

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	Andri Fajar S/1164065	2
1.4.1	TEORI	2
1.4.2	Instalasi	4
1.4.3	Mencoba Learning and predicting	4
1.4.4	Mencoba Model Persistance	5
1.4.5	Mencoba Conventions	9
1.5	Penanganan Error	12
1.6	Yusniar Nur Syarif Sidiq/1164089	13
1.6.1	Teori	13
1.6.2	Instalasi	15
1.6.3	Learning And Predicting	16
1.6.4	Model Persistence	17
1.6.5	Conventions	19
1.6.6	Refitting And Updating Parameters	20
1.6.7	Multiclass VS Multilabel Fitting	21
1.6.8	Penjelasan Error	23
1.7	Imron Sumadireja / 1164076	23
1.7.1	Teori	23
1.7.2	Instalasi	26
1.7.2.1	Proses Instalasi Anaconda dan Library Scikit	26
1.7.3	Mencoba Loading Dataset	28
1.7.4	Learning and Predicting	29
1.7.5	Model Persistence	30
1.7.6	Conventions	32

1.7.7	Penanganan Error	40
2	Related Works	42
2.1	Same Topics	42
2.1.1	Topic 1	42
2.1.2	Topic 2	42
2.2	Same Method	42
2.2.1	Method 1	42
2.2.2	Method 2	42
3	Methods	43
3.1	The data	43
3.2	Method 1	43
3.3	Method 2	43
4	Experiment and Result	44
4.1	Experiment	44
4.2	Result	44
5	Conclusion	45
5.1	Conclusion of Problems	45
5.2	Conclusion of Method	45
5.3	Conclusion of Experiment	45
5.4	Conclusion of Result	45
6	Discussion	46
7	Discussion	47
8	Discussion	48
9	Discussion	49
10	Discussion	50
11	Discussion	51
12	Discussion	52
13	Discussion	53

14 Discussion	54
A Form Penilaian Jurnal	55
B FAQ	58
Bibliography	60

List of Figures

1.1	conda install scikit-learn.	4
1.2	Melihat Version.	5
1.3	Install pip.	5
1.4	Hasil Kompile.	6
1.5	Hasil Kompile.	6
1.6	Hasil Kompile.	7
1.7	Membuka Python	7
1.8	Estimator Sklearn	8
1.9	Mendefinisikan Classifier	8
1.10	Memanggil Classifier	8
1.11	Memprediksi Nilai Baru	8
1.12	Hasil Classifier	8
1.13	Hasil Classifier	8
1.14	Pickle Python	9
1.15	Classifier Pickle	9
1.16	Joblib	9
1.17	Deklarasi Numpy	10
1.18	Contoh Casting	10
1.19	FitTransform	10
1.20	Regresi Yang Dilempar	11
1.21	Memperbaharui Parameter	11
1.22	MultiClass	12
1.23	MultiClass biner 2D	12
1.24	MultiLabel	13
1.25	Error Import	13
1.26	Instal Library Joblib	14
1.27	Import Library Joblib	14
1.28	conda install scikit-learn.	16

1.29	Melihat Version.	17
1.30	Install pip.	17
1.31	Hasil Kompile.	18
1.32	Dataset.	18
1.33	Joblib Errorr.	23
1.34	Hasil Joblib.	24
1.35	Download Aplikasi Anaconda	27
1.36	Proses Instalasi Aplikasi	27
1.37	Proses Instalasi Aplikasi	28
1.38	Proses Instalasi Aplikasi	29
1.39	Proses Instalasi Aplikasi	30
1.40	Proses Instalasi Aplikasi	31
1.41	Proses Instalasi Aplikasi	32
1.42	Proses Instalasi Aplikasi	33
1.43	Proses Instalasi Aplikasi	34
1.44	Proses Instalasi Aplikasi	35
1.45	Instalasi Library	35
1.46	Instalasi Library	36
1.47	Instalasi Library	36
1.48	Instalasi Library	37
1.49	Loading dataset	37
1.50	Error	40
1.51	Instalasi	40
1.52	Hasil	41
A.1	Form nilai bagian 1.	56
A.2	form nilai bagian 2.	57

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*[3]. 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 Andri Fajar S/1164065

1.4.1 TEORI

1. Definisi, Sejarah, Dan Perkembangan Sejarah AI

Didefinisikan kecerdasan yang ditunjukkan oleh suatu entitas buatan. Umumnya dianggap komputer. Kecerdasan Buatan (Artificial Intelligence atau AI) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan dimasukkan ke

dalam mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Kecerdasan Buatan (Artificial Intelligence atau AI) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya di anggap komputer. Kecerdasan diciptakan dan dimasukkan melakukan pekerjaan seperti yang dapat dilakukan manusia.

Sejarah dan perkembangan kecerdasan buatan terjadi pada musim panas tahun 1956 tercatat adanya seminar mengenai AI di Darmouth College. Seminar pada waktu itu dihadiri oleh sejumlah pakar komputer dan membahas potensi komputer dalam meniru kepandaian manusia. Akan tetapi perkembangan yang sering terjadi semenjak diciptakannya LISP, yaitu bahasa kecerdasan buatan yang dibuat tahun 1960 oleh John McCarthy. Istilah pada kecerdasan buatan atau Artificial Intelligence diambil dari Marvin Minsky dari MIT. Dia menulis karya ilmiah berjudul Step towards Artificial Intelligence, The Institute of radio Engineers Proceedings 49, January 1961[?].

Supervised learning merupakan sebuah pendekatan dimana sudah terdapat data yang dilatih, dan terdapat variable yang ditargetkan sehingga tujuan dari pendekatan ini adalah mengelompokkan suatu data ke data yang sudah ada. Sedangkan unsupervised learning tidak memiliki data latih, sehingga dari data yang ada, kita mengelompokkan data tersebut menjadi 2 bagian atau 3 bagian dan seterusnya.

Klasifikasi adalah salah satu topik utama dalam data mining atau machine learning. Klasifikasi yaitu suatu pengelompokan data dimana data yang digunakan tersebut mempunyai kelas label atau target.

Regresi adalah Supervised learning tidak hanya mempelajari classifier, tetapi juga mempelajari fungsi yang dapat memprediksi suatu nilai numerik. Contoh, ketika diberi foto seseorang, kita ingin memprediksi umur, tinggi, dan berat orang yang ada pada foto tersebut.

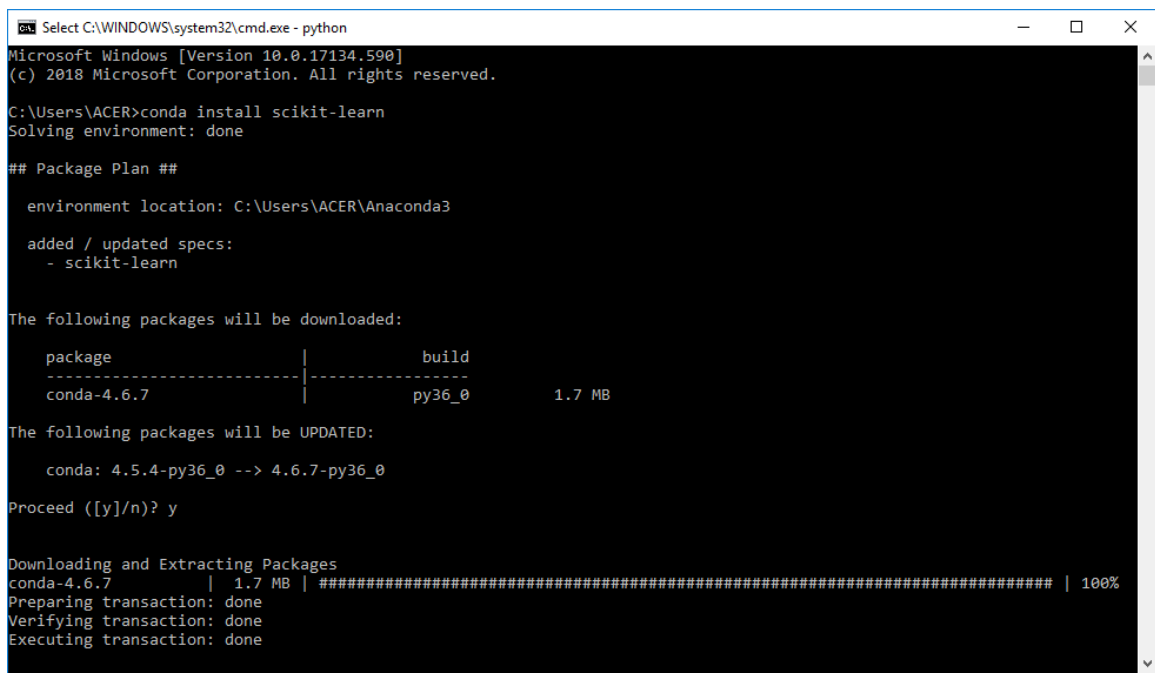
Data set adalah cabang aplikasi dari Artificial Intelligence/Kecerdasan Buatan yang fokus pada pengembangan sebuah sistem yang mampu belajar sendiri tanpa harus berulang kali di program oleh manusia.

2. Training set yaitu jika pasangan objek, dan kelas yang menunjuk pada objek tersebut adalah suatu contoh yang telah diberi label akan menghasilkan suatu algoritma pembelajaran.

Testing set digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar.

1.4.2 Instalasi

- Memberikan perintah conda install scikit-learn di cmd, lihat gambar 1.1
- Melihat versinya dengan memberikan perintah conda -version dan python -version, lihat gambar 1.2
- Install pip, lihat pada gambar 1.3
- Hasil Kompile, lihat gambar 1.4
- Import dataset kemudian load iris dan data dari digits, lihat gambar 1.5
- Melihat data digits



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

C:\Users\ACER>conda install scikit-learn
Solving environment: done

## Package Plan ##

  environment location: C:\Users\ACER\Anaconda3

  added / updated specs:
    - scikit-learn

The following packages will be downloaded:

  package | build | size
  -----|-----|-----
  conda-4.6.7 | py36_0 | 1.7 MB

The following packages will be UPDATED:

  conda: 4.5.4-py36_0 --> 4.6.7-py36_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
conda-4.6.7 | 1.7 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Figure 1.1: conda install scikit-learn.

1.4.3 Mencoba Learning and predicting

1. Buka CMD lalu ketikkan perintah Python.
2. "from sklearn import svm" artinya akan memanggil dan menggunakan estimator dari kelas sklearn.svm.SVC

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

C:\Users\ACER>python --version
Python 3.6.5
```

Figure 1.2: Melihat Version.

```
C:\Users\ACER>pip install -U scikit-learn
Collecting scikit-learn
  Downloading https://files.pythonhosted.org/packages/ee/c8/c89ebdc0d7dbba6e6fd222daabd257da3c28a967dd7c352d4272b2e1cef6
/scikit_learn-0.20.2-cp36-cp36m-win32.whl (4.3MB)
    100% |#####| 4.3MB 622kB/s
Collecting numpy>=1.8.2 (from scikit-learn)
  Downloading https://files.pythonhosted.org/packages/84/10/f1f99ba67aff4c3fb033571e87876ed0403114b13bc70cc125372b0c1dcb
/numpy-1.16.1-cp36-cp36m-win32.whl (10.0MB)
    100% |#####| 10.0MB 863kB/s
Collecting scipy>=0.13.3 (from scikit-learn)
  Downloading https://files.pythonhosted.org/packages/b7/b6/eedfa8b002ffae365c3f957154a9c29cb91a8e657808749c78758f0f8110
/scipy-1.2.1-cp36-cp36m-win32.whl (26.9MB)
    100% |#####| 27.0MB 89kB/s
Installing collected packages: numpy, scipy, scikit-learn
Successfully installed numpy-1.16.1 scikit-learn-0.20.2 scipy-1.2.1
You are using pip version 18.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Figure 1.3: Install pip.

3. disini gamma didefinisikan secara manual
4. Estimator clf (for classifier) pertama kali dipasang pada model. Ini dilakukan dengan melewati training set ke metode fit. Untuk training set, akan menggunakan semua gambar dari set data yang ada, kecuali untuk gambar terakhir, yang dicadangan untuk prediksi. Pada skrip dibawah memilih training set dengan sintaks Python [: -1], yang menghasilkan array baru yang berisi semua kecuali item terakhir dari digits.data
5. Pada penggalan skrip dibawah, ini menunjukan prediksi nilai baru menggunakan gambar terakhir dari digits.data.

1.4.4 Mencoba Model Persistence

1. "from sklearn import svm" artinya akan mengimport sebuah Support Vector Machine(SVM) yang merupakan algoritma classification yang akan diambil dari Scikit-Learn.

```
C:\Users\ACER>python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('andri')
andri
```

Figure 1.4: Hasil Kompile.

```
>>> from sklearn import datasets
>>> iris = datasets.load_iris()
>>> digits = datasets.load_digits()
>>>
```

Figure 1.5: Hasil Kompile.

2. "from sklearn import datasets" artinya akan mengambil package datasets dari Scikit-Learn.
3. ketikan, `clf = svm.SVC(gamma='scale')` berfungsi untuk mendeklarasikan suatu value yang bernama `clf` yang berisi `gamma`.
4. Ketikan, `X, y = iris.data, iris.target`, artinya `X` sebagai data iris, dan `y` merupakan larik target.
5. Ketikan, `clf.fit(X, y)` berfungsi untuk melakukan pengujian classifier. hasilnya seperti ini

Dari gambar diatas dapat dijelaskan bahwa akan mengimport Pickle dari Python. Pickle digunakan untuk serialisasi dan de-serialisasi struktur objek Python. Objek apa pun dengan Python dapat di-Pickle sehingga dapat disimpan di disk. kemudian menyimpan data objek ke file CLF sebelumnya dengan menggunakan function `pickle.dumps(clf)`.

7. Setelah mengetikan fungsi fungsi diatas, selanjutnya ketikan "`clf2 = pickle.loads(s)`" yang artinya `pickle.loads` digunakan untuk memuat data pickle dari string byte. "S" dalam loads mengacu pada fakta bahwa dalam Python 2, data dimuat dari string.

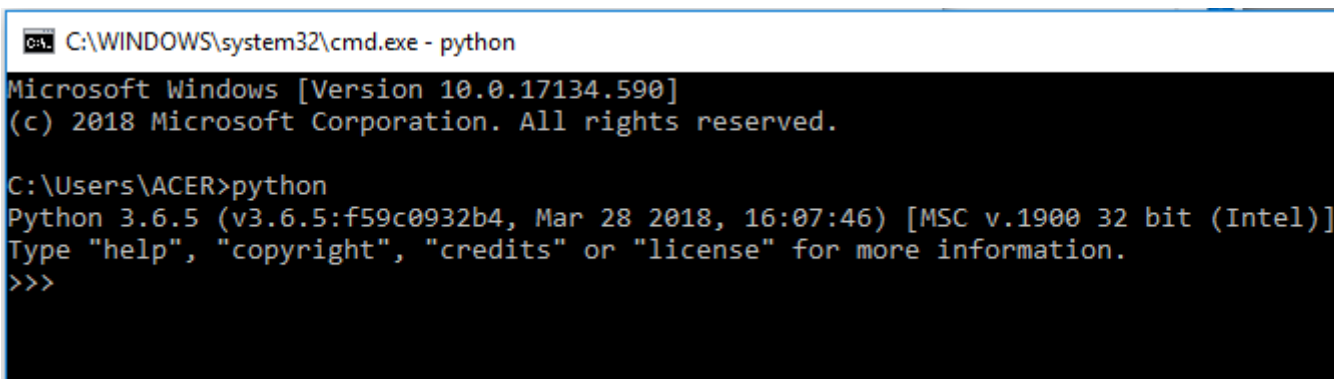
Pada gambar diatas dilakukan pengujian nilai baru dengan menggunakan "`clf2.predict(X[0:1])`" dengan target asumsinya (0,1) hasilnya berbentuk array.

```
>>> 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.]]
>>>
```

Figure 1.6: Hasil Kompile.

9. "from joblib import dump , load" yang artinya akan Merekonstruksi objek Python dari file yang sudah ada.

dump(clf, 'filename.joblib') akan merekontruksi file CLF yang tadi sudah dideklarasikan.
 clf = load('filename.joblib') untuk mereload model yang sudah di Pickle



```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ACER>python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)]
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figure 1.7: Membuka Python


```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ACER>python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)]
Type "help", "copyright", "credits" or "license" for more information.
>>> from sklearn import svm
>>>
```

Figure 1.8: Estimator Sklearn

```
>>> clf = svm.SVC(gamma=0.001, C=100.)
>>>
```

Figure 1.9: Mendefinisikan Classifier

```
>>> 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=0.001, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Figure 1.10: Memanggil Classifier

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

Figure 1.11: Memprediksi Nilai Baru

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

Figure 1.12: Hasil Classifier

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

6.

Figure 1.13: Hasil Classifier

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

Figure 1.14: Pickle Python

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

Figure 1.15: Classifier Pickle

1.4.5 Mencoba Conventions

1. Import numpy as np, digunakan untuk mengimport Numpy sebagai np.
From sklearn import randomprojection artinya modul yang mengimplementasikan cara sederhana dan efisien secara komputasi untuk mengurangi dimensi data dengan memperdagangkan sejumlah akurasi yang terkendali (sebagai varian tambahan) untuk waktu pemrosesan yang lebih cepat dan ukuran model yang lebih kecil.

Pada gambar diatas dapat dijelaskan bahwa :

rng = np.random.RandomState(0), digunakan untuk menginisialisasikan random number generator.

X = rng.rand(10, 2000) artinya akan merandom value antara 10 sampai 2000.

X = np.array(X, dtype='float32') Array numpy terdiri dari buffer memori "mentah" yang diartikan sebagai array melalui "views". Anda dapat menganggap semua array numpy sebagai tampilan. Mendeklarasikan X sebagai float32.

3. Dalam contoh ini, X adalah float32, yang dilemparkan ke float64 oleh fittransform (X).
4. Target regresi dilemparkan ke float64 dan target klasifikasi dipertahankan.
list(clf.predict(irisdata[:3])), akan memprediksi 3 data dari iris.
clf.fit irisdata, iristargetnames[iristarget] menguji classifier dengan ada targetnya yaitu irisnya sendiri.

```
>>> from joblib import dump, load
>>> dump(clf, 'filename.joblib')
['filename.joblib']
>>> clf = load('filename.joblib')
```

Figure 1.16: Joblib

```
>>> import numpy as np
>>> from sklearn import random_projection
```

Figure 1.17: Deklarasi Numpy

```
>>> rng = np.random.RandomState(0)
>>> X = rng.rand(10, 2000)
>>> X = np.array(X, dtype='float32')
>>> X.dtype
dtype(<'float32'>)
```

Figure 1.18: Contoh Casting

`list(clf.predict(irisdata[:3]))`, setelah diuji maka akan muncul datanya seperti dibawah ini

Di sini, prediksi pertama () mengembalikan array integer, karena `iristarget` (array integer) yang digunakan sesuai. Prediksi kedua () mengembalikan array string, karena `iristargetnames` cocok.

5. Refitting dan Memperbaharui Parameter

`y = rngbinomial(1, 0.5, 100)` , random value dengan angka binomial atau suku dua untuk `y`

`clfsetparams(kernel='linear')fit(X, y)` mengubah kernel default menjadi linear
`clfsetparams(kernel='rbf', gamma='scale')fit(X, y)` Di sini, kernel default rbf pertama kali diubah menjadi linear melalui

`SVCsetparams()` setelah estimator dibuat, dan diubah kembali ke rbf untuk mereparasi estimator dan membuat prediksi kedua.

6. MultiClass VS MultiLabel Classifier

`from sklearn.multiclass import OneVsRestClassifier` , adalah ketika kita ingin melakukan klasifikasi multiclass atau multilabel dan baik untuk menggunakan `OneVsRestClassifier` per kelas. Untuk setiap classifier, kelas tersebut dipasang terhadap semua kelas lainnya. (Ini cukup jelas dan itu berarti bahwa masalah klasifikasi multiclass / multilabel dipecah menjadi beberapa masalah klasifikasi biner).

```
>>> transformer = random_projection.GaussianRandomProjection()
>>> X_new = transformer.fit_transform(X)
>>> X_new.dtype
dtype(<'float64'>)
```

Figure 1.19: FitTransform

```

>>> from sklearn import datasets
>>> from sklearn.svm import SVC
>>> iris = datasets.load_iris()
>>> clf = SVC(gamma='scale')
>>> clf.fit(iris.data, 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)
>>> list(clf.predict(iris.data[:3]))
[0, 0, 0]
>>> clf.fit(iris.data, iris.target_names[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)
>>> list(clf.predict(iris.data[:3]))
['setosa', 'setosa', 'setosa']

```

Figure 1.20: Regresi Yang Dilempar

```

>>> import numpy as np
>>> from sklearn.svm import SVC
>>> rng = np.random.RandomState(0)
>>> X = rng.rand(100, 10)
>>> y = rng.binomial(1, 0.5, 100)
>>> X_test = rng.rand(5, 10)
>>> clf = SVC()
>>> clf.set_params(kernel='linear').fit(X, y)
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)
>>> clf.predict(X_test)
array([1, 0, 1, 1, 0])
>>> clf.set_params(kernel='rbf', gamma='scale').fit(X, 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)
>>> clf.predict(X_test)
array([1, 0, 1, 1, 0])

```

Figure 1.21: Memperbaharui Parameter

```

>>> from sklearn.svm import SVC
>>> from sklearn.multiclass import OneVsRestClassifier
>>> from sklearn.preprocessing import LabelBinarizer
>>> X = [[1, 2], [2, 4], [4, 5], [3, 2], [3, 1]]
>>> y = [0, 0, 1, 1, 2]
>>> classif = OneVsRestClassifier(estimator=SVC(gamma='scale',
... random_state=0))
>>> classif.fit(X, y).predict(X)
array([0, 0, 1, 1, 2])

```

Figure 1.22: MultiClass

```

>>> y = LabelBinarizer().fit_transform(y)
>>> classif.fit(X, y).predict(X)
array([[1, 0, 0],
       [1, 0, 0],
       [0, 1, 0],
       [0, 0, 0],
       [0, 0, 0]])

```

Figure 1.23: MultiClass biner 2D

from sklearn.preprocessing import LabelBinarizer ,adalah kelas utilitas untuk membantu membuat matriks indikator label dari daftar label multi-kelas Dalam gambar dibawah, classifier cocok pada array 1d label multiclass dan oleh karena itu metode predict () memberikan prediksi multiclass yang sesuai.

7. Di sini, classifier cocok () pada representasi label biner 2d dari y, menggunakan LabelBinarizer. Dalam hal ini predict () mengembalikan array 2d yang mewakili prediksi multilabel yang sesuai.
8. from sklearn.preprocessing import MultiLabelBinarizer , artinya Transformasi antara iterable dari iterables dan format multilabel.
Dalam hal ini, penggolongnya sesuai pada setiap instance yang diberi beberapa label. MultiLabelBinarizer digunakan untuk membuat binarize array 2d dari multilabel agar sesuai. Hasilnya, predict () mengembalikan array 2d dengan beberapa label yang diprediksi untuk setiap instance.

1.5 Penanganan Error

1. Berikut ini merupakan eror yang ditemui pada saat melakukan percobaan skrip.
2. Pada gambar eror diatas, kode erornya adalah "ImportError: No Module Named" artinya mengalami masalah saat mengimpor modul yang ditentukan.

```

>>> from sklearn.preprocessing import MultiLabelBinarizer
>>> y = [[0, 1], [0, 2], [1, 3], [0, 2, 3], [2, 4]]
>>> y = MultiLabelBinarizer().fit_transform(y)
>>> 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]])
>>>

```

Figure 1.24: MultiLabel

```

>>> from joblib import dump, load
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named joblib

```

Figure 1.25: Error Import

3. Solusinya bisa dilakukan seperti berikut :
 eror diatas terjadi dikarenakan Library Joblib belum terinstal pada PC. Maka dari itu sekarang kita harus menginstalnya dulu.
4. Buka CMD, kemudian ketikan "pip install joblib" tunggu sampai instalasi berhasil seperti gambar berikut.
5. Apabila sudah terinstall, dapat dilakukan lagi import library joblib, maka akan berhasil seperti dibawah berikut

1.6 Yusniar Nur Syarif Sidiq/1164089

1.6.1 Teori

1. Definisi, Sejarah, Dan Perkembangan Sejarah AI

Kecerdasan buatan merupakan sebuah bidang dalam ilmu computer yang begitu penting di zaman ini dan masa yang akan datang guna mewujudkan sebuah sistem computer yang begitu cerdas. Kecerdasan buatan sudah berkembang begitu pesat dalam 20 tahun terakhir seiring dengan adanya kebutuhan perangkat yang cerdas pada bidang industry dan rumah tangga.

Artificial Intelligence atau biasa di singkat dengan AI berasal dari bahasa latin yang dimana intelligence berarti saya paham. AI dimulai dari kemunculan sebuah komputer pada tahun 1940-an, akan tetapi perkembangannya

```
C:\Users\ACER>pip install joblib
Collecting joblib
  Downloading https://files.pythonhosted.org/packages/cd/c1/50a758e8247561e58cb8730
/joblib-0.13.2-py2.py3-none-any.whl (278kB)
    100% |████████████████████████████████████████| 286kB 535kB/s
Installing collected packages: joblib
Successfully installed joblib-0.13.2
You are using pip version 18.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command
C:\Users\ACER>
```

Figure 1.26: Instal Library Joblib

```
>>> from joblib import dump, load
>>> dump(clf, 'filename.joblib')
['filename.joblib']
>>> clf = load('filename.joblib')
>>>
```

Figure 1.27: Import Library Joblib

dapat dilacak pada zaman Mesir Kuno. Dalam masa ini dimana perhatian difokuskan dengan kemampuan komputer dalam mengerjakan sesuatu yang dapat dilakukan oleh manusia sehingga kompute tersebut dapat meniru kemampuan dan prilaku manusia secara cerdas.

Pada tahun 1955, Newell dan juga Simon telah mengembangkan The Logic Theorist, yaitu program AI pertama. Dimana program tersebut mempresentasikan sebuah masalah sebagai model pohon, lalu diselesaikan dengan cara memilih cabang yang akan mewujudkan kesimpulan terbenar dan tepat. Program AI tersebut berdampak sangat besar dan dapat mendeda batu loncatan yang cukup penting dalam mengembangkan bidang AI. Sekitar tahun 1956 dimana orang yang dianggap sebagai bapak AI yaitu John McCarthy telah menyelenggarakan konferensi guna menarik para ahli dibidang komputer untuk bertemu, dengan acara yang diberi nama The Dartmouth Summer Research Project On Artificial Intelligence. Dalam konferensi tersebut telah mempertemukan pendiri dan pengembang AI. Pada konferensi tersebut bapak AI John McCarthy mengusulkan definisi AI yaitu merupakan cabang dari sebuah ilmu komputer yang dapat berfokus terhadap pengembangan computer sehingga da-

pat memiliki kemampuan dan juga perilaku seperti manusia.[1].

2. Definisi Supervised Learning, Unsupervised Learning, Klasifikasi, Dan Regresi

Supervised Learning merupakan sebuah pendekatan yang dimana terdapat data dan variable yang telah ditargetkan sehingga pendekatan tersebut bertujuan untuk mengelompokkan sebuah data ke data yang sudah ada, beda dengan Unsupervised learning yang tidak mempunyai data, sehingga data yang ada harus di kelompokkan menjadi beberapa bagian.

Klasifikasi merupakan sebuah kegiatan penggolongan atau pengelompokkan. Menurut kamus besar bahasa Indonesia yang dimana klasifikasi merupakan penyusunan sistem di dalam kelompok atau golongan berdasarkan kaidah atau standar yang telah ditetapkan. Regresi merupakan sebuah metode analisis statistic yang akan digunakan untuk melihat pengaruh variable.

3. Definisi Dataset, Training Set, Dan Testing Set

Dataset merupakan sebuah objek yang akan mempresentasikan sebuah data dan relasinya di memory. Struktur pada dataset ini mirip dengan data yang ada di dalam database. Training set merupakan bagian dari dataset yang berperan dalam membuat prediksi atau algoritma sesuai tujuan masing – masing. Testing set merupakan bagian dari dataset yang akan di tes guna melihat keakuratan atau ketepatan datanya.

1.6.2 Instalasi

- Memberikan perintah conda install scikit-learn di cmd, lihat gambar 1.1
- Melihat versinya dengan memberikan perintah conda –version dan python –version, lihat gambar 1.2
- Install pip, lihat pada gambar 1.3
- Hasil Kompile, lihat gambar 1.4

Dataset adalah objek seperti kamus yang menyimpan semua data dan berupa metadata tentang data. Data tersebut disimpan di .data anggota yang merupakan array. Misalnya dalam kasus dataset digit, memberikan akses ke fitur yang dapat digunakan untuk mengklarifikasikan sampel digit, lihat gambar 1.5.


```
C:\Users\NS>conda install scikit-learn
Solving environment: done

## Package Plan ##

  environment location: C:\Users\NS\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
```

Figure 1.28: conda install scikit-learn.

1.6.3 Learning And Predicting

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

Pada baris pertama akan melakukan import datasets dari folder sklearn. Pada baris kedua akan mengambil datasets iris dari folder iris Pada baris ketiga yaitu akan mengambil datasets digit dari folder digit Pada baris keempat akan melakukan import svm dari folder sklearn Pada baris Kelimla akan melakukan

```
Command Prompt

Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\NS>conda --version
conda 4.5.4

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

Figure 1.29: Melihat Version.

```
C:\Users\NS>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:\users\ns\anaconda3\lib\site-packages (from scikit-learn) (1.14.3)
Requirement not upgraded as not directly required: scipy>=0.13.3 in c:\users\ns\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 1.30: Install pip.

deklarasi gamma Pada baris kenam akan membaca data digits dan target digits
Pada baris ketujuh akan melakukan prediksi data

1.6.4 Model Persistence

```
from sklearn import svm
from sklearn import datasets
clf = svm.SVC(gamma='scale')
iris = datasets.load_iris()
X, y = iris.data, iris.target
clf.fit(X, 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)
```

```

C:\Users\NS>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('Yusniar')
Yusniar
>>>

```

Figure 1.31: Hasil Kompile.

```

>>> 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.]]
>>>

```

Figure 1.32: Dataset.

```

import pickle
s = pickle.dumps(clf)
clf2 = pickle.loads(s)
clf2.predict(X[0:1])
array([0])
y[0]
0

```

Pada baris pertama akan melakukan import svm dari folder sklearn Pada baris kedua akan melakukan import datasets dari folder sklearn Pada baris ketiga akan mendeklarasikan gamma dengan scale Pada baris keempat akan

mengambil datasets iris dari folder iris Pada bari kelima akan mendeklarasikan data iris dan target iris dengan X dan y Pada baris ke delapan akan melakukan import pickle Pada baris kesembilan akan memanggil deklarasi scale dengan s Pada baris kesepuluh akan mengambil data pickle Pada baris berikutnya akan melakukan prediksi dan memunculkan hasilnya

1.6.5 Conventions

1.

```
import numpy as np
from sklearn import random_projection

rng = np.random.RandomState(0)
X = rng.rand(10, 2000)
X = np.array(X, dtype='float32')
X.dtype
dtype('float32')

transformer = random_projection.GaussianRandomProjection()
X_new = transformer.fit_transform(X)
X_new.dtype
dtype('float64')
```

Pada baris pertama akan melakukan import numpy. Pada baris berikutnya akan melakukan import akan tetapi dengan data yang random pada folder sklearn. Pada baris berikutnya akan menentukan nilai array dan radian pada X. Pada baris berikutnya menentukan type data X. Pada baris berikutnya dijelaskan bahwa typenya yaitu float 32. Pada baris berikutnya akan menentukan random project dengan menggunakan transformer. Baris berikutnya membuat nilai X baru dan mengirimnya dengan menggunakan transformer.fit_transform. Pada baris berikutnya membuat type baru pada X. Pada baris berikutnya hasil dari type tersebut adalah float64.

2.

```

from sklearn import datasets
from sklearn.svm import SVC
iris = datasets.load_iris()
clf = SVC(gamma='scale')
clf.fit(iris.data, 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)

list(clf.predict(iris.data[:3]))
[0, 0, 0]

clf.fit(iris.data, iris.target_names[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)

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

```

Pada baris pertama melakukan import datasets dari folder sklearn. Pada baris berikutnya akan melakukan import SVC dari folder sklearn. Pada baris berikutnya akan mengambil datasets iris dari folder iris. Pada baris berikutnya akan melakukan deklarasi pada gamma Pada baris berikutnya membaca data iris dan target iris. Pada baris berikutnya akan melakukan prediksi dan hasilnya adalah 0,0,0. Pada baris berikutnya akan memberikan nama pada setiap data iris. Pada baris berikutnya akan melakukan prediksi dan akan memunculkan hasilnya.

1.6.6 Refitting And Updating Parameters

```
import numpy as np
```

```

from sklearn.svm import SVC

rng = np.random.RandomState(0)
X = rng.rand(100, 10)
y = rng.binomial(1, 0.5, 100)
X_test = rng.rand(5, 10)

clf = SVC()
clf.set_params(kernel='linear').fit(X, y)
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) clf.predict(X_test)
array([1, 0, 1, 1, 0])

clf.set_params(kernel='rbf', gamma='scale').fit(X, 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)
clf.predict(X_test)
array([1, 0, 1, 1, 0])

```

Pada baris pertama akan melakukan import pada numpy. Pada baris selanjutnya akan melakukan import SVC dari folder sklearn. Pada baris berikutnya akan menentukan nilai dari RandomState. Pada baris berikutnya akan menentukan metode yang digunakan yaitu linier dan akan membaca data SVC. Hasilnya berupa da array.

1.6.7 Multiclass VS Multilabel Fitting

1.

```

from sklearn.svm import SVC
from sklearn.multiclass import OneVsRestClassifier
from sklearn.preprocessing import LabelBinarizer

```

```

X = [[1, 2], [2, 4], [4, 5], [3, 2], [3, 1]]
y = [0, 0, 1, 1, 2]
classif = OneVsRestClassifier(estimator=SVC(gamma='scale', random_state=0))
classif.fit(X, y).predict(X)
array([0, 0, 1, 1, 2])

```

Pada baris pertama akan melakukan import SVC pada folder sklearn. Pada baris berikutnya akan melakukan import class. Class yang digunakan adalah OneVsRestClassifier sehingga akan menciptakan outputan array.

2.

```

y = LabelBinarizer().fit_transform(y)
classif.fit(X, y).predict(X)
array([[1, 0, 0],
       [1, 0, 0],
       [0, 1, 0],
       [0, 0, 0],
       [0, 0, 0]])

```

Codingan tersebut akan membaca output pada label dan akan mengeluarkan data array.

3.

```

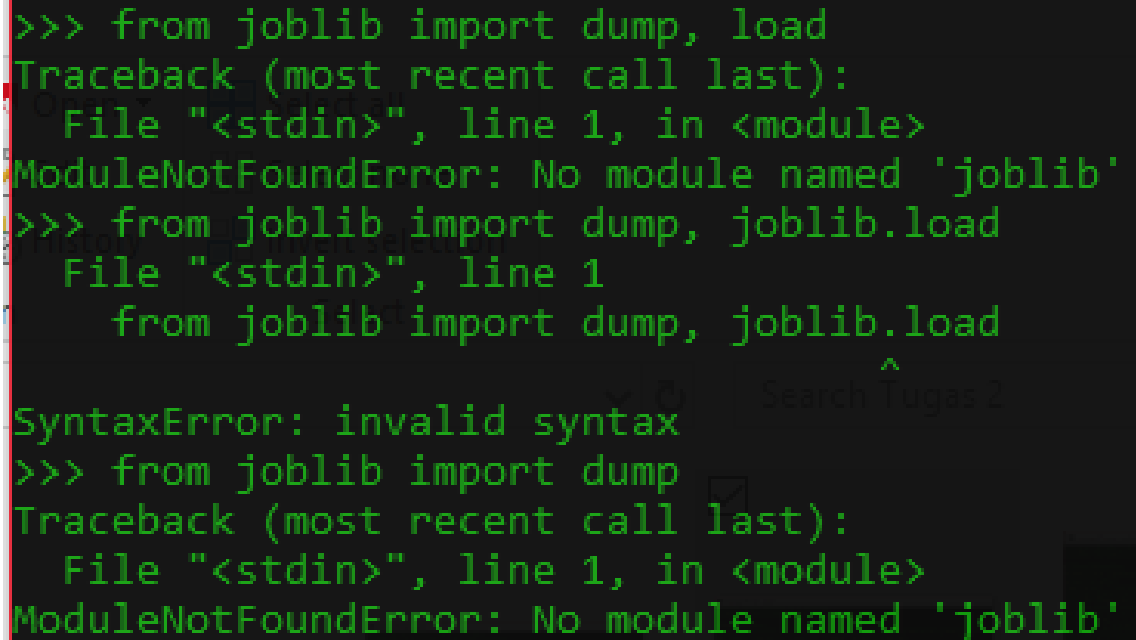
from sklearn.preprocessing import MultiLabelBinarizer
y = [[0, 1], [0, 2], [1, 3], [0, 2, 3], [2, 4]]
y = MultiLabelBinarizer().fit_transform(y)
classif.fit(X, y).predict(X)
array([[1, 1, 0, 0, 0],
       [1, 0, 1, 0, 0],
       [0, 1, 0, 1, 0],
       [1, 0, 1, 0, 0],
       [1, 0, 1, 0, 0]])

```

Pada baris pertama akan melakukan import Multilabelbinarizer yang berisikan data-data lalu di transform sehingga memunculkan data array.

1.6.8 Penjelasan Error

Dimana Errorr Tersebut dapat di lihat pada gambar Figure Joblib Errorr



```
>>> from joblib import dump, load
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'joblib'
>>> from joblib import dump, joblib.load
  File "<stdin>", line 1
    from joblib import dump, joblib.load
                                ^
SyntaxError: invalid syntax
>>> from joblib import dump
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'joblib'
```

Errorr Y.PNG

Figure 1.33: Joblib Errorr.

Codingan yang eror tersebut adalah

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

Hal ini dikarenakan belum terinstallnya file joblib di dalam pc anda. Untuk menginstallnya cukup berikan perintah

```
pip install joblib
```

maka hasilnya akan nampak seperti pada gambar hasil joblib

=====

1.7 Imron Sumadireja / 1164076

1.7.1 Teori

1. Pengertian

Joblib Y.PNG

```
>>> from joblib import dump, load
>>> dump(clf, 'filename.joblib')
['filename.joblib']
>>> clf = load('filename.joblib')
>>>
```

Figure 1.34: Hasil Joblib.

Kecerdasan Buatan Artificial Intelligence merupakan salah satu bagian dari ilmu komputer yang mempelajari cara membuat mesin komputer dapat melakukan pekerjaan sebaik bahkan lebih baik dari yang dilakukan oleh manusia. Agar mesin dapat bekerja layaknya manusia maka perlu diberi bekal pengetahuan, sehingga mempunyai kemampuan untuk menalar. Menurut para ahli kecerdasan buatan seperti berikut:

- H. A. Simon: Kecerdasan buatan Artificial Intelligence merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas.
- Rich and Knight: Kecerdasan buatan Artificial Intelligence merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia.

2. Sejarah dan Perkembangan

Kata intelligence berasal dari bahasa latin intelligo yang memiliki arti saya paham. Arti dasar dari intelligence merupakan kemampuan untuk memahami dan melakukan aksi. Area Kecerdasan Buatan Artificial Intelligence, bermula pada saat kemunculan komputer sekitar tahun 1940-an, walaupun sejarah perkembangannya dapat dilacak sejak zaman Mesir kuno. Pada masa saat ini, perhatian difokuskan pada kemampuan komputer mengerjakan sesuatu yang dapat dilakukan oleh manusia. Dalam hal ini, komputer tersebut dapat meniru kemampuan kecerdasan dan perilaku manusia dengan akurasi yang cukup baik [5].

Pada akhir tahun 1955, Newell dan Simon mengembangkan The Logic Theorist, program AI pertama, program ini merepresentasikan masalah sebagai

model pohon, lalu penyelesaiannya dengan memilih cabang yang akan menghasilkan kesimpulan yang paling benar. Pada tahun 1956 John McCarthy dari Massachusetts Institute of Technology dianggap sebagai bapak AI, menyelenggarakan konferensi untuk menarik para ahli komputer bertemu, dengan nama kegiatan The Dartmouth Summer Research Project on Artificial Intelligence. Konferensi Dartmouth itu mempertemukan para pendiri AI, dan bertugas untuk meletakkan dasar bagi masa depan pengembangan dan penelitian AI. John McCarthy saat itu mengusulkan definisi AI adalah AI merupakan cabang dari ilmu komputer yang berfokus pada pengembangan komputer untuk dapat memiliki kemampuan dan berperilaku seperti manusia[2].

3. Supervised Learning dan Unsupervised Learning

Supervised Learning merupakan suatu pendekatan dimana sudah terdapat data yang dilatih, dan terdapat variable yang ditargetkan sehingga tujuan dari pendekatan ini adalah mengelompokkan suatu data ke data yang sudah ada. Sebagai contoh, ketika Anda memiliki sejumlah buku yang sudah dibeli dengan beberapa kategori. Misalnya, kategori buku akademik, dan buku novel. Selanjutnya Anda membeli sejumlah buku baru, maka Anda harus mengidentifikasi buku tersebut, dan memasukkannya dalam kategori yang sudah ada.

Unsupervised Learning merupakan suatu pendekatan namun tidak memiliki data yang dilatih, sehingga dari data yang ada, kita dapat mengelompokkan data tersebut menjadi 2 bagian atau 3 bagian dan seterusnya. Sebagai contoh, Anda belum pernah membeli sejumlah buku, suatu hari Anda membeli sejumlah buku dan ingin membaginya kedalam beberapa kategori agar mudah dicari. Anda akan mengidentifikasi buku mana yang mirip. Dalam hal ini, kita memilih buku berdasarkan isinya.

4. Klasifikasi dan Regresi

Klasifikasi merupakan penempatan objek-objek ke salah satu dari beberapa kategori yang telah ditentukan sebelumnya. Klasifikasi banyak digunakan untuk memprediksi kelas pada suatu label atau atribut tertentu, yaitu dengan mengklasifikasi data membangun model berdasarkan training set dan nilai-nilai dalam mengklasifikasikan data yang baru. Regresi dibedakan menjadi 2, diantaranya regresi linear dan regresi nonlinear.

- Regresi Linear Regresi Linear merupakan bentuk hubungan di mana variabel bebas x maupun variabel tergantung y sebagai faktor yang berpangkat satu.
- Regresi Nonlinear Regresi Nonlinear merupakan bentuk hubungan atau fungsi di mana variabel x dan variabel tidak bebas y dapat berfungsi sebagai faktor atau variabel dengan pangkat tertentu.

5. Data set, Training set, dan Testing set

Untuk melakukan data set, training set, dan testing set diperlukan beberapa langkah, diantaranya:

- Membuat model atau mesin untuk memeriksa data,
- Membuat model atau mesin belajar dari kesalahannya,
- Membuat kesimpulan tentang seberapa baik kinerja model atau mesin tersebut.

(a) Data set

Data set ini mencakup sekumpulan contoh input yang modelnya akan cocok atau dilatih dengan menyesuaikan parameter.

(b) Training set

Training set diperlukan oleh model atau mesin agar dapat dilatih. Dengan menghitung kerugian tingkat kesalahan yang dilakukan model atau mesin menghasilkan pada set validasi pada titik tertentu, agar kita tahu seberapa akuratnya. Selanjutnya, model akan menyesuaikan parameternya berdasarkan hasil evaluasi yang sering pada training set ini.

(c) Testing set

Testing set sangat penting untuk menguji generalisasi model atau mesin. Dengan testing set ini, kita bisa mendapatkan akurasi kinerja model atau mesin.

1.7.2 Instalasi

1.7.2.1 Proses Instalasi Anaconda dan Library Scikit

1. Pertama kita unduh terlebih dahulu aplikasi anaconda, seperti gambar berikut

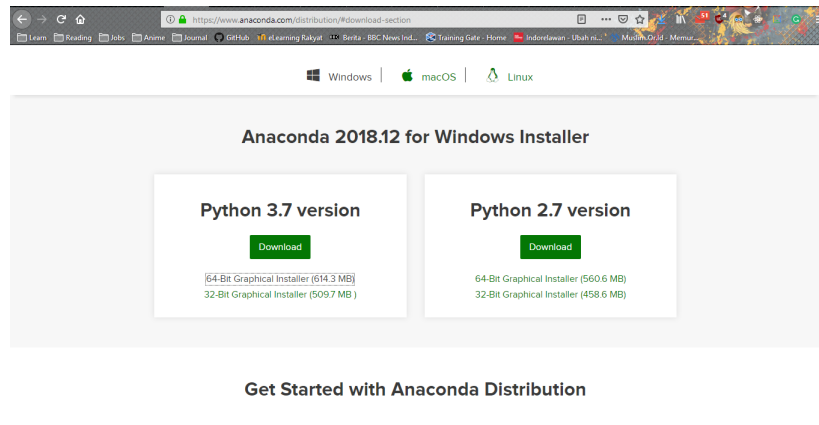


Figure 1.35: Download Aplikasi Anaconda

2. Setelah di unduh, selanjutnya buka aplikasi tersebut. Lalu klik next untuk melanjutkan.

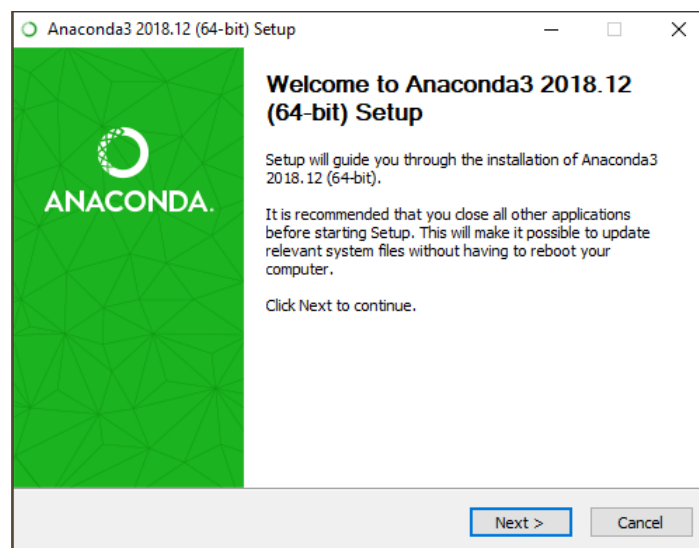


Figure 1.36: Proses Instalasi Aplikasi

3. Lalu klik I Agree untuk melanjutkan.
4. Selanjutnya pilih Just me agar aplikasi tersebut hanya dapat digunakan oleh user yang login pada laptop tersebut.
5. Lalu tentukan direktori penyimpanan file tersebut
6. Selanjutnya akan muncul pop up box tentang advance installation options, ceklis keduanya.

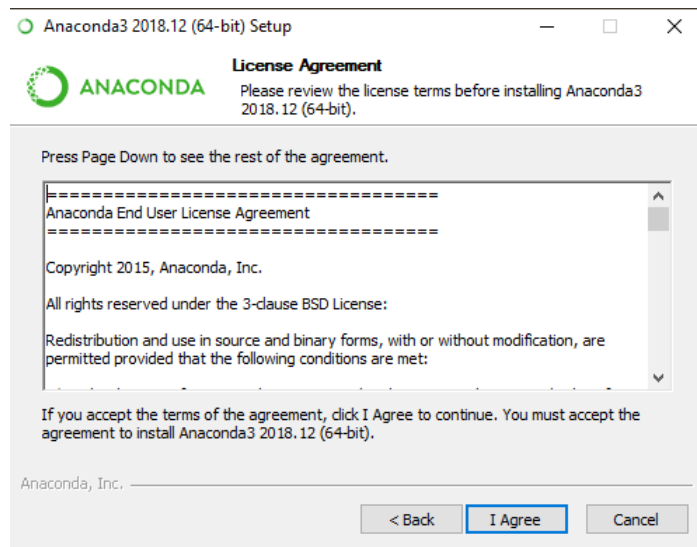


Figure 1.37: Proses Instalasi Aplikasi

7. Tunggu hingga proses install selesai
8. Setelah proses instalasi selesai, klik next
9. Pada bagian selanjutnya akan muncul box dengan memberikan pilihan untuk install VS Code, jika tidak klik skip.
10. Setelah selesai, klik finish
11. Setelah proses instalasi selesai, selanjutnya buka cmd dan ketikan seperti berikut.
12. Selanjutnya ketikan perintah berikut untuk mengunduh library scikit
13. Jika sudah berhasil selanjutnya, ketikan perintah seperti gambar berikut untuk melakukan cek versi conda dan python
14. Mencoba dan mengcompile source code, hasilnya seperti berikut

1.7.3 Mencoba Loading Dataset

1. Berikut source code yang menjelaskan tentang loading dataset. Pada baris pertama code tersebut berfungsi untuk import library datasets dari sklearn. Baris kedua berfungsi untuk menampilkan data secara berurutan. Baris ketiga untuk menampilkan data tersebut berupa angka dan baris keempat untuk menampilkan data tersebut.

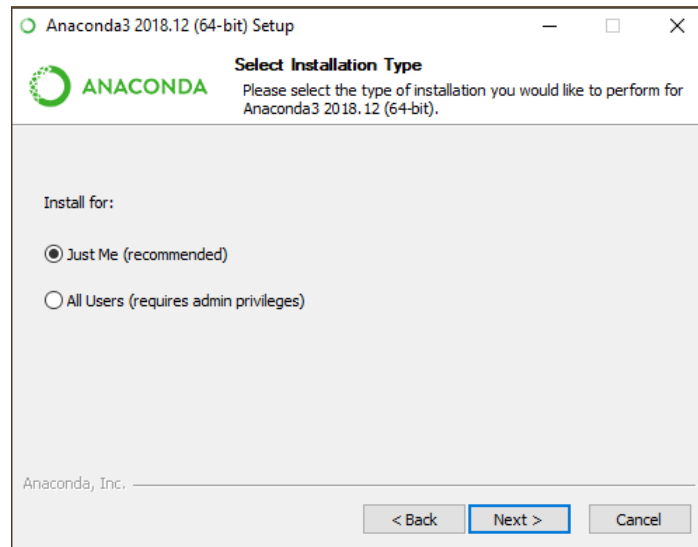


Figure 1.38: Proses Instalasi Aplikasi

1.7.4 Learning and Predicting

```
>>> from sklearn import datasets
>>> iris = datasets.load_iris()
>>> digits = datasets.load_digits()
>>> print(digits.data)
```

- import datasets dari package sklearn
- loading dataset iris
- loading dataset digits
- menampilkan data dari loading dataset digits

```
>>> from sklearn import svm
>>> clf = svm.SVC(gamma=0.001, C=100.)
>>> clf.fit(digits.data[:-1], digits.target[:-1])
```

- Baris tersebut menjelaskan bahwa dalam project ini kita menggunakan source dari sklearn dengan mengambil/import dari svm
- classifier svc dengan atribur gamma dan c
- classifier tersebut akan dijalankan dengan menggunakan metode fit

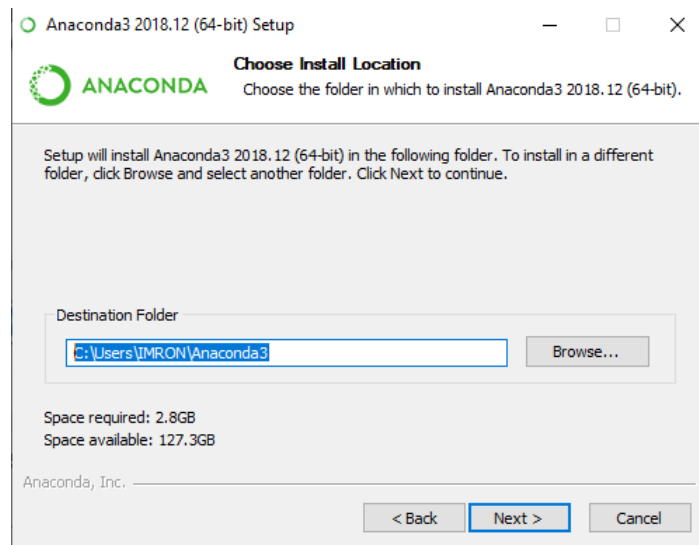


Figure 1.39: Proses Instalasi Aplikasi

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

- hasilnya seperti di atas

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

- classifier predict loading data digits

```
array([8])
```

- hasilnya seperti di atas

1.7.5 Model Persistence

```
>>> from sklearn import svm
>>> from sklearn import datasets
>>> clf = svm.SVC(gamma='scale')
>>> iris = datasets.load_iris()
>>> X, y = iris.data, iris.target
>>> clf.fit(X, y)
```

- import svm dari package sklearn

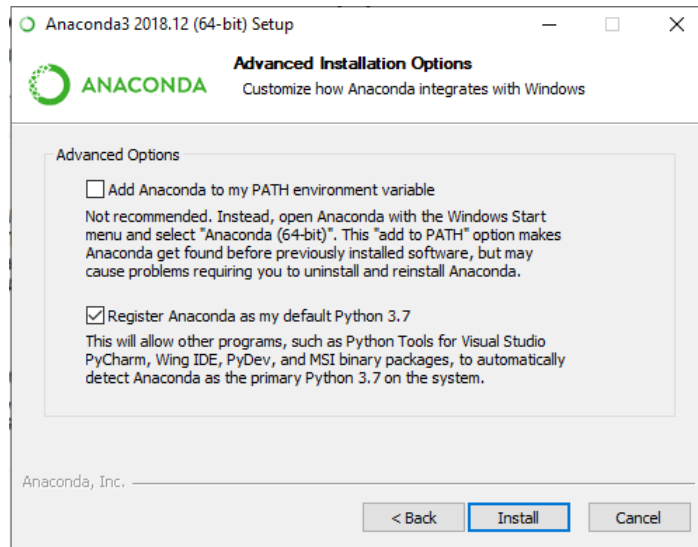


Figure 1.40: Proses Instalasi Aplikasi

- importt datasets dari package sklearn
- classifier svc dengan atribut gamma
- loading dataset iris
- parameter x dan y dengan key iris data dan iris target
- classifier akan dijalankan menggunakan metode fit

```
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)
```

- hasilnya seperti di atas

```
>>> import pickle
>>> s = pickle.dumps(clf)
>>> clf2 = pickle.loads(s)
>>> clf2.predict(X[0:1])
```

- import package pickle
- pickle akan melakukan dumps pada classifier

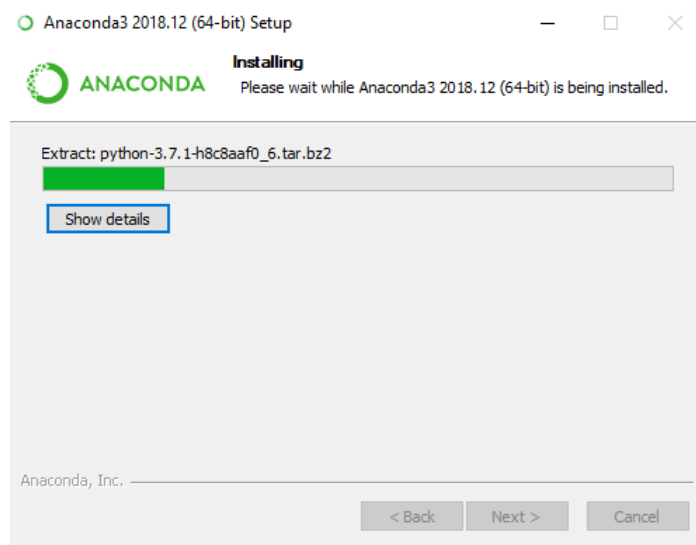


Figure 1.41: Proses Instalasi Aplikasi

- classifier2 akan mengambil data pada classifier pertama
- classifier2 akan memprediksi hasilnya dengan menggunakan syntax python

```
array([0])
```

- hasilnya seperti diatas

```
>>> y[0]
```

- parameter y dengan atribut 0

```
0
```

- hasil seperti di atas

1.7.6 Conventions

```
>>> import numpy as np
>>> from sklearn import random_projection
>>> rng = np.random.RandomState(0)
>>> X = rng.rand(10, 2000)
>>> X = np.array(X, dtype='float32')
>>> X.dtype
```

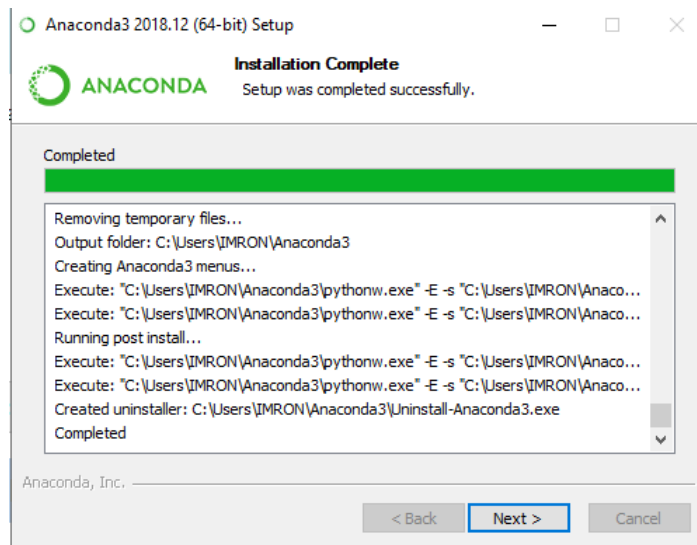


Figure 1.42: Proses Instalasi Aplikasi

- `import numpy` dengan alias `np`
- `import random_projection` pada package `sklearn`
- `rng` parameter dan akan melakukan proses random dalam menentukan hasil
- parameter `x` memiliki `rand` dengan nilai 10, 2000
- parameter `x` dengan `numpy array` akan memunculkan kata `float32` pada hasil terakhir

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

- hasilnya seperti diatas

```
>>> transformer = random_projection.GaussianRandomProjection()
>>> X_new = transformer.fit_transform(X)
>>> X_new.dtype
```

- `transformer` parameter yang di gunakan untuk melakukan pencarian data dengan `gaussianrandomprojection`
- `x new` parameter dan akan dijalankan dengan menggunakan metode `fit`
- `x dtype` akan menampilkan hasilnya

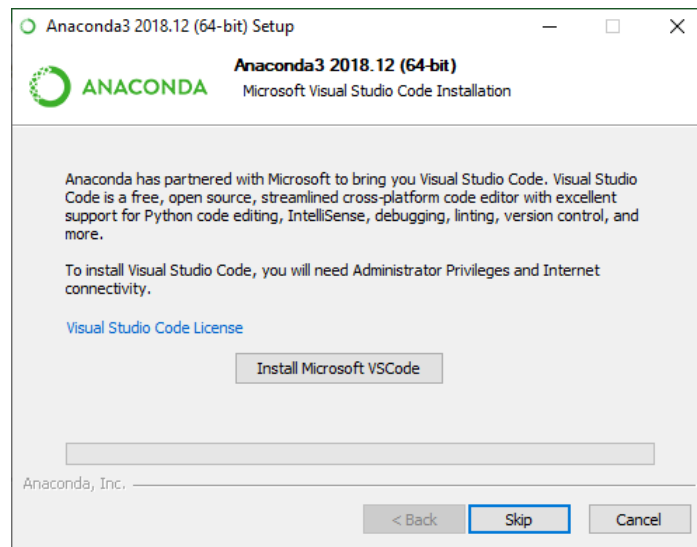


Figure 1.43: Proses Instalasi Aplikasi

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

- hasilnya seperti di atas

```
>>> from sklearn import datasets
>>> from sklearn.svm import SVC
>>> iris = datasets.load_iris()
>>> clf = SVC(gamma='scale')
>>> clf.fit(iris.data, iris.target)
```

- import datasets pada package sklearn
- import svc pada package sklearn
- loading dataset iris
- classifier dengan atribut gamma
- classifier dengan metode fit pada key iris dan 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)
```

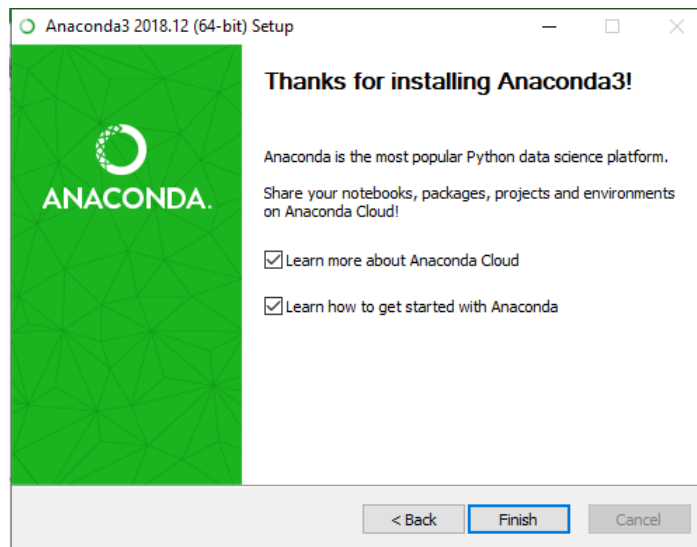


Figure 1.44: Proses Instalasi Aplikasi

```
Administrator: Command Prompt
(c) 2018 Microsoft Corporation. All rights reserved.

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: scipy>=0.13.3 in c:\programdata\anaconda3\lib\site-packages (from
m scikit-learn) (1.1.0)
Requirement not upgraded as not directly required: numpy>=1.8.2 in c:\programdata\anaconda3\lib\site-packages (from
scikit-learn) (1.14.3)
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 1.45: Instalasi Library

- hasilnya seperti di atas

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

- untuk list classifier predict pada loading dataset iris

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

- hasilnya seperti diatas

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

- classifier dengan menggunakan metode fit dan key data dan target

```

Administrator: Command Prompt
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 downloaded:

package | build | size
-----|-----|-----
conda-4.6.7 | py36_0 | 1.7 MB

The following packages will be UPDATED:

conda: 4.5.4-py36_0 --> 4.6.7-py36_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
conda-4.6.7 | 1.7 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done

```

Figure 1.46: Instalasi Library

```

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

C:\WINDOWS\system32>conda --version
conda 4.6.7

```

Figure 1.47: Instalasi Library

```

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)

```

- hasilnya seperti diatas

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

- list untuk classifier pada predict loading data iris

```
['setosa', 'setosa', 'setosa']
```

- hasilnya seperti diatas

```

>>> import numpy as np
>>> from sklearn.svm import SVC
>>> rng = np.random.RandomState(0)
>>> X = rng.rand(100, 10)

```

```

Administrator: Command Prompt - python
C:\WINDOWS\system32>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.]]
>>> digits.target
array([0, 1, 2, ..., 8, 9, 8])
>>> digits.images[0]
array([[ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.],
       [ 0.,  0., 13., 15., 10., 15.,  5.,  0.],
       [ 0.,  3., 15.,  2.,  0., 11.,  8.,  0.],
       [ 0.,  4., 12.,  0.,  0.,  8.,  8.,  0.],
       [ 0.,  5.,  8.,  0.,  0.,  9.,  8.,  0.],
       [ 0.,  4., 11.,  0.,  1., 12.,  7.,  0.],
       [ 0.,  2., 14.,  5., 10., 12.,  0.,  0.],
       [ 0.,  0.,  6., 13., 10.,  0.,  0.,  0.]])
>>> from sklearn.preprocessing import MultiLabelBinarizer
>>> y = [[0,1],[0,2],[1,3],[0,2,3],[2,4]]
>>> y = MultiLabelBinarizer().fit_transform(y)
>>> clf = MultiLabelBinarizer().fit(X,y).predict(X)

```

Figure 1.48: Instalasi Library

```

C:\WINDOWS\system32>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.]]
>>> digits.target
array([0, 1, 2, ..., 8, 9, 8])
>>>

```

Figure 1.49: Loading dataset

```

>>> y = rng.binomial(1, 0.5, 100)
>>> X_test = rng.rand(5, 10)
>>> clf = SVC()
>>> clf.set_params(kernel='linear').fit(X, y)

```

- import numpy alias np
- import svc dari package sklearn
- rng parameter untuk mencari data pada atribut randomstate
- X memiliki jangkauan rand 100, 10
- y memiliki binominal 5,10
- x memiliki rand 5,10
- classifier dengan atribut svc

- classifier parameter dengan atribut linear menggunakan metode fit

```
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)
```

- hasilnya seperti di atas

```
>>> clf.predict(X_test)
```

- classifier untuk memprediksi nilai x

```
array([1, 0, 1, 1, 0])
```

- hasilnya seperti di atas

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

- classifier parameters dengan atribut gamma menggunakan metode fit

```
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)
```

- hasilnya seperti di atas

```
>>> clf.predict(X_test)
```

- classifier prediksi dari x

```
array([1, 0, 1, 1, 0])
```

- hasilnya seperti di atas

```

>>> from sklearn.svm import SVC
>>> from sklearn.multiclass import OneVsRestClassifier
>>> from sklearn.preprocessing import LabelBinarizer
>>> X = [[1,2],[2,4],[4,5],[3,2],[3,1]]
>>> y = [0,0,1,1,2]
>>> classif = OneVsRestClassifier(estimator=SVC(gamma='scale', random_state=0))
>>> classif.fit(X, y).predict(X)
array([0, 0, 1, 1, 2])
>>> y = LabelBinarizer().fit_transform(y)
>>> classif.fit(X, y).predict(X)

```

- import svc dari package sklearn
- import OneVsRestClassifier dari package sklearn
- import LabelBinarizer dari package sklearn
- atribut x
- atribut y
- classifier dengan atribut OneVsRestClassifier dan estimator svc

```

array([[1, 0, 0],
       [1, 0, 0],
       [0, 1, 0],
       [0, 0, 0],
       [0, 0, 0]])

```

- hasilnya seperti di atas

```

>>> from sklearn.preprocessing import MultiLabelBinarizer
>>> y = [[0,1],[0,2],[1,3],[0,2,3],[2,4]]
>>> y = MultiLabelBinarizer().fit_transform(y)
>>> classif.fit(X, y).predict(X)

```

- import MultiLabelBinarizer dari package sklearn
- atribut y
- atribut y akan dijamkan dengan metode fit pada tampilan MultiLabelBinarizer

- classifier dengan metode fit pada x dan y untuk memprediksi

```
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]])
```

- hasilnya seperti di atas

1.7.7 Penanganan Error

Dari percobaan yang telah dilakukan terdapat error yang di dapatkan pada bagian joblib model persistence

```
>>> from joblib import dump, load
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'joblib'
>>> dump(clf, 'filename.joblib')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'dump' is not defined
>>>
```

Figure 1.50: Error

- Error tersebut dikarenakan saya belum install package joblib sehingga error pun terjadi dengan source code sebagai berikut: from joblib import dump, load

```
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 5.4MB/s
Distributed 1.21.0 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 1.51: Instalasi

- Untuk itu solusinya saya install terlebih dahulu joblib agar error pun tidak terjadi kembali.

```
>>> from joblib import dump, load
>>> dump(clf, 'filename.joblib')
['filename.joblib']
>>>
```

Figure 1.52: Hasil

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] Abdillah Baraja. Kecerdasan buatan tinjauan historikal. *Speed-Sentra Penelitian Engineering dan Edukasi*, 1(1), 2008.
- [2] Youssef Bassil. Expert pc troubleshooter with fuzzy-logic and self-learning support. *arXiv preprint arXiv:1204.0181*, 2012.
- [3] 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.
- [4] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.
- [5] Kevin Warwick. *Artificial intelligence: the basics*. Routledge, 2013.