

# Tugas

## Kecerdasan Buatan 1



Dzul Jalali Wal Ikram

1194011

Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering

*Politeknik Pos Indonesia*

Bandung 2022

‘Jika Kamu tidak dapat menahan lelahnya belajar,  
Maka kamu harus sanggup menahan perihnya Kebodohan.’  
Imam Syafi’i

## **Acknowledgements**

Pertama-tama kami panjatkan puji dan syukur kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga Buku Pedoman Tingkat Akhir ini dapat diselesaikan.

## **Abstract**

Buku Pedoman ini dibuat dengan tujuan memberikan acuan, bagi mahasiswa Tingkat Akhir dan dosen Pembimbing. Pada intinya buku ini menjelaskan secara lengkap tentang Standar pengerjaan Intership dan Tugas Akhir di Program Studi D4 Teknik Informatika, dan juga mengatur mekanisme, teknik penulisan, serta penilaiannya. Dengan demikian diharapkan semua pihak yang terlibat dalam aktivitas Bimbingan Mahasiswa Tingkat Akhir berjalan lancar dan sesuai dengan standar.

# Contents

<b>1</b>	<b>Mengenal Kecerdasan Buatan dan Scikit-Learn</b>	<b>1</b>
1.1	Teori . . . . .	1
1.2	Instalasi . . . . .	2
1.3	Penanganan Error . . . . .	5
<b>2</b>	<b>Membangun Model Prediksi</b>	<b>7</b>
2.1	Teori . . . . .	7
2.2	scikit-learn . . . . .	10
2.3	Penanganan Error . . . . .	18

# List of Figures

# Chapter 1

## Mengenal Kecerdasan Buatan dan Scikit-Learn

### 1.1 Teori

Praktek teori penunjang yang dikerjakan :

1. Sejarah Perkembangan dan penjelasan Definisi *Artifical Intelligence*. *Artifical Intelligence* merupakan suatu ilmu pada bidang komputer yang memiliki kemampuan untuk meniru, melakukan sesuatu sama seperti yang dilakukan oleh manusia.

Pada akhir tahun 1955 program *Artifical Intelligence* pertama kali muncul berkat adanya perkembangan *The Logic Theorist* oleh Newell dan Simon. Program tersebut menunjukkan masalah dengan menggunakan model pohon, dimana untuk menyelesaikannya hanya perlu memilih sebuah cabang yang akan menghasilkan sebuah kesimpulan yang paling benar. Dengan adanya program tersebut pengembangan dibidang *Artifical Intelligence* mengalami peningkatan yang besar. Di tahun 1956, seseorang bernama John McCarthy dari Massacuhetts Institute of Technology yang juga dikenal sebagai Bapak AI, menyelenggarakan sebuah konferensi dengan nama *The Dartmouth summer research project on artificial intelligence*, konferensi itu diselenggarakan untuk menarik minat para ahli komputer untuk berkumpul dan bertemu. Konferensi tersebut mempertemukan para pendiri dibidang AI dan bertugas untuk meletakkan dasar untuk penelitian dan pengembangan *Artifical Intelligence* dimasa depan.

2. Definisi supervised learning, klasifikasi, regresi dan unsupervised learning. Data set, training set dan testing set

- Supervised Learning

*Supervised learning* adalah suatu pendekatan machine learning yang ditentukan berdasarkan penggunaan dataset, supervised learning menggunakan dataset berlabel atau labeled dataset. Supervised Learning digunakan untuk melakukan klasifikasi data atau memprediksi hasil secara akurat sesuai dengan output berdasarkan pola yang ada didalam data training dan berupa data yang memiliki label yang sudah ditentukan terlebih dahulu

- Unsupervised Learning

*Unsupervised Learning* adalah pendekatan machine learning yang digunakan untuk menganalisa dan juga mengelompokkan kumpulan - kumpulan data yang tidak berlabel.

- Klasifikasi

*Klasifikasi* adalah sebuah proses menggunakan algoritma untuk secara akurat memasukan data kedalam kategori yang spesifik.

- Regresi

*Regresi* adalah sebuah proses menggunakan algoritma untuk memahami hubungan antara 2 variabel, yaitu variabel dependen dan variabel independen, Regresi dapat memprediksi nilai numerik variabel dependen berdasarkan variabel independen.

- Dataset

*Dataset* adalah suatu kumpulan data yang berisi informasi-informasi lama, dan dapat dikelola sehingga menjadi sebuah informasi baru.

- Training set

*Training set* adalah bagian dari dataset yang dilatih untuk kemudian digunakan untuk memprediksi sesuatu atau menjalankan fungsi dari algoritma.

- Testing set

*Testing set* adalah bagian dari dataset yang digunakan untuk melihat tingkat keakuratan dan performa dari algoritma.

## 1.2 Instalasi

1. Melakukan instalasi pada anaconda prompt dengan perintah " pip install -U scikit-learn".



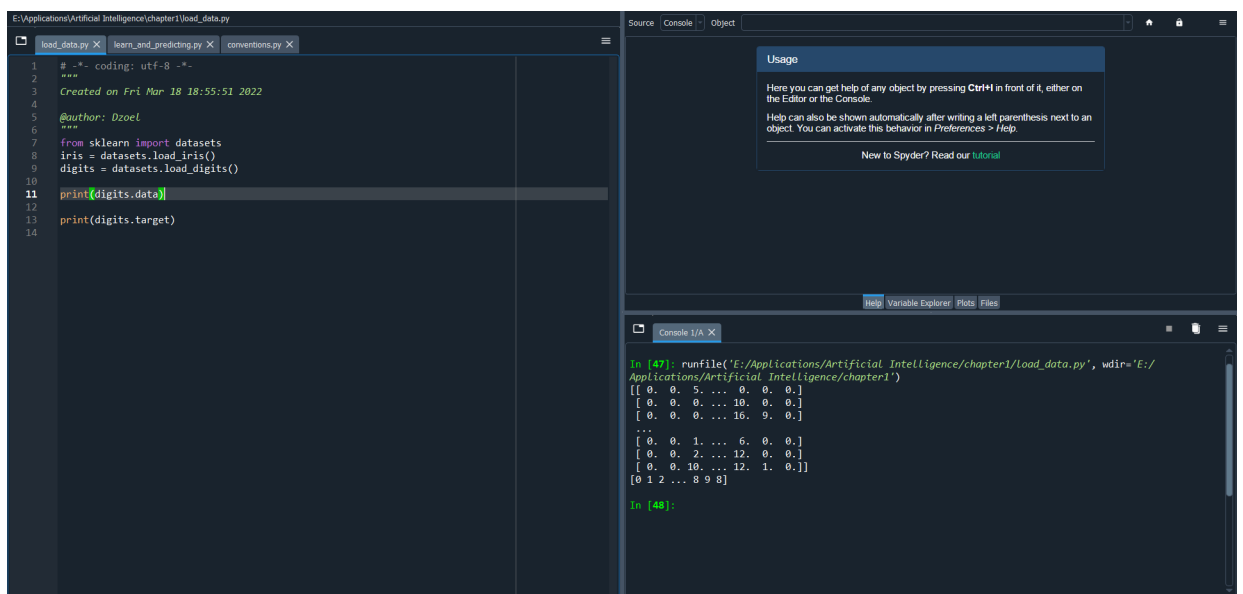
```
Anaconda Prompt (anaconda3)

(base) C:\Users\Dzoel>pip install -U scikit-learn
Requirement already satisfied: scikit-learn in c:\users\dzoel\anaconda3\lib\site-packages (0.24.2)
Collecting scikit-learn
  Downloading scikit_learn-1.0.2-cp39-cp39-win_amd64.whl (7.2 MB)
    | 7.2 MB 242 KB/s
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\dzoel\anaconda3\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: scipy>=1.1.0 in c:\users\dzoel\anaconda3\lib\site-packages (from scikit-learn) (1.7.1)
Requirement already satisfied: numpy>=1.14.6 in c:\users\dzoel\anaconda3\lib\site-packages (from scikit-learn) (1.20.3)
Requirement already satisfied: joblib>=0.11 in c:\users\dzoel\anaconda3\lib\site-packages (from scikit-learn) (1.1.0)
Installing collected packages: scikit-learn
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 0.24.2
    Uninstalling scikit-learn-0.24.2:
      Successfully uninstalled scikit-learn-0.24.2
Successfully installed scikit-learn-1.0.2

(base) C:\Users\Dzoel>
```

2. Kemudian klik link berikut ini untuk melakukan basic tutorial *scikit-learn* ”<https://scikit-learn.org/stable/tutorial/basic/tutorial.html>”.

3. Mencoba Loading an example dataset



```
E:\Applications\Artificial Intelligence\chapter1\load_data.py
load_data.py X learn_and_predicting.py X conventions.py X

1 # -*- coding: utf-8 -*-
2 """
3 Created on Fri Mar 18 18:55:51 2022
4
5 @author: Dzoel
6 """
7 from sklearn import datasets
8 iris = datasets.load_iris()
9 digits = datasets.load_digits()
10
11 print(digits.data)
12
13
14 print(digits.target)
```

Usage

Here you can get help of any object by pressing **Ctrl+H** in front of it, either on the Editor or the Console.

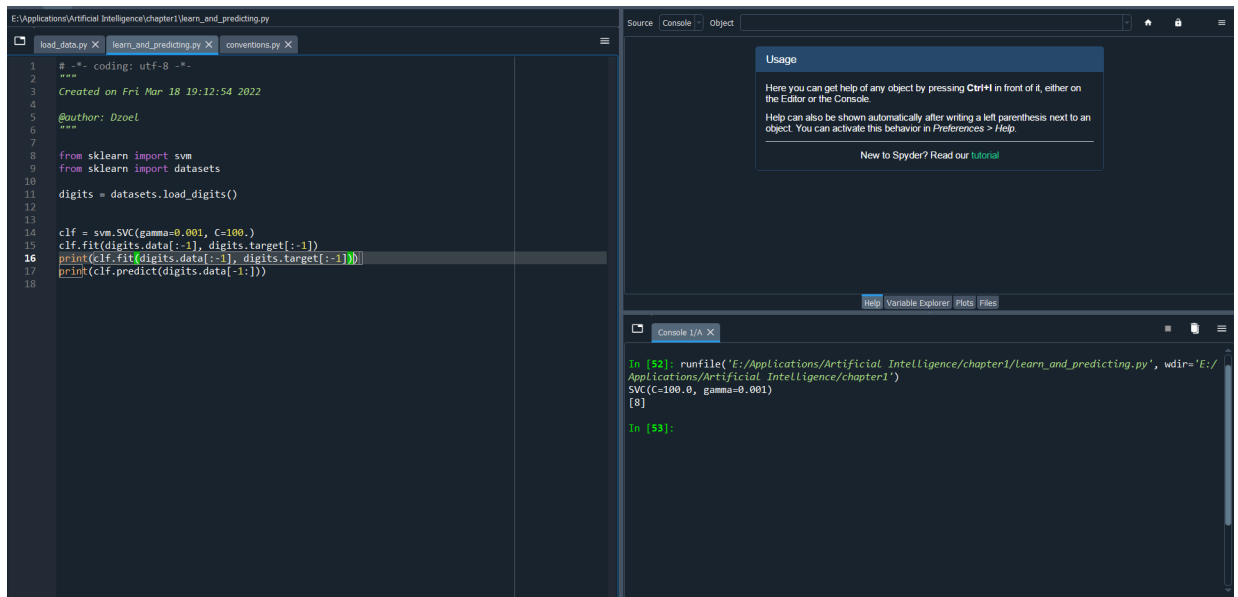
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in **Preferences > Help**.

New to Spyder? Read our [tutorial](#)

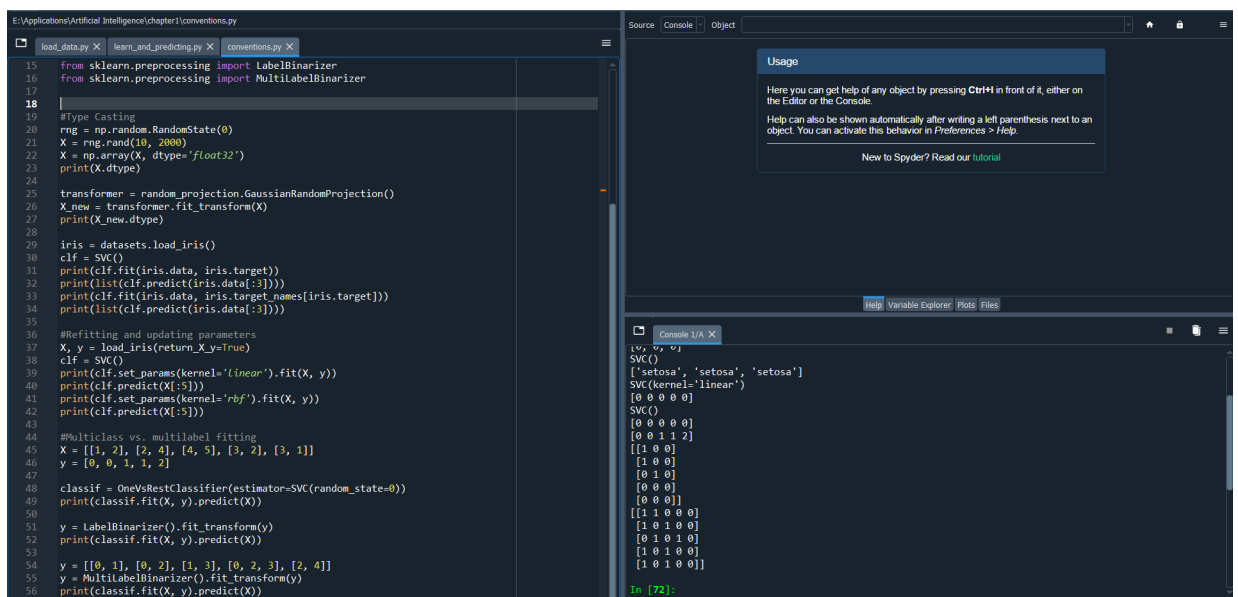
```
In [47]: runfile('E:/Applications/Artificial Intelligence/chapter1/load_data.py', wdir='E:/Applications/Artificial Intelligence/chapter1')
[[ 0.  0.  5. ...  0.  0.  0.]
 [ 0.  0.  0. ... 10.  0.  0.]
 [ 0.  0.  0. ... 16.  9.  0.]
 ...
 [ 0.  0.  1. ...  6.  0.  0.]
 [ 0.  0.  2. ... 12.  0.  0.]
 [ 0.  0. 10. ... 12.  1.  0.]]
[0 1 2 ... 8 9 8]
```

```
In [48]:
```

4. Mencoba Learning and predicting



## 5. Mencoba Conventions

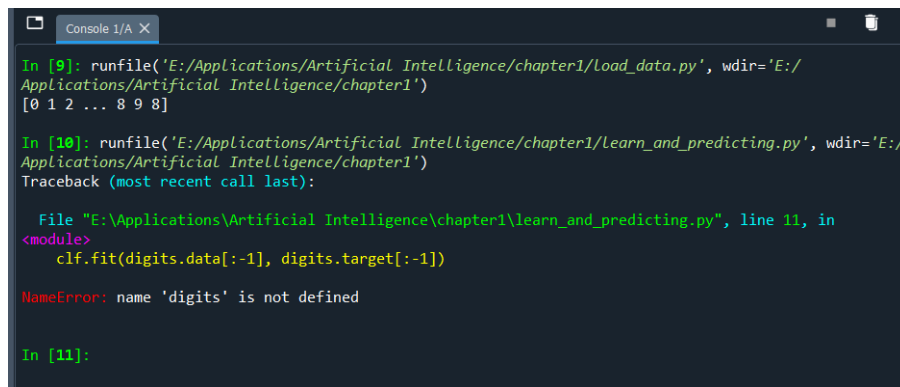


6. Link Youtube praktikum : <https://www.youtube.com/watch?v=tFsRdpQ-VUg>

## 1.3 Penanganan Error

Dari percobaan yang dilakukan di atas, apabila mendapatkan error maka:

### 1. Screenshoot Error



```
Console 1/A X
In [9]: runfile('E:/Applications/Artificial Intelligence/chapter1/load_data.py', wdir='E:/
Applications/Artificial Intelligence/chapter1')
[0 1 2 ... 8 9 8]

In [10]: runfile('E:/Applications/Artificial Intelligence/chapter1/learn_and_predicting.py', wdir='E:/
Applications/Artificial Intelligence/chapter1')
Traceback (most recent call last):

  File "E:\Applications\Artificial Intelligence\chapter1\learn_and_predicting.py", line 11, in
<module>
    clf.fit(digits.data[:-1], digits.target[:-1])

NameError: name 'digits' is not defined

In [11]:
```



```
Console 1/A X

  File "E:\Applications\Artificial Intelligence\chapter1\learn_and_predicting.py", line 10, in
<module>
    clf.predict(digits.data[:-1])

  File "C:\Users\Dzoel\anaconda3\lib\site-packages\sklearn\base.py", line 778, in predict
    check_is_fitted(self)

  File "C:\Users\Dzoel\anaconda3\lib\site-packages\sklearn\utils\validation.py", line 1222, in
check_is_fitted
    raise NotFittedError(msg % {"name": type(estimator).__name__})

NotFittedError: This SVC instance is not fitted yet. Call 'fit' with appropriate arguments before
using this estimator.

In [15]:
```

2. Tuliskan kode error dan jenis error

NameError = name 'digits' is not defined

NotFittedError = This SVC instance is not fitted yet. Call 'fit' with appropriate arguments before using this estimator.

3. Solusi pemecahan masalah error tersebut

NameError = solusinya adalah membuat variabel dengan nama digits

NotFittedError = solusinya adalah memanggil parameter dengan method fit, sebelum menggunakan method predict.

# Chapter 2

## Membangun Model Prediksi

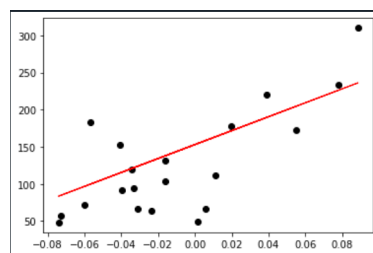
### 2.1 Teori

Praktek teori penunjang yang dikerjakan(nilai 5 per nomor, untuk hari pertama) :

1. Jelaskan apa itu binary classification dilengkapi ilustrasi gambar sendiri  
*Binary Classification* adalah mengklasifikasikan elemen - elemen himpunan menjadi 2 kelompok berdasarkan aturan klasifikasi.
2. Jelaskan apa itu supervised learning dan unsupervised learning dan clustering dengan ilustrasi gambar sendiri.

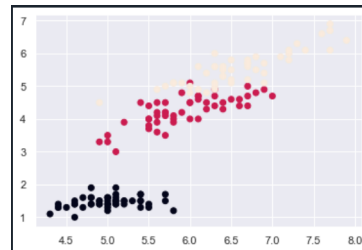
- Supervised Learning

*Supervised Learning* adalah suatu pendekatan machine learning yang ditentukan berdasarkan penggunaan dataset, supervised learning menggunakan dataset berlabel atau labeled dataset. Supervised Learning digunakan untuk melakukan klasifikasi data atau memprediksi hasil secara akurat sesuai dengan output berdasarkan pola yang ada didalam data training dan berupa data yang memiliki label yang sudah ditentukan terlebih dahulu.



- Unsupervised Learning

*Unsupervised Learning* adalah pendekatan machine learning yang digunakan untuk menganalisa dan juga mengelompokkan kumpulan - kumpulan data yang tidak berlabel.



- Clustering

*Clustering* adalah sebuah proses pengelompokan data kedalam beberapa cluster sehingga data-data disuatu cluster memiliki kemiripan.



- Jelaskan apa itu evaluasi dan akurasi dari buku dan disertai ilustrasi contoh dengan gambar sendiri
- *Evaluasi dan Akurasi* Evaluasi adalah bagaimana cara agar dapat mengevaluasi seberapa baiknya model bekerja dengan cara mengukur akurasi. Akurasi memiliki definisi sebagai persentase kasus yang diklasifikasikan dengan benar.  
Kesalahan dari model dapat dianalisis menggunakan confusion matrix.
- Jelaskan bagaimana cara membuat dan membaca confusion matrix, buat confusion matrix buatan sendiri.
- *Confusion Matrix*
  1. Import dependencies yang dibutuhkan.

2. Kemudian buat variabel untuk menampung value atau Nilai yang diprediksi.
3. Buat variabel untuk menampung value atau nilai sebenarnya.
4. Kemudian print confusion matrixnya dengan menggunakan variabel dari nilai sebenarnya dan variabel nilai yang diprediksi dengan label misalnya a,b,c.
5. Kemudian print precision dan recall diantara matrix lainnya.

```
from sklearn import metrics

y_pred = ["a", "b", "c", "a", "b"]
y_act = ["a", "b", "c", "c", "a"]

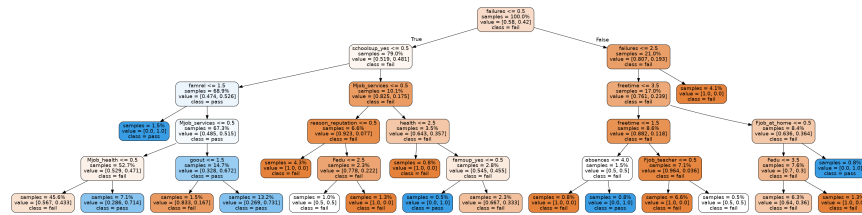
print(metrics.confusion_matrix(y_act, y_pred, labels=["a", "b", "c"]))

print(metrics.classification_report(y_act, y_pred, labels=["a", "b", "c"]))
```

```
In [1]: runfile('E:/Applications/Artificial Intelligence/chapter2/confusion_matrix.py', wdir='E:/Applications/Artificial Intelligence/chapter2')
[[1 1 0]
 [0 1 0]
 [1 0 1]]
```

	precision	recall	f1-score	support
a	0.50	0.50	0.50	2
b	0.50	1.00	0.67	1
c	1.00	0.50	0.67	2
accuracy			0.60	5
macro avg	0.67	0.67	0.61	5
weighted avg	0.70	0.60	0.60	5

- Jelaskan bagaimana K-fold cross validation bekerja dengan gambar ilustrasi contoh buatan sendiri.
- 1. Mengacak dataset secara random.
  2. Membagi dataset tersebut kedalam k-group, yaitu sebagai test dataset dan sisanya sebagai training dataset.
  3. Pasang model di set pelatihan dan evaluasi di set tes.
  4. Simpan skor evaluasi dan buang modelnya.
  5. Meringkas keterampilan model menggunakan sampel skor menggunakan model evaluasi.
- Jelaskan apa itu decision tree dengan gambar ilustrasi contoh buatan sendiri.
- *Decision Tree* adalah metode yang digunakan untuk membantu membuat keputusan dengan menggunakan model pohon untuk menampilkan keputusan, kemungkinan resiko, dan lainnya.



- Jelaskan apa itu information gain dan entropi dengan gambar ilustrasi buatan sendiri.
- *Information Gain* adalah sekumpulan informasi yang didapatkan dari variabel acak.  
*Entropi* adalah tingkat keacakan dari informasi yang diperoleh dari information gain yang sedang diproses.

## 2.2 scikit-learn

Dataset ambil di <https://github.com/PacktPublishing/Python-Artificial-Intelligence-Projects-for-Beginners> folder Chapter01. Tugas anda adalah, dataset ganti menggunakan **student-mat.csv** dan mengganti semua nama variabel dari kode di bawah ini dengan nama-nama makanan (NPM mod 3=0), kota (NPM mod 3=1), buah (NPM mod 3=2), . Jalankan satu per satu kode tersebut di spyder dengan menggunakan textitRun current cell. Kemudian Jelaskan dengan menggunakan bahasa yang mudah dimengerti dan bebas plagiat dan wajib skrinsut dari komputer sendiri masing masing nomor di bawah ini(nilai 5 masing masing pada hari kedua).

### 1. #1 Load dataset

```
import pandas as pd
apel = pd.read_csv('student-mat.csv', sep=';')
len(apel)
```

Output :

```
In [2]: import pandas as pd
...: apel = pd.read_csv('student-mat.csv', sep=';')
...: len(apel)
Out[2]: 395
In [3]:
```

### 2. #2 generate binary label (pass)



```
# (test grades, each 0-20 pts); threshold for passing is sum>=30
```

```
apel['pass'] = apel.apply(lambda row: 1 if (row['G1']+row['G2']+row['G3']) >= 35 else 0,
                           axis=1)
apel = apel.drop(['G1', 'G2', 'G3'], axis=1)
apel.head()
```

Output :

```
Console 1/A X
```

```
In [8]:
.....
.....
..... apel['pass'] = apel.apply(lambda row: 1 if (row['G1']+row['G2']+row['G3']) >= 35 else 0,
axis=1)
..... apel = apel.drop(['G1', 'G2', 'G3'], axis=1)
..... apel.head()
..... # print(apel.head())
Out[8]:
  school sex  age address famsize ... Dalc  Walc  health absences pass
0     GP  F   18      U    GT3 ...   1     1     3         6     0
1     GP  F   17      U    GT3 ...   1     1     3         4     0
2     GP  F   15      U    LE3 ...   2     3     3        10     0
3     GP  F   15      U    GT3 ...   1     1     5         2     1
4     GP  F   16      U    GT3 ...   1     2     5         4     0

[5 rows x 31 columns]

In [9]:
```

3. # 3 use one-hot encoding on categorical columns

```
apel = pd.get_dummies(apel, columns=['sex', 'school', 'address', 'famsize', 'reason', 'guardian', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'romantic'])
apel.head()
```

Output :



```
Console 1/A X
[5 rows x 51 columns]

In [9]:
...: apel = pd.get_dummies(apel, columns=['sex', 'school', 'address', 'famsize', 'Pstatus', 'Mjob',
...:                                     'Fjob',
...:                                     'reason', 'guardian', 'schoolsup', 'famsup', 'paid',
...:                                     'activities',
...:                                     'nursery', 'higher', 'internet', 'romantic'])
...: apel.head()
Out[9]:
   age  Medu  Fedu  ...  internet_yes  romantic_no  romantic_yes
0   18     4     4  ...              0             1              0
1   17     1     1  ...              1             1              0
2   15     1     1  ...              1             1              0
3   15     4     2  ...              1             0              1
4   16     3     3  ...              0             1              0

[5 rows x 57 columns]

In [10]:
```

4. # 4 shuffle rows

```
apel = apel.sample(frac=1)
# split training and testing data
apel_train = apel[:500]
apel_test = apel[500:]

apel_train_att = apel_train.drop(['pass'], axis=1)
apel_train_pass = apel_train['pass']

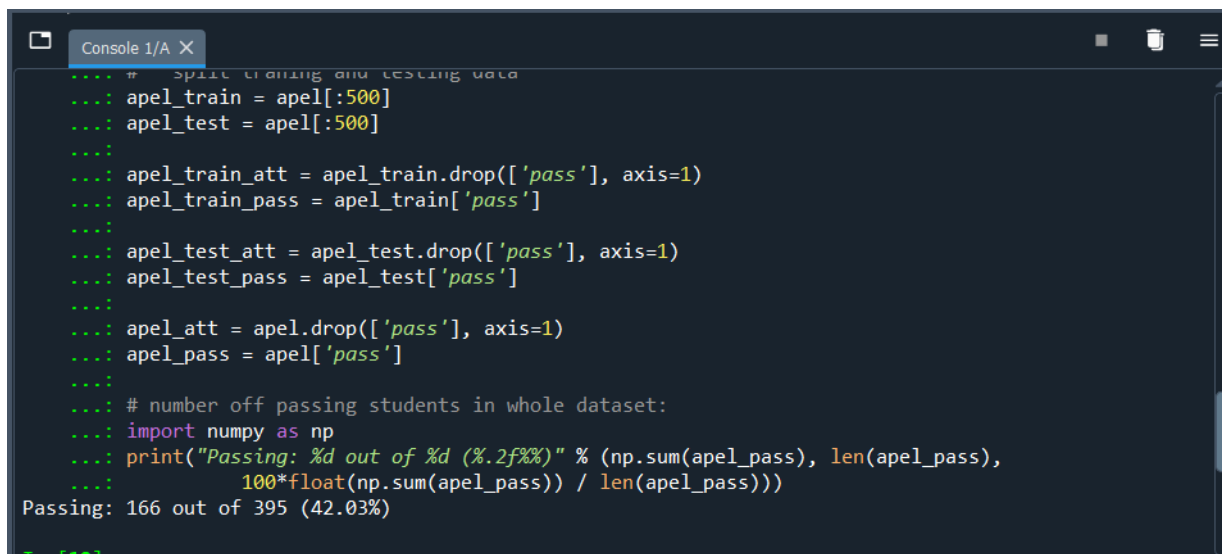
apel_test_att = apel_test.drop(['pass'], axis=1)
apel_test_pass = apel_test['pass']

apel_att = apel.drop(['pass'], axis=1)
apel_pass = apel['pass']

# number of passing students in whole dataset:
```

```
import numpy as np
print("Passing: %d out of %d (%.2f%%)" % (np.sum(apel_pass), len(apel_pass),
100*float(np.sum(apel_pass)) / len(apel_pass)))
```

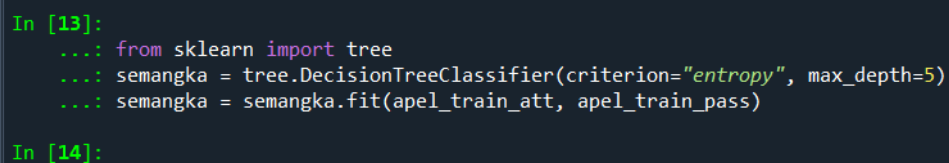
Output :



```
Console 1/A X
.... # Split training and testing data
.... apel_train = apel[:500]
.... apel_test = apel[:500]
....
.... apel_train_att = apel_train.drop(['pass'], axis=1)
.... apel_train_pass = apel_train['pass']
....
.... apel_test_att = apel_test.drop(['pass'], axis=1)
.... apel_test_pass = apel_test['pass']
....
.... apel_att = apel.drop(['pass'], axis=1)
.... apel_pass = apel['pass']
....
.... # number off passing students in whole dataset:
.... import numpy as np
.... print("Passing: %d out of %d (%.2f%%)" % (np.sum(apel_pass), len(apel_pass),
....     100*float(np.sum(apel_pass)) / len(apel_pass)))
Passing: 166 out of 395 (42.03%)
```

5. # 5 fit a decision tree

```
from sklearn import tree
semangka = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
semangka = semangka.fit(apel_train_att, apel_train_pass)
```

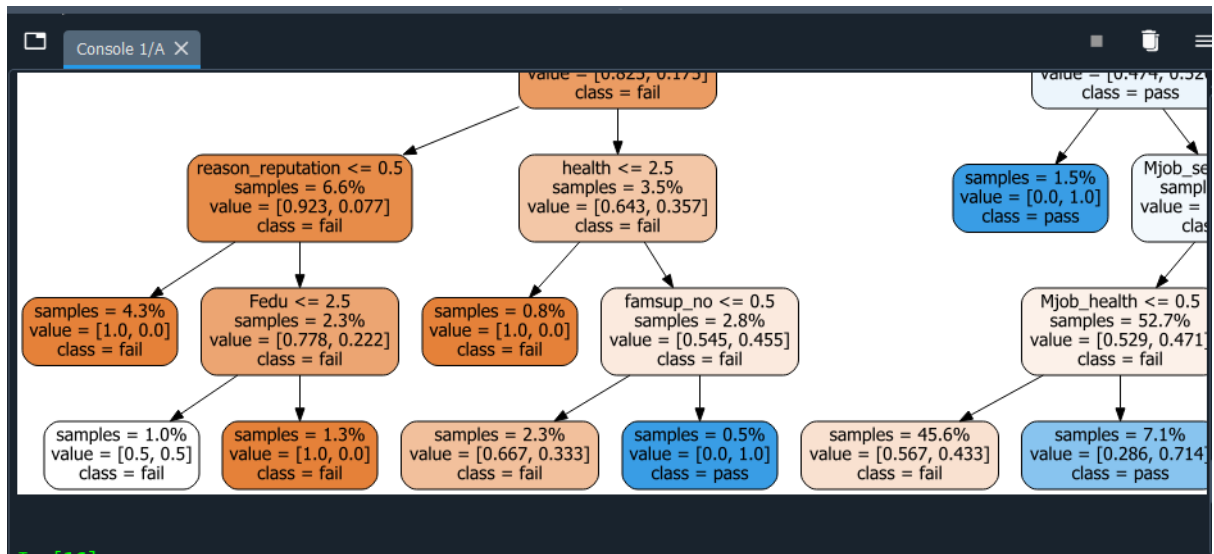


```
In [13]:
.... from sklearn import tree
.... semangka = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
.... semangka = semangka.fit(apel_train_att, apel_train_pass)

In [14]:
```

6. # 6 visualize tree

```
import graphviz
dot_data = tree.export_graphviz(semangka, out_file=None, label="all",
impurity=False, proportion=True,
feature_names=list(apel_train_att),
class_names=["fail", "pass"],
filled=True, rounded=True)
graph = graphviz.Source(dot_data)
graph
```



7. # 7 save tree

```
tree.export_graphviz(semangka, out_file="student-performance.dot",
label="all", impurity=False,
proportion=True,
feature_names=list(apel_train_att),
class_names=["fail", "pass"],
filled=True, rounded=True)
```

```
In [16]:
...: tree.export_graphviz(semangka, out_file="student-performance.dot",
...:                       label="all", impurity=False,
...:                       proportion=True,
...:                       feature_names=list(apel_train_att),
...:                       class_names=["fail", "pass"],
...:                       filled=True, rounded=True)
In [17]:
```

8. # 8

```
semangka.score(apel_test_att, apel_test_pass)
```

```
In [16]:
...: tree.export_graphviz(semangka, out_file="student-performance.dot",
...:                       label="all", impurity=False,
...:                       proportion=True,
...:                       feature_names=list(apel_train_att),
...:                       class_names=["fail", "pass"],
...:                       filled=True, rounded=True)
In [17]: semangka.score(apel_test_att, apel_test_pass)
Out[17]: 0.7063291139240506
```

9. # 9

```
from sklearn.model_selection import cross_val_score
salak = cross_val_score(semangka, apel_att, apel_pass, cv=5)
# show average score and +/- two standard deviations away
#(covering 95% of scores)
print("Accuracy: %0.2f (+/- %0.2f)" % (salak.mean(), salak.std() * 2))
```

```
In [18]:
...: from sklearn.model_selection import cross_val_score
...: salak = cross_val_score(semangka, apel_att, apel_pass, cv=5)
...: # show average score and +/- two standard deviations away
...: #(covering 95% of scores)
...: print("Accuracy: %0.2f (+/- %0.2f)" % (salak.mean(), salak.std() * 2))
Accuracy: 0.57 (+/- 0.06)
```

```
In [19]:
```

10. # 10

```
for max_depth in range(1, 20):
    semangka = tree.DecisionTreeClassifier(criterion="entropy",
    max_depth=max_depth)
    salak = cross_val_score(semangka, apel_att, apel_pass, cv=5)
    print("Max depth: %d, Accuracy: %0.2f (+/- %0.2f)" %
    (max_depth, salak.mean(), salak.std() * 2)
    )
```

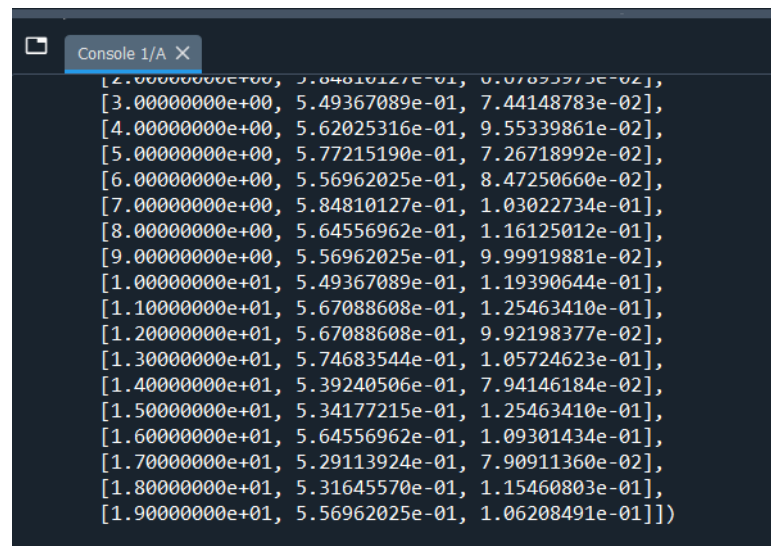
```
Console 1/A X
Max depth: 2, Accuracy: 0.56 (+/- 0.07)
Max depth: 3, Accuracy: 0.55 (+/- 0.07)
Max depth: 4, Accuracy: 0.56 (+/- 0.09)
Max depth: 5, Accuracy: 0.58 (+/- 0.07)
Max depth: 6, Accuracy: 0.55 (+/- 0.08)
Max depth: 7, Accuracy: 0.59 (+/- 0.09)
Max depth: 8, Accuracy: 0.57 (+/- 0.14)
Max depth: 9, Accuracy: 0.55 (+/- 0.14)
Max depth: 10, Accuracy: 0.55 (+/- 0.13)
Max depth: 11, Accuracy: 0.55 (+/- 0.15)
Max depth: 12, Accuracy: 0.53 (+/- 0.13)
Max depth: 13, Accuracy: 0.55 (+/- 0.11)
Max depth: 14, Accuracy: 0.55 (+/- 0.09)
Max depth: 15, Accuracy: 0.54 (+/- 0.08)
Max depth: 16, Accuracy: 0.55 (+/- 0.10)
Max depth: 17, Accuracy: 0.55 (+/- 0.09)
Max depth: 18, Accuracy: 0.53 (+/- 0.07)
Max depth: 19, Accuracy: 0.54 (+/- 0.08)

In [20]:
```

```

11. # 11
    depth_acc = np.empty((19,3), float)
    i = 0
    for max_depth in range(1, 20):
        semangka = tree.DecisionTreeClassifier(criterion="entropy",
        max_depth=max_depth)
        salak = cross_val_score(semangka, apel_att, apel_pass, cv=5)
        depth_acc[i,0] = max_depth
        depth_acc[i,1] = salak.mean()
        depth_acc[i,2] = salak.std() * 2
        i += 1
    depth_acc

```



```

12. # 12
    import matplotlib.pyplot as plt
    fig, ax = plt.subplots()
    ax.errorbar(depth_acc[:,0], depth_acc[:,1], yerr=depth_acc[:,2])
    plt.show()

```

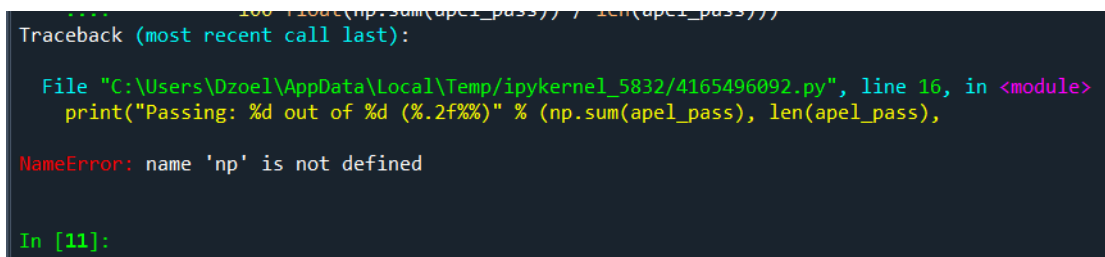


Link Youtube : <https://youtu.be/2aM5RLBPp1c>

## 2.3 Penanganan Error

Dari percobaan yang dilakukan di atas, error yang kita dapatkan di dokumentasikan dan di selesaikan (nilai 5 hari kedua):

1. skrinsut error



```
Traceback (most recent call last):
  File "C:\Users\Dzoel\AppData\Local\Temp\ipykernel_5832\4165496092.py", line 16, in <module>
    print("Passing: %d out of %d (%.2f%%)" % (np.sum(apel_pass), len(apel_pass),
NameError: name 'np' is not defined

In [11]:
```

2. Tuliskan kode eror dan jenis errornya  
NameError : name 'np' is not defined
3. Solusi pemecahan masalah error tersebut  
Solusinya adalah import numpy dengan alias np.