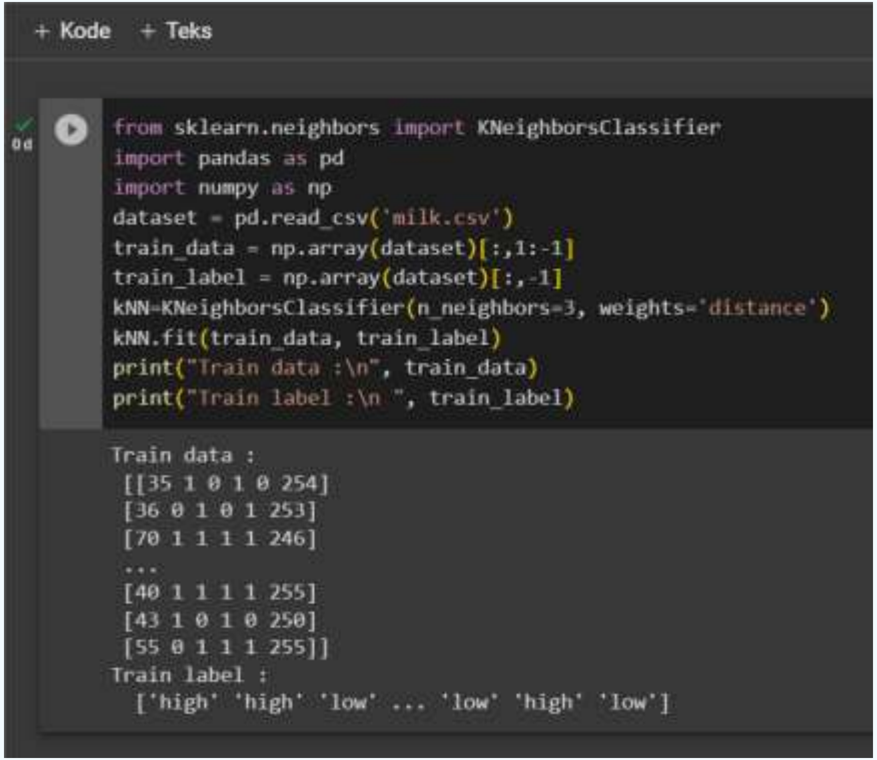
k – Nearest Neightbors

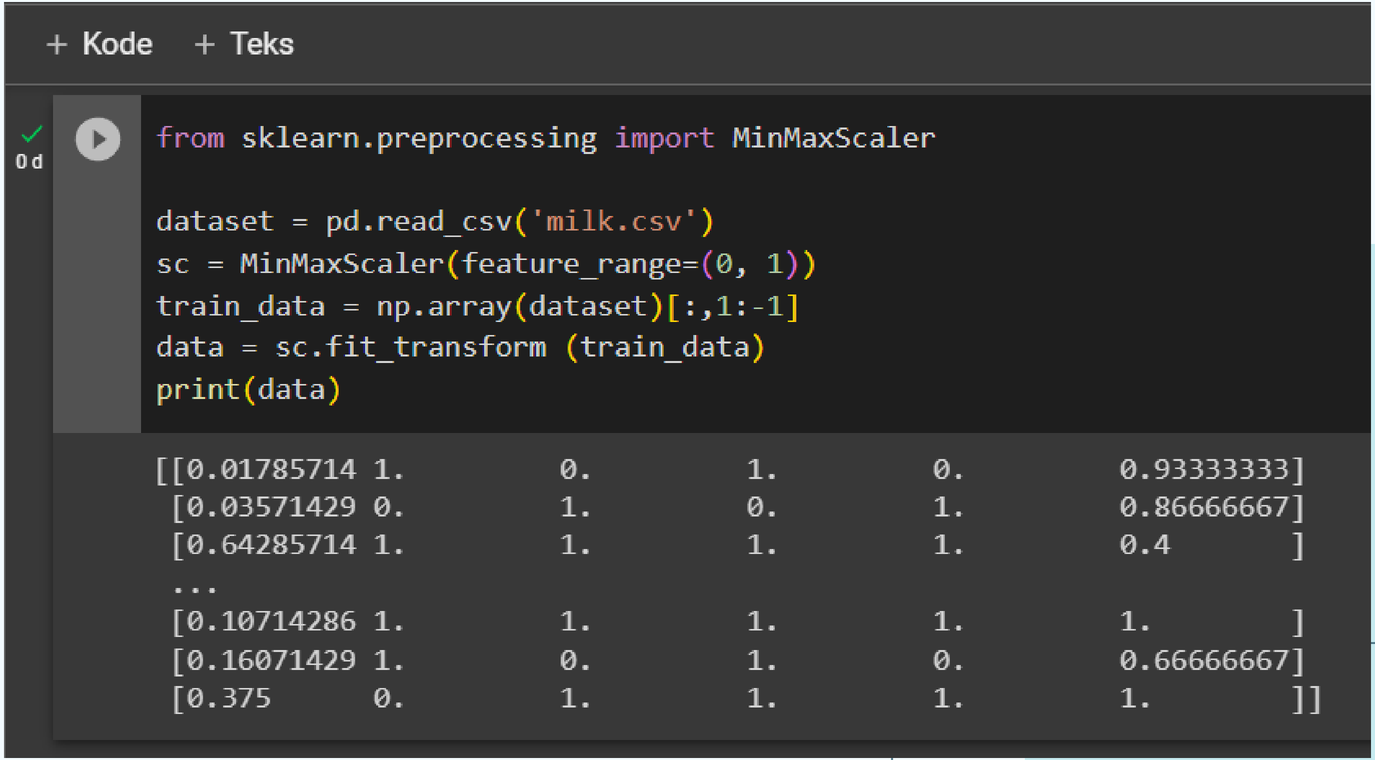
1. Dataset ← milk.csv, ambil train_data & train_label, tampilkan



```
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import numpy as np
dataset = pd.read_csv('milk.csv')
train_data = np.array(dataset)[:,1:-1]
train_label = np.array(dataset)[:,-1]
kNN=KNeighborsClassifier(n_neighbors=3, weights='distance')
kNN.fit(train_data, train_label)
print("Train data :\n", train_data)
print("Train label :\n ", train_label)
```

```
Train data :
 [[35 1 0 1 0 254]
 [36 0 1 0 1 253]
 [70 1 1 1 1 246]
 ...
 [40 1 1 1 1 255]
 [43 1 0 1 0 250]
 [55 0 1 1 1 255]]
Train label :
  ['high' 'high' 'low' ... 'low' 'high' 'low']
```

2. Lakukan normalisasi terhadap train_label dengan metode min-max(0-1)

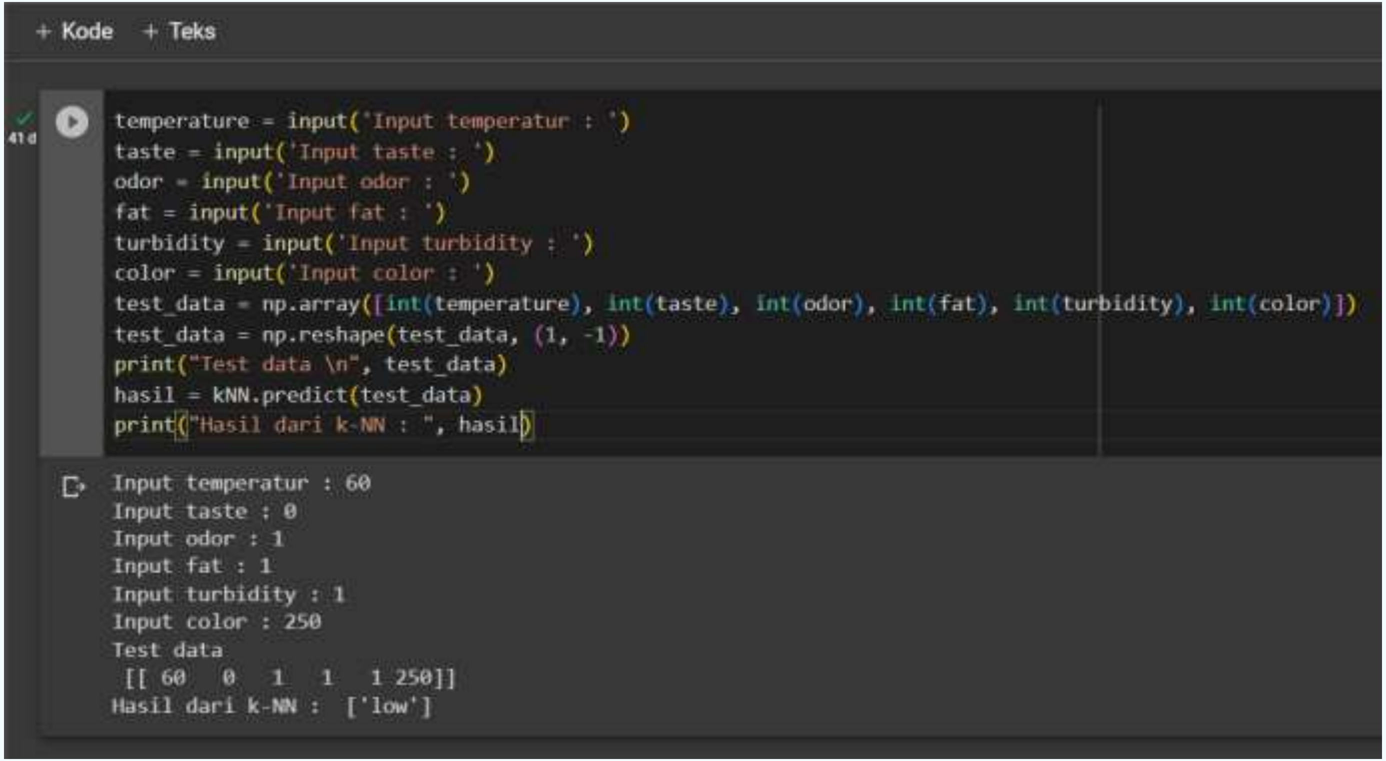


```
from sklearn.preprocessing import MinMaxScaler

dataset = pd.read_csv('milk.csv')
sc = MinMaxScaler(feature_range=(0, 1))
train_data = np.array(dataset)[:,1:-1]
data = sc.fit_transform (train_data)
print(data)
```

```
[[0.01785714 1.         0.         1.         0.         0.93333333]
 [0.03571429 0.         1.         0.         1.         0.86666667]
 [0.64285714 1.         1.         1.         1.         0.4       ]
 ...
 [0.10714286 1.         1.         1.         1.         1.        ]
 [0.16071429 1.         0.         1.         0.         0.66666667]
 [0.375      0.         1.         1.         1.         1.        ]]
```

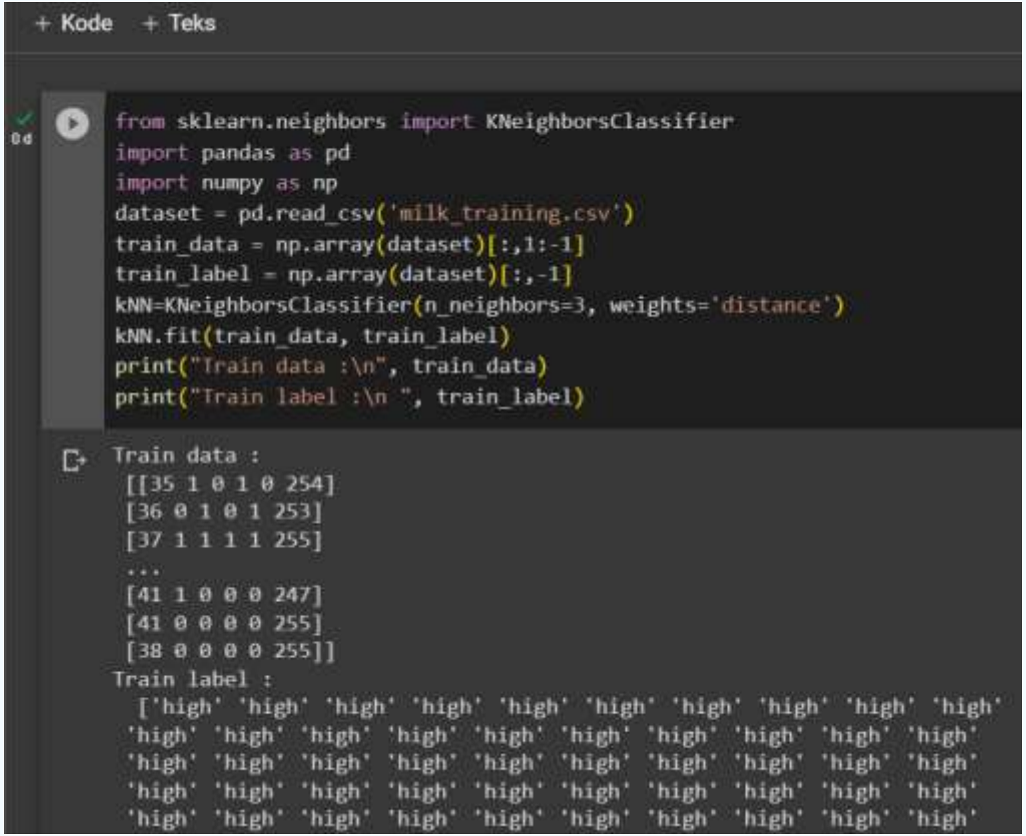
3. Lakukan klasifikasi menggunakan k-NN untuk 1 input data test



```
temperature = input('Input temperatur : ')
taste = input('Input taste : ')
odor = input('Input odor : ')
fat = input('Input fat : ')
turbidity = input('Input turbidity : ')
color = input('Input color : ')
test_data = np.array([int(temperature), int(taste), int(odor), int(fat), int(turbidity), int(color)])
test_data = np.reshape(test_data, (1, -1))
print("Test data \n", test_data)
hasil = kNN.predict(test_data)
print("Hasil dari k-NN : ", hasil)
```

```
Input temperatur : 60
Input taste : 0
Input odor : 1
Input fat : 1
Input turbidity : 1
Input color : 250
Test data
 [[ 60   0   1   1   1 250]]
Hasil dari k-NN :  ['low']
```

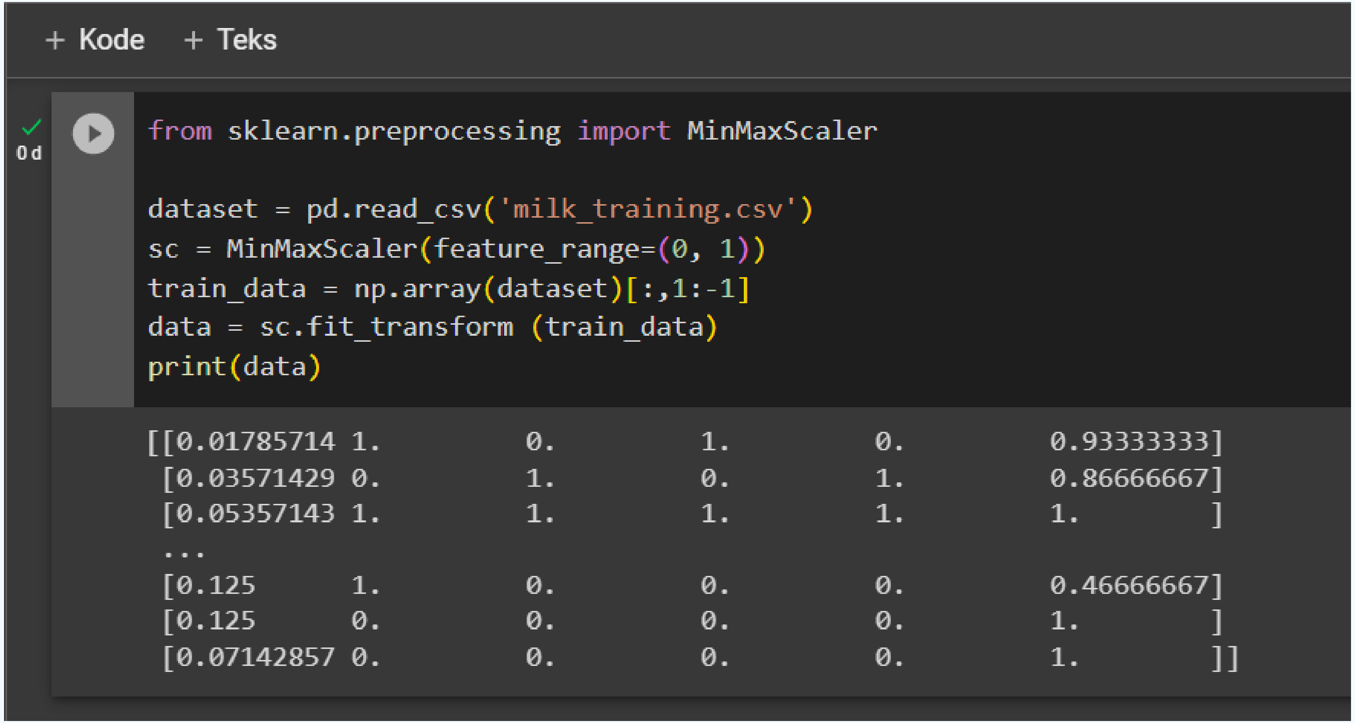
4. dataset ← milk_training.csv, ambil train_data & train_label, tampilkan



```python
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import numpy as np
dataset = pd.read_csv('milk_training.csv')
train_data = np.array(dataset)[:,1:-1]
train_label = np.array(dataset)[:,-1]
kNN=KNeighborsClassifier(n_neighbors=3, weights='distance')
kNN.fit(train_data, train_label)
print("Train data :\n", train_data)
print("Train label :\n ", train_label)
```

```
Train data :
 [[35 1 0 1 0 254]
 [36 0 1 0 1 253]
 [37 1 1 1 1 255]
 ...
 [41 1 0 0 0 247]
 [41 0 0 0 0 255]
 [38 0 0 0 0 255]]
Train label :
  ['high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high'
 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high'
 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high'
 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high'
 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high'
```

5. Lakukan normalisasi terhadap train_data dengan metode min – max(0-1)

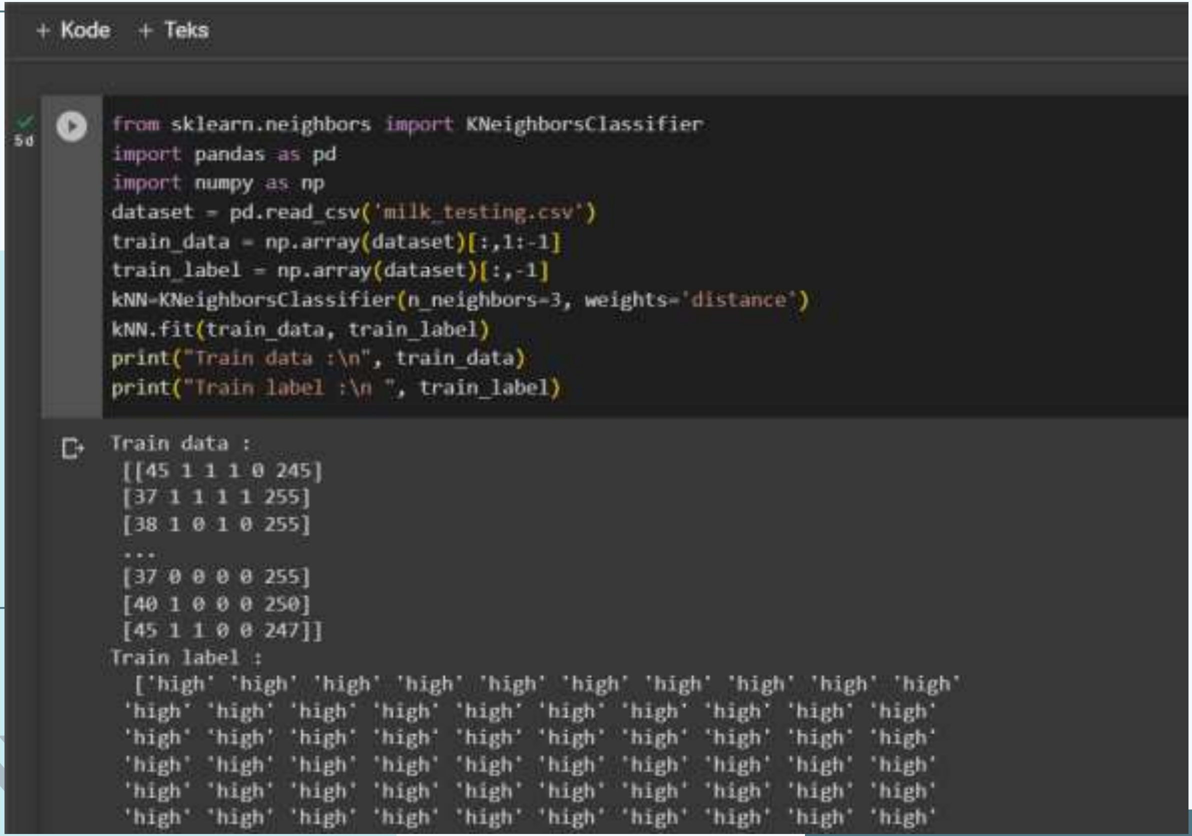


```
from sklearn.preprocessing import MinMaxScaler

dataset = pd.read_csv('milk_training.csv')
sc = MinMaxScaler(feature_range=(0, 1))
train_data = np.array(dataset)[:,1:-1]
data = sc.fit_transform (train_data)
print(data)
```

```
[[0.01785714 1.         0.         1.         0.         0.93333333]
 [0.03571429 0.         1.         0.         1.         0.86666667]
 [0.05357143 1.         1.         1.         1.         1.        ]
 ...
 [0.125      1.         0.         0.         0.         0.46666667]
 [0.125      0.         0.         0.         0.         1.        ]
 [0.07142857 0.         0.         0.         0.         1.        ]]
```

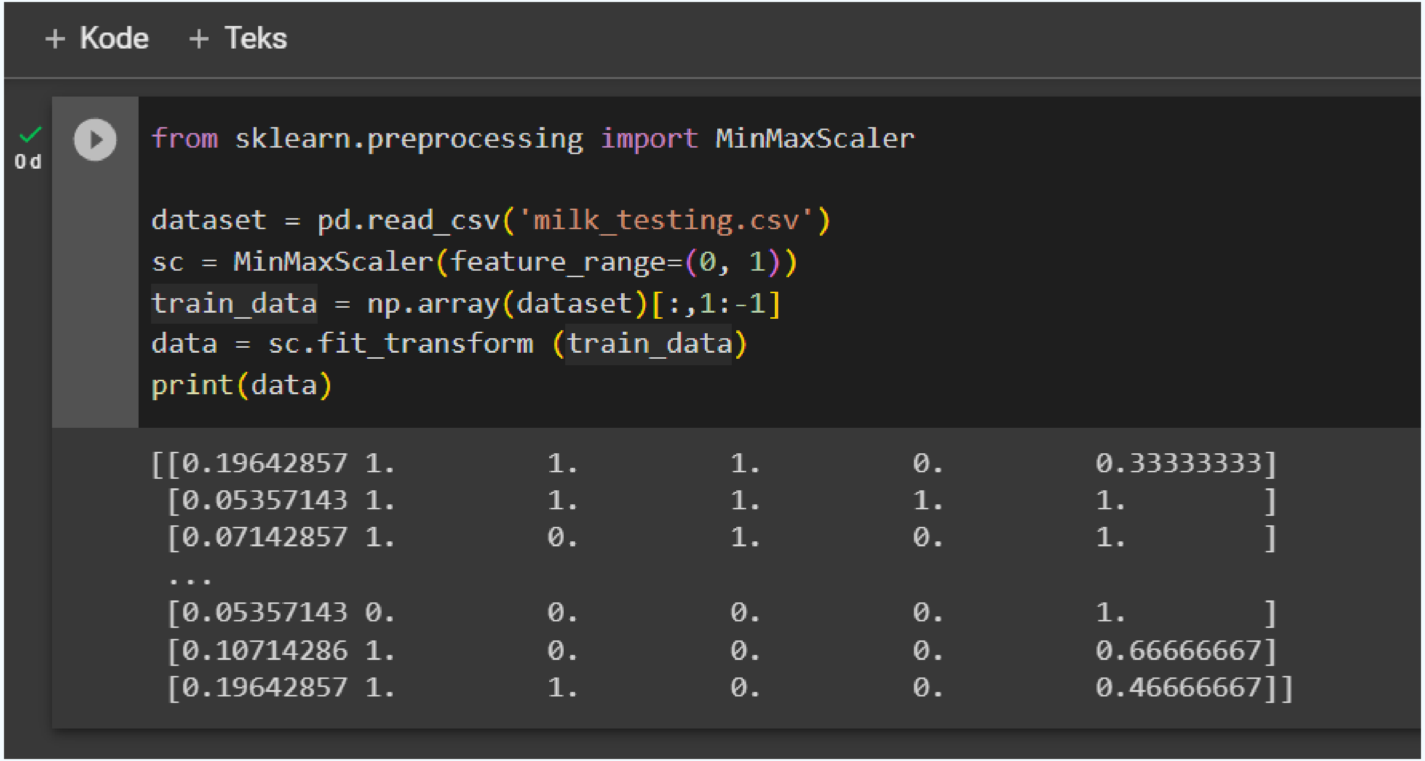
6. test_dataset ← milk_testing.csv, ambil test_data & test_label, tampilkan



```
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import numpy as np
dataset = pd.read_csv('milk_testing.csv')
train_data = np.array(dataset)[:,1:-1]
train_label = np.array(dataset)[:,-1]
kNN=KNeighborsClassifier(n_neighbors=3, weights='distance')
kNN.fit(train_data, train_label)
print("Train data :\n", train_data)
print("Train label :\n ", train_label)
```

```
Train data :
 [[45 1 1 1 0 245]
 [37 1 1 1 1 255]
 [38 1 0 1 0 255]
 ...
 [37 0 0 0 0 255]
 [40 1 0 0 0 250]
 [45 1 1 0 0 247]]
Train label :
  ['high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high'
 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high'
 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high'
 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high'
 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high'
 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high' 'high'
```

7. Lakukan normalisasi terhadap test_data dengan metode min – max (0-1)

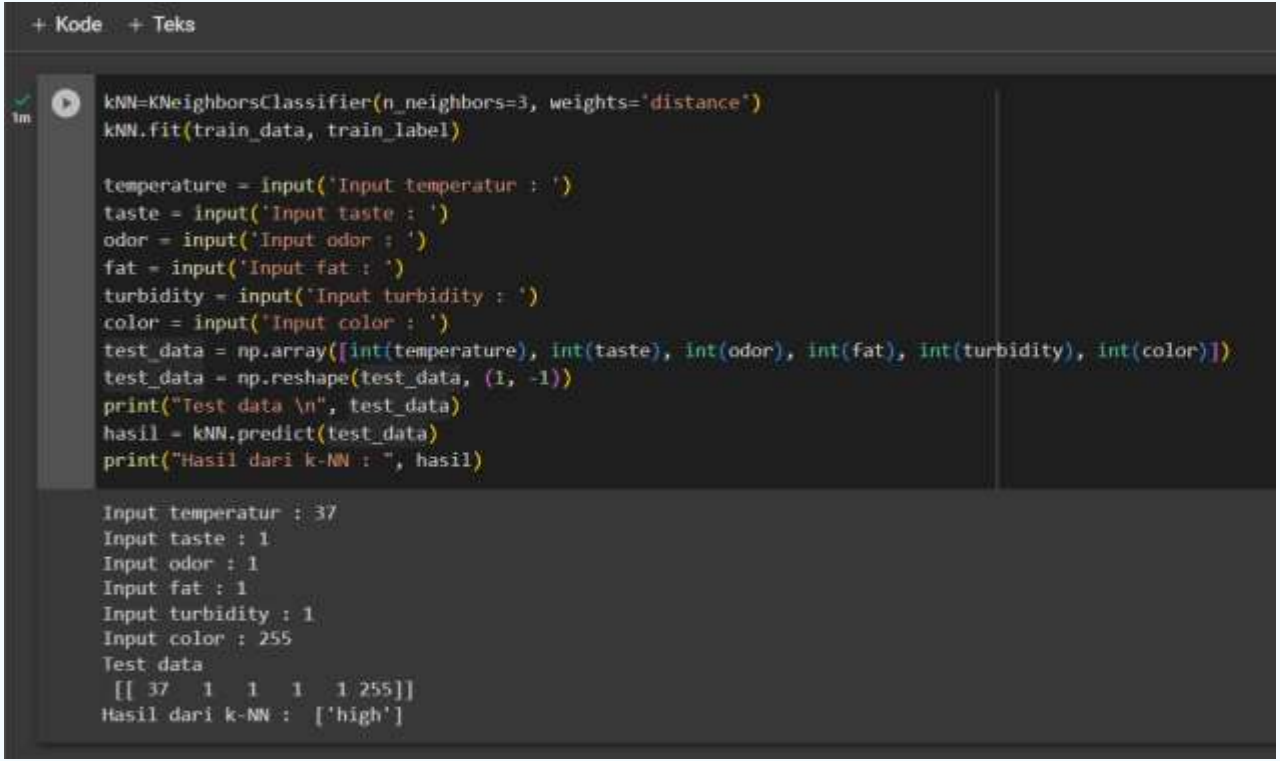


```
from sklearn.preprocessing import MinMaxScaler

dataset = pd.read_csv('milk_testing.csv')
sc = MinMaxScaler(feature_range=(0, 1))
train_data = np.array(dataset)[:,1:-1]
data = sc.fit_transform (train_data)
print(data)
```

```
[[0.19642857 1.         1.         1.         0.         0.33333333]
 [0.05357143 1.         1.         1.         1.         1.        ]
 [0.07142857 1.         0.         1.         0.         1.        ]
 ...
 [0.05357143 0.         0.         0.         0.         1.        ]
 [0.10714286 1.         0.         0.         0.         0.66666667]
 [0.19642857 1.         1.         0.         0.         0.46666667]]
```

8. Lakukan klasifikasi menggunakan k-NN untuk test_data, bandingkan hasilnya dengan test_label



```python
kNN=KNeighborsClassifier(n_neighbors=3, weights='distance')
kNN.fit(train_data, train_label)

temperature = input('Input temperatur : ')
taste = input('Input taste : ')
odor = input('Input odor : ')
fat = input('Input fat : ')
turbidity = input('Input turbidity : ')
color = input('Input color : ')
test_data = np.array([int(temperature), int(taste), int(odor), int(fat), int(turbidity), int(color)])
test_data = np.reshape(test_data, (1, -1))
print("Test data \n", test_data)
hasil = kNN.predict(test_data)
print("Hasil dari k-NN : ", hasil)
```

```
Input temperatur : 37
Input taste : 1
Input odor : 1
Input fat : 1
Input turbidity : 1
Input color : 255
Test data
 [[ 37   1   1   1   1 255]]
Hasil dari k-NN :  ['high']
```