd64.whl (2.5 MB)
Installing collected packages: pillow
Successfully installed pillow-9.3.0
```

Analisa : Disini terdapat pip install pilow dan kita instal agar bisa tersedia format gambar yang terdiri dari JPEG,PNG dan pcara proses input gambar.

7.)

```
@app.route('/', methods=['GET'])
def index():
    return "<h1>Hello World</h1>"
@app.route('/predict', methods=['GET', 'POST'])
def upload():
    data={"success":False}
    namafile=''
    if request.method == 'POST':
        file = request.files['file']

        if file.filename == '':
            data["file"]="Tidak ada file"
        else:
            data["file"]="File berhasil di simpan"
            filename = secure_filename(file.filename)
            file.save('data_test/'+file.filename)
            namafile='data_test/'+file.filename
            img = load_img(namafile, target_size=(150, 150))

            x = img_to_array(img)
            x = x.reshape((1,) + x.shape)
            x /= 255.0
            predict=model.predict(x)
            temp=0
            label=0
            hasil=[]

            for y in range(4):
                presentase=predict[0][y]*100
                hasil.append(predict[0][y]*100)
                if presentase>temp:
                    temp=presentase
                    label=y
                data["success"]=True
                data["label"]=label
            print(data)
            return flask.jsonify(data)
        else:
            return "<h1>Method salah</h1>"
app.run()

3m 31.8s Python

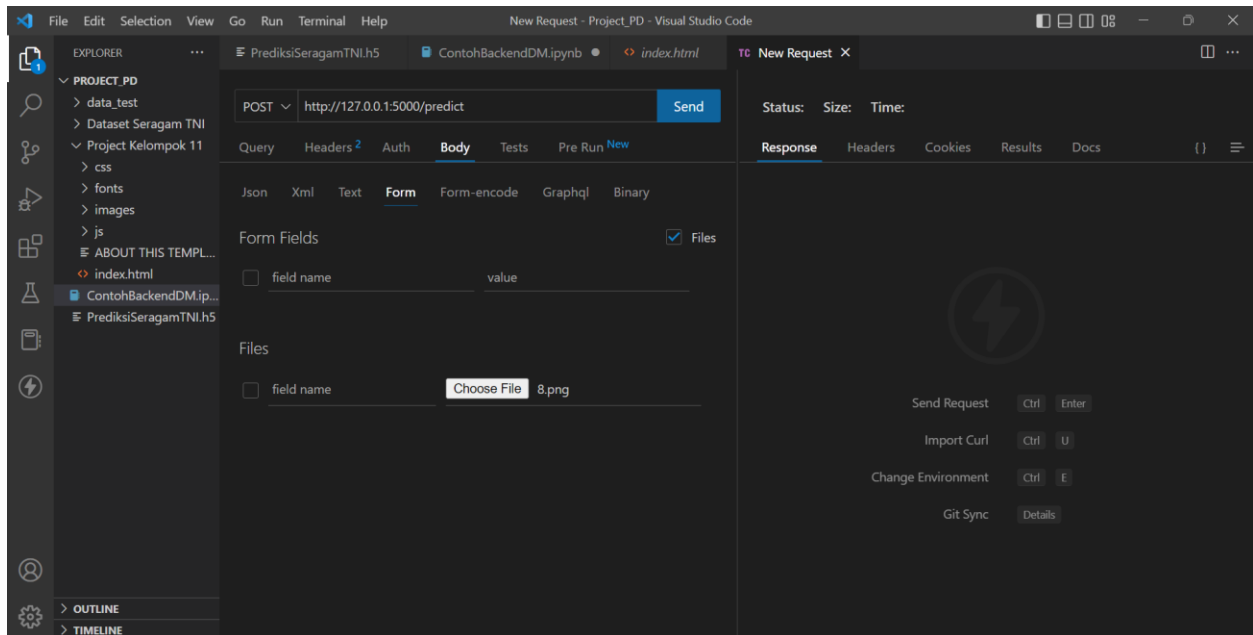
* Serving Flask app '__main__'
* Debug mode: off

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit

* Running on http://017a-113-212-122-130.ngrok.io
* Traffic stats available on http://127.0.0.1:4040
```

Analisa : Disini ada app route yang memiliki fungsi untuk menjadi pengatur dari setiap jalannya aplikasi.

8.)



Analisa : Merupakan penggabungan dari back end dan front end dengan menggunakan library thunder layent. Dan menggunakan url yang diambil dari front end sehingga perintah didalam back end ketika dijalankan yang mana perintah tersebut telah tersambung kedalam database yang menyimpan semua dari data data dari dataset yang telah disediakan, yang mana ketika front end dijalankan maka dengan menggunakan thuder layent dan data testing diinputkan kedalam filenya maka akan menampilkan output berdasarkan data yang telah terhubung di back end.