Lab 5 KNN

No.	Program

1	`# import tools`
	`import numpy as np`
	`import pandas as pd`
	`import matplotlib.pyplot as plt`
	`import seaborn as sns`

2	`# import dataset`
	`diabetes_data = pd.read_csv('diabetes.csv')`
	`diabetes_data.head()`

3	`# deskripsi data` `diabetes_data.describe().T`

4	`# replace zeros dengan nan supaya cleaning lebih mudah`
	`diabetes_data_copy = diabetes_data.copy()`
	`diabetes_data_copy[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']] =`
	`diabetes_data_copy[['Glucose','BloodPressure','SkinThickness','Insulin',`
	`'BMI']].replace(0,np.NaN)`

5	`## tampilkan NaN` `print(diabetes_data_copy.isnull().sum())`
6	`# cek korelasi` `p=sns.pairplot(diabetes_data_copy, hue = 'Outcome')`
7	`plt.figure(figsize=(12,10))` `p=sns.heatmap(diabetes_data_copy.corr(), annot=True,cmap ='RdYlGn')`

8	`# optimasi data menggunakan scaler`
	`from sklearn.preprocessing import StandardScaler`
	`sc_X = StandardScaler()`
	`X =  pd.DataFrame(sc_X.fit_transform(diabetes_data_copy.drop(["Outcome"],axis = 1),),`
	`columns=['Pregnancies', 'Glucose', 'BloodPressure','SkinThickness','Insulin',`
	`'BMI', 'DiabetesPedigreeFunction', 'Age'])`

9	`# cek data` `X.head()`
10	`# buat variabel target(y)` `y = diabetes_data_copy.Outcome`

11	`#import train_test_split dan split data`
	`from sklearn.model_selection import train_test_split`
	`X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=1/3,random_state=42,`
	` stratify=y)`

12	`# membuat model untuk mencari jumlah K`
	`from sklearn.neighbors import KNeighborsClassifier`
	
	`test_scores = []`
	`train_scores = []`
	
	`for i in range(1,15):`
	
	`    knn = KNeighborsClassifier(i)`
	`    knn.fit(X_train,y_train)`
	
	`    train_scores.append(knn.score(X_train,y_train))`
	`    test_scores.append(knn.score(X_test,y_test))`

13	`#skor yang diperoleh dari data yang sama untuk training dan testing`
	`max_train_score = max(train_scores)`
	`train_scores_ind = [i for i, v in enumerate(train_scores) if v == max_train_score]`
	`print('Max train score {} % and k = {}'.format(max_train_score*100,`
	`    list(map(lambda x: x+1, train_scores_ind))))`

14	`#skor yang diperoleh dari pengetesan menggunakan data testing`
	`max_test_score = max(test_scores)`
	`test_scores_ind = [i for i, v in enumerate(test_scores) if v == max_test_score]`
	`print('Max test score {} % and k = {}'.format(max_test_score*100,`
	`    list(map(lambda x: x+1, test_scores_ind))))`

15	`# visualisasi`
	`plt.figure(figsize=(12,5))`
	`p = sns.lineplot(range(1,15),train_scores,marker='*',label='Train Score')`
	`p = sns.lineplot(range(1,15),test_scores,marker='o',label='Test Score')`

16	`#membuat model KNN dengan K yang sudah diperoleh`
	`knn = KNeighborsClassifier(7)`
	
	`knn.fit(X_train,y_train)`
	`knn.score(X_test,y_test)`

17	`#import confusion_matrix`
	`from sklearn.metrics import confusion_matrix`
	`y_pred = knn.predict(X_test)`

```
confusion_matrix(y_test,y_pred)
```