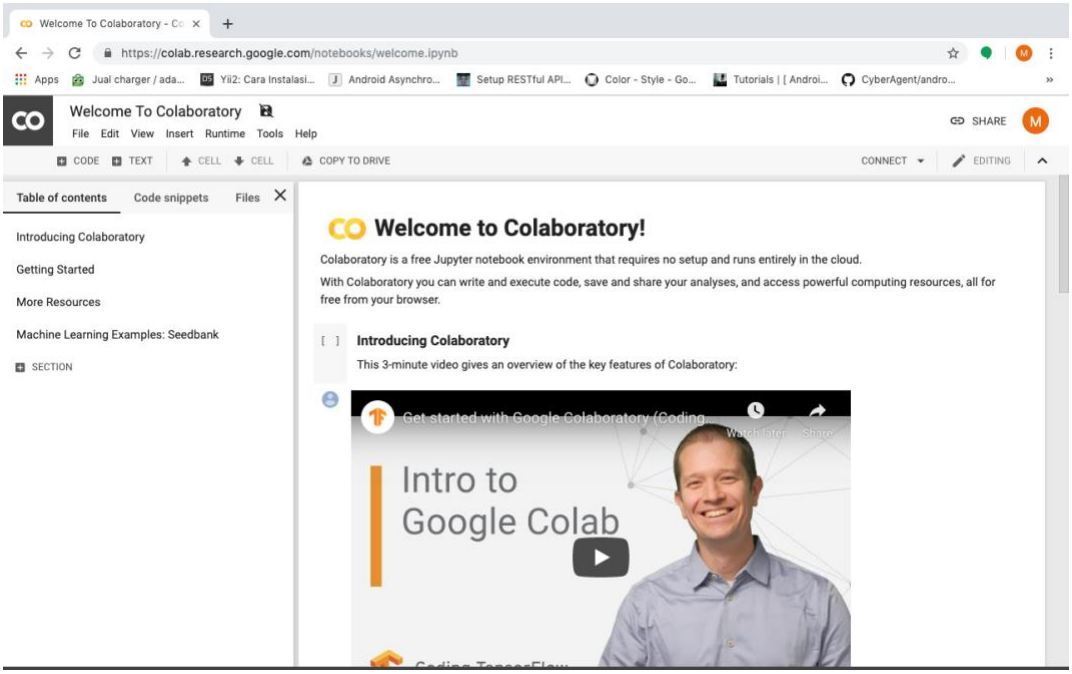
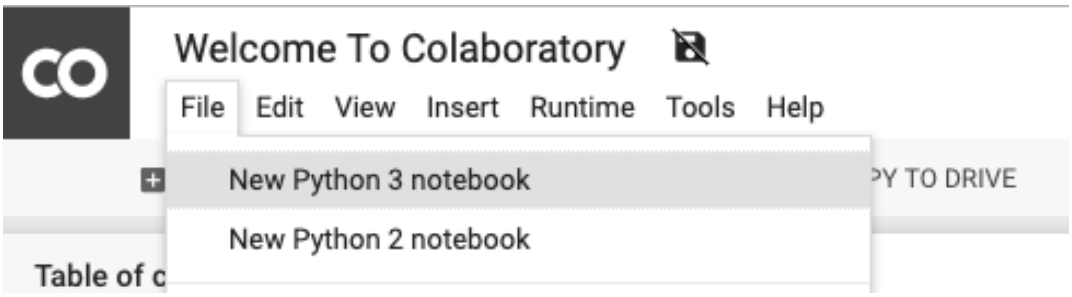
ARTIFICIAL INTELLIGENCE WORKSHOP

1. Membuat Project pada Google Colabs

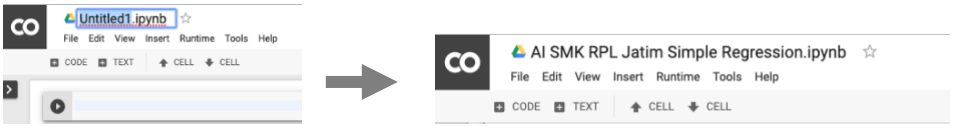
a. Buka browser dan ketikkan link berikut : https://colab.research.google.com/notebooks/welcome.ipynb



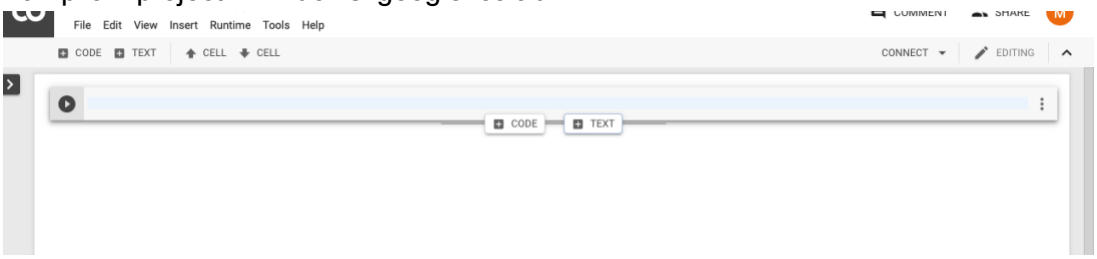
b. Buat sebuah project dengan klik menu **File -> New Phython 3 Notebook**



c. Ganti nama dokumen menjadi **AI SMK RPL Jatim Simple Regression** dengan double klik pada judul dokumen



d. Tampilan project windows google colab



e. Lokasi file project

Buka **Google Drive -> My Drive -> Colab Notebooks -> AI SMK RPL Jatim Simple Regression.ipyb**

2. Python Basic

a. Komentar

```
# Komentar dimulai dengan tanda '#'
# Python tidak membutuhkan tanda semi colon ";" di akhir setiap baris seperti R, Java, atau C
n = 5
b = 3
print('b={}, n={}'.format(b, n)) # new in Python 3
```

b. Identiting

```
# Python menggunakan "indenting" dan bukan tanda kurung seperti C, R, PHP, atau Java

if True:
    print('ini menggunakan tab')
    print('ini menggunakan spasi x4')
print('teks ini sudah diluar "IF"')
```

c. Teks

```
# Teks di python bisa menggunakan ' atau "

print(' "hi", text enclosed by \'  ')
print(" 'hi', text enclosed by \" ")

# tanda "\" digunakan untuk merubah character khusus di python menjadi character biasa
# Sangat berguna bagi Data Scientist saat menangani data dari database
```

d. Variabel

```
# Di Python variabel tidak perlu deklarasi (seperti matlab/Julia)
a = 2
b = 3.4
c = 'teks'
d = True
```

e. Type

```
# Untuk mengetahui tipe suatu variabel kita bisa menggunakan perintah "type"
type(a)
```

f. Array

```
# "Array" dasar di Python bisa berupa List, Tuple, Set atau Dictionary
# Mind with NameSpace
L = [1, 5, 7, 2] # ~vector in C
T = (1,4,2) #immutable
S = set(L) # no order, fast for checking if element 'exist'
D = {1:'satu', 2:'dua', 3:'tiga'}
D[1]
```

g. Function

```
def f(x, y=2):
    """
    Komentar >1 baris dilakukan seperti ini
    Sangat baik dilakukan untuk memberi keterangan suatu fungsi
    """
    return x**y
f(3)
```

h. Condition

```
x = 0
y = 5

if x < y:
  print('yes')
```

i. Loop

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

3. Simple Liniear Regression

Sebuah contoh dari membangun sebuah model liniear regression di google colabs untuk mempredeksi nilai prestasi siswa dari jarak rumah siswa

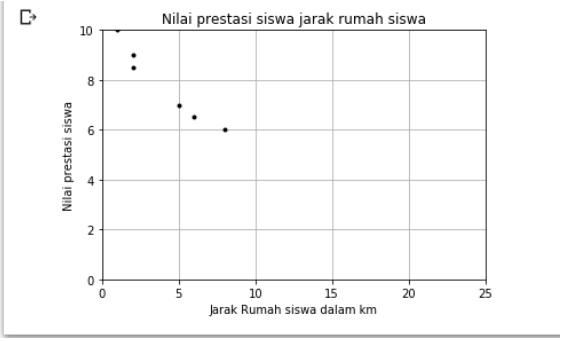
a. Mengimport sebuah library **matplotlib** untuk menampilkan grafik

```
import matplotlib.pyplot as plt
```

b. Membuat sebuah sample data terhadap variabel x dan variabel y beserta keterangan informasi dalam grafik

```
X = [[1], [2], [2], [5], [6], [8]]
y = [[10], [9], [8.5], [7], [6.5], [6]]
plt.figure()
plt.title('Nilai prestasi siswa jarak rumah siswa')
plt.xlabel('Jarak Rumah siswa dalam km')
plt.ylabel('Nilai prestasi siswa')
plt.plot(X, y, 'k.')
plt.axis([0, 25, 0, 10])
plt.grid(True)
plt.show()
```

Hasil run :



c. Membuat sebuah model linear regresi dari sample data yang telah dibuat sebelumnya

```
from sklearn.linear_model import LinearRegression

# Create and fit the model
model = LinearRegression()
model.fit(X, y)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

d. Melakukan prediksi nilai prestasi siswa dari sebuah nilai jarak rumah siswa (12 km)

```
print('Siswa dengan jarak rumah 12 Km dari sekolah diprediksi akan mempunyai nilai prestasi %.2f' % model.predict([[9]]))
```

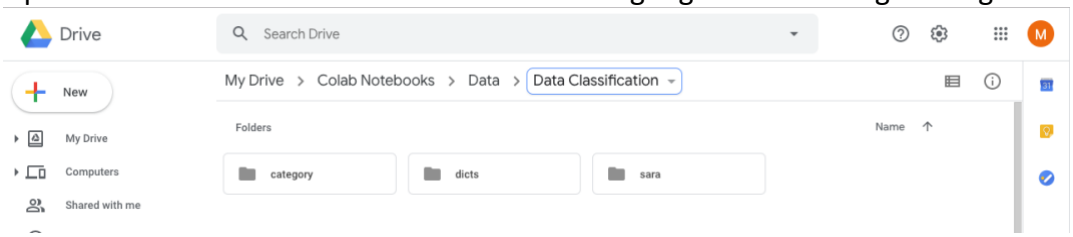
```
Siswa dengan jarak rumah 12 Km dari sekolah diprediksi akan mempunyai nilai prestasi 5.07
```

4. Classification dan Regression Logistic (Text)

a. Buat sebuah project baru dengan nama **AI SMK RPL Jatim Classification (Text)**

b. Download data untuk diolah di https://github.com/moryku/Artificial-Intelligence-Workshop

c. Upload folder Data Classification ke dalam akun google drive masing-masing



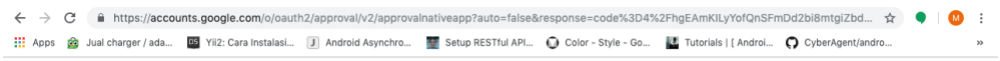
d. Kembali ke project google colabs dan tambahkan kode berikut untuk menghubungkan project colabs dengan akun google drive

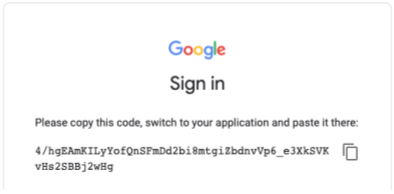
```
# Mount Google Drive
from google.colab import drive
drive.mount('/content/gdrive')
```

Lalu klik tombol run

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.goo

Enter your authorization code:
```






```
Enter your authorization code:
••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
```

e. Lalu arahkan list directory ke folder dict di dalam google drive dari cell colabs

```
import os
os.listdir(os.path.join('gdrive', 'My Drive', 'Colab Notebooks', 'Data', 'Data Classification', 'dicts'))
```

```
['stop_words.txt', 'id_full.txt']
```

f. Menyimpan file normal comment dan sara comment ke dalam variabel di colab cell

```
data_dir = os.path.join('gdrive', 'My Drive', 'Colab Notebooks', 'Data', 'Data Classification')
# Reading training texts
cmt_norm_file = os.path.join(data_dir, 'sara', 'normal_comments.txt')
cmt_sara_file = os.path.join(data_dir, 'sara', 'sara_comments.txt')
```

g. Karena isi dari file tersebut masih bersifat raw text lalu di ubah ke dalam bentuk List

```
# Read texts into list
def read_lines(filepath):
    with open(filepath) as fp:
        content = fp.readlines()
        content = [x.strip() for x in content]
    return content
```

```
cmt_norm = read_lines(cmt_norm_file)
cmt_sara = read_lines(cmt_sara_file)
print('data set size: ',len(cmt_norm), len(cmt_sara))
```

```
data set size:  67222 17724
```

h. Mengimport library pandas yang akan digunakan untuk membuat sebuah dataframe dari sebuah list dengan menambahkan sebuah label yang bernilai 1 untuk comment sara dan bernilai 0 untuk comment normal

```
import pandas as pd
# create a dataframe for all training texts, with their labels
def create_dataframe_with_label(cmt_norm, cmt_sara):
  cmt_all = cmt_norm + cmt_sara
  # make label
  label = []
  for _ in cmt_norm:
    label.append(0)
  for _ in cmt_sara:
    label.append(1)
  # create a pandas dataframe using texts and labels
  trainDF = pd.DataFrame()
  trainDF['text'] = cmt_all
  trainDF['label'] = label
  return trainDF
```

Lalu memanggil fungsi tersebut dan menampilkan hasil 20 data teratas

```
[ ]  data = create_dataframe_with_label(cmt_norm, cmt_sara)
     print(data.head(20))
```

```
                                                 text  label
0                                                           0
1   In sya Allah.....mereka berhasil ubtuk menguba...      0
2   Kasian,akibat terpojok,fitnah sana sini,salut ...      0
3                          Allahu Akbar.....hidup FPI      0
4                                               pamer      0
5   Mantap Tegakkan Hukum dgn seadil-adilnya Benar...      0
6                            AMIEN RAIS DALANGNYA...!      0
7   Silakan usut pak asal yg bener ngusutnya... Jg...      0
8   Heheehhee, rizieq pintar kali berkelit yaa..!!...      0
9   AHOK BATAL CAGUB???? DR SKG AJ DBUAT ATURAN SP...      0
10  Bapak  10  thn  nga  afa  yg  bapak  buat  din...      0
11  Wkwkwkwk nasier makin ngawur ya ... Percaya nu...      0
12  Baru terjadi di dunia penjahat jadi penjabat, ...      0
13                                                  0      0
14  Ini negara hukum, tangkap dan selesaikan sesua...      0
15  Prass @   Gua udah 2 tahun main babe   Insya A...      0
16                                        Aamin YRA      0
17        Perbuatanmu. ..kata-kata mu aja penuh dosa      0
18                         Klo milih ankmu yg salah pk      0
19                                 Khusus tukang ojek      0
```

Jika ingin memanggil 30 data secara random

```
print(data.sample(30))
```

```
                                                    text  label
34799  Ini cuma karena Ahok itu Non Muslim coba kalau...      0
32168  Betul pak SBY atas nasihat nya,maksih pak SBY ...      0
1437                  Klu bukan beking apa ya nama'y.....      0
8345                                 Pemimpin yang letoy      0
23798  Giliran massa ahok yg jauh dari tertib di biar...      0
55671  Semoga diberi umur pjg ngak mati2 supaya struk...      0
59758  lah buktinya ada aja yg mau kerja di situ, knp...      0
3302   Tenang Bae Pak Ahok Gusti Alloh Maha Segalanya...      0
8270                                     Yusdhy se77777.      0
73827  Matinya insya Allah msk surga dr pd lu di godo...      1
48085                            Ahok kapan di penjara?      0
43774  Kasian smp rengek2.. hhhh... PPP aj mrapat krn...      0
50889  Nawarin melulu kapan pelaksananya.\tNgak usah ...      0
```

i. Menyimpan data stop words dari google drive dimana data tersebut merupakan data yang tidak boleh ada di dalam dataframe teks comment maupun sara

```
[17] def load_stop_words():
       # Get the set of stopwords
       stop_words_f = os.path.join(data_dir, 'dicts', 'stop_words.txt')

       flines = read_lines(stop_words_f)
       return set([x.strip() for x in flines])
```

```
stop_words = load_stop_words()
print(len(stop_words))
```

```
7193
```

j. Membuat sebuah fungsi Cleaning data

```
#install beautifulSoup
# For cleaning
from bs4 import BeautifulSoup
import re
# text cleansing function
def raw_to_words(raw_text, stop_words=None):
    # 1. Remove HTML
    text_1 = BeautifulSoup(raw_text).get_text()

    # 2. Remove non-letters with regex
    letters_only = re.sub("[^a-zA-Z]", " ", text_1)

    # 3. Convert to lower case, split into individual words
    words = letters_only.lower().split()

    # 4. Remove stop words
    if stop_words:
        meaningful_words = [w for w in words if not w in stop_words]
    else:
        meaningful_words = words

    # 5. Join the words back into one string separated by space & return
    return(" ".join(meaningful_words))
```

k. Melihat salah satu teks hasil sebelum dan sesudah cleaning data

```
# check to see how the cleansing function works
print(cmt_norm[1])
print(raw_to_words(cmt_norm[1], stop_words=stop_words))
print(raw_to_words(cmt_norm[1], stop_words=stop_words))
```

```
In sya Allah.....mereka berhasil ubtuk mengubah kehidupan keluarganya. Aamiin yaa rabbal'aalamiin.
allah berhasil ubtuk mengubah kehidupan keluarganya aamiin
allah berhasil ubtuk mengubah kehidupan keluarganya aamiin
```

l. Melakukan cleaning data terhadap semua teks pada data frame

```
clean_data = data
clean_data['text'] = clean_data['text'].apply(raw_to_words, stop_words=stop_words)
clean_data = clean_data.loc[clean_data['text']!='']
print(clean_data.head())
```

```
/usr/local/lib/python3.6/dist-packages/bs4/__init__.py:273: UserWarning: "b'..'
  ' Beautiful Soup.' % markup)
                                                text  label
1  allah berhasil ubtuk mengubah kehidupan keluar...      0
2  kasian akibat terpojok fitnah salut tau dri ah...      0
3                             allahu akbar hidup fpi      0
4                                              pamer      0
5  mantap tegakkan hukum dgn seadil adilnya benar...      0
```

m. Mengimport library sklearn untuk pembuatan model klasifikasi

```
[ ]  from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
     from sklearn import model_selection, preprocessing, linear_model, naive_bayes, metrics, svm
     from sklearn import decomposition, ensemble
     from sklearn.model_selection import cross_val_score
```

n. Melakukan split data untuk data training dan data validasi

```
# split the dataset into training and validation datasets
train_x, valid_x, train_y, valid_y = model_selection.train_test_split(clean_data['text'], clean_data['label'])
print(len(train_x), len(valid_x))
```

```
62821 20941
```

o. Mengubah data teks ke dalam bentuk vector

```
# create a count vectorizer
count_vect = CountVectorizer(analyzer='word', token_pattern=r'\w{1,}')
count_vect.fit(clean_data['text'])
# transform the training and validation data using count vectorizer
xtrain_count = count_vect.transform(train_x)
xvalid_count = count_vect.transform(valid_x)
print(xtrain_count.shape, xvalid_count.shape)

# word level tf-idf
tfidf_vect = TfidfVectorizer(analyzer='word', token_pattern=r'\w{1,}')
tfidf_vect.fit(clean_data['text'])
xtrain_tfidf = tfidf_vect.transform(train_x)
xvalid_tfidf = tfidf_vect.transform(valid_x)
print(xtrain_tfidf.shape, xvalid_tfidf.shape)
```

```
(62821, 53647) (20941, 53647)
(62821, 53647) (20941, 53647)
```

p. Pembuatan model klasifikasi dengan logisticregression dari library sklearn model linear

```
#train model
logistic_r = linear_model.LogisticRegression()
logistic_r.fit(xtrain_tfidf, train_y)
```

q. Mendapatkan data akurasi setelah model diuji ke dalam data validasi yang sudah di split sebelumnya

```
# get the predictions
pred_valid = logistic_r.predict(xvalid_tfidf)
print("Validation set")
print("Accuracy:", metrics.accuracy_score(pred_valid, valid_y))
print("Report:\n", metrics.classification_report(valid_y, pred_valid))
```

```
Validation set
Accuracy: 0.8635690750202951
Report:
              precision    recall  f1-score   support

           0       0.86      0.98      0.92     16489
           1       0.87      0.42      0.57      4452

    accuracy                           0.86     20941
   macro avg       0.87      0.70      0.74     20941
weighted avg       0.86      0.86      0.84     20941
```

5. Classification Image

a. Buat sebuah project baru dengan nama **AI SMK RPL Jatim Klasifikasi (Image)**

b. Mengimport library yang diperlukan

```
[1]  from __future__ import print_function

     import numpy as np
     import matplotlib.pyplot as plt

     from sklearn.model_selection import train_test_split

     import tensorflow as tf
     from tensorflow import keras
```

c. Mendapatkan data dari dataset keras

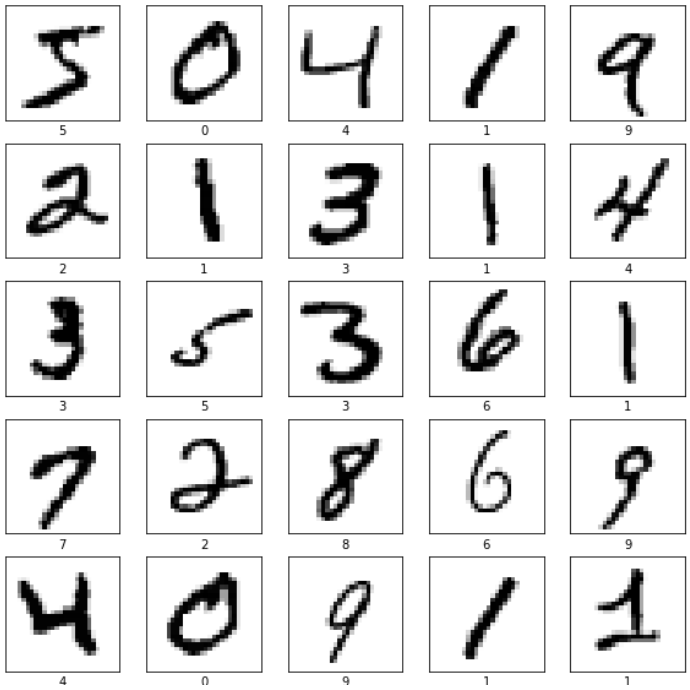
```
# GET the data
## Memuat Data Training dan Testing
mnist = keras.datasets.mnist
(train_data, train_labels), (test_data, test_labels) = mnist.load_data()
```

d. Eksplorasi dan melihat sample data

```
[3]  # EXPLORE the data
     ## Menampilkan 25 Gambar Pertama dari Data Training
     plt.figure(figsize=(10,10))
     for i in range(25):
         plt.subplot(5,5,i+1)
         plt.xticks([])
         plt.yticks([])
         plt.grid(False)
         plt.imshow(train_data[i], cmap=plt.cm.binary)
         plt.xlabel(train_labels[i])
     plt.show()
```

Hasil:

[3]



e. Menampilkan ukuran data training dan testing

```
[4]  ## Menampilkan Ukuran Data Training
     train_data.shape
```

```
(60000, 28, 28)
```

```
[5]  ## Menampilkan Ukuran Data Testing
     test_data.shape
```

```
(10000, 28, 28)
```

f. Membuat sebuah model dengan algoritma neural network + activation function

```
[ ]  ## Pemilihan Model (Arsitektur dan Parameter)
     units=128
     learning_rate=1e-3

     model = keras.Sequential([
         keras.layers.Flatten(input_shape=(28, 28)),
         keras.layers.Dense(units, activation=tf.nn.relu),
         keras.layers.Dense(10, activation=tf.nn.softmax)
     ])

     model.compile(optimizer=tf.keras.optimizers.Adam(lr=learning_rate),
                   loss='sparse_categorical_crossentropy',
                   metrics=['accuracy'])

     fit_data = train_data[:48001]
     val_data = train_data[48001:]
     fit_labels = train_labels[:48001]
     val_labels = train_labels[48001:]
     callbacks = [keras.callbacks.EarlyStopping(monitor='val_acc', patience=2)]
     model.fit(fit_data, fit_labels, batch_size=32, epochs=40,
               validation_data=(val_data, val_labels), callbacks=callbacks)
```

hasil :

```
WARNING: Logging before flag parsing goes to stderr.
W0714 05:25:55.316253 139967407486848 deprecation.py:506] From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/init_ops.py
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
Train on 48001 samples, validate on 11999 samples
Epoch 1/40
48001/48001 [==============================] - 4s 81us/sample - loss: 0.2831 - acc: 0.9196 - val_loss: 0.1548 - val_acc: 0.9552
Epoch 2/40
48001/48001 [==============================] - 4s 73us/sample - loss: 0.1268 - acc: 0.9629 - val_loss: 0.1227 - val_acc: 0.9637
Epoch 3/40
48001/48001 [==============================] - 3s 71us/sample - loss: 0.0881 - acc: 0.9739 - val_loss: 0.1068 - val_acc: 0.9686
Epoch 4/40
48001/48001 [==============================] - 3s 68us/sample - loss: 0.0647 - acc: 0.9810 - val_loss: 0.0975 - val_acc: 0.9696
Epoch 5/40
48001/48001 [==============================] - 3s 68us/sample - loss: 0.0503 - acc: 0.9842 - val_loss: 0.0841 - val_acc: 0.9741
Epoch 6/40
48001/48001 [==============================] - 3s 69us/sample - loss: 0.0399 - acc: 0.9875 - val_loss: 0.0928 - val_acc: 0.9720
Epoch 7/40
48001/48001 [==============================] - 3s 69us/sample - loss: 0.0317 - acc: 0.9908 - val_loss: 0.0848 - val_acc: 0.9752
Epoch 8/40
48001/48001 [==============================] - 3s 70us/sample - loss: 0.0235 - acc: 0.9928 - val_loss: 0.1006 - val_acc: 0.9732
Epoch 9/40
48001/48001 [==============================] - 4s 73us/sample - loss: 0.0205 - acc: 0.9936 - val_loss: 0.0885 - val_acc: 0.9763
Epoch 10/40
48001/48001 [==============================] - 4s 73us/sample - loss: 0.0166 - acc: 0.9947 - val_loss: 0.0933 - val_acc: 0.9747
Epoch 11/40
48001/48001 [==============================] - 4s 75us/sample - loss: 0.0139 - acc: 0.9960 - val_loss: 0.1104 - val_acc: 0.9718
<tensorflow.python.keras.callbacks.History at 0x7f4c6e30eef0>
```

g. Evaluasi Model terhadapt data test

```
[ ]  ## Evaluasi Model
     test_loss, test_acc = model.evaluate(test_data, test_labels)
     print('Test accuracy:', test_acc)
```

Hasil :

```
10000/10000 [==============================] - 0s 26us/sample - loss: 0.1079 - acc: 0.9734
Test accuracy: 0.9734
```

h. Import and mount google drive

```
## Penyimpanan Model
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Drive alre  Run cell (⌘/Ctrl+Enter)
            cell executed since last change
                                    ve; to attempt to forcibly remount, call drive.mou
```

i. Save model to google drive

```
model.save('/content/gdrive/My Drive/Colab Notebooks/Data/Result/klasifikasi_image_hand_nn.h5')
```

j. Load model from google drive

```
# COMMUNICATE the result
model = keras.models.load_model('/content/gdrive/My Drive/Colab Notebooks/Data/Result/klasifikasi_image_hand_nn.h5')
```

k. Melakukan prediksi dari sebuah gambar tulisan tangan ke model yang telah dibuat, namun sebelumnya import library terkait dengan proses gambar dan data array/matriks

```
from PIL import Image
import numpy
```

l. Mengimpor gambar tulisan tangan anda yang sebelumnya telah di upload di google drive

```
[107] im = Image.open("/content/gdrive/My Drive/Colab Notebooks/Data/HandWriteImage/6.png")
```

m. Meresize ukuran pixel gambar menjadi 28 x 28

```
temp=numpy.array(resized_image)
print(temp.shape)
```

```
(28, 28, 4)
```

n. Mengubah raw gambar ke dalam bentuk array menggunakan library **numpy**

```
[114] temp=numpy.array(resized_image)
```

o. Melakukan transformasi ke bentu 2 dimensi array

```
[111] x=temp.shape[0]
      y=temp.shape[1]

      temp.resize((x,y)) # a 2D array
      print(temp.shape)
```

```
(28, 28)
```

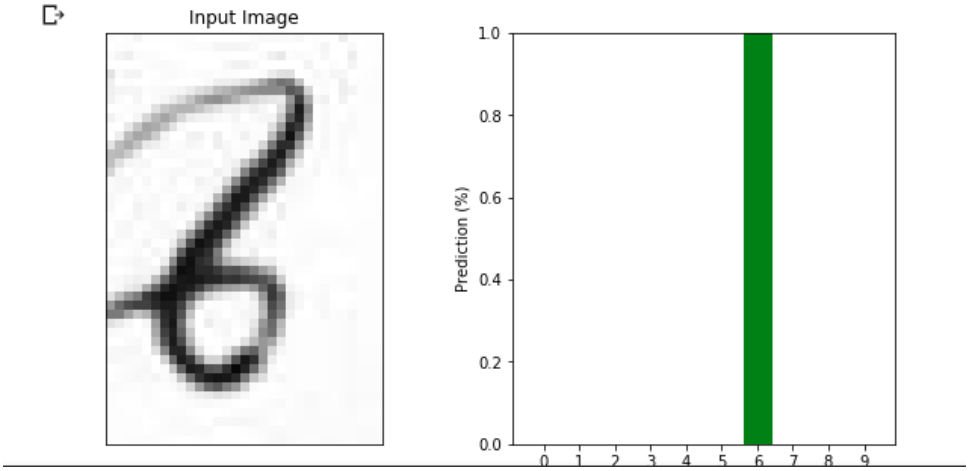
p. Melakukan prediksi terhadap array yang telah dibuat dari sebuah model yang telah dibuat sebelumnya

```
temp = temp / 255.0
predictions = model.predict(np.expand_dims(temp,0))
print(np.argmax(predictions))
```

```
6
```

q. Menampilkan dalam bentuk grafik

```
[113] ## Visualisasi Hasil
      plt.figure(figsize=(10,5))
      plt.subplot(1,2,1)
      plt.xticks([])
      plt.yticks([])
      plt.title('Input Image')
      plt.imshow(im)
      plt.subplot(1,2,2)
      plt.xticks(range(10), ['0','1','2','3','4','5','6','7','8','9'])
      plt.ylim([0, 1])
      plt.xlabel('Class Image')
      plt.ylabel('Prediction (%)')
      thisplot = plt.bar(range(10), predictions[0], color="red")
      thisplot[np.argmax(predictions)].set_color('green')
```



Sumber

- Python basic: https://www.python-course.eu/python3_history_and_philosophy.php
- Data Science Basic: https://scikit-learn.org/stable/tutorial/index.html
- Advance Python: http://andy.terrel.us/blog/2012/09/27/starting-with-python/
- Visualisasi di Python: https://matplotlib.org/gallery.html
- AI Academy from BABE (Pemkot Malang)