

Nama : Ravi Mukti  
Kelas : TIF RM 17 A  
NPM : 17111247  
Mata Kuliah : Data Mining  
Dosen : Deden Komaludin, S.T., M.T.

Dengan menggunakan dataset yang kita miliki dari soal kita mampu memprediksi apakah seorang siswa itu layak atau tidak untuk mendapatkan beasiswa berdasarkan data training yang sudah kita miliki, dan kita olah dengan menggunakan algoritma decision tree yang disediakan oleh library scikit learn. Dengan hal ini tentu mempermudah bagi kita dalam mengambil sebuah keputusan.

Berikut adalah screenshot kode program dan juga hasilnya menggunakan Jupyter Notebook. Adapun untuk lebih lengkapnya bisa mengunjungi link project aplikasi di <https://github.com/raviMukti/simple-decision-tree>

```
In [14]: import os
import pandas as pd

BASEDIR = os.getcwd()
df = pd.read_csv(os.path.join(BASEDIR, '../resource/Data-Penerima-Beasiswa-Sekolah-A.csv'))
df.head()
```

```
Out[14]:
```

	nama	jk	jt	at	pka	pha	pki	phi	status_kelayakan
0	S-1	L	0.50	1.000	0.083	0.625	0.833	1.000	Tidak
1	S-2	L	0.25	0.143	0.667	0.375	0.833	1.000	Tidak
2	S-3	L	0.25	1.000	0.667	0.500	0.833	1.000	Tidak
3	S-4	L	0.25	0.286	0.500	0.250	0.833	1.000	Tidak
4	S-5	L	0.25	0.000	0.000	0.500	0.917	0.875	Tidak

```
In [17]: # menghapus kolom status_kelayakan dan nama
inputs = df.drop('status_kelayakan', axis='columns')
# menentukan target hasil
target = df['status_kelayakan']

# karena library hanya mampu menganalisa angka
# maka kita perlu mengubah kolom jk menjadi angka
# kita ubah menggunakan labelencoder

from sklearn.preprocessing import LabelEncoder

le_jk = LabelEncoder()
inputs['jk_n'] = le_jk.fit_transform(inputs['jk'])
print()

print("### 5 Data Teratas ###")
print(inputs.head())
print("#####")
```

```
### 5 Data Teratas ###
```

	nama	jk	jt	at	pka	pha	pki	phi	jk_n
0	S-1	L	0.50	1.000	0.083	0.625	0.833	1.000	0
1	S-2	L	0.25	0.143	0.667	0.375	0.833	1.000	0
2	S-3	L	0.25	1.000	0.667	0.500	0.833	1.000	0
3	S-4	L	0.25	0.286	0.500	0.250	0.833	1.000	0
4	S-5	L	0.25	0.000	0.000	0.500	0.917	0.875	0

```
#####
```

```
In [18]: inputs_n = inputs.drop(['nama', 'jk'], axis='columns')
inputs_n
```

```
Out[18]:
```

	jt	at	pka	pha	pki	phi	jk_n
0	0.50	1.000	0.083	0.625	0.833	1.000	0

	jt	at	pka	pha	pki	phi	jk_n
1	0.25	0.143	0.667	0.375	0.833	1.000	0
2	0.25	1.000	0.667	0.500	0.833	1.000	0
3	0.25	0.286	0.500	0.250	0.833	1.000	0
4	0.25	0.000	0.000	0.500	0.917	0.875	0
5	0.25	1.000	0.333	0.500	0.083	0.875	1
6	0.25	0.286	0.667	0.250	0.833	1.000	0
7	0.25	0.000	0.083	0.500	0.833	1.000	0
8	0.25	0.286	0.917	0.375	0.833	1.000	0
9	0.25	1.000	0.083	0.625	0.667	0.625	1
10	0.25	0.571	0.083	0.625	0.667	0.625	1
11	0.25	0.143	0.917	0.500	0.500	1.000	0
12	0.25	1.000	0.667	0.750	0.667	0.625	1
13	0.25	1.000	0.917	0.500	0.833	1.000	1
14	0.25	1.000	0.083	0.625	0.833	1.000	0
15	0.25	1.000	1.000	0.625	1.000	0.500	0
16	0.25	0.000	0.917	0.500	0.833	1.000	0
17	0.25	1.000	0.333	0.500	0.833	1.000	0
18	0.25	1.000	0.917	0.500	0.833	1.000	0
19	0.25	1.000	0.083	0.875	0.833	1.000	1
20	0.25	1.000	0.917	0.500	0.833	1.000	0

```
In [20]: from sklearn import tree

# membuat model decision tree
model = tree.DecisionTreeClassifier()
# kemudian kita train model dengan data inputs_n yg sudah bernilai angka semua
model.fit(inputs_n, target)
# kemudian kita akan menentukan score / nilai pengaruh prediksi jika 1 maka akan sangat akurat
model.score(inputs_n, target)
```

Out[20]: 1.0

```
In [25]: # untuk memprediksi tidak layak, kita akan masukan nilai yg sudah ada pada data training, harusnya menghasilkan tidak layak
model.predict([[0.25, 0, 0.917, 0.5, 0.833, 1, 0]])
```

Out[25]: array(['Tidak'], dtype=object)

```
In [26]: # untuk memprediksi tidak layak, kita akan masukan nilai yg sudah ada pada data training, harusnya menghasilkan layak
model.predict([[0.25, 1, 0.083, 0.875, 0.833, 1, 1]])
```

```
Out[26]: array(['Ya'], dtype=object)
```

