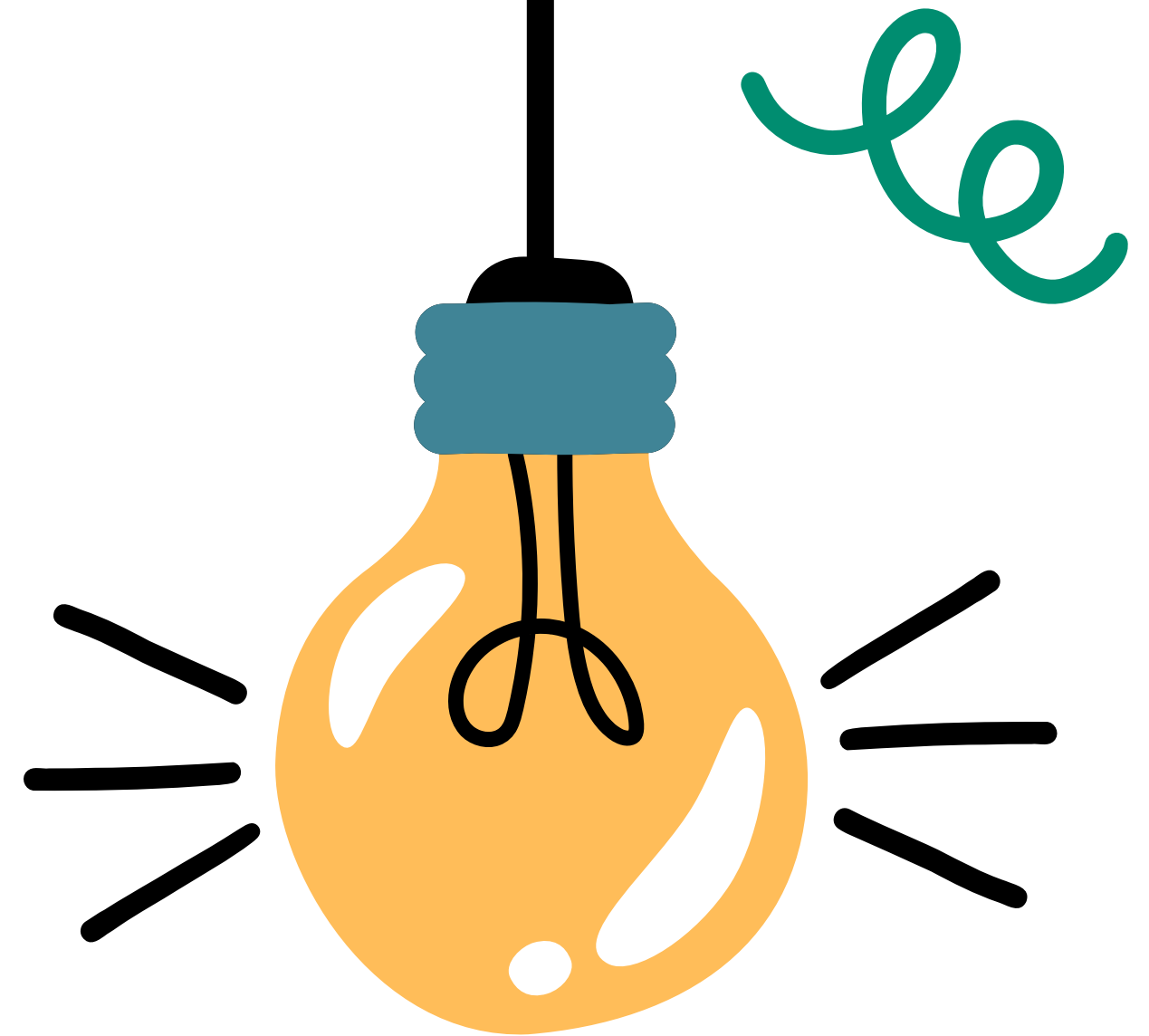




# DECISION TREE



Dosen Pengampu : Entin Martiana Kusumaningtyas S.Kom,  
M.Kom.

Bayu Kurniawan / 3322600019

## Praktikum Decission Tree

```
# Assignment 1
import pandas as pd
import numpy as np
# menampilkan data milk_traning.csv
train_dataset = pd.read_csv("C:/Users/bayuk/OneDrive/Documents/AI/pens/smtr3/Machine Learning/Data/milk_training.csv")
train_dataset
```

	pH	Temprature	Taste	Odor	Fat	Turbidity	Colour	Grade
0	6.6	35	1	0	1	0	254	high
1	6.6	36	0	1	0	1	253	high
2	6.6	37	1	1	1	1	255	high
3	6.8	45	0	1	1	1	255	high
4	6.6	45	0	1	1	1	250	high
...	...	...	...	...	...	...	...	...
737	6.7	45	1	1	1	0	245	medium
738	6.5	38	1	0	1	0	255	medium
739	6.7	41	1	0	0	0	247	medium
740	6.8	41	0	0	0	0	255	medium
741	6.8	38	0	0	0	0	255	medium

742 rows × 8 columns

## Praktikum Decission Tree

# Assignment 2

```
train_data = train_dataset[['pH', 'Temprature', 'Taste', 'Odor', 'Fat', 'Turbidity', 'Colour']]  
print(train_data)
```

	pH	Temprature	Taste	Odor	Fat	Turbidity	Colour
0	6.6	35	1	0	1	0	254
1	6.6	36	0	1	0	1	253
2	6.6	37	1	1	1	1	255
3	6.8	45	0	1	1	1	255
4	6.6	45	0	1	1	1	250
..	...	...	...	...	...	...	...
737	6.7	45	1	1	1	0	245
738	6.5	38	1	0	1	0	255
739	6.7	41	1	0	0	0	247
740	6.8	41	0	0	0	0	255
741	6.8	38	0	0	0	0	255

[742 rows x 7 columns]

## Praktikum Decission Tree

```
# Assignment 3
# menampilkan data milk_traning.csv
test_dataset = pd.read_csv("C:/Users/bayuk/OneDrive/Documents/AI/pens/smtr3/Machine Learning/Data/milk_testing.csv")
test_dataset
```

	pH	Temprature	Taste	Odor	Fat	Turbidity	Colour	Grade
0	6.8	45	1	1	1	0	245	high
1	6.6	37	1	1	1	1	255	high
2	6.7	38	1	0	1	0	255	high
3	6.8	45	0	1	1	1	255	high
4	6.6	37	1	1	1	1	255	high
...	...	...	...	...	...	...	...	...
312	6.5	36	0	0	0	0	247	medium
313	6.6	38	0	0	0	0	255	medium
314	6.5	37	0	0	0	0	255	medium
315	6.5	40	1	0	0	0	250	medium
316	6.7	45	1	1	0	0	247	medium

317 rows × 8 columns

## Praktikum Decission Tree

```
# Assignment 4
test_data = test_dataset[['pH', 'Temprature', 'Taste', 'Odor', 'Fat', 'Turbidity', 'Colour']]
print(test_data)
```

	pH	Temprature	Taste	Odor	Fat	Turbidity	Colour
0	6.8	45	1	1	1	0	245
1	6.6	37	1	1	1	1	255
2	6.7	38	1	0	1	0	255
3	6.8	45	0	1	1	1	255
4	6.6	37	1	1	1	1	255
..	...	...	...	...	...	...	...
312	6.5	36	0	0	0	0	247
313	6.6	38	0	0	0	0	255
314	6.5	37	0	0	0	0	255
315	6.5	40	1	0	0	0	250
316	6.7	45	1	1	0	0	247

[317 rows x 7 columns]

## Praktikum Decission Tree

```
# Assignment 5  
train_label = train_dataset['Grade']  
print(train_label)
```

```
0      high  
1      high  
2      high  
3      high  
4      high
```

```
...
```

```
737    medium  
738    medium  
739    medium  
740    medium  
741    medium
```

```
Name: Grade, Length: 742, dtype: object
```

## Praktikum Decision Tree

```
# Assignment 6
test_label = test_dataset['Grade']
print(test_label)
```

```
0      high
1      high
2      high
3      high
4      high
...
312    medium
313    medium
314    medium
315    medium
316    medium
Name: Grade, Length: 317, dtype: object
```

## ●●● Praktikum Decision Tree

```
# Assignment 7
from sklearn.tree import DecisionTreeClassifier as dtc
from sklearn.metrics import accuracy_score

dtc = dtc()
dtc.fit(train_data, train_label)
dtc_predict = dtc.predict(test_data)

acc_dtc = 1 - accuracy_score(test_label, dtc_predict)
print("Decision Tree Error Rate: ", acc_dtc)
```

Decision Tree Error Rate: 0.009463722397476393

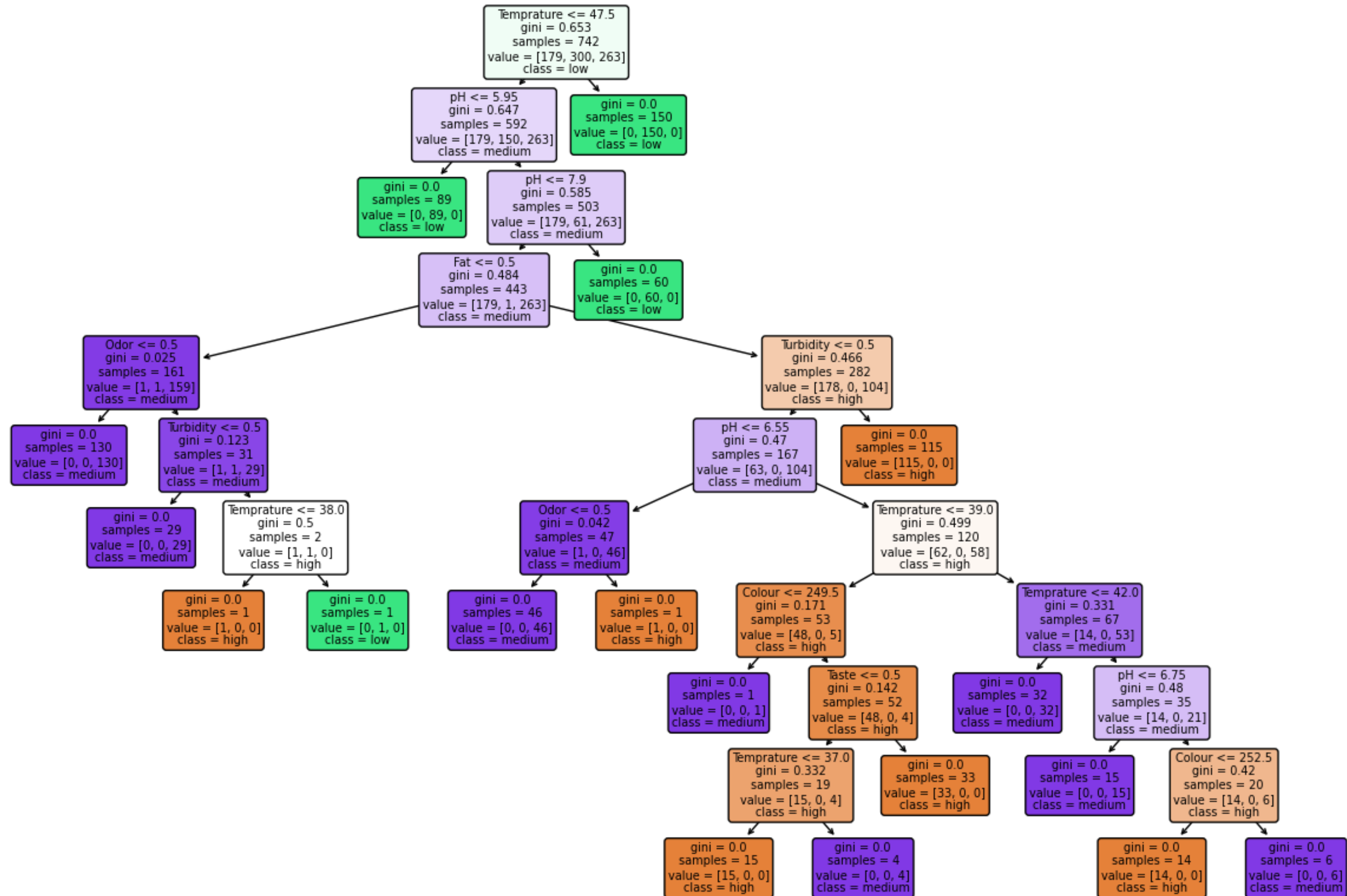


## Praktikum Decission Tree

```
# Assignment 8
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt

# Menampilkan pohon keputusan
plt.figure(figsize=(15, 10))
plot_tree(dtc, filled=True, feature_names=train_data.columns, class_names=train_label.unique(), rounded=True)
plt.show()
```

# Praktikum Decission Tree



## Praktikum Decision Tree

```
# Assignment 9
from sklearn.naive_bayes import GaussianNB as GNB
from sklearn.neighbors import KNeighborsClassifier
# Assignment 8
# Classification using k-NN
kNN = KNeighborsClassifier(n_neighbors=3, weights='distance')
kNN.fit(train_data, train_label)

kNN_predict = kNN.predict(test_data)
acc_kNN = accuracy_score(test_label, kNN_predict)
print("Decision Tree Error Rate: ", acc_dtc)
print("k-NN Error Rate :", acc_kNN)

# Classification using bayesian
classifier = GNB()
classifier.fit(train_data, train_label.values.ravel())

byn_pradict = classifier.predict(test_data)
acc_byn = accuracy_score(test_label, byn_pradict)
print("Bayesian Error Rate :", acc_byn)
```

```
Decision Tree Error Rate: 0.009463722397476393
k-NN Error Rate : 0.9873817034700315
Bayesian Error Rate : 0.9558359621451105
```