

Modul Praktikum Kecerdasan Buatan



Rolly Maulana Awangga
0410118609

Applied Bachelor of Informatics Engineering
Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering
Politeknik Pos Indonesia

Bandung 2019

‘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

1	Mengenal Kecerdasan Buatan dan Scikit-Learn	1
1.1	Teori	1
1.2	Instalasi	2
1.3	Penanganan Error	2
1.4	Cokro Edi Prawiro/1164069	2
1.4.1	Praktek teori penunjang	2
1.4.2	Instalisasi	5
1.4.3	Penanganan Error	15
2	Related Works	20
2.1	Same Topics	20
2.1.1	Topic 1	20
2.1.2	Topic 2	20
2.2	Same Method	20
2.2.1	Method 1	20
2.2.2	Method 2	20
3	Methods	21
3.1	The data	21
3.2	Method 1	21
3.3	Method 2	21
4	Experiment and Result	22
4.1	Experiment	22
4.2	Result	22

5 Conclusion	23
5.1 Conclusion of Problems	23
5.2 Conclusion of Method	23
5.3 Conclusion of Experiment	23
5.4 Conclusion of Result	23
6 Discussion	24
7 Discussion	25
8 Discussion	26
9 Discussion	27
10 Discussion	28
11 Discussion	29
12 Discussion	30
13 Discussion	31
14 Discussion	32
A Form Penilaian Jurnal	33
B FAQ	36
Bibliography	38

List of Figures

1.1	Tampilan website Scikit 1.	5
1.2	Tampilan website Scikit 2.	6
1.3	Tampilan Versi Python dan Anaconda	16
1.4	Instalisasi Library Sikic.	16
1.5	Instalasi Library Sikic Melalui Conda	17
1.6	Console Python Include Anaconda	17
1.7	Contoh Codingan Dataset	17
1.8	Error Coding 1	18
1.9	Error Coding 2	18
1.10	Codingan Solusi Untuk Error digits	18
1.11	Codingan Solusi Untuk Error Joblib	18
1.12	Hasil Solusi Error Joblib	19
A.1	Form nilai bagian 1.	34
A.2	form nilai bagian 2.	35

Chapter 1

Mengenai Kecerdasan Buatan dan Scikit-Learn

Buku umum yang digunakan adalah [4] dan untuk sebelum UTS menggunakan buku *Python Artificial Intelligence Projects for Beginners*[1]. Dengan praktek menggunakan python 3 dan editor anaconda dan library python scikit-learn. Tujuan pembelajaran pada pertemuan pertama antara lain:

1. Mengerti definisi kecerdasan buatan, sejarah kecerdasan buatan, perkembangan dan penggunaan di perusahaan
2. Memahami cara instalasi dan pemakaian sci-kit learn
3. Memahami cara penggunaan variabel explorer di spyder

Tugas dengan cara dikumpulkan dengan pull request ke github dengan menggunakan latex pada repo yang dibuat oleh asisten riset.

1.1 Teori

Praktek teori penunjang yang dikerjakan :

1. Buat Resume Definisi, Sejarah dan perkembangan Kecerdasan Buatan, dengan bahasa yang mudah dipahami dan dimengerti. Buatan sendiri bebas plagiat[hari ke 1](10)
2. Buat Resume mengenai definisi supervised learning, klasifikasi, regresi dan unsupervised learning. Data set, training set dan testing set.[hari ke 1](10)

1.2 Instalasi

Membuka <https://scikit-learn.org/stable/tutorial/basic/tutorial.html>. Dengan menggunakan bahasa yang mudah dimengerti dan bebas plagiat. Dan wajib skrinsut dari komputer sendiri.

1. Instalasi library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer[hari ke 1](10)
2. Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 1](10)
3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)
4. mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)
5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)

1.3 Penanganan Error

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

1. skrinsut error[hari ke 2](10)
2. Tuliskan kode eror dan jenis errornya [hari ke 2](10)
3. Solusi pemecahan masalah error tersebut[hari ke 2](10)

1.4 Cokro Edi Prawiro/1164069

1.4.1 Praktek teori penunjang

1. Kecerdasan Buatan *Artificial Intelligence* adalah suatu cabang dalam bidang sains komputer yang mengkaji bagaimana untuk melengkapi sebuah komputer dengan kemampuan atau kepintaran saperti manusia. Komputer tersebut di harapkan dapat belajar sendiri dengan cara mengumpulkan data-data yang diterimanya, yang berguna sebagai parameter untuk memecahkan masalah. Jadi

kecerdasan buatan merupakan kecerdasan yang di program dalam komputer untuk memecahkan masalah secara tepat dan cepat atau untuk memberikan kemungkinan keberhasilan dan kegagalan pada solusi dari suatu masalah.

Adapun kecerdasan buatan menurut para ahli adalah sebagai berikut :

- Kecerdasan Buatan merupakan Kawasan penelitian, aplikasi dan intruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas (H. A. Simon[1997]).
- Kecerdasan buatan adalah bidang studi yang berhubungan dengan penangkapan, pemodelan, dan penyimpanan kecerdasan manusia dalam sebuah sistem teknologi sehingga sistem tersebut dapat memfasilitasi proses pengambilan keputusan yang biasanya dilakukan oleh manusia (Haag dan Keen[1996]).

Sejarah dan Perkembangan Kecerdasan Buatan.

ketika *Rene Descartes* mengemukakan gagasan yang menjadi cikal bakal kecerdasan buatan pada abad 17 mengemukakan bahwa hewan bukan apa-apa melainkan hanya mesin yang rumit yang dilanjutkan oleh *Blaise Pascal* yang telah menciptakan mesin penghitung digital mekanis pertama pada tahun 1642. Lalu pada abad 19 *Charles Babbage* dan *Ada Lovelace* bekerja sama membuat mesin penghitung mekanis yang dapat di program.

Perkembangan kecerdasan buatan inipun terus berlanjut, *Bertrand Russell* dan *Alfred North Whitehead* menerbitkan *mathematica*, yang merombak logika formal. Setelah itu dilanjutkan dengan penemuan oleh *Warren McCulloch* dan *Walter Pitts* menerbitkan “Kalkulus Logis Gagasan yang tetap ada dalam Aktivitas” pada 1943 yang meletakkan pondasi awal berupa jaringan syaraf. Kemudian pada tahun 1950-an adalah periode awal usaha aktif kecerdasan buatan. Program Kecerdasan buatan pertama yang bekerja diciptakan pada tahun 1951 untuk menjalankan mesin Ferranti Mark I di University of Manchester (UK) yaitu sebuah program permainan naskah yang ditulis oleh *Christopher Strachey* dan program permainan catur yang ditulis oleh *Dietrich Prinz*. Kemudian pada konferensi pertama tahun 1956 *John McCarthy* mengemukakan istilah “kecerdasan buatan” kemudian dia juga menemukan bahasa pemrograman lisp. *Joseph Weizenbaum* menciptakan ELIZA, sebuah chatterbot yang menerapkan psikoterapi Rogerian.

Selama rentang waktu tahun 1960-an dan 1970-an, *Joel Moses* mendemonstrasikan kekuatan pertimbangan simbolis untuk mengintegrasikan masalah di

dalam program Macsyma, yang merupakan program yang pertamakali sukses dalam bidang matematika. Kemudian pada tahun 1980-an industri kecerdasan buatan ini berkembang walau sudah di mulai pada tahun 1970-an Evolusi kecerdasan buatan berjalan dalam dua jalur yang berbeda yaitu meniru proses berpikir manusia untuk menyelesaikan masalah umum. Kedua mengkombinasikan pemikiran terbaik para ahli pada sepotong software yang dirancang untuk memecahkan persoalan yang spesifik.

2. Supervised learning adalah sebuah pendekatan dengan syarat sudah terdapat data yang dilatih kemudian harus terdapat variable yang ditargetkan sehingga tujuan dari pendekatan ini adalah pengelompokan data terhadap data yang telah ada. Ciri khas dari Supervised learning yaitu terdapat label atau nama kelas pada data latih (supervisi) dan data baru di klasifikasikan berdasarkan data latih. Data latih dikelompokkan berdasarkan ukuran kemiripan pada suatu kelas. Berdasarkan keluaran dari fungsi, Supervised learning dibagi menjadi 2, regresi dan klasifikasi. Regresi terjadi jika output dari fungsi merupakan nilai yang kontinyu, sedangkan klasifikasi terjadi jika keluaran dari fungsi adalah nilai tertentu dari suatu atribut (tidak kontinyu). Tujuan dari Supervised learning adalah untuk memprediksi nilai dari fungsi untuk sebuah data masuk yang sah setelah melihat sejumlah data latih[2].

Adapun pengertian klasifikasi dan regresi adalah sebagai berikut :

- Klasifikasi merupakan pengelompokan berdasarkan parameter tertentu yang tidak konstan contoh pada makhluk hidup yaitu persamaan ciri cara hidup dan tempat hidup.
- Regresi yaitu pengeluaran nilai output yang konstan jika dipicu dengan parameter tertentu biasanya regresi disini berbentuk regresi linier. Regresi linier yaitu metode statistika yang digunakan untuk membentuk model hubungan antara variabel terikat(dependen, respon, Y) dengan satu atau lebih variabel bebas(independent, prediktor, X). Apabila banyaknya variabel bebas hanya ada satu, disebut sebagai regresi linier sederhana, sedangkan apabila terdapat lebih dari satu variabel bebas, disebut sebagai regresi linier berganda [3].

unsupervised learning adalah pendekatan yang tidak memerlukan data latih atau data training untuk melakukan prediksi atau klasifikasi. Berdasarkan

model secara matematisnya, algoritma ini tidak memiliki target variabel. Tujuan dari algoritma ini yaitu pengelompokan objek yang memiliki kesamaan atau hampirsama dalam satu cakupan wilayah tertentu. Kemudian pada unsupervised learning tidak terdapat label atau nama kelas pada data latih. Kemudian dataset merupakan objek yang menggambarkan data itu sendiri dan relasinya di memory. Struktur datanya mirip dengan struktur data di basis-data. Jadi strikturnya terdiri atas baris kolom dan juga ada sejenis relasi data. Pada dataset terdiri bagian bagian yaitu training set dan Testing set. Adapun pengertian dari training set dan Testing set adalah sebagai berikut :

- Training set adalah bagian dari dataset itu sendiri yang dilatih untuk membuat prediksi atau algoritma mesin learning lainnya sesuai keinginan atau tujuan data itu dibuat.
- Testing set adalah bagian dari dataset yang di tes atau diujicoba untuk melihat keakuratannya dengan katalain melihat peformanya.

1.4.2 Instalasi

Pada proses instalasi ini langkah pertama yaitu mengakses website scikit dengan mengakses link berikut <https://scikit-learn.org/stable/tutorial/basic/tutorial.html> maka hasilnya dapat dilihat pada gambar 1.1 kemudian setelah itu klik button installation maka akan muncul tampilan yang dapat dilihat pada gambar 1.2.

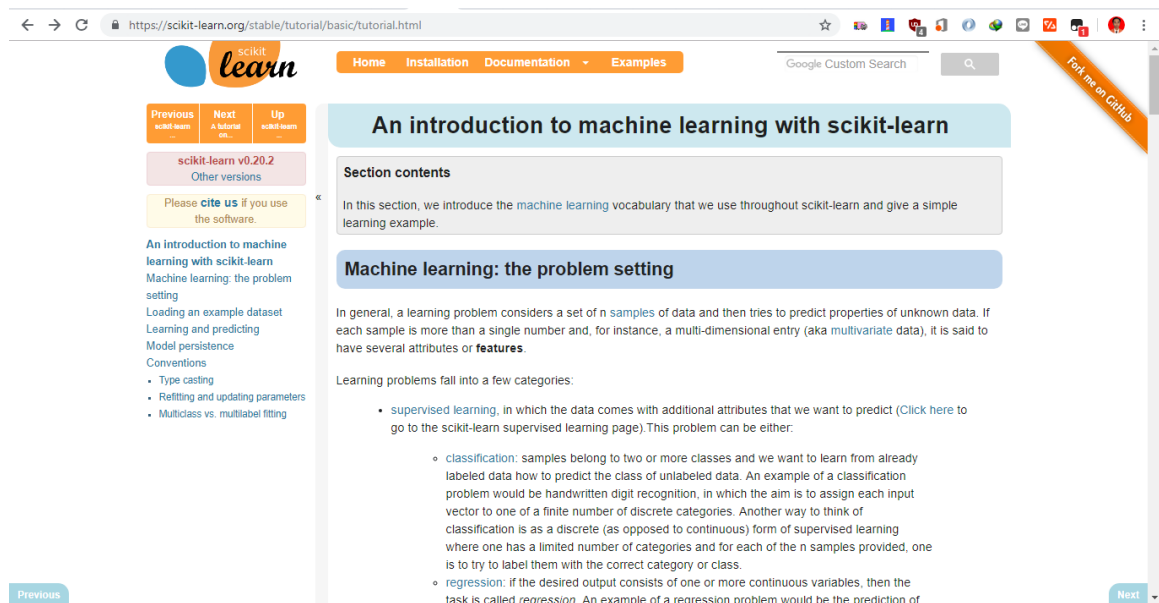


Figure 1.1: Tampilan website Scikit 1.



Figure 1.2: Tampilan website Scikit 2.

1. cara instalisasi Instalasi library scikit dari anaconda langkah pertama instal terlebih dahulu anacondanya dikarenakan anaconda sudah include dengan python maka codingan python dapat digunakan di anaconda dan ketika diperiksa versinya maka akan muncul tampilan seperti Gambar 1.3

kemudian pada cmd administrator install library scikit dengan cara memasukkan codingan `pip install -U scikit-learn` maka hasilnya seperti seperti pada gambar 1.4 berikut.

setelah itu masukan kembali perintah berikut di cmd `conda install scikit-learn` jika hasilnya seperti pada gambar 1.5 maka librari scikit telah terinstal dan siap untuk di gunakan.

kemudian untuk mencobanya tuliskan perintah python pada cmd lalu masukan codingan `print("Hello Anaconda!")` maka hasilnya terlihat seperti gambar 1.6 codingan `print("Hello Anaconda!")` yaitu berfungsi mencetak nilai yang ada di dalam kurung dan di antara kutip.

2. cara mencoba dataset yaitu dengan cara memasukkan perintah berikut pada cmd seperti pada gambar 1.7
 - pada codingan `from sklearn import datasets` menjelaskan inport librari dataset dari librari scikit pada gambar 1.7

- pada codingan `iris = datasets.load_iris()` berarti parameter atau acuan bernama `iris` kemudian di load ke dalam dataset sebagai perbandingan kalau dalam diagram iris bisa disebut X nya pada gambar 1.7
- pada codingan `digits = datasets.load_digits()` berarti parameter hampir mirip seperti iris tadi namun `digits` merupakan kebalikannya kalau di dalam diagram dia bernilai Y pada gambar 1.7
- kemudian pada codingan `print(digits.data)` yaitu mencetak data `digits` yang di bandingkan dengan data iris pada gambar 1.7
- pada codingan `print(iris.data)` yaitu mencetak data iris yang dibandingkan dengan data `digits` pada gambar 1.7

3. Mencoba Learning and Predicting

Pada kasus ini dataset digit digunakan untuk memprediksi yang mana telah di berikan gambar untuk mewakili sampel masing-masing dari 10 kelas yang dimulai dari digit nol hingga sembilan yang digunakan untuk memprediksi sampel yang tidak terlihat untuk lebih jelasnya dapat di praktikan codingan berikut ini.

```
>>> from sklearn import datasets
```

pada baris ini dapat diartikan bahwa librari `sklearn` mengimport package dataset

```
>>> iris = datasets.load_iris()
```

pada baris ini dimasukan parameter `iris` yang di sandingkan dengan dataset load sehingga `iris` berisi nilai dataset

```
>>> digits = datasets.load_digits()
```

pada baris ini dimasukan parameter `digits` yang di sandingkan dengan dataset load sehingga `digits` berisi nilai dataset

```
>>> from sklearn import svm
```

pada baris ini librari `sklearn` mengimport package `svm`

```
>>> clf = svm.SVC(gamma=0.0001, C=100.)
```

pada codingan diatas dibuat variabel clf yang di isi dengan nilai svm dengan nilai gama 0.0001 dan 100.

```
>>> clf.fit(digits.data[:-1], digits.target[:-1])
```

pada codingan clf di implementasikan dengan perintah fit

```
SVC(C=100.0, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape='ovr', degree=3, gamma=0.0001, kernel='rbf',  
    max_iter=-1, probability=False, random_state=None, shrinking=True,  
    tol=0.001, verbose=False)
```

yang hasilnya seperti codingan diatas yang menjabarkan isi dari SVC itu sendiri.

```
>>> clf.predict(digits.data[-1:])  
array([8])  
>>>
```

kemudain pada codingan diatas digunakan perintah predic yang merupakan perintah untuk mengimplementasikan method digits.

4. Mencoba model pertistence

model pertistence yaitu model yang digunakan untuk mengolah data sehingga data tersebut konstan atau konsisten terhadap parameter tertentu contoh pada codingan di bagawah nilai y akan konstan di nol walaupun telah di isi nilai lebih dari nol.

```
>>> from sklearn import svm
```

pada baris ini librari sklearn mengimport package svm.

```
>>> from sklearn import datasets
```

pada baris ini librari sklearn mengimport package datasets.

```
>>> clf = svm.SVC(gamma='scale')
```

pada codingan diatas dibuat variabel clf yang di isi dengan nilai svm dengan gamma sama dengan scale.

```
>>> iris = datasets.load_iris()
```

pada baris ini dimasukan parameter iris yang di sandingkan dengan dataset load sehingga iris berisi nilai dataset.

```
>>> X, y = iris.data, iris.target
```

pada codingan diatas X berisi nilai iris.data dan y berisi nilai iris.target.

```
>>> clf.fit(X, y)
```

method clf di implementasikan dengan perintah fit dengan X, y sebagai nilai untuk implementasinya.

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',  
    max_iter=-1, probability=False, random_state=None, shrinking=True,  
    tol=0.001, verbose=False)
```

maka hasilnya penjabaran dari SVC seperti codingan diatas.

```
>>> import pickle
```

mengimport library atau package pickle.

```
>>> s = pickle.dumps(clf)
```

kemudian di buat variabel s yang di load oleh package pickle dengan di isi nilai clf.

```
>>> clf2 = pickle.loads(s)
```

setelah itu pada codingan diatas dibuat lagi variabel clf2 kemudian di load pickle.

```
>>> clf2.predict(X[0:1])
```

kemudian variabel clf2 di implementasikan dengan parameter X dengan nilai 0 berbanding 1 maka hasilnya nilainya array bernilai nol dan y bernialai nol.


```
array([0])
>>> y[0]
0
```

5. mencoba conventions conventions merupakan aturan aturan dasar atau kesepakatan kesepakatan dalam pemrograman sikit python dan anaconda berikut merupakan jenis-jenis codingan conventions :

- Type casting yaitu tipe pelemparan parameter atau variabel kedalam variabel baru.

```
>>> import numpy as np
```

codingan diatas yaitu import librari numpy yang di inisialisasi menjadi np

```
>>> from sklearn import random_projection
```

import librari random_projection

```
>>> rng = np.random.RandomState(0)
```

membuat variabel baru dengan nama rng dengan nilai random

```
>>> X = rng.rand(10, 2000)
```

memasukan nilai rng kedalam variabel X dengan rad nilai 10 sampai 2000

```
>>> X = np.array(X, dtype='float32')
```

menambahkan nilai np berupa array yaitu X dan float 32

```
>>> X.dtype
```

```
dtype('float32')
```

X.dtype di running menghasilkan nilai dtype float

```
>>> transformer = random_projection.GaussianRandomProjection()
```

membuat variabel transformer dengan nilai random

```
>>> X_new = transformer.fit_transform(X)
```

membuat variabel X_new dan di isi nilai transformer kemudian di implementasikan

```
>>> X_new.dtype
```

merunning variabel X_new

```
dtype('float64')
```

hasil running X_new

```
>>> from sklearn import datasets
```

mengimport library dataset

```
>>> from sklearn.svm import SVC
```

mengimport library SVC

```
>>> iris = datasets.load_iris()
```

membuat variabel iris dengan nilai load dataset

```
>>> clf = SVC(gamma='scale')
```

membuat variabel clf bernilai SVC dengan gamma menggunakan nilai skala

```
>>> clf.fit(iris.data, iris.target)
```

merunning variabel atau method clf dengan isian nilai iris data dan iris target

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',  
    max_iter=-1, probability=False, random_state=None, shrinking=True,  
    tol=0.001, verbose=False)
```

detail hasil runing clf

```
>>> list(clf.predict(iris.data[:3]))
```

```
[0, 0, 0]
```

memunculkan detail atau lis sebanyak tiga nilai

```
>>> clf.fit(iris.data, iris.target_names[iris.target])
```

merunning kembali clf dengan nilai iris data, iris target name

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',  
    max_iter=-1, probability=False, random_state=None, shrinking=True,  
    tol=0.001, verbose=False)
```

hasil dari running clf

```
>>> list(clf.predict(iris.data[:3]))  
['setosa', 'setosa', 'setosa']
```

memunculkan tiga nilai yang telah dilempar dari SVC

- Refitting and updating parameters atau pengisian ulang atau memperbaiki parameter merupakan cara untuk merubah nilai dari sebuah parameter contoh nilai x adalah 10 jika di perbaharui bisa menjadi 15 begitu juga dalam codingan berikut hal ini dapat dilakukan untuk lebih jelasnya dapat dilihat codingan dibawah ini :

```
>>> import numpy as np
```

codingan diatas yaitu import librari numpy yang di inisialisasi menjadi np

```
>>> from sklearn.svm import SVC
```

mengimport library SVC

```
>>> rng = np.random.RandomState(0).
```

membuat variabel baru dengan nama rng dengan nilai random.

```
>>> X = rng.rand(100, 10)
```

parameter X dengan nilai dari variabel rng dan rad dari 100 sampai 10.

```
>>> y = rng.binomial(1, 0.5, 100)
```

parameter y dengan nilai rng binominal dari 1 0,5 sampai 100.

```
>>> X_test = rng.rand(5, 10)
```

parameter X_test dengan nilai rng dan rad dari 5 ke 10

```
>>> clf = SVC()
```

parameter clf bernilai SVC

```
>>> clf.set_params(kernel='linear').fit(X, y)
```

parameter clf di set dengan mengompile atau mengekstrak nilai X dan y dengan kernel linear.

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='linear', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

penjabaran nilai SVC hasil running CLF

```
>>> clf.predict(X_test)
array([1, 0, 1, 1, 0])
```

meranning clf dengan nilai X_test

```
>>> clf.set_params(kernel='rbf', gamma='scale').fit(X, y)
```

parameter clf di set dengan kernel rbf dan gama skala dan mengkompilasi nilai X dan y.

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

penjabaran nilai SVC hasil running CLF

```
>>> clf.predict(X_test)
array([1, 0, 1, 1, 0])
```

Hasil dari running clf.

- Multiclass vs. multilabel fitting perbandingan antara banyak kelas dan pelabelan yang tepat berikut merupakan codingannya.

```
>>> from sklearn.svm import SVC
```

mengimport library SVC

```
>>> from sklearn.multiclass import OneVsRestClassifier
```

memasukan librari OneVsRestClassifier dengan kondisi multi class.

```
>>> from sklearn.preprocessing import LabelBinarizer
```

memasukan librari LabelBinarizer

```
>>> X = [[1, 2], [2, 4], [4, 5], [3, 2], [3, 1]]
```

pemberian nilai pada parameter X

```
>>> y = [0, 0, 1, 1, 2]
```

pemberian nilai pada parameter y

```
>>> classif = OneVsRestClassifier(estimator=SVC(gamma='scale',
... random_state=0))
```

opsi untuk class dengan ketentuan estimator SVC gamma berbentuk skala dan random (acak).

```
>>> classif.fit(X, y).predict(X)
array([0, 0, 1, 1, 2])
```

hasil array dari running classif

```
>>> y = LabelBinarizer().fit_transform(y)
```

memberikan nilai pada parameter y

```
>>> classif.fit(X, y).predict(X)
array([[1, 0, 0],
       [1, 0, 0],
       [0, 1, 0],
       [0, 0, 0],
       [0, 0, 0]])
```

hasil running classif

```
>>> from sklearn.preprocessing import MultiLabelBinarizer

import librari multi label
```

```
>>> y = [[0, 1], [0, 2], [1, 3], [0, 2, 3], [2, 4]]
```

memberikan nilai pada parameter y

```
>>> y = MultiLabelBinarizer().fit_transform(y)
```

membuat parameter y menjadi multi label.

```
>>> classif.fit(X, y).predict(X)
array([[1, 1, 0, 0, 0],
       [1, 0, 1, 0, 0],
       [0, 1, 0, 1, 0],
       [1, 0, 1, 0, 0],
       [1, 0, 1, 0, 0]])
```

hasil running classif setelah nilai parameter y telah di ganti.

1.4.3 Penanganan Error

1. Screenshot Error untuk lebih jelasnya Screenshot codingan dapat dilihat pada gambar 1.8 dan 1.9
2. Kode error pada screenshot untuk kode error pada gambar 1.8 yaitu pada source code berikut

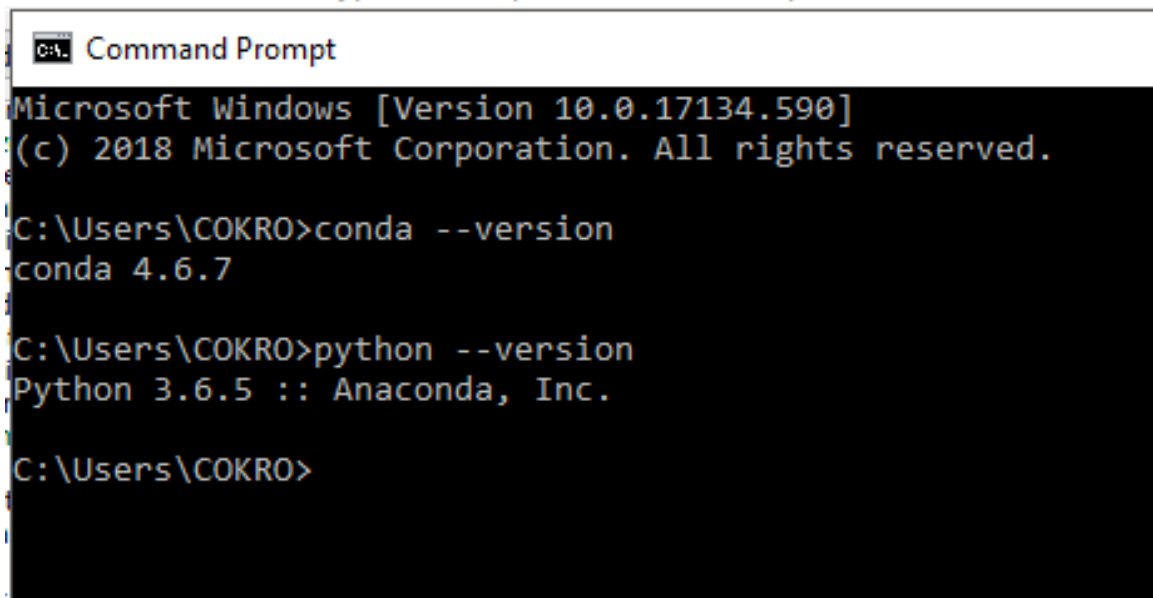
```
clf.fit(digits.data[:-1], digits.target[:-1])
```

pada codingan tersebut menjadi error dikarenakan variabel atau method digits belum di definisikan. sedangkan untuk kode error pada gambar 1.9 yaitu pada source code

```
from joblib import dump, load
```

pada codingantersebut terjadi error dikarenakan module joblib belum di install atau modul tersebut tidak ada di library python.

3. Solusi Pemecahan Masalah Error
 - untuk memperbaiki error pada gambar 1.8 tinggal mendefinisikan variabel atau method digits, untuk lebih lengkapnya dapat dilihat pada gambar 1.10 dengan cara tersebut maka masalah error dapat diselesaikan.
 - sedangkan untuk error joblib bisa dilakukan dengan cara masuk ke cmd administrator kemudian isikan perintah pip install joblib kemudiantekan enter sehingga hasilnya terlihat seperti gambar 1.11 setelah itu coba masuk kembali ke python di cmd dan ketikan perintah from joblib import dump, load maka hasilnya seperti gambar 1.12.

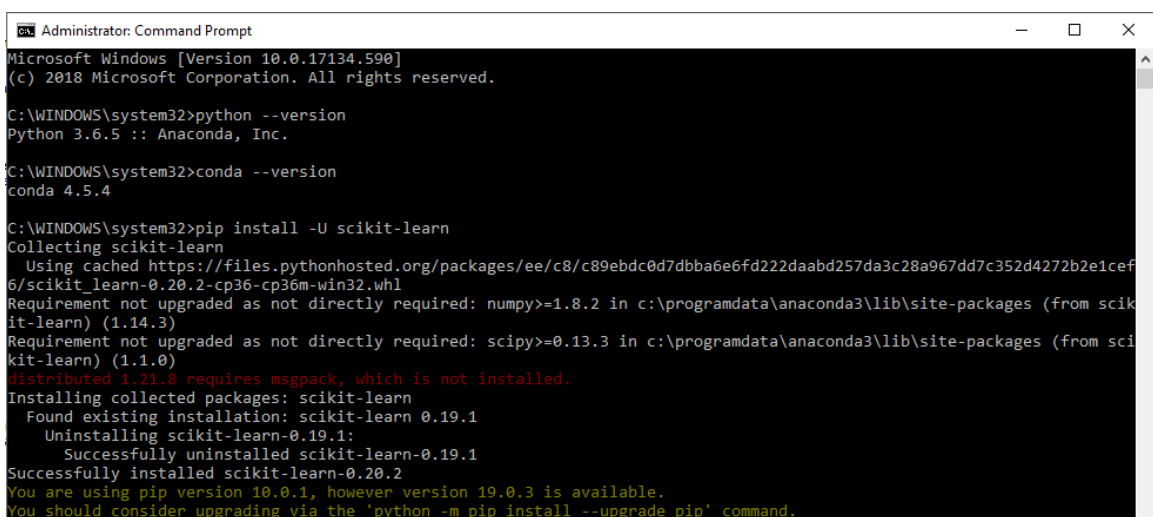


```
C:\Users\COKRO>conda --version
conda 4.6.7

C:\Users\COKRO>python --version
Python 3.6.5 :: Anaconda, Inc.

C:\Users\COKRO>
```

Figure 1.3: Tampilan Versi Python dan Anaconda .



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>python --version
Python 3.6.5 :: Anaconda, Inc.

C:\WINDOWS\system32>conda --version
conda 4.5.4

C:\WINDOWS\system32>pip install -U scikit-learn
Collecting scikit-learn
  Using cached https://files.pythonhosted.org/packages/ee/c8/c89ebdc0d7dbba6e6fd222daabd257da3c28a967dd7c352d4272b2e1cef6/scikit_learn-0.20.2-cp36-cp36m-win32.whl
Requirement not upgraded as not directly required: numpy>=1.8.2 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (1.14.3)
Requirement not upgraded as not directly required: scipy>=0.13.3 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (1.1.0)
distributed 1.21.8 requires msgpack, which is not installed.
Installing collected packages: scikit-learn
  Found existing installation: scikit-learn 0.19.1
    Uninstalling scikit-learn-0.19.1:
      Successfully uninstalled scikit-learn-0.19.1
Successfully installed scikit-learn-0.20.2
You are using pip version 10.0.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Figure 1.4: Instalikasi Library Sikic.

```

C:\WINDOWS\system32>conda install scikit-learn
Solving environment: done

## Package Plan ##

  environment location: C:\ProgramData\Anaconda3

  added / updated specs:
    - scikit-learn

The following packages will be UPDATED:

    conda: 4.5.4-py36_0 --> 4.6.7-py36_0

Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

C:\WINDOWS\system32>

```

Figure 1.5: Instalasi Library Sikic Melalui Conda

```

C:\Users\COKRO>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello Anaconda!")
Hello Anaconda!
>>>

```

Figure 1.6: Console Python Include Anaconda

```

Command Prompt - python

C:\Users\COKRO>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from sklearn import datasets
>>> iris = datasets.load_iris()
>>> digits = datasets.load_digits()
>>> print(digits.data)
[[ 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.]]
>>> print(iris.data)
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]
 [5.4 3.7 1.5 0.2]
 [4.8 3.4 1.6 0.2]
 [4.8 3.  1.4 0.1]
 [4.3 3.  1.1 0.1]
 [5.8 4.  1.2 0.2]
 [5.7 4.4 1.5 0.4]
 [5.4 3.9 1.3 0.4]
 [5.1 3.5 1.4 0.3]
 [5.7 3.8 1.7 0.3]
 [5.1 3.8 1.5 0.3]
 [5.4 3.4 1.7 0.2]
 [5.1 3.7 1.5 0.4]
 [4.6 3.6 1.  0.2]
 [5.1 3.3 1.7 0.5]
 [4.8 3.4 1.9 0.2]
 [5.  3.  1.6 0.2]
 [5.  3.4 1.6 0.4]
 [5.2 3.5 1.5 0.2]]

```

Figure 1.7: Contoh Codingan Dataset


```
Command Prompt - python
C:\Users\COKRO>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from sklearn import svm
>>> clf = svm.SVC(gamma=0.001, C=100.)
>>> clf.fit(digits.data[:-1], digits.target[:-1])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'digits' is not defined
>>>
```

Figure 1.8: Error Coding 1

```
Command Prompt - python
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\COKRO>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from joblib import dump, load
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'joblib'
>>>
```

Figure 1.9: Error Coding 2

```
Command Prompt - python
C:\Users\COKRO>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from sklearn import datasets
>>> iris = datasets.load_iris()
>>> digits = datasets.load_digits()
>>> from sklearn import svm
>>> clf = svm.SVC(gamma=0.0001, C=100.)
>>> clf.fit(digits.data[:-1], digits.target[:-1])
SVC(C=100.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.0001, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
>>> clf.predict(digits.data[-1:])
array([8])
>>>
```

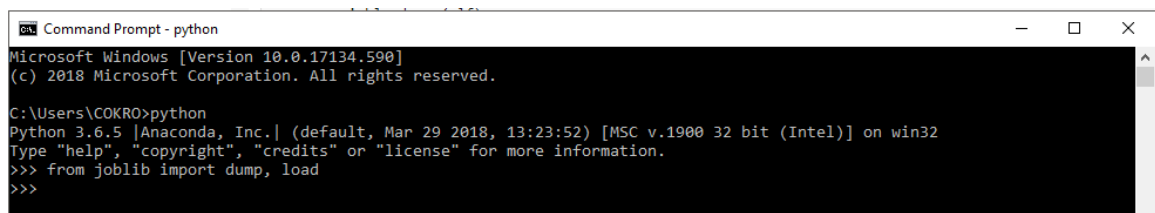
Figure 1.10: Codingan Solusi Untuk Error digits

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>pip install joblib
Collecting joblib
  Downloading https://files.pythonhosted.org/packages/cd/c1/50a758e8247561e58cb87305b1e90b171b8c767b15b12a1734001f41d356
/joblib-0.13.2-py2.py3-none-any.whl (278kB)
    100% |#####| 286kB 2.3MB/s
distributed 1.21.8 requires msgpack, which is not installed.
Installing collected packages: joblib
Successfully installed joblib-0.13.2
You are using pip version 10.0.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\WINDOWS\system32>
```

Figure 1.11: Codingan Solusi Untuk Error Joblib



```
Command Prompt - python
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\COKRO>python
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from joblib import dump, load
>>>
```

Figure 1.12: Hasil Solusi Error Joblib

Chapter 2

Related Works

Your related works, and your purpose and contribution which must be different as below.

2.1 Same Topics

Cite every latest journal with same topic

2.1.1 Topic 1

cite for first topic

2.1.2 Topic 2

if you have two topics you can include here to

2.2 Same Method

write and cite latest journal with same method

2.2.1 Method 1

cite and paraphrase method 1

2.2.2 Method 2

cite and paraphrase method 2 if you have more method please add new subsection.

Chapter 3

Methods

3.1 The data

Please tell where is the data come from, a little brief of company can be put here.

3.2 Method 1

Definition, steps, algorithm or equation of method 1 and how to apply into your data

3.3 Method 2

Definition, steps, algorithm or equation of method 2 and how to apply into your data

Chapter 4

Experiment and Result

brief of experiment and result.

4.1 Experiment

Please tell how the experiment conducted from method.

4.2 Result

Please provide the result of experiment

Chapter 5

Conclusion

brief of conclusion

5.1 Conclusion of Problems

Tell about solving the problem

5.2 Conclusion of Method

Tell about solving using method

5.3 Conclusion of Experiment

Tell about solving in the experiment

5.4 Conclusion of Result

tell about result for purpose of this research.

Chapter 6

Discussion

Please tell more about conclusion and how to the next work of this study.

Chapter 7

Discussion

Please tell more about conclusion and how to the next work of this study.

Chapter 8

Discussion

Please tell more about conclusion and how to the next work of this study.

Chapter 9

Discussion

Please tell more about conclusion and how to the next work of this study.

Chapter 10

Discussion

Please tell more about conclusion and how to the next work of this study.

Chapter 11

Discussion

Please tell more about conclusion and how to the next work of this study.

Chapter 12

Discussion

Please tell more about conclusion and how to the next work of this study.

Chapter 13

Discussion

Please tell more about conclusion and how to the next work of this study.

Chapter 14

Discussion

Please tell more about conclusion and how to the next work of this study.

Appendix A

Form Penilaian Jurnal

gambar A.1 dan A.2 merupakan contoh bagaimana reviewer menilai jurnal kita.

NO	UNSUR	KETERANGAN	MAKS	KETERANGAN
1	Keefektifan Judul Artikel	Maksimal 12 (dua belas) kata dalam Bahasa Indonesia atau 10 (sepuluh) kata dalam Bahasa Inggris	2	a. Tidak lugas dan tidak ringkas (0) b. Kurang lugas dan kurang ringkas (1) c. Ringkas dan lugas (2)
2	Pencantuman Nama Penulis dan Lembaga Penulis		1	a. Tidak lengkap dan tidak konsisten (0) b. Lengkap tetapi tidak konsisten (0,5) c. Lengkap dan konsisten (1)
3	Abstrak	Dalam Bahasa Indonesia dan Bahasa Inggris yang baik, jumlah 150-200 kata. Isi terdiri dari latar belakang, metode, hasil, dan kesimpulan. Isi tertuang dengan kalimat yang jelas.	2	a. Tidak dalam Bahasa Indonesia dan Bahasa Inggris (0) b. Abstrak kurang jelas dan ringkas, atau hanya dalam Bahasa Inggris, atau dalam Bahasa Indonesia saja (1) c. Abstrak yang jelas dan ringkas dalam Bahasa Indonesia dan Bahasa Inggris (2)
4	Kata Kunci	Maksimal 5 kata kunci terpenting dalam paper	1	a. Tidak ada (0) b. Ada tetapi kurang mencerminkan konsep penting dalam artikel (0,5) c. Ada dan mencerminkan konsep penting dalam artikel (1)
5	Sistematika Pembahasan	Terdiri dari pendahuluan, tinjauan pustaka, metode penelitian, hasil dan pembahasan, kesimpulan dan saran, daftar pustaka	1	a. Tidak lengkap (0) b. Lengkap tetapi tidak sesuai sistematika (0,5) c. Lengkap dan bersistem (1)
6	Pemanfaatan Instrumen Pendukung	Pemanfaatan Instrumen Pendukung seperti gambar dan tabel	1	a. Tidak dimanfaatkan (0) b. Kurang informatif atau komplementer (0,5) c. Informatif dan komplementer (1)
7	Cara Pengacuan dan Pengutipan		1	a. Tidak baku (0) b. Kurang baku (0,5) c. Baku (1)
8	Penyusunan Daftar Pustaka	Penyusunan Daftar Pustaka	1	a. Tidak baku (0) b. Kurang baku (0,5) c. Baku (1)
9	Peristilahan dan Kebahasaan		2	a. Buruk (0) b. Baik (1) c. Cukup (2)
10	Makna Sumbangan bagi Kemajuan		4	a. Tidak ada (0) b. Kurang (1) c. Sedang (2) d. Cukup (3) e. Tinggi (4)

Figure A.1: Form nilai bagian 1.

11	Dampak Ilmiah		7	a. Tidak ada (0) b. Kurang (1) c. Sedang (3) d. Cukup (5) e. Besar (7)
12	Nisbah Sumber Acuan Primer berbanding Sumber lainnya	Sumber acuan yang langsung merujuk pada bidang ilmiah tertentu, sesuai topik penelitian dan sudah teruji.	3	a. < 40% (1) b. 40-80% (2) c. > 80% (3)
13	Derajat Kemutakhiran Pustaka Acuan	Derajat Kemutakhiran Pustaka Acuan	3	a. < 40% (1) b. 40-80% (2) c. > 80% (3)
14	Analisis dan Sintesis	Analisis dan Sintesis	4	a. Sedang (2) b. Cukup (3) c. Baik (4)
15	Penyimpulan	Sangat jelas relevasinya dengan latar belakang dan pembahasan, dirumuskan dengan singkat	3	a. Kurang (1) b. Cukup (2) c. Baik (3)
16	Unsur Plagiat		0	a. Tidak mengandung plagiat (0) b. Terdapat bagian-bagian yang merupakan plagiat (-5) c. Keseluruhannya merupakan plagiat (-20)
TOTAL			36	
Catatan : Nilai minimal untuk diterima 25				

Figure A.2: form nilai bagian 2.

Appendix B

FAQ

M : Kalo Intership II atau TA harus buat aplikasi ? D : Ga harus buat aplikasi tapi harus ngoding

M : Pa saya bingung mau ngapain, saya juga bingung mau presentasi apa? D : Makanya baca de, buka jurnal topik ‘ganteng’ nah kamu baca dulu sehari 5 kali ya, 4 hari udah 20 tuh. Bingung itu tanda kurang wawasan alias kurang baca.

M : Pa saya sudah cari jurnal terindeks scopus tapi ga nemu. D : Kamu punya mata de? coba dicolok dulu. Kamu udah lakuin apa aja? tolong di list laporkan ke grup Tingkat Akhir. Tinggal buka google scholar klik dari tahun 2014, cek nama jurnalnya di scimagojr.com beres.

M : Pa saya belum dapat tempat intership, jadi ga tau mau presentasi apa? D : kamu kok ga nyambung, yang dipresentasikan itu yang kamu baca bukan yang akan kamu lakukan.

M : Pa ini jurnal harus yang terindex scopus ga bisa yang lain ? D : Index scopus menandakan artikel tersebut dalam standar semantik yang mudah dipahami dan dibaca serta bukan artikel asal jadi. Jika diluar scopus biasanya lebih sukar untuk dibaca dan dipahami karena tidak adanya proses review yang baik dan benar terhadap artikel.

M : Pa saya tidak mengerti D : Coba lihat standar alasan

M : Pa saya bingung D : Coba lihat standar alasan

M : Pa saya sibuk D : Mbahmu....

M : Pa saya ganteng D : Ndasmu....

M : Pa saya kece D : wes karepmu lah....

Biasanya anda memiliki alasan tertentu jika menghadapi kendala saat proses bimbingan, disini saya akan melakukan standar alasan agar persepsi yang diterima sama dan tidak salah kaprah. Penggunaan kata alasan tersebut antara lain :

1. Tidak Mengerti : anda boleh menggunakan alasan ini jika anda sudah melakukan tahapan membaca dan meresumekan 15 jurnal. Sudah mencoba dan mempraktekkan teorinya dengan mencari di youtube dan google minimal 6 jam sehari selama 3 hari berturut-turut.

2. Bingung : anda boleh mengatakan alasan bingung setelah maksimal dalam berusaha menyelesaikan tugas bimbingan dari dosen(sudah dilakukan semua). Anda belum bisa mengatakan alasan bingung jika anda masih belum menyelesaikan tugas bimbingan dan poin nomor 1 diatas. Setelah anda menyelesaikan tugas bimbingan secara maksimal dan tahap 1 poin diatas, tapi anda masih tetap bingung maka anda boleh memakai alasan ini.

Bibliography

- [1] Joshua Eckroth. *Python Artificial Intelligence Projects for Beginners: Get up and running with Artificial Intelligence using 8 smart and exciting AI applications*. Packt Publishing Ltd, 2018.
- [2] Herny Februariyanti and Eri Zuliarso. Klasifikasi dokumen berita teks bahasa indonesia menggunakan ontologi. *Dinamik*, 17(1), 2012.
- [3] Deny Kurniawan. Regresi linier. *R-Foundation for Statistical Computing. Vienna, Austria*, 17, 2008.
- [4] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.