```python
# Import library pandas
import pandas as pd

# Mendefinisikan nama colom pada tabel
col_names = ['age', 'income', 'credit_rating', 'buys_car']

# Fungsi untuk membaca file .csv dengan menggunakan library pandas
df = pd.read_csv('dataset.csv', header=None, names=col_names)
df
```

	age	income	credit_rating	buys_car
0	25-30	high	fair	no
1	25-30	high	excellent	no
2	31-40	high	fair	no
3	>40	medium	fair	yes
4	>40	low	fair	yes
5	31-40	low	excellent	yes
6	31-40	medium	excellent	yes
7	25-30	low	fair	yes
8	>40	medium	excellent	no
9	>40	high	excellent	yes
10	25-30	medium	excellent	yes
11	25-30	medium	fair	no

```python
# Mendefinisikan variabel fitur yang digunakan
feature_cols = ['age', 'income', 'credit_rating']

# Features Variable
feature_variable = df[feature_cols]
# Target variable
target_variable = df.buys_car

# Menghapus kolom buys_car pada table
inputs = df.drop('buys_car', axis='columns')
```

```python
# Menampilkan target variable
target_variable
```

```
0      no
1      no
2      no
3     yes
4     yes
5     yes
6     yes
7     yes
8      no
9     yes
10    yes
11     no
Name: buys_car, dtype: object
```

```python
# Import library LabelEncoder untuk mengubah bilangan string ke bilangan integer
from sklearn.preprocessing import LabelEncoder

le_age = LabelEncoder() # Age label
le_income = LabelEncoder() # Income label
le_credit_rating = LabelEncoder() # Credit rating label
```

```python
print('Keterangan:')
print('age_n : 0 = 25-30, 1 = 31-40, 2 = >40')
print('income_n : 0 = high, 1 = low, 2 = medium')
print('credit_rating_n : 0 = excellent, 1 = fair')
print('-------------------------')

# Menambahkan kolom age label pada table
inputs['age_n'] = le_age.fit_transform(inputs['age'])
# Menambahkan kolom income label pada table
inputs['income_n'] = le_age.fit_transform(inputs['income'])
# Menambahkan credit rating age label pada table
inputs['credit_rating_n'] = le_age.fit_transform(inputs['credit_rating'])

# Menampilkan table yang sudah ditambah kolom baru
inputs
```

```
Keterangan:
age_n : 0 = 25-30, 1 = 31-40, 2 = >40
income_n : 0 = high, 1 = low, 2 = medium
credit_rating_n : 0 = excellent, 1 = fair
-------------------------
```

	age	income	credit_rating	age_n	income_n	credit_rating_n
0	25-30	high	fair	0	0	1
1	25-30	high	excellent	0	0	0

```python
# Menghapus kolom age, income, credit rating untuk menampilkan semua kolom yang sudah di encoder
inputs_n = inputs.drop(['age', 'income', 'credit_rating'], axis='columns')

# Menampilkan table yang kolom tertentu dihapus
inputs_n
```

	age_n	income_n	credit_rating_n
0	0	0	1
1	0	0	0
2	1	0	1
3	2	2	1
4	2	1	1
5	1	1	0
6	1	2	0
7	0	1	1
8	2	2	0
9	2	0	0
10	0	2	0
11	0	2	1

```python
# Import library matplotlib
import matplotlib.pyplot as plt

# Import library tree dari sklearn
%matplotlib inline
from sklearn import tree

# Mendifinisikan model dengan fungsi DecisionTreeClassifier()
model = tree.DecisionTreeClassifier()
model.fit(inputs_n,target_variable)
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

```python
# Menghitung hasil perhitungan score antara target dengan data yang sudah di encoder
model.score(inputs_n,target_variable)
```

```
1.0
```

```python
# Memprediksi hasil target dari nilai encoder yang dimasukkan (age, income, credit rating)
model.predict([[2,1,1]])
```

```
array(['yes'], dtype=object)
```

✓ 0s completed at 10:39 AM ● ×