Classification (2)

Import Library dan load Dataset

In [1]:
```python
import pandas as pd
import numpy as np
```

In [2]:
```python
dataset = pd.read_csv(r"D:\SEMESTER 4\IS411 Data Modelling\LAB\Bahan Modul 10\diabetes.csv", header=0)
dataset
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

In [3]:
```python
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Modelling

In [4]:
```python
#split dataset in features and target variable
feature_cols = ['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedigreeFunction','Age']
X = dataset[feature_cols] #features
y = dataset['Outcome'] #Target variables
```

In [5]:
```python
#split dataset into training set and test set
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
```

Menggunakan teknik pruning

Decision Tree biasa

In [6]:
```python
from sklearn.tree import DecisionTreeClassifier

# Create Decision Tree classifer object
clf = DecisionTreeClassifier()

# Train Decision Tree Classifier
clf = clf.fit(X_train, y_train)

# Predict the response for test dataset
y_pred = clf.predict(X_test)
```

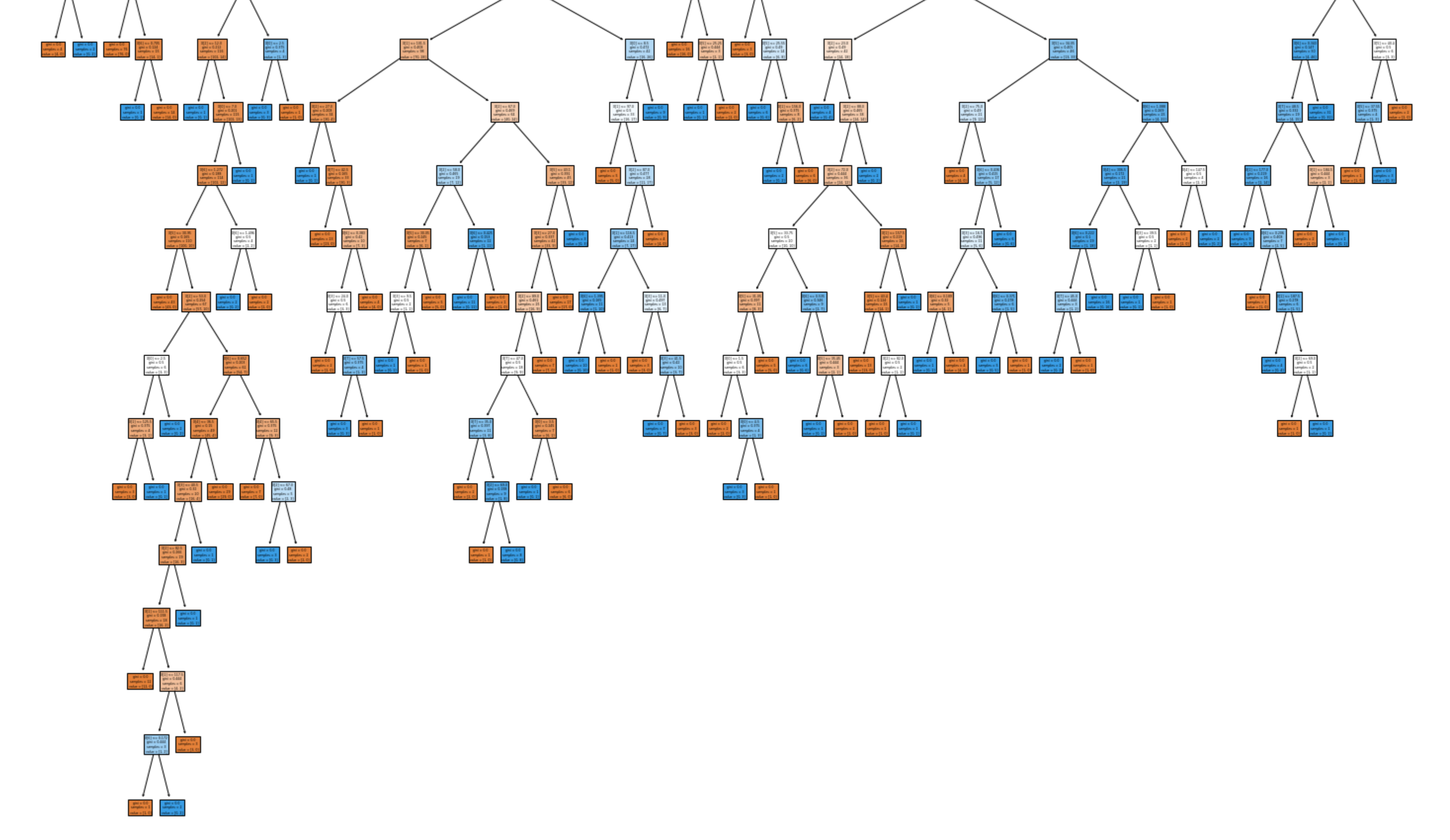
In [7]:
```python
from sklearn import metrics

# Model Accuracy, how often is the classifier correct?
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.7012987012987013
```

In [8]:
```python
from matplotlib import pyplot as plt
from sklearn import tree

fig = plt.figure(figsize=(25,20))
tree.plot_tree(clf, filled=True)
```

Out[8]:

```
[Text(0.4008620689655174, 0.9705882352941176, 'X[1] <= 129.5\ngini = 0.449\nsamples = 537\nvalue = [354, 183]'),
 Text(0.14906609195402298, 0.9117647058823529, 'X[5] <= 26.3\ngini = 0.329\nsamples = 357\nvalue = [283, 74]'),
 Text(0.04597701149425287, 0.8529411764705882, 'X[5] <= 9.1\ngini = 0.06\nsamples = 97\nvalue = [94, 3]'),
 Text(0.022988505747126436, 0.7941176470588235, 'X[0] <= 7.5\ngini = 0.444\nsamples = 6\nvalue = [4, 2]'),
 Text(0.011494252873563218, 0.7352941176470589, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.034482758620689655, 0.7352941176470589, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.06896551724137931, 0.7941176470588235, 'X[6] <= 0.669\ngini = 0.022\nsamples = 91\nvalue = [90, 1]'),
 Text(0.05747126436781609, 0.7352941176470589, 'gini = 0.0\nsamples = 76\nvalue = [76, 0]'),
 Text(0.08045977011494253, 0.7352941176470589, 'X[6] <= 0.705\ngini = 0.124\nsamples = 15\nvalue = [14, 1]'),
 Text(0.06896551724137931, 0.6764705882352942, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.09195402298850575, 0.6764705882352942, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]'),
 Text(0.2521551724137931, 0.8529411764705882, 'X[7] <= 27.5\ngini = 0.397\nsamples = 260\nvalue = [189, 71]'),
 Text(0.14942528735632185, 0.7941176470588235, 'X[5] <= 45.4\ngini = 0.243\nsamples = 120\nvalue = [103, 17]'),
 Text(0.12643678160919541, 0.7352941176470589, 'X[2] <= 12.0\ngini = 0.212\nsamples = 116\nvalue = [102, 14]'),
 Text(0.11494252873563218, 0.6764705882352942, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.13793103448275862, 0.6764705882352942, 'X[6] <= 1.272\ngini = 0.201\nsamples = 115\nvalue = [102, 13]'),
 Text(0.12643678160919541, 0.6176470588235294, 'X[4] <= 7.0\ngini = 0.188\nsamples = 114\nvalue = [102, 12]'),
 Text(0.10344827586206896, 0.5588235294117647, 'X[5] <= 30.95\ngini = 0.165\nsamples = 110\nvalue = [100, 10]'),
 Text(0.09195402298850575, 0.5, 'gini = 0.0\nsamples = 43\nvalue = [43, 0]'),
 Text(0.11494252873563218, 0.5, 'X[2] <= 53.0\ngini = 0.254\nsamples = 67\nvalue = [57, 10]'),
 Text(0.08620689655172414, 0.4411764705882353, 'X[0] <= 2.5\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(0.07471264367816093, 0.38235294117647056, 'X[1] <= 125.5\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.06321839080459771, 0.3235294117647059, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.08620689655172414, 0.3235294117647059, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.09770114942528736, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.14367816091954022, 0.4411764705882353, 'X[6] <= 0.652\ngini = 0.203\nsamples = 61\nvalue = [54, 7]'),
 Text(0.1206896551724138, 0.38235294117647056, 'X[4] <= 36.5\ngini = 0.15\nsamples = 49\nvalue = [45, 4]'),
 Text(0.10919540229885058, 0.3235294117647059, 'X[5] <= 32.0\ngini = 0.32\nsamples = 20\nvalue = [16, 4]'),
 Text(0.09770114942528736, 0.2647058823529412, 'X[1] <= 110.0\ngini = 0.266\nsamples = 19\nvalue = [16, 3]'),
 Text(0.08620689655172414, 0.20588235294117646, 'X[1] <= 111.5\ngini = 0.198\nsamples = 18\nvalue = [16, 2]'),
 Text(0.07471264367816093, 0.14705882352941177, 'gini = 0.0\nsamples = 12\nvalue = [12, 0]'),
 Text(0.09770114942528736, 0.14705882352941177, 'X[1] <= 117.5\ngini = 0.444\nsamples = 6\nvalue = [4, 2]'),
 Text(0.08620689655172414, 0.08823529411764706, 'X[6] <= 0.172\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.07471264367816093, 0.029411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.09770114942528736, 0.029411764705882353, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.10919540229885058, 0.08823529411764706, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.10919540229885058, 0.20588235294117646, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.1206896551724138, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.13218390804597702, 0.3235294117647059, 'gini = 0.0\nsamples = 29\nvalue = [29, 0]'),
 Text(0.16666666666666666, 0.38235294117647056, 'X[4] <= 65.5\ngini = 0.375\nsamples = 12\nvalue = [9, 3]'),
 Text(0.15517241379310345, 0.3235294117647059, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.1781609195402299, 0.3235294117647059, 'X[2] <= 67.0\ngini = 0.48\nsamples = 5\nvalue = [2, 3]'),
 Text(0.16666666666666666, 0.2647058823529412, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.1896551724137931, 0.2647058823529412, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.14942528735632185, 0.5588235294117647, 'X[6] <= 1.496\ngini = 0.5\nsamples = 4\nvalue = [2, 2]'),
 Text(0.13793103448275862, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.16091954022988506, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.14942528735632185, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.1724137931034483, 0.7352941176470589, 'X[0] <= 2.5\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.16091954022988506, 0.6764705882352942, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.1839080459770115, 0.6764705882352942, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.35488505747126436, 0.7941176470588235, 'X[6] <= 0.563\ngini = 0.474\nsamples = 140\nvalue = [86, 54]'),
 Text(0.27298850574712646, 0.7352941176470589, 'X[1] <= 101.5\ngini = 0.408\nsamples = 98\nvalue = [70, 28]'),
 Text(0.20689655172413793, 0.6764705882352942, 'X[2] <= 27.0\ngini = 0.208\nsamples = 34\nvalue = [30, 4]'),
 Text(0.19540229885057472, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.21839080459770116, 0.6176470588235294, 'X[7] <= 42.5\ngini = 0.165\nsamples = 33\nvalue = [30, 3]'),
 Text(0.20689655172413793, 0.5588235294117647, 'gini = 0.0\nsamples = 23\nvalue = [23, 0]'),
 Text(0.22988505747126436, 0.5588235294117647, 'X[6] <= 0.383\ngini = 0.42\nsamples = 10\nvalue = [7, 3]'),
 Text(0.21839080459770116, 0.5, 'X[3] <= 24.0\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(0.20689655172413793, 0.4411764705882353, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.22988505747126436, 0.4411764705882353, 'X[7] <= 57.5\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.21839080459770116, 0.38235294117647056, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.2413793103448276, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.2413793103448276, 0.5, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.3390804597701149, 0.6764705882352942, 'X[1] <= 125.5\ngini = 0.469\nsamples = 64\nvalue = [40, 24]'),
 Text(0.29885057471264365, 0.6176470588235294, 'X[2] <= 58.0\ngini = 0.465\nsamples = 19\nvalue = [7, 12]'),
 Text(0.27586206896551724, 0.5588235294117647, 'X[5] <= 30.05\ngini = 0.245\nsamples = 7\nvalue = [6, 1]'),
 Text(0.26436781609195403, 0.5, 'X[3] <= 9.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.25287356321839083, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.27586206896551724, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.28735632183908044, 0.5, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.3218390804597701, 0.5588235294117647, 'X[6] <= 0.425\ngini = 0.153\nsamples = 12\nvalue = [1, 11]'),
 Text(0.3103448275862069, 0.5, 'gini = 0.0\nsamples = 11\nvalue = [0, 11]'),
 Text(0.3333333333333333, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.3793103448275862, 0.6176470588235294, 'X[5] <= 43.1\ngini = 0.391\nsamples = 45\nvalue = [33, 12]'),
 Text(0.367816091954023, 0.5588235294117647, 'X[3] <= 27.0\ngini = 0.337\nsamples = 42\nvalue = [33, 9]'),
 Text(0.3563218390804598, 0.5, 'X[2] <= 89.0\ngini = 0.461\nsamples = 25\nvalue = [16, 9]'),
 Text(0.3448275862068966, 0.4411764705882353, 'X[7] <= 47.0\ngini = 0.5\nsamples = 18\nvalue = [9, 9]'),
 Text(0.3218390804597701, 0.38235294117647056, 'X[4] <= 48.0\ngini = 0.397\nsamples = 11\nvalue = [3, 8]'),
 Text(0.3103448275862069, 0.3235294117647059, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.3333333333333333, 0.3235294117647059, 'X[2] <= 69.0\ngini = 0.198\nsamples = 9\nvalue = [1, 8]'),
 Text(0.3218390804597701, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.3448275862068966, 0.2647058823529412, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(0.367816091954023, 0.38235294117647056, 'X[0] <= 3.5\ngini = 0.245\nsamples = 7\nvalue = [6, 1]'),
 Text(0.3563218390804598, 0.3235294117647059, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3793103448275862, 0.3235294117647059, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.367816091954023, 0.4411764705882353, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.3793103448275862, 0.5, 'gini = 0.0\nsamples = 17\nvalue = [17, 0]'),
 Text(0.39080459770114945, 0.5588235294117647, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.4367816091954023, 0.7352941176470589, 'X[0] <= 8.5\ngini = 0.472\nsamples = 42\nvalue = [16, 26]'),
 Text(0.4252873563218391, 0.6764705882352942, 'X[6] <= 0.1\ngini = 0.5\nsamples = 33\nvalue = [16, 17]'),
 Text(0.41379310344827586, 0.6176470588235294, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.4367816091954023, 0.6176470588235294, 'X[2] <= 87.0\ngini = 0.477\nsamples = 28\nvalue = [11, 17]'),
 Text(0.4252873563218391, 0.5588235294117647, 'X[1] <= 116.5\ngini = 0.413\nsamples = 24\nvalue = [7, 17]'),
 Text(0.40229885057471265, 0.5, 'X[6] <= 1.395\ngini = 0.165\nsamples = 11\nvalue = [1, 10]'),
 Text(0.39080459770114945, 0.4411764705882353, 'gini = 0.0\nsamples = 10\nvalue = [0, 10]'),
 Text(0.41379310344827586, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.4482758620689655, 0.5, 'X[3] <= 11.0\ngini = 0.497\nsamples = 13\nvalue = [6, 7]'),
 Text(0.4367816091954023, 0.4411764705882353, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.45977011494252873, 0.4411764705882353, 'X[3] <= 41.5\ngini = 0.42\nsamples = 10\nvalue = [3, 7]'),
 Text(0.4482758620689655, 0.38235294117647056, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(0.47126436781609193, 0.38235294117647056, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.4482758620689655, 0.5588235294117647, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.4482758620689655, 0.6764705882352942, 'gini = 0.0\nsamples = 9\nvalue = [0, 9]'),
 Text(0.6525804597701149, 0.9117647058823529, 'X[5] <= 27.85\ngini = 0.478\nsamples = 180\nvalue = [71, 109]'),
 Text(0.5172413793103449, 0.8529411764705882, 'X[1] <= 145.5\ngini = 0.375\nsamples = 36\nvalue = [27, 9]'),
 Text(0.47701149425287354, 0.7941176470588235, 'X[7] <= 59.5\ngini = 0.1\nsamples = 19\nvalue = [18, 1]'),
 Text(0.46551724137931033, 0.7352941176470589, 'gini = 0.0\nsamples = 16\nvalue = [16, 0]'),
 Text(0.4885057471264368, 0.7352941176470589, 'X[5] <= 25.25\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.47701149425287354, 0.6764705882352942, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5, 0.6764705882352942, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.5229885057471264, 0.7941176470588235, 'X[5] <= 23.1\ngini = 0.498\nsamples = 17\nvalue = [9, 8]'),
 Text(0.5114942528735632, 0.7352941176470589, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.5344827586206896, 0.7352941176470589, 'X[5] <= 25.55\ngini = 0.49\nsamples = 14\nvalue = [6, 8]'),
 Text(0.5229885057471264, 0.6764705882352942, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.5459770114942529, 0.6764705882352942, 'X[1] <= 156.0\ngini = 0.375\nsamples = 8\nvalue = [6, 2]'),
 Text(0.5344827586206896, 0.6176470588235294, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.5574712643678161, 0.6176470588235294, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.805316091954023, 0.8529411764705882, 'X[1] <= 158.5\ngini = 0.424\nsamples = 144\nvalue = [44, 100]'),
 Text(0.6623563218390804, 0.7941176470588235, 'X[7] <= 30.5\ngini = 0.487\nsamples = 88\nvalue = [37, 51]'),
 Text(0.5804597701149425, 0.7352941176470589, 'X[2] <= 23.0\ngini = 0.49\nsamples = 42\nvalue = [24, 18]'),
 Text(0.5689655172413793, 0.6764705882352942, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.5919540229885057, 0.6764705882352942, 'X[2] <= 88.0\ngini = 0.465\nsamples = 38\nvalue = [24, 14]'),
 Text(0.5804597701149425, 0.6176470588235294, 'X[2] <= 72.0\ngini = 0.444\nsamples = 36\nvalue = [24, 12]'),
 Text(0.5402298850574713, 0.5588235294117647, 'X[5] <= 33.75\ngini = 0.5\nsamples = 20\nvalue = [10, 10]'),
 Text(0.5172413793103449, 0.5, 'X[5] <= 31.05\ngini = 0.397\nsamples = 11\nvalue = [8, 3]'),
 Text(0.5057471264367817, 0.4411764705882353, 'X[0] <= 1.5\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(0.4942528735632184, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.5172413793103449, 0.38235294117647056, 'X[0] <= 4.5\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.5057471264367817, 0.3235294117647059, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.5287356321839081, 0.3235294117647059, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5287356321839081, 0.4411764705882353, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.5632183908045977, 0.5, 'X[6] <= 0.535\ngini = 0.346\nsamples = 9\nvalue = [2, 7]'),
 Text(0.5517241379310345, 0.4411764705882353, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.5747126436781609, 0.4411764705882353, 'X[5] <= 35.45\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.5632183908045977, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5862068965517241, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.6206896551724138, 0.5588235294117647, 'X[1] <= 157.5\ngini = 0.219\nsamples = 16\nvalue = [14, 2]'),
 Text(0.6091954022988506, 0.5, 'X[5] <= 43.4\ngini = 0.124\nsamples = 15\nvalue = [14, 1]'),
 Text(0.5977011494252874, 0.4411764705882353, 'gini = 0.0\nsamples = 13\nvalue = [13, 0]'),
 Text(0.6206896551724138, 0.4411764705882353, 'X[2] <= 82.0\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.6091954022988506, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.632183908045977, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.632183908045977, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.603448275862069, 0.6176470588235294, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.7442528735632183, 0.7352941176470589, 'X[5] <= 34.05\ngini = 0.405\nsamples = 46\nvalue = [13, 33]'),
 Text(0.6781609195402298, 0.6764705882352942, 'X[2] <= 75.0\ngini = 0.49\nsamples = 21\nvalue = [9, 12]'),
 Text(0.6666666666666666, 0.6176470588235294, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.6896551724137931, 0.6176470588235294, 'X[3] <= 16.5\ngini = 0.415\nsamples = 17\nvalue = [5, 12]'),
 Text(0.6551724137931034, 0.5588235294117647, 'X[6] <= 0.189\ngini = 0.496\nsamples = 11\nvalue = [5, 6]'),
 Text(0.6436781609195402, 0.5, 'X[6] <= 0.371\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(0.632183908045977, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6551724137931034, 0.4411764705882353, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.6666666666666666, 0.5, 'X[6] <= 0.371\ngini = 0.278\nsamples = 6\nvalue = [1, 5]'),
 Text(0.6551724137931034, 0.4411764705882353, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(0.6781609195402298, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7241379310344828, 0.5588235294117647, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.8103448275862069, 0.6764705882352942, 'X[6] <= 1.088\ngini = 0.269\nsamples = 25\nvalue = [4, 21]'),
 Text(0.7816091954022989, 0.6176470588235294, 'X[7] <= 48.5\ngini = 0.172\nsamples = 21\nvalue = [2, 19]'),
 Text(0.7586206896551724, 0.5588235294117647, 'X[6] <= 0.222\ngini = 0.1\nsamples = 19\nvalue = [1, 18]'),
 Text(0.7471264367816092, 0.5, 'X[7] <= 45.0\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.735632183908046, 0.4411764705882353, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.7586206896551724, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7701149425287356, 0.5, 'gini = 0.0\nsamples = 16\nvalue = [0, 16]'),
 Text(0.8045977011494253, 0.5588235294117647, 'X[3] <= 39.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.7931034482758621, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8160919540229885, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8390804597701149, 0.6176470588235294, 'X[4] <= 147.5\ngini = 0.5\nsamples = 4\nvalue = [2, 2]'),
 Text(0.8275862068965517, 0.5588235294117647, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.8505747126436781, 0.5588235294117647, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.9482758620689655, 0.7941176470588235, 'X[6] <= 1.157\ngini = 0.219\nsamples = 56\nvalue = [7, 49]'),
 Text(0.9195402298850575, 0.7352941176470589, 'X[6] <= 0.343\ngini = 0.147\nsamples = 50\nvalue = [4, 46]'),
 Text(0.8850574712643678, 0.6764705882352942, 'X[7] <= 48.5\ngini = 0.332\nsamples = 19\nvalue = [4, 15]'),
 Text(0.8735632183908046, 0.6176470588235294, 'X[1] <= 177.0\ngini = 0.219\nsamples = 16\nvalue = [2, 14]'),
 Text(0.8620689655172413, 0.5588235294117647, 'gini = 0.0\nsamples = 9\nvalue = [0, 9]'),
 Text(0.896551724137931, 0.5588235294117647, 'X[6] <= 0.206\ngini = 0.408\nsamples = 7\nvalue = [2, 5]'),
 Text(0.8850574712643678, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9080459770114943, 0.5, 'X[1] <= 187.5\ngini = 0.278\nsamples = 6\nvalue = [1, 5]'),
 Text(0.896551724137931, 0.4411764705882353, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.9195402298850575, 0.4411764705882353, 'X[2] <= 69.0\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.9080459770114943, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9310344827586207, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.896551724137931, 0.6176470588235294, 'X[1] <= 184.5\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.8850574712643678, 0.5588235294117647, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.9080459770114943, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9540229885057471, 0.6764705882352942, 'gini = 0.0\nsamples = 31\nvalue = [0, 31]'),
 Text(0.9770114942528736, 0.7352941176470589, 'X[5] <= 40.4\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(0.9655172413793104, 0.6764705882352942, 'X[7] <= 37.5\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.9540229885057471, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9770114942528736, 0.6176470588235294, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.9885057471264368, 0.6764705882352942, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]')]
```



Decision Tree with pruning

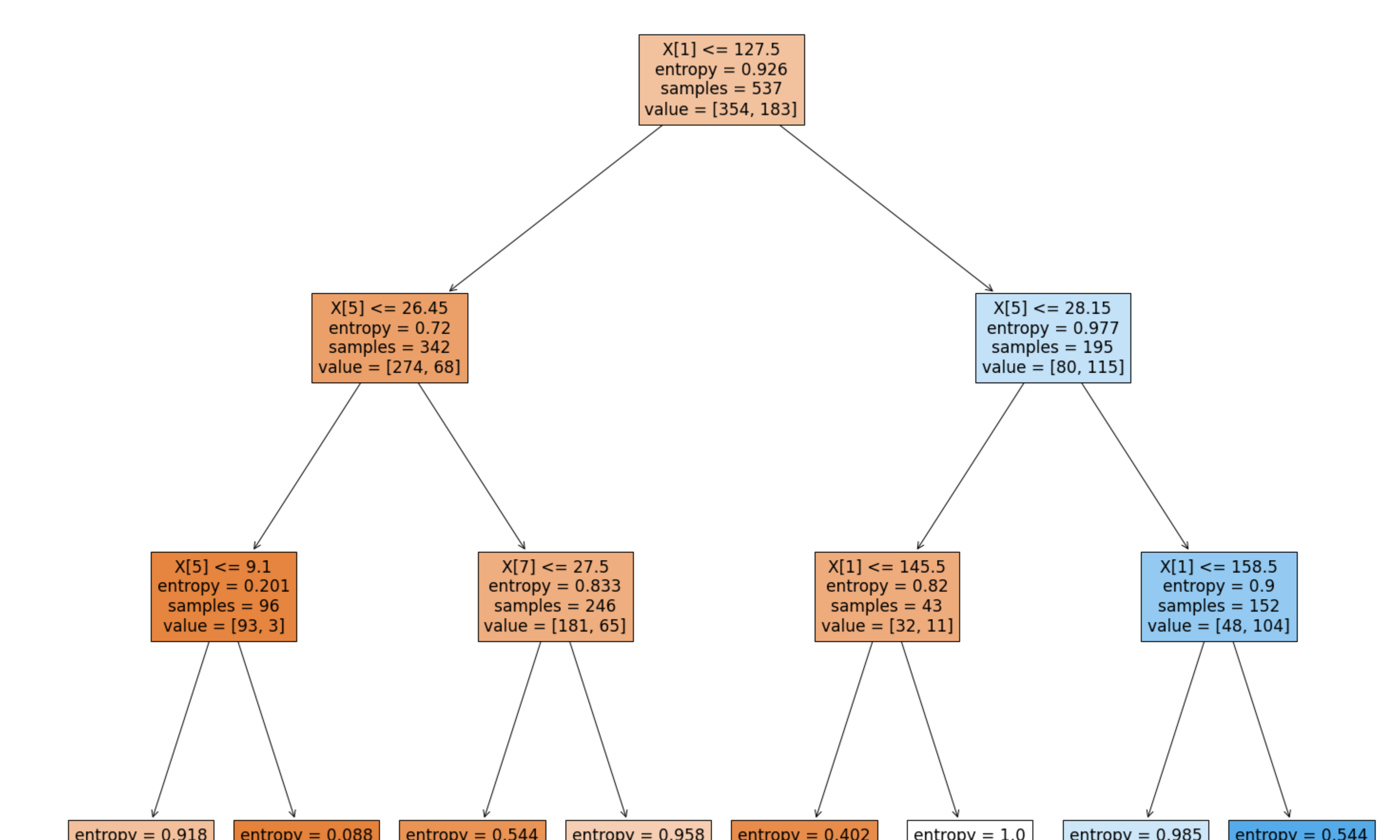
In [9]:
```python
# Create Decision Tree Classifier
clf_p = DecisionTreeClassifier(criterion="entropy", max_depth=3)

# Train Decision Tree Classifier
clf_p = clf_p.fit(X_train, y_train)

#Predict the response for test dataset
y_pred_p = clf_p.predict(X_test)

# Model Accuracy, how often is the classifier correct?
print("Accuracy:", metrics.accuracy_score(y_test, y_pred_p))
```

```
Accuracy: 0.7705627705627706
```

In [10]:
```python
from matplotlib import pyplot as plt
from sklearn import tree

fig = plt.figure(figsize=(25,20))
tree.plot_tree(clf_p, filled=True)
```

Out[10]:

```
[Text(0.5, 0.875, 'X[1] <= 127.5\nentropy = 0.926\nsamples = 537\nvalue = [354, 183]'),
 Text(0.25, 0.625, 'X[5] <= 26.45\nentropy = 0.72\nsamples = 342\nvalue = [274, 68]'),
 Text(0.125, 0.375, 'X[5] <= 9.1\nentropy = 0.201\nsamples = 96\nvalue = [93, 3]'),
 Text(0.0625, 0.125, 'entropy = 0.918\nsamples = 6\nvalue = [4, 2]'),
 Text(0.1875, 0.125, 'entropy = 0.088\nsamples = 90\nvalue = [89, 1]'),
 Text(0.375, 0.375, 'X[7] <= 27.5\nentropy = 0.833\nsamples = 246\nvalue = [181, 65]'),
 Text(0.3125, 0.125, 'entropy = 0.544\nsamples = 112\nvalue = [98, 14]'),
 Text(0.4375, 0.125, 'entropy = 0.958\nsamples = 134\nvalue = [83, 51]'),
 Text(0.75, 0.625, 'X[5] <= 28.15\nentropy = 0.977\nsamples = 195\nvalue = [80, 115]'),
 Text(0.625, 0.375, 'X[1] <= 145.5\nentropy = 0.82\nsamples = 43\nvalue = [32, 11]'),
 Text(0.5625, 0.125, 'entropy = 0.402\nsamples = 25\nvalue = [23, 2]'),
 Text(0.6875, 0.125, 'entropy = 1.0\nsamples = 18\nvalue = [9, 9]'),
 Text(0.875, 0.375, 'X[1] <= 158.5\nentropy = 0.9\nsamples = 152\nvalue = [48, 104]'),
 Text(0.8125, 0.125, 'entropy = 0.985\nsamples = 96\nvalue = [41, 55]'),
 Text(0.9375, 0.125, 'entropy = 0.544\nsamples = 56\nvalue = [7, 49]')]
```



Menggunakan random forest tanpa pruning

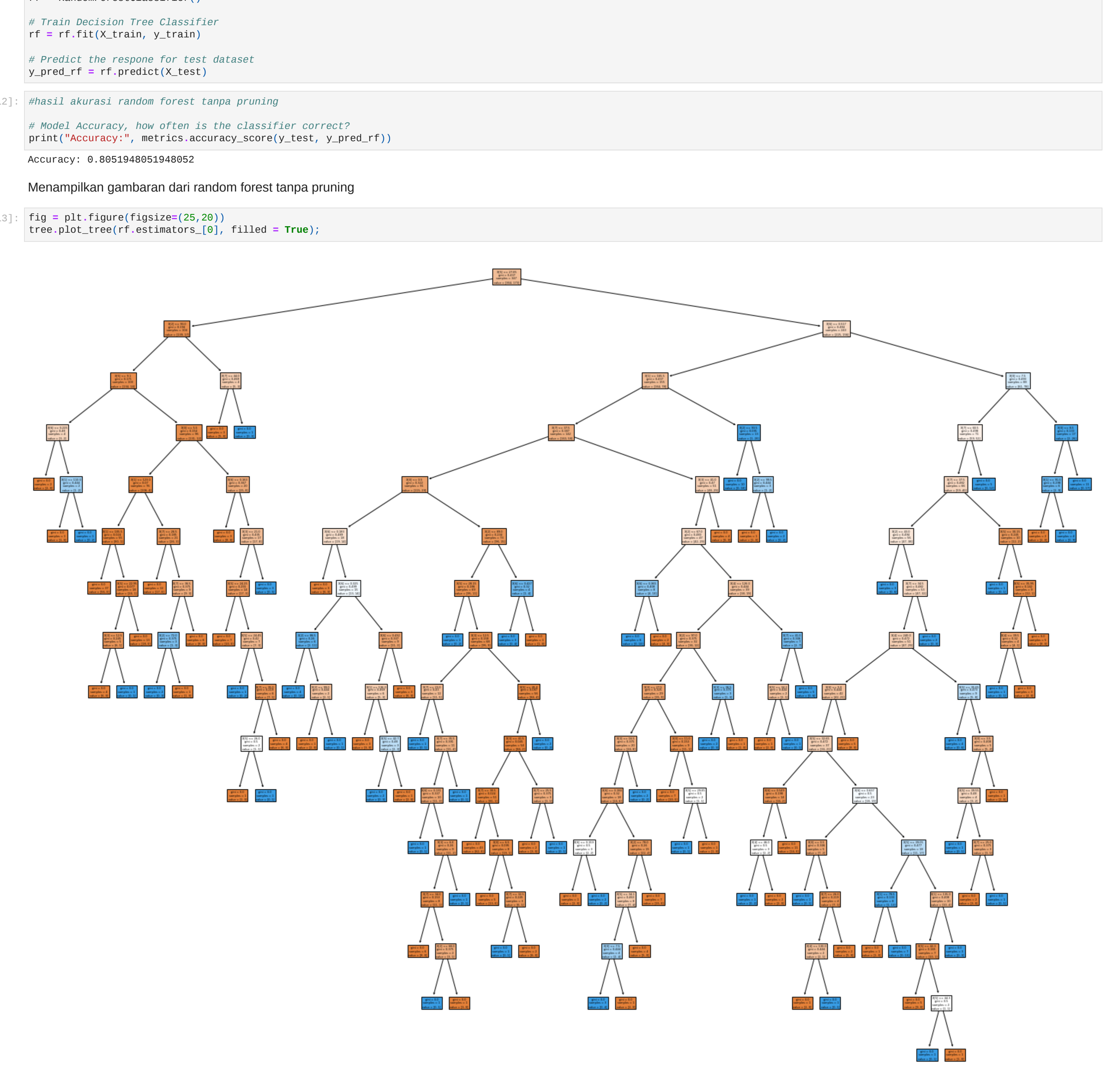
In [11]:
```python
from sklearn.ensemble import RandomForestClassifier

# Create Decision Tree Classifier object
rf = RandomForestClassifier()

# Train Decision Tree Classifier
rf = rf.fit(X_train, y_train)

# Predict the respone for test dataset
y_pred_rf = rf.predict(X_test)
```

In [12]:
```python
#hasil akurasi random forest tanpa pruning

# Model Accuracy, how often is the classifier correct?
print("Accuracy:", metrics.accuracy_score(y_test, y_pred_rf))
```

```
Accuracy: 0.8051948051948052
```

Menampilkan gambaran dari random forest tanpa pruning

In [13]:
```python
fig = plt.figure(figsize=(25,20))
tree.plot_tree(rf.estimators_[0], filled = True);
```



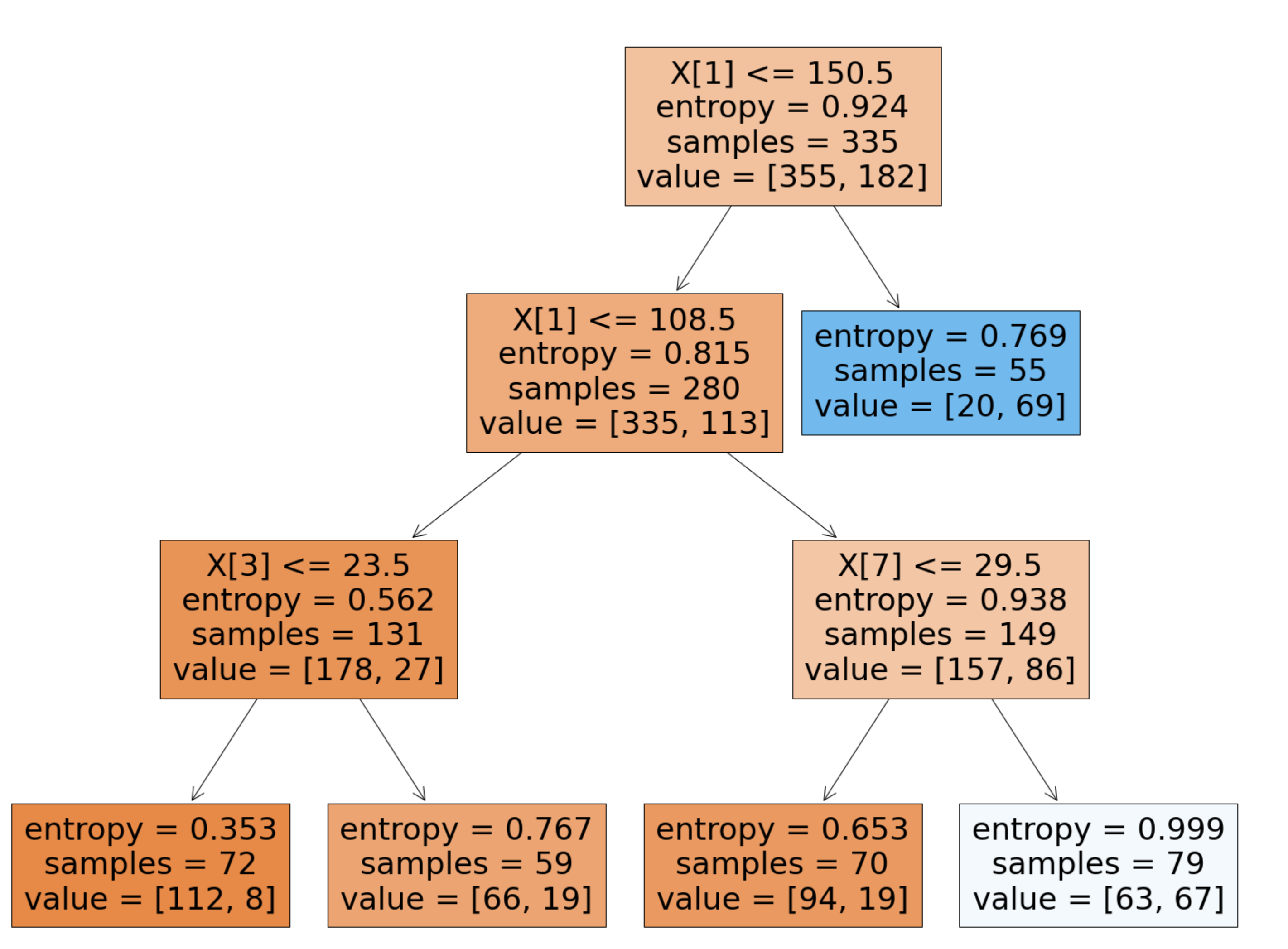
Random Forest menggunakan pruning

In [14]:
```python
from sklearn.ensemble import RandomForestClassifier

# Create Decision Tree classifier object
rf_p = RandomForestClassifier(criterion="entropy", max_depth=3, max_leaf_nodes=5, class_weight=None)

# Train Decision Tree Classifier
rf_p = rf_p.fit(X_train, y_train)

# Predict the response for test dataset
y_pred_rfp = rf_p.predict(X_test)

print("Accuracy:", metrics.accuracy_score(y_test, y_pred_rfp))
```

```
Accuracy: 0.7792207792207793
```

Menampilkan gambaran dari Random Forest dengan pruning

In [15]:
```python
fig = plt.figure(figsize=(25,20))
tree.plot_tree(rf_p.estimators_[0], filled = True);
```



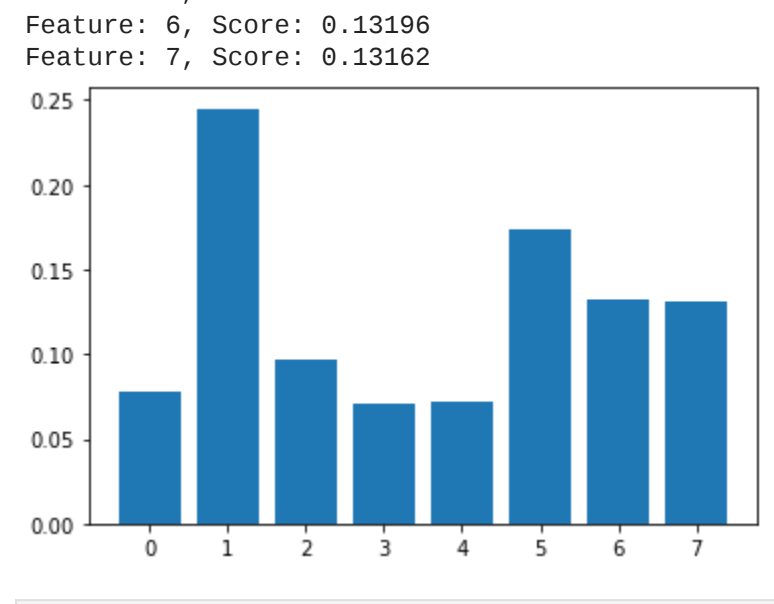
Feature Importance

Feature Importance merupakan salah satu cara tuning Random Forest dengan menentukan variable penting yang mempengaruhi hasil klasifikasi pada Random Forest.

Menggunakan feature importance

In [16]:
```python
importance = rf.feature_importances_

for i,v in enumerate(importance):
    print("Feature: %0d, Score: %.5f" % (i,v))

# menampilkan plot feature importance
plt.bar([x for x in range(len(importance))], importance)
plt.show()
```

```
Feature: 0, Score: 0.07742
Feature: 1, Score: 0.24498
Feature: 2, Score: 0.09707
Feature: 3, Score: 0.07120
Feature: 4, Score: 0.07192
Feature: 5, Score: 0.17382
Feature: 6, Score: 0.13196
Feature: 7, Score: 0.13162
```



In [17]:
```python
#split dataset in features and target variable
importance_cols = ['Pregnancies','Glucose','BloodPressure','BMI','DiabetesPedigreeFunction','Age']
X1 = dataset[importance_cols] # Features
y1 = dataset['Outcome'] # Target variable

# split dataset into training set and test set
X1_train, X1_test, y1_train, y1_test = train_test_split(X1, y1, test_size=0.3, random_state=1)
```

Hasil akurasi setelah menggunakan feature importance

In [18]:
```python
# Create Decision Tree classifier object
rf1 = RandomForestClassifier()

# Train Decision Tree Classifier
rf1 = rf1.fit(X1_train, y1_train)

# Predict the response for test dataset
y1_pred = rf1.predict(X1_test)

print("Accuracy:", metrics.accuracy_score(y1_test, y1_pred))
```

```
Accuracy: 0.7965367965367965
```

Perhatikan, apakah hasil akurasi dengan Random Forest sebelum dan sesudah menggunakan feature importance mengalami kenaikan atau penurunan?

Hasilnya akurasinya naik

In [ ]:
```python
!jupyter nbconvert --to html "./00000054804_Christopher Darren_Week10.ipynb" --output-dir="./"
```