

# Tugas

# Kecerdasan Buatan



Nur Hanifah Amatullah

1184086

Applied Bachelor of Informatics Engineering

Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering

*Politeknik Pos Indonesia*

Bandung 2021

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

## **Acknowledgements**

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

## **Abstract**

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

# Contents

<b>1</b>	<b>Mengenal Kecerdasan Buatan dan Scikit-Learn</b>	<b>1</b>
1.1	Teori . . . . .	1
1.2	Instalasi . . . . .	2
1.3	Penanganan Error . . . . .	6
	<b>Bibliography</b>	<b>7</b>

# List of Figures

# Chapter 1

## Mengenai Kecerdasan Buatan dan Scikit-Learn

### 1.1 Teori

Praktek teori penunjang yang dikerjakan :

1. Sejarah Perkembangan dan Definisi *Artifical Intelligence*. Artifical Intelligence atau biasa juga disebut sebagai kecerdasan buatan ini merupakan suatu ilmu pada bidang komputer yang mana dapat membuat suatu sistem yang cerdas untuk menyelesaikan suatu permasalahan.

Pada tahun 1956 istilah dari *Artifical Intelligence* pertama kali diciptakan. Pada awal-awal munculnya kecerdasan buatan ini mengalami banyak kesuksesan, yang mana diawali dengan kesuksesan Newell dan Simon dengan sebuah program yang disebut dengan *General Problem Solver*, program ini dirancang untuk menyelesaikan masalah secara manusiawi. Kemudian pada tahun 1960-an Departement Pertahanan Amerika Serikat juga ingin mengembangkan dan melatih komputer agar memiliki penalaran seperti manusia secara mendasar. Setelah itu tahun 1970-an, proyek DARPA(*Defence Advanced Research Project Agency*) berhasil menyelesaikan studi kasus tentang pemetaan jalan. Selanjutnya awal abad ke 21 atau lebih tepatnya pada tahun 2003, DARPA juga berhasil menciptakan asisten pribadi cerdas. Sejak itulah kecerdasan buatan mengalami perkembangan hingga saat ini menjadi program yang sangat kompleks dengan menerapkan algoritma dan *deep learning*. Oleh karena itu kecerdasan buatan mampu memberikan solusi yang kompleks untuk berbagai macam permasalahan dengan inovasi yang bervariasi.

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

- Supervised Learning dan Unsupervised Learning

*Supervised learning* merupakan pembelajaran yang ada supervisornya maksudnya yaitu adanya tag dari data yang ditambahkan dalam machine learning tersebut, guna untuk memprediksi suatu pola yang mana pola tersebut sudah ada contoh datanya, jadi pola tersebut merupakan hasil dari data yang sudah lengkap tadi. Sedangkan *Unsupervised Learning* yaitu tidak menggunakan label dalam memprediksi targetnya melainkan dengan melihat kesamaan dari atribut yang dimiliki. Jika memiliki kesamaan dari atribut atau sifat data yang diekstrak maka akan dimasukkan kedalam satu kelompok (*clustering*). Sehingga dapat menimbulkan banyak kelompok.

- Klasifikasi dan Regresi

Klasifikasi merupakan suatu teknik mengklasifikasikan atau mengelompokkan beberapa item item yang belum ada berlabel kedalam sebuah kelas distrit. Sedangkan Regresi merupakan suatu teknik analisis untuk mendefinisikan suatu relasi diantara dua variabel atau lebih, guna menemukan suatu fungsi model data dengan meminimalkan error atau selisih nilai prediksi dengan nilai sebenarnya.

- Data set, Training set dan Testing set

Data set merupakan suatu objek yang akan mempresentasikan sebuah data dengan relasinya di memory. Kemudian Training set yaitu merupakan bagian dari data set yang bertujuan membuat suatu algoritma sesuai tujuan yang diinginkan sedangkan Testing set yaitu bagian dari data set yang bertujuan untuk melihat keakuratan dari data yang di tes.

## 1.2 Instalasi

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



```
Anaconda Prompt (Anaconda3)

(base) C:\Users\User>pip install -U scikit-learn
Requirement already up-to-date: scikit-learn in c:\users\user\anaconda3\lib\site-packages (0.24.1)
Requirement already satisfied, skipping upgrade: threadpoolctl>=2.0.0 in c:\users\user\anaconda3\lib\site-packages (from scikit-learn) (2.1.0)
Requirement already satisfied, skipping upgrade: scipy>=0.19.1 in c:\users\user\anaconda3\lib\site-packages (from scikit-learn) (1.2.2)
Requirement already satisfied, skipping upgrade: joblib>=0.11 in c:\users\user\anaconda3\lib\site-packages (from scikit-learn) (0.13.2)
Requirement already satisfied, skipping upgrade: numpy>=1.13.3 in c:\users\user\anaconda3\lib\site-packages (from scikit-learn) (1.16.4)
(base) C:\Users\User>
```

2. Setelah itu masuk ke link website yang telah diberikan yaitu "<https://scikit-learn.org/stable/tutorial/basic/tutorial.html>".

```
1 from sklearn.linear_model import LogisticRegression
2 from sklearn import set_config
3
4
5 lr = LogisticRegression(penalty='l1')
6 print('Default representation:')
7 print(lr)
8 # LogisticRegression(C=1.0, class_weight=None, dual=False,
9 #                    fit_intercept=True,
10 #                    intercept_scaling=1, l1_ratio=None, max_iter
11 #                    =100,
12 #                    multi_class='auto', n_jobs=None, penalty='l1',
13 #                    random_state=None, solver='warn', tol=0.0001,
14 #                    verbose=0,
15 #                    warm_start=False)
16 set_config(print_changed_only=True)
17 print('\nWith changed_only option:')
18 print(lr)
19 # LogisticRegression(penalty='l1')
```

3. Mencoba Loading example dataset

```
1 from sklearn import datasets #mengimport class dataset dari scikit
  learn library
2 iris = datasets.load_iris() #memuat dan memasukkan dataset iris ke
  variabel bernama iris
3 digits = datasets.load_digits() #membuat dan memasukkan dataset
  digits ke variabel digits
4
5 print(digits.data) #memberikan akses ke fitur yang dapat digunakan
  untuk mengklasifikasikan sampel digit dan menampilkan diconsole
6
7 digits.target #memberikan informasi tentang data yang berhubungan
  atau juga dapat dijadikan sebagai label
8
9 digits.images[0] #Data selalu berupa array 2D, shape (n.samples, n.
  features), meskipun data aslinya mungkin memiliki bentuk yang
  berbeda.
```

4. Mencoba Learning and predicting

```

1 from sklearn import svm #perintah untuk mengimport class svm dari
  package sklearn
2
3 digits = datasets.load_digits() #memasukkan dan memuat dataset
  digits ke variabel digits
4
5 clf = svm.SVC(gamma=0.001, C=100.) #elf sebagai estimator/parameter
  , svm.SVC sebagai class , gamma sebagai parameter untuk
  menerapkan nilai secara manual
6
7 clf.fit(digits.data[:-1], digits.target[:-1]) #elf sebagai
  estimator/parameter, fit sebagai metode, digits.data sebagai
  item,[:-1] sebagai syntax python dan menampilkan outputnya
8
9
10 print(clf.predict(digits.data[-1:])) #predict sebagai metode
  lainnya , digit.data sebagai item menampilkan outputnya

```

## 5. Mencoba Model Persistence

```

1 from sklearn import svm #mengimport class svm dari scikit learn
  library
2 from sklearn import datasets #mengimport class dataset dari scikit
  learn library
3
4 clf = svm.SVC(gamma=0.001, C=100.) #memanggil class SVC dan menset
  argument constructor SVC serta ditampung di variabel clf
5 X, y = datasets.load_iris(return_X_y=True) #meload datasets iris dan
  ditampung di variabel x untuk data sedangkan y untuk target
6
7 clf.fit(X, y) #memanggil method fit untuk melakukan training data
  dengan argumen data dan target dari database iris
8
9 import pickle #mengimport pickle (agar dapat terbaca)
10 s = pickle.dumps(clf) #memanggil method dumps dengan argumen clf
  dan ditampung pada variabel s
11 clf2 = pickle.loads(s) #memanggil method loads dengan argumen s dan
  ditampung di variabel clf2
12 clf2.predict(X[0:1]) #menampilkan hasil dari method predict dengan
  argumen data variabel x
13
14 from joblib import dump, load #mengimport dump dan load dari
  library joblib
15 dump(clf, '1184086.joblib') #memanggil method dumps dengan argumen
  clf dari nama file joblib
16 clf3 = load('1184086.joblib') #memanggil method load dengan argumen
  nama file joblibnya
17 print(clf3.predict(X[0:1])) #menampilkan hasil dari method predict
  dengan argumen data variabel

```

## 6. Mencoba Conventions

```

1 #Type casting

```

```

2 import numpy as np
3 from sklearn import random_projection
4
5 rng = np.random.RandomState(0)
6 X = rng.rand(10, 2000)
7 X = np.array(X, dtype='float32')
8 print(X.dtype)
9
10
11 transformer = random_projection.GaussianRandomProjection()
12 X_new = transformer.fit_transform(X)
13 print(X_new.dtype)
14
15 from sklearn import datasets
16 from sklearn.svm import SVC
17 iris = datasets.load_iris()
18 clf = SVC(gamma=0.001, C=100.)
19 clf.fit(iris.data, iris.target)
20 print(list(clf.predict(iris.data[:3])))
21 clf.fit(iris.data, iris.target_names[iris.target])
22 print(list(clf.predict(iris.data[:3])))
23
24 #refitting and updating parameters
25 import numpy as np
26 from sklearn.datasets import load_iris
27 from sklearn.svm import SVC
28 X, y = load_iris(return_X_y=True)
29 clf = SVC(gamma=0.001, C=100.)
30 clf.set_params(kernel='linear').fit(X, y)
31 clf.set_params(kernel='rbf').fit(X, y)
32 print(clf.predict(X[:5]))
33
34 #multiclass vs multilabel fitting
35 from sklearn.svm import SVC
36 from sklearn.multiclass import OneVsRestClassifier
37 from sklearn.preprocessing import LabelBinarizer
38
39 X = [[1, 2], [2, 4], [4, 5], [3, 2], [3, 1]]
40 y = [0, 0, 1, 1, 2]
41
42 classif = OneVsRestClassifier(estimator=SVC(random_state=0, gamma
    =0.001, C=100.))
43 print(classif.fit(X, y).predict(X))
44 y = LabelBinarizer().fit_transform(y)
45 print(classif.fit(X, y).predict(X))
46
47 from sklearn.preprocessing import MultiLabelBinarizer
48 y = [[0, 1], [0, 2], [1, 3], [0, 2, 3], [2, 4]]
49 y = MultiLabelBinarizer().fit_transform(y)
50 print(classif.fit(X, y).predict(X))

```

## 1.3 Penanganan Error

### 1. Screenshoot Error

```
File "C:\Users\User\Anaconda3\lib\site-packages\joblib\numpy_pickle.py", line 590, in
load
    with open(filename, 'rb') as f:
FileNotFoundError: [Errno 2] No such file or directory: '2284086.joblib'
```

```
File "E:/KECERDASAN BUATAN/TUGAS/chapter1_Nurhanifah_1184086/conventions.py", line 5,
in <module>
    X, y = datasets.load.iris(return_X_y=True) #meload datasets iris dan ditampung
divariabel x untuk data dan y untuk target
AttributeError: module 'sklearn.datasets' has no attribute 'load'
```

### 2. Tuliskan kode error dan jenis error

FileNotFoundError ( file tidak ditemukan pada saat kita menjalankan program).  
AttribututeError

### 3. Solusi dari error tersebut

pertama yaitu salah dalam penulisan filenya kemudian typo pada penulisan codenya. jadi kita dalam membuat code harus lebih teliti lagi.

# Bibliography