

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsional Layanan Teknologi Informasi

DATA MINING FOR SMART CITY

Laporan ini merupakan bagian dari Assignment EL5224 – IT Service Operational Management ITB2015. Source code dan laporan juga tersedia di link github [berikut](#).

1. Menentukan Model untuk Sistem Penunjang Keputusan (SPK) penerima beasiswa.

Tabel 1. Data Mahasiswa

IPK	Tingkat Kemiskinan	Perolehan Beasiswa
0.8	0.8	Tidak Dapat
1.2	1	Tidak Dapat
1	1.2	Tidak Dapat
2	1.5	Tidak Dapat
0.7	1.5	Tidak Dapat
2.5	3	Dapat
3	2.5	Dapat
3	2	Dapat
3	3	Dapat
3.5	3.9	Dapat
2.5	3.5	Dapat

- a) Tentukan komponen-komponen utama dari proses klasifikasi, antara lain :
- Kelas, atau biasa disebut target, adalah variabel dependent. Dalam hal ini, variabel kelasnya : [Perolehan Beasiswa](#).
 - Prediktor, atau biasa disebut features, adalah sekumpulan variabel independent. Dalam hal ini, variabel prediktornya : [IPK dan Tingkat Kemiskinan](#).
 - Set data latih. Beberapa artikel mengatakan, perbandingan terbaik untuk data latih : data uji adalah 70:30, artinya 70% data yang tersedia untuk data latih, sementara sisanya 30% untuk data uji. Secara random, 70% data latih adalah sebagai berikut :

Tabel 2. data latih

IPK	Tingkat Kemiskinan	Perolehan Beasiswa
1	1.2	1 Tidak Dapat

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

5	2.5	3	Dapat
6	3	2.5	Dapat
0	0.8	0.8	Tidak Dapat
3	2	1.5	Tidak Dapat
1	2.5	3.5	Dapat
0			
8	3	3	Dapat

- iv. Set data uji. Data uji diambil sebanyak 30% dari data yang tidak beririsan dengan data latih :

Tabel 3. data uji

	IPK	Tingkat Kemiskinan	Perolehan Beasiswa
7	3	2	Dapat
4	0.7	1.5	Tidak Dapat
9	3.5	3.9	Dapat
2	1	1.2	Tidak Dapat

- b) Tentukan metode yang akan digunakan : [Decision Tree](#).
c) Jelaskan tahapan-tahapan sesuai model yang digunakan, dan tuliskan aturan-aturan yang diperoleh.

Decision Tree adalah sebuah metode yang menggunakan grafik mirip pohon, yang terdiri dari beberapa cabang dalam berbagai tingkat. Path yang berawal dari akar sampai ke cabang merupakan *classification rules*. Terdapat banyak tools atau software yang mendukung decision tree, seperti Rapid Miner dan Weka, tetapi agar rules yang dijalankan dapat di-debug, untuk itu digunakan **Python** dan **Scikit-learn**. Scikit-learn adalah sebuah library yang ditulis dalam bahasa Python (beserta C dan C++) yang memiliki fitur klasifikasi, regresi, dan clustering. Decision Tree Algorithm yang digunakan oleh Scikit-learn adalah **Optimized CART**. Classification and regression trees (CART) adalah sebuah teknik *non-parametric decision tree learning* yang menghasilkan pohon klasifikasi atau regresi, tergantung variabel dependent-nya kategorik atau numerik. Berikut tahapan Decision Tree dengan menggunakan Scikit-learn :

- i. Install dan import library yang dibutuhkan :
1. Numpy, untuk menghandle array dan matrix, serta operasi matematika

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsional Layanan Teknologi Informasi

- untuk memanipulasi matrix tersebut,
2. Pandas, untuk manipulasi data dan analisis,
 3. Sklearn (Scikit-learn), untuk klasifikasi, regresi, dan clustering,
 4. PyGraphviz dan Pydot, untuk plotting hasil yang diperoleh,
 5. Jupyter dan Ipython, sebagai environment pengolahan.

```
# Import necessary library
from __future__ import print_function

import os
import subprocess

from IPython.display import Image, display

from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.naive_bayes import GaussianNB
from sklearn.cross_validation import train_test_split
from sklearn.externals.six import StringIO

import pandas as pd
import numpy as np

import statistics
import pydot
```

- ii. Baca file csv, dan definisikan kolom-kolom yang akan menjadi target dan features :

```
file_csv = "1_beasiswa.csv"

# Define class (target) and features
# Column 0 = IPK
# Column 1 = Tingkat Kemiskinan
target_column = "Perolehan Beasiswa"
feature_columns = [0,1]

# Read data from csv
file_csv = file_csv.replace(".csv", "")
df = pd.read_csv(file_csv + ".csv")
```

- iii. Encode (Kategorikan dalam bentuk integer) data pada kolom target, sehingga values ["Dapat", "Tidak"] akan diencode menjadi [0, 1]

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

```
# Function to encode target into integer
def encode_target(df, target_column, encoded_target_column_name):
    df_mod = df.copy()
    targets = df_mod[target_column].unique()
    map_to_int = {name: n for n, name in enumerate(targets)}
    df_mod[encoded_target_column_name] =
df_mod[target_column].replace(map_to_int)

    return (df_mod, targets)
```

```
# Encode target (class) as unique integer
df, targets = encode_target(df, target_column, target_column + "
Encoded")
```

- iv. Pilih kolom feature, berdasarkan index kolom yang didefinisikan pada poin (ii) :

```
# Select features (predictors)
features = list(df.columns[feature_columns])
```

- v. Bagi data menjadi data latih dan data uji dengan komposisi latih:uji = 70:30 :

```
# Create data train and test
# Data test size = 30% of all data
train, test = train_test_split(df, test_size = 0.3)
```

- vi. Debug untuk mengetahui baris data yg menjadi data latih dan data uji :

```
# Print data train and data test
print("Data latih iterasi ke-%s" % iter)
print("-----")
print(train)
print("\r\n")
print("Data uji iterasi ke-%s" % iter)
print("-----")
print(test)
print("\r\n")
```

Hasil yang diperoleh :

```
Data latih iterasi ke-1
-----
      IPK  Tingkat Kemiskinan  Perolehan Beasiswa  Perolehan Beasiswa Encoded
1    1.2                1.0      Tidak Dapat                0
5    2.5                3.0        Dapat                1
6    3.0                2.5        Dapat                1
0    0.8                0.8      Tidak Dapat                0
3    2.0                1.5      Tidak Dapat                0
10   2.5                3.5        Dapat                1
8    3.0                3.0        Dapat                1
```

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

Data uji iterasi ke-1

	IPK	Tingkat Kemiskinan	Perolehan Beasiswa	Perolehan Beasiswa Encoded
7	3.0	2.0	Dapat	1
4	0.7	1.5	Tidak Dapat	0
9	3.5	3.9	Dapat	1
2	1.0	1.2	Tidak Dapat	0

vii. Gunakan data training untuk membentuk model :

```
# Use train data to construct model
# Define target and features column
y = train[target_column + " Encoded"]
X = train[features]
```

viii. Bentuk modelnya dengan kriteria “gini”. Perbedaan antara gini dan entropy :

1. Gini digunakan untuk atribut yang bersifat kontinyu, sementara
2. Entropy digunakan untuk atribut yang bersifat kategorik

```
# Construct model
# criterion = gini -> features are continues
# criterion = entropy -> features are categorical
# dt = GaussianNB()
dt = DecisionTreeClassifier(criterion="gini")
dt.fit(X, y)
```

ix. Print model yang diperoleh :

```
def get_code(tree, feature_names, target_names,
            spacer_base="  "):
    left      = tree.tree_.children_left
    right     = tree.tree_.children_right
    threshold = tree.tree_.threshold
    features  = [feature_names[i] for i in tree.tree_.feature]
    value     = tree.tree_.value

    def recurse(left, right, threshold, features, node, depth):
        spacer = spacer_base * depth
        if (threshold[node] != -2):
            print(spacer + "if " + features[node] + " <= " +
                  str(threshold[node]) + " :")
            if left[node] != -1:
                recurse(left, right, threshold, features,
                        left[node], depth+1)
            print(spacer + "else :")
            if right[node] != -1:
                recurse(left, right, threshold, features,
                        right[node], depth+1)
        else:
            target = value[node]
            for i, v in zip(np.nonzero(target)[1],
```

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

```
                target[np.nonzero(target)]:  
                target_name = target_names[i]  
                target_count = int(v)  
                print(spacer + "return " + str(target_name) + \  
                      " (count=" + str(target_count) + ")")  
  
recurse(left, right, threshold, features, 0, 0)
```

```
# Get code for model  
print("Code iterasi ke-%s" % iter)  
print("-----")  
get_code(dt, features, targets)  
print("\r\n")
```

Hasil yang diperoleh :

```
Code iterasi ke-1  
-----  
if IPK <= 2.25 :  
    return Tidak Dapat (count=3)  
else :  
    return Dapat (count=4)
```

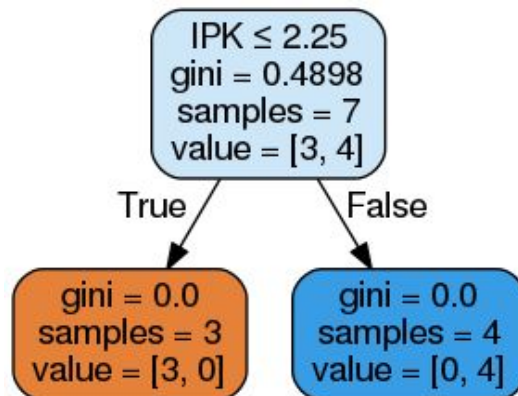
x. Tampilkan grafik tree yang terbentuk :

```
def create_tree(dt, features):  
    dot_data = StringIO()  
    export_graphviz(dt, out_file=dot_data,  
                    feature_names=features,  
                    filled=True, rounded=True,  
                    special_characters=True)  
    graph = pydot.graph_from_dot_data(dot_data.getvalue())  
    return graph.create_png()
```

```
# Construct tree image  
print("Grafik Tree iterasi ke-%s" % iter)  
print("-----")  
tree_image = create_tree(dt, features)  
display(Image(tree_image))  
print("\r\n")
```

Hasil yang diperoleh :

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi



Dari model tersebut dapat disimpulkan, penentuan apakah seorang mahasiswa mendapatkan beasiswa hanya tergantung pada variabel IPK, dimana jika IPK kurang dari sama dengan 2,25 maka mendapat beasiswa, dan jika lebih dari 2,25 tidak mendapat beasiswa.

d) Gunakan data uji untuk mengukur akurasi model :

```
# Check accuracy of the model using test data
y = test[target_column + " Encoded"]
X = test[features]
accuracy = dt.score(X,y)
accuracies.append(accuracy)
```

```
# Print accuracy
print("Akurasi iterasi ke-%s : %s" % (iter, accuracy))
print("-----")
print("\r\n")
```

Hasil yang diperoleh :

```
Akurasi iterasi ke-1 : 1.0
```

Lakukan pengulangan langkah (c) sebanyak 10 kali, kemudian hitung rata-rata akurasi :

```
# Print the mean of accuracies of all iterations
accuracy_mean = statistics.mean(accuracies)
print("Rata-rata akurasi dari %s kali percobaan : %s" %
      (len(range(1,11)), accuracy_mean))
```

Hasil yang diperoleh :

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsional Layanan Teknologi Informasi

Rata-rata akurasi dari 10 kali percobaan : 0.95

2. Cari permasalahan yang berkaitan dengan Smart City yang solusinya menggunakan pembelajaran mesin. Berikan penjelasan ringkas untuk poin-poin berikut :

a) Permasalahan yang ditemukan :

Pendataan **Potensi Desa (Podes)** merupakan pendataan yang dilakukan 3 (tiga) kali dalam 10 tahun. Podes merupakan satu-satunya data kewilayahan yang terlengkap dalam skala nasional. Tujuan Podes antara lain : Menyediakan data tentang keberadaan, ketersediaan dan perkembangan potensi yang dimiliki setiap wilayah administrasi pemerintahan yang meliputi: sarana dan prasarana wilayah serta potensi ekonomi, sosial, budaya, dan aspek kehidupan masyarakat lainnya untuk berbagai keperluan yang berkaitan dengan perencanaan wilayah di tingkat nasional dan daerah; Menyediakan data dasar bagi keperluan penentuan klasifikasi/tipologi wilayah (seperti: perkotaan-perdesaan, wilayah tertinggal, wilayah pesisir dan sebagainya) dan penyusunan statistik wilayah kecil; Melengkapi penyusunan kerangka sampling untuk kegiatan statistik lain lebih lanjut. Pada kasus ini, permasalahan yang akan diangkat dan diteliti dengan pembelajaran mesin adalah bagaimana model dalam penentuan **Klasifikasi Desa Pedesaan/Perkotaan**.

b) Pendekatan yang dilakukan untuk memecahkan masalah tersebut :

i. Install dan import library yang dibutuhkan :

```
# Import necessary library
from __future__ import print_function

import os
import subprocess

from IPython.display import Image, display

from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.naive_bayes import GaussianNB
from sklearn.cross_validation import train_test_split
from sklearn.externals.six import StringIO

import pandas as pd
import numpy as np

import statistics
import pydot
```

ii. Baca file csv, dan definisikan kolom-kolom yang akan menjadi target dan

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

features :

```
file_csv = "2_podes.csv"

# Define class (target) and features
# Column 0 = 1.KEPADATAN PENDUDUK
# Column 1 = 2.PERSENTASE RT PERTANIAN
# Column 2 = TK
# Column 3 = SMP
# Column 4 = SMU
# Column 5 = PASAR
# Column 6 = PERTOKOAN
# Column 7 = BIOSKOP
# Column 8 = RUMAH SAKIT
# Column 9 = HOTEL/BILYAR/DISKOTEK
# Column 10 = % RT TELEPON
# Column 11 = % RT LISRIK
target_column = "KRITERIA KOTA/DESA"
feature_columns = list(range(0,11))

# Read data from csv
file_csv = file_csv.replace(".csv", "")
df = pd.read_csv(file_csv + ".csv")
```

iii. Encode (Kategorikan dalam bentuk integer) data pada kolom target, sehingga values ["Pedesaan", "Perkotaan"] akan diencode menjadi [0, 1]

```
# Function to encode target into integer
def encode_target(df, target_column, encoded_target_column_name):
    df_mod = df.copy()
    targets = df_mod[target_column].unique()
    map_to_int = {name: n for n, name in enumerate(targets)}
    df_mod[encoded_target_column_name] =
df_mod[target_column].replace(map_to_int)

    return (df_mod, targets)
```

```
# Encode target (class) as unique integer
df, targets = encode_target(df, target_column, target_column + "
Encoded")
```

iv. Pilih kolom feature, berdasarkan index kolom yang didefinisikan pada poin (ii) :

```
# Select features (predictors)
features = list(df.columns[feature_columns])
```

v. Bagi data menjadi data latih dan data uji dengan komposisi latih:uji = 70:30 :

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

```
# Create data train and test
# Data test size = 30% of all data
train, test = train_test_split(df, test_size = 0.3)
```

vi. Debug untuk mengetahui baris data yg menjadi data latih dan data uji :

```
# Print data train and data test
print("Data latih iterasi ke-%s" % iter)
print("-----")
print(train)
print("\r\n")
print("Data uji iterasi ke-%s" % iter)
print("-----")
print(test)
print("\r\n")
```

Hasil yang diperoleh :

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsional Layanan Teknologi Informasi

Data latih iterasi ke-1

	1.KEPADATAN PENDUDUK	2.PERSENTASE RT PERTANIAN	TK	SMP	SMU	PASAR	\
54	1		3	1	0	0	0
20	1		2	1	1	0	0
46	1		2	1	0	0	0
51	1		1	1	1	1	0
24	2		7	0	0	1	1
4	1		2	1	0	0	0
55	1		5	1	1	0	0
56	1		2	1	1	0	1
60	1		1	1	1	0	0
2	1		2	1	1	1	0
19	1		1	0	0	0	0
49	1		5	1	1	0	0
8	2		1	1	1	1	1
61	1		1	1	0	0	0
30	1		3	1	1	1	1
29	1		3	0	0	0	0
45	1		2	1	0	0	0
1	3		2	1	1	1	0
17	1		1	1	1	1	0
62	1		2	1	1	0	1
36	1		5	1	1	0	0
40	1		1	1	1	1	1
43	1		1	1	1	1	0
23	1		4	1	1	0	0
39	1		2	1	0	0	0
33	1		1	1	1	0	1
22	1		3	1	1	1	1
12	1		1	1	0	0	0
28	2		2	1	1	0	1
50	1		1	1	1	0	0
18	1		1	1	1	0	0
21	2		5	1	0	0	1
9	1		4	1	1	0	1
35	1		2	1	0	0	0
38	1		2	1	0	0	0
25	2		3	1	1	1	1
53	1		3	1	0	0	0
58	1		1	1	0	0	0
10	2		7	1	1	1	1
7	1		2	1	0	0	0
32	1		2	1	1	0	1
37	1		2	1	1	1	1

Nama : Aris Prawisudatama
 NIP : 23215131
 Jurusan : Teknik Elektro
 Opsi Layanan Teknologi Informasi

16			1		3	1	0	0	1
44			1		2	1	0	0	0
	PERTOKOAN	BIOSKOP	RUMAH SAKIT	HOTEL/BILYAR/DISKOTEK	% RT	TELEPON	\		
54	0	0	0		0			0	
20	1	0	0		0			0	
46	0	0	0		0			0	
51	0	0	0		0			0	
24	1	0	1		1			0	
4	0	0	0		0			0	
55	0	0	0		0			0	
56	0	0	0		0			0	
60	1	0	0		0			0	
2	1	0	0		0			0	
19	0	0	0		0			0	
49	0	0	0		0			0	
8	1	0	1		0			0	
61	0	0	0		0			0	
30	1	0	0		0			0	
29	1	0	1		0			0	
45	0	0	0		0			0	
1	1	0	1		1			1	
17	1	0	0		0			0	
62	0	0	0		0			0	
36	0	0	0		0			0	
40	0	0	0		0			0	
43	0	0	0		0			0	
23	1	0	1		1			0	
39	0	0	0		0			0	
33	0	0	0		0			0	
22	1	0	0		0			0	
12	0	0	0		0			0	
28	1	0	0		1			0	
50	0	0	0		0			0	
18	0	0	0		0			0	
21	1	0	0		1			0	
9	1	0	1		1			0	
35	0	0	0		0			0	
38	0	0	0		0			0	
25	1	0	1		1			0	
53	0	0	0		0			0	
58	0	0	0		0			0	
10	1	0	0		0			0	
7	0	0	0		0			0	
32	0	0	0		0			0	
37	0	0	0		0			0	
16	1	0	0		1			0	
44	0	0	0		0			0	
	% RT	LISRIK	KRITERIA KOTA/DESA	KRITERIA KOTA/DESA	Encoded				
54		1	PEDESAAN					0	
20		1	PEDESAAN					0	
46		1	PEDESAAN					0	
51		1	PEDESAAN					0	
24		1	PERKOTAAN				1		
4		1	PEDESAAN				0		
55		1	PEDESAAN				0		
56		1	PEDESAAN				0		
60		1	PEDESAAN				0		
2		1	PEDESAAN				0		
19		1	PEDESAAN				0		

Nama : Aris Prawisudatama
 NIP : 23215131
 Jurusan : Teknik Elektro
 Opsi Layanan Teknologi Informasi

49	1	PEDESAAN	0
8	1	PERKOTAAN	1
61	1	PEDESAAN	0
30	1	PERKOTAAN	1
29	1	PEDESAAN	0
45	1	PEDESAAN	0
1	1	PERKOTAAN	1
17	1	PEDESAAN	0
62	1	PEDESAAN	0
36	1	PEDESAAN	0
40	1	PEDESAAN	0
43	1	PEDESAAN	0
23	1	PERKOTAAN	1
39	1	PEDESAAN	0
33	1	PEDESAAN	0
22	1	PERKOTAAN	1
12	1	PEDESAAN	0
28	1	PERKOTAAN	1
50	1	PEDESAAN	0
18	1	PEDESAAN	0
21	1	PERKOTAAN	1
9	1	PERKOTAAN	1
35	1	PEDESAAN	0
38	1	PEDESAAN	0
25	1	PERKOTAAN	1
53	1	PEDESAAN	0
58	1	PEDESAAN	0
10	1	PERKOTAAN	1
7	1	PEDESAAN	0
32	1	PEDESAAN	0
37	1	PEDESAAN	0
16	1	PEDESAAN	0
44	1	PEDESAAN	0

Data uji iterasi ke-1

	1.KEPADATAN PENDUDUK	2.PERSENTASE RT PERTANIAN	TK	SMP	SMU	PASAR	\
6	1		1	1	0	0	1
42	1		2	1	0	0	0
47	1		3	1	1	1	1
48	1		1	1	0	0	0
5	1		3	1	1	0	0
52	1		2	1	0	0	0
15	1		1	1	0	0	1
14	1		1	1	0	0	0
13	1		1	1	1	0	0
11	1		1	0	0	0	0
34	1		1	1	0	1	0
31	2		7	1	1	0	1
3	1		1	0	0	0	0
57	1		2	1	0	0	0
41	1		3	1	1	0	0
26	1		4	1	0	1	0
0	1		3	1	0	0	1
27	1		8	1	1	1	0
59	1		3	1	1	0	0

	PERTOKOAN	BIOSKOP	RUMAH SAKIT	HOTEL/BILYAR/DISKOTEK	% RT	TELEPON	\
6	0	0	0		0		0
42	0	0	0		0		0

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

47	0	0	0	0	0
48	0	0	0	0	0
5	0	0	0	0	0
52	0	0	0	0	0
15	0	0	0	0	0
14	0	0	0	0	0
13	0	0	0	1	0
11	0	0	0	0	0
34	0	0	0	0	0
31	1	0	0	1	0
3	0	0	0	0	0
57	0	0	0	0	0
41	0	0	0	1	0
26	1	0	0	1	0
0	1	0	0	0	0
27	1	0	1	1	0
59	1	0	1	0	0

	% RT LISRIK	KRITERIA	KOTA/DESA	KRITERIA	KOTA/DESA	Encoded
6	1		PEDESAAN			0
42	1		PEDESAAN			0
47	1		PEDESAAN			0
48	1		PEDESAAN			0
5	1		PEDESAAN			0
52	1		PEDESAAN			0
15	1		PEDESAAN			0
14	1		PEDESAAN			0
13	1		PEDESAAN			0
11	1		PEDESAAN			0
34	1		PEDESAAN			0
31	1		PERKOTAAN			1
3	1		PEDESAAN			0
57	1		PEDESAAN			0
41	1		PEDESAAN			0
26	1		PERKOTAAN			1
0	1		PEDESAAN			0
27	1		PERKOTAAN			1
59	1		PEDESAAN			0

vii. Gunakan data training untuk membentuk model :

```
# Use train data to construct model
# Define target and features column
y = train[target_column + " Encoded"]
X = train[features]
```

viii. Bentuk modelnya dengan kriteria “entropy”, karena atribut yang digunakan bersifat kategorik.

```
# Construct model
# criterion = gini -> features are continues
# criterion = entropy -> features are categorical
# dt = GaussianNB()
dt = DecisionTreeClassifier(criterion="entropy")
dt.fit(X, y)
```

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

ix. Print model yang diperoleh :

```
def get_code(tree, feature_names, target_names,
            spacer_base="    "):
    left      = tree.tree_.children_left
    right     = tree.tree_.children_right
    threshold = tree.tree_.threshold
    features  = [feature_names[i] for i in tree.tree_.feature]
    value     = tree.tree_.value

    def recurse(left, right, threshold, features, node, depth):
        spacer = spacer_base * depth
        if (threshold[node] != -2):
            print(spacer + "if " + features[node] + " <= " +
                  str(threshold[node]) + " :")
            if left[node] != -1:
                recurse(left, right, threshold, features,
                        left[node], depth+1)
            print(spacer + "else :")
            if right[node] != -1:
                recurse(left, right, threshold, features,
                        right[node], depth+1)
        else:
            target = value[node]
            for i, v in zip(np.nonzero(target)[1],
                           target[np.nonzero(target)]):
                target_name = target_names[i]
                target_count = int(v)
                print(spacer + "return " + str(target_name) + \
                      " (count=" + str(target_count) + ")")

    recurse(left, right, threshold, features, 0, 0)
```

```
# Get code for model
print("Code iterasi ke-%s" % iter)
print("-----")
get_code(dt, features, targets)
print("\r\n")
```

Hasil yang diperoleh :

```
if 1.KEPADATAN PENDUDUK <= 1.5 :
    if 2.PERSentase RT PERTANIAN <= 2.5 :
        return PEDESAAN (count=26)
    else :
        if 2.PERSentase RT PERTANIAN <= 4.5 :
            if 2.PERSentase RT PERTANIAN <= 3.5 :
                return PEDESAAN (count=4)
                return PERKOTAAN (count=2)
            else :
                return PERKOTAAN (count=2)
        else :
            return PEDESAAN (count=3)
```

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

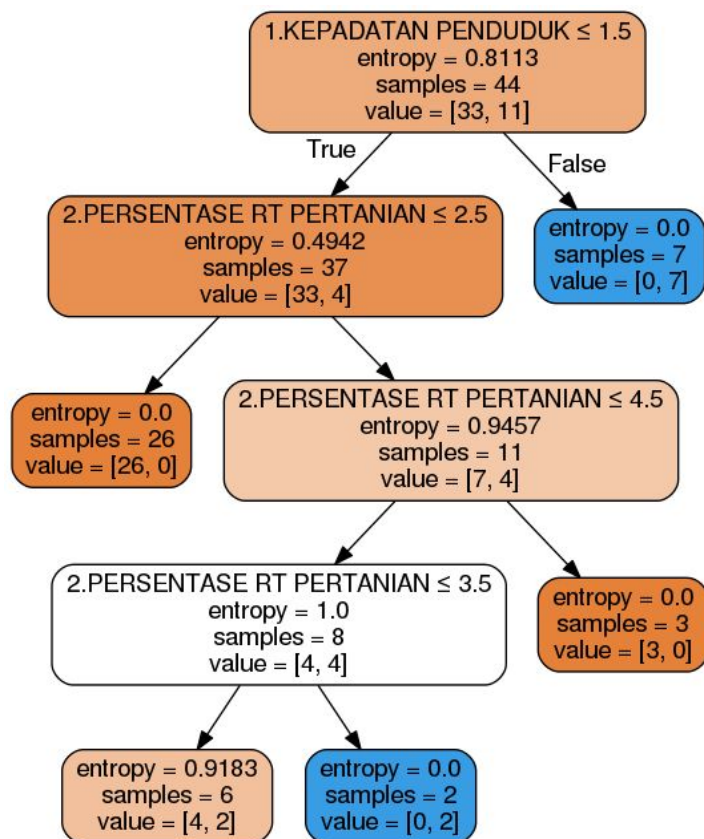
```
else :  
    return PERKOTAAN (count=7)
```

x. Tampilkan grafik tree yang terbentuk :

```
def create_tree(dt, features):  
    dot_data = StringIO()  
    export_graphviz(dt, out_file=dot_data,  
                    feature_names=features,  
                    filled=True, rounded=True,  
                    special_characters=True)  
    graph = pydot.graph_from_dot_data(dot_data.getvalue())  
    return graph.create_png()
```

```
# Construct tree image  
print("Grafik Tree iterasi ke-%s" % iter)  
print("-----")  
tree_image = create_tree(dt, features)  
display(Image(tree_image))  
print("\r\n")
```

Hasil yang diperoleh :



Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

xi. Gunakan data uji untuk mengukur akurasi model :

```
# Check accuracy of the model using test data
y = test[target_column + " Encoded"]
X = test[features]
accuracy = dt.score(X,y)
accuracies.append(accuracy)
```

```
# Print accuracy
print("Akurasi iterasi ke-%s : %s" % (iter, accuracy))
print("-----")
print("\r\n")
```

Hasil yang diperoleh :

```
Akurasi iterasi ke-1 : 0.947368421053
```

Lakukan pengulangan langkah (c) sebanyak 10 kali, kemudian hitung rata-rata akurasinya :

```
# Print the mean of accuracies of all iterations
accuracy_mean = statistics.mean(accuracies)
print("Rata-rata akurasi dari %s kali percobaan : %s" %
      (len(range(1,11)), accuracy_mean))
```

Hasil yang diperoleh :

```
Rata-rata akurasi dari 10 kali percobaan : 0.921052631579
```

- c) Penjelasan keterkaitan data seperti sumber data, jumlah data, dll :
Data dari Pendataan Potensi Desa (Podes) tidak tersedia setiap tahun, karena Podes hanya dilaksanakan 3 kali dalam 10 tahun.