



1. Kerja lab di pandu oleh asisten dosen (asdos) dalam bentuk coding implementasi dari salah satu teknik learning.
2. Peserta kerja lab diminta untuk mengikuti dengan seksama rangkaian penjelasan yang disampaikan oleh asdos, kemudian mengumpulkan keseluruhan jawaban pada lembar ini (termasuk file keseluruhan code dari google colab) melalui akun LMS masing-masing pada waktu yang telah ditetapkan.
3. Pastikan gambar screenshot output maupun isian baris coding yang dituliskan dapat dipahami dengan jelas, dan merupakan hasil karya sendiri, dengan menyertakan link google colab yang dapat di akses di akhir jawaban.
4. Peserta ujian remedial menyimpan lembar jawaban ujian ini dalam format .pdf dengan penamaan file yaitu <NIM> <Nama> Kerja Lab

NIM: 1304211013	Nama Mahasiswa: Syahratul Muthi'ah M. Masiming	Kelas: IF-45- GAB.1PJJ	Ruang:	Nilai: <optional, bisa diisi jika tidak menerapkan CLO>
---	--	--	------------------------	---

Capaian Pembelajaran Mata Kuliah (<i>Course Learning Outcome</i>) yang akan dicapai		Total Nilai Kerja Lab (maksimal 6)
CLO 3	Mahasiswa mampu mengimplementasikan, dan mendesain teknik Learning.	

```
import pandas as pd
import numpy as np
```

Load Data

```
# Load Iris Dataset
dwn_url='https://drive.google.com/uc?id=' + '1C5-s_COUWjrjn52wzs-QtEYc6QE_I1R20'
df = pd.read_csv(dwn_url)
```

Data Explorations

```
# Membaca beberapa baris pertama dari data
df.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
# Membaca beberapa baris terakhir dari data
df.tail()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

```
# Menampilkan unique value pada kolom Species
# (bisa kita lihat bahwa kolom Species menjadi class yang memiliki 3 kelas)
df.Species.unique()
```

```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

```
# Menampilkan metadata dari data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype  
---  --
 0   Id                  150 non-null   int64  
 1   SepalLengthCm       150 non-null   float64
 2   SepalWidthCm        150 non-null   float64
 3   PetalLengthCm       150 non-null   float64
 4   PetalWidthCm        150 non-null   float64
 5   Species             150 non-null   object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

```
[51] # Menampilkan summary statistik dari data
df.shape
```

```
(150, 6)
```

```
[50] # Menampilkan jumlah baris dan kolom pada dataframe
df.shape
```

```
(150, 6)
```

Preprocessing

```
# Menghapus kolom 'Id' dari dataframe
df = df.drop('Id', axis=1)
```

```
# Shuffle Data
df = df.sample(150).reset_index(drop=True)
df
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	7.9	3.8	6.4	2.0	Iris-virginica
1	6.9	3.2	5.7	2.3	Iris-virginica
2	4.8	3.1	1.6	0.2	Iris-setosa
3	4.7	3.2	1.6	0.2	Iris-setosa
4	5.0	3.5	1.3	0.3	Iris-setosa
...
145	5.2	3.4	1.4	0.2	Iris-setosa
146	6.8	2.8	4.8	1.4	Iris-versicolor
147	5.1	2.5	3.0	1.1	Iris-versicolor
148	5.6	3.0	4.1	1.3	Iris-versicolor
149	5.8	2.6	4.0	1.2	Iris-versicolor

150 rows x 5 columns

```
# Melakukan folding pada data
#data test #data train
fold1 = (df.iloc[0:50].reset_index(drop=True), df.iloc[50:150].reset_index(drop=True))
fold2 = (df.iloc[50:100].reset_index(drop=True), pd.concat([df.iloc[0:50], df.iloc[100:150]]).reset_index(drop=True))
fold3 = (df.iloc[100:150].reset_index(drop=True), df.iloc[0:100].reset_index(drop=True))
```

```
# Menampilkan data fold2 untuk training sebanyak 100 baris
test, train = fold2
print(train)
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	7.9	3.8	6.4	2.0	Iris-virginica
1	6.9	3.2	5.7	2.3	Iris-virginica
2	4.8	3.1	1.6	0.2	Iris-setosa
3	4.7	3.2	1.6	0.2	Iris-setosa
4	5.0	3.5	1.3	0.3	Iris-setosa
...
95	5.2	3.4	1.4	0.2	Iris-setosa
96	6.8	2.8	4.8	1.4	Iris-versicolor
97	5.1	2.5	3.0	1.1	Iris-versicolor
98	5.6	3.0	4.1	1.3	Iris-versicolor
99	5.8	2.6	4.0	1.2	Iris-versicolor

[100 rows x 5 columns]

```
# Normalizations
# Melakukan normalisasi data dengan min-max normalization
def norm(df):
    df = (df - df.min()) / (df.max() - df.min())
    return df
```

```
# Assign fitur dan kelas
# Menghapus kolom species(karena akan kita gunakan sebagai class)
X = df.drop('Species', axis=1)
# Species sebagai kelas
Y = df.Species
```

```
# Melakukan normalisasi untuk data fitur
X = norm(X)
```

```
X
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	1.000000	0.750000	0.915254	0.791667
1	0.722222	0.500000	0.796610	0.916667
2	0.138889	0.458333	0.101695	0.041667
3	0.111111	0.500000	0.101695	0.041667
4	0.194444	0.625000	0.050847	0.083333
...
145	0.250000	0.583333	0.067797	0.041667
146	0.694444	0.333333	0.644068	0.541667
147	0.222222	0.208333	0.338983	0.416667
148	0.361111	0.416667	0.525424	0.500000
149	0.416667	0.250000	0.508475	0.458333

150 rows x 4 columns

KNN Model

```
[59] # Menghitung jarak menggunakan euclidean distance
def euclidean(x1,x2):
    return np.sqrt(np.sum((x1-x2)**2))
```

```
# Euclidean distance dari data baris pertama dan kedua
euclidean(X.iloc[0],X.iloc[1])

0.41153603566454183
```

```
# Training KNN
def knn(X_train, y_train, X_test, k): # k sebagai banyaknya neighbors yang ditentukan
    dist = []
    # Menghitung distance dari data training dan data testing
    for row in range(X_train.shape[0]):
        dist.append(euclidean(X_train.iloc[row],X_test))

    # Menambahkan data distance pada data
    data = X_train.copy()
    data['Dist'] = dist
    # Menambahkan class pada data
    data['Class'] = y_train
    # Mengurutkan data berdasarkan distance
    data = data.sort_values(by= 'Dist').reset_index(drop=True)
    y_pred = data.iloc[:k].Class.mode()
    # Mengambil label kelas yang paling sering muncul diantara k-NN
    return y_pred[0]
```

Evaluation

```
# Menghitung akurasi dari output berdasarkan label kelas
def acc(y_pred, y_true):
    true = 0
    for i in range(len(y_pred)):
        if y_pred[i] == y_true[i]:
            true+=1
    return true/len(y_pred)
```

```
# Evaluasi model dengan menggunakan data fold
def evaluate(fold, k):
    test, train = fold
    X_train, Y_train = train.drop('Species',axis=1), train.Species
    X_test, Y_test = test.drop('Species', axis=1), test.Species
    X_train = norm(X_train)
    X_test = norm(X_test)
    y_preds = []
    for row in range(X_test.shape[0]):
        y_preds.append(knn(X_train,Y_train,X_test.iloc[row],k))
    return (acc(y_preds, Y_test))
```

```
#Main Program
k = 5
accs = []
folds = [fold1, fold2, fold3]
for i in range(len(folds)):
    accs.append(evaluate(folds[i], k))
print(f'Menggunakan k : {k}, dengan rata-rata akurasi : {sum(accs)/3}')

Menggunakan k : 5, dengan rata-rata akurasi : 0.32
```

Link google colab jawaban masing-masing:

https://colab.research.google.com/drive/1X0FVP8JK3_aHBnPCChDA9TYFaBx4ZLLfk?usp=sharing