DECISION TREE

NADILA IMAARAH

D4 SAINS DATA TERAPAN A

3323600015

A MEMBACA DATASET

MENGIMPOR LIBRARY

```
import os
import pandas as pd
from sklearn import tree
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
```

LOAD DATASET

```
dataset = pd.read_csv("titanic.csv")
dataset.head()
```

OUTPUT

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Dataset yang digunakan adalah dataset titanic yang terdiri dari beberapa fitur yakni passengerid,survived,pclass,name,sex,age, Sibsp,parch,ticket,fare,cabin,dan embarked. Namun untuk klasifikasi saat ini tidak akan menggunakan semua fitur ini namun hanya mengambil beberapa fitur saja

B

MENAMPILKAN DATA TEST

```
1  test_dataset = pd.read_csv("titanic_test.csv")
2  test_dataset.head()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN

Untuk klasifikasi ini membutuhkan data tes, di sini saya menampilkan data tes untuk kelasifikasi. Data tes ini memiliki jumlah fitur yang sama dengan dataset awal

C

MEMBUAT DATA TRAIN

```python
train_data = dataset[['Sex', 'Age', 'Pclass', 'Fare']].copy()
train_data['Age'] = train_data.groupby('Pclass')['Age'].transform(lambda x: x.fillna(x.mean()))

train_data.isnull().sum()
```

```
Sex       0
Age       0
Pclass    0
Fare      0
dtype: int64
```

Setelah membuat data tes, di sini saya membuat data trian, namun untuk data train ini hanya terdiri dari fitur sex,age,pclass dan fare saja. Kemudian saya memastikan bahwa tidak ada missing value di data train ini. Jika ada missing value maka akan saya imputasi berdasarkan nilai rata-rata dari masing-masing class nya yakni Pclass. Dapat dilihat bahwa tidak ada fitur yang memilki missing value

E

MEMBUAT DATA TES

```
1  test_data = test_dataset[['Sex', 'Age', 'Pclass', 'Fare']].copy()
2  test_data.head()
```

	Sex	Age	Pclass	Fare
0	male	34.5	3	7.8292
1	female	47.0	3	7.0000
2	male	62.0	2	9.6875
3	male	27.0	3	8.6625
4	female	22.0	3	12.2875

Di sini saya membuat data tes lagi namun dengan hanya memilih beberapa fitur saja yakni fitur sex,age,pclass dan fare.

F MEMBUAT DATA TRAIN LABEL DAN TES LABEL

DATA TRAIN LABEL

```
train_label = dataset[['Survived']].copy()
train_label.head()
```

	Survived
0	0
1	1
2	1
3	1
4	0

Selanjutnya membuat data train_label. Data train ini hanya terdiri dari fitur Survived yang di ambil dari dataset awal

DATA TEST LABEL

```
test_label = pd.read_csv("titanic_testlabel.csv")
test_label = test_label.drop(columns=['PassengerId'])
test_label.head()
```

	Survived
0	0
1	1
2	0
3	0
4	1

Kemudian membuat data test_label. Data test_label ini didapatkan dari file titanic_test_label. Pada file ini kita menghapus fitur PassengerId sehingga hanya ada fitur Survived saja

ENCODING

```
train_data["Sex"] = train_data["Sex"].map({"male": 0, "female": 1})
test_data["Sex"] = test_data["Sex"].map({"male": 0, "female": 1})
```

Untuk mempermudah klasifikasi di sini saya merubah fitur sex yang awalnya berupa kategorik yakni sex dan female akan saya rubah menjadi fitur numerik, angka 0 untuk kategori male dan angka 1 untuk kategori famale

H

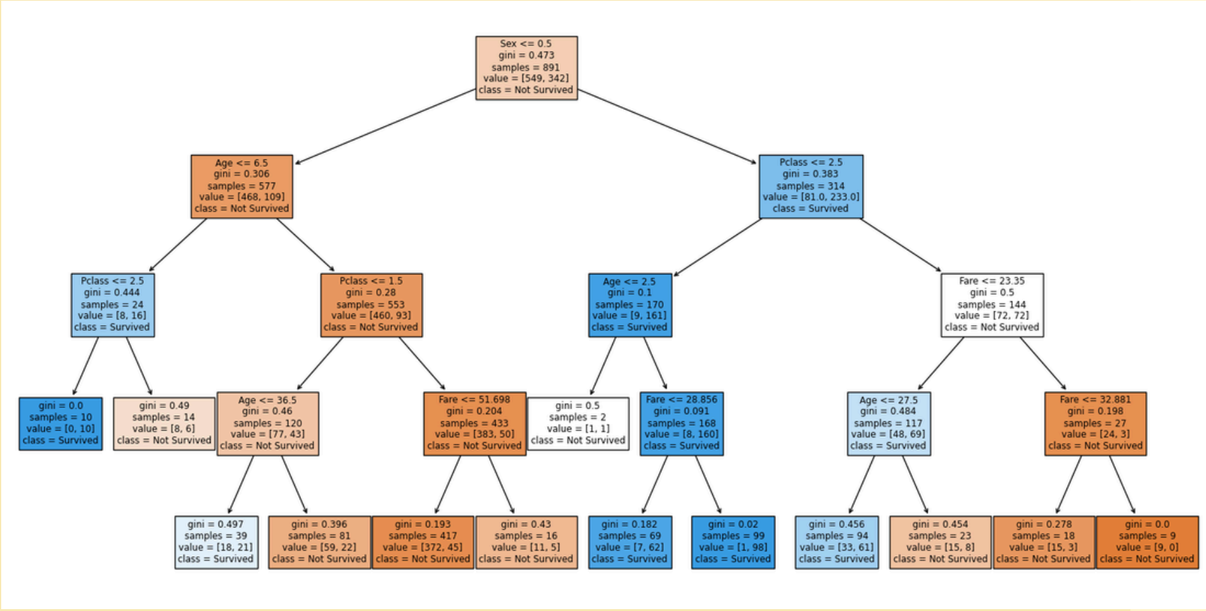
KLASIFIKASI DAN MENAMPILKAN ERROR RATIO

```
clf = DecisionTreeClassifier(max_depth=4,min_samples_split=15,random_state=42)
clf.fit(train_data, train_label)
y_pred = clf.predict(test_data)
accuracy = accuracy_score(test_label, y_pred)
error_ratio = 1 - accuracy
print(f"Error Ratio: {error_ratio:.2%}")
print(f'Accuracy: {accuracy:.2%}')
```

Ditahap ini saya melakukan klasifikasi dengan menggunakan metode Decision Tree dengan parameter max_depth=4 dan min_samples_split=10, yang akan dilatih menggunakan data train kemudian diuji dengan test_data. Kemudian menampilkan error rasio sebesar 10.05% dan akurasi sebesar 89.95%

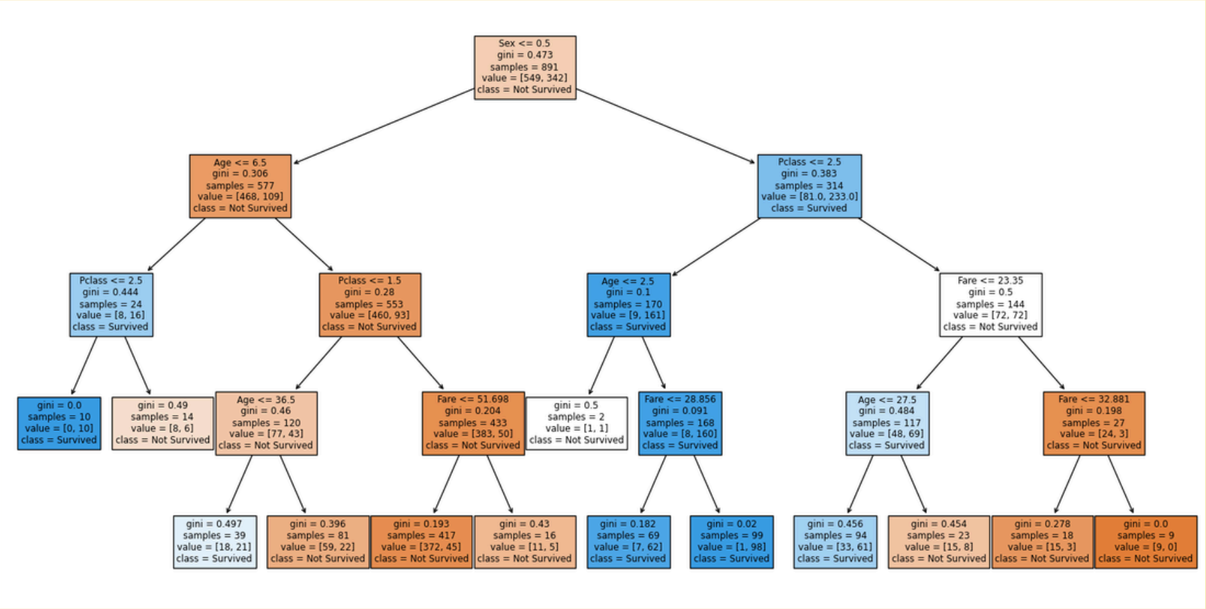
MENAMPILKAN HIERARKI TREE

```
plt.figure(figsize=(20, 10))
tree.plot_tree(clf, feature_names=train_data.columns,
               class_names=["Not Survived", "Survived"], filled=True)
plt.show()
```



Setelah melakukan klasifikasi, di sini saya akan menampilkan plot hierarki dari model decision tree yang telah di buat.

MENAMPILKAN HIERARKI TREE



Setiap node warna ini memiliki warna yang berbeda, yakni seperti berikut :

1. **warna biru : semakin tua warna biru menandakan bahwa hampir semua penumpang survived dengan nilai gini mendekati 0**
2. **warna orange : semakin tua warna orange menandakan hampir semua penumpang not survived dengan nilai gini yang lebih tinggi**
3. **warna putih : menandakan node ini murni yang menandakan hanya 1 kelas saja yakni survived semua atau not survived semua**

TERIMA KASIH