

# 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

<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 . . . . .	2
1.4	andi muh aslam/1164064 . . . . .	2
1.4.1	sejarah dan perkembangan kecerdasan buatan . . . . .	2
1.5	Instalasi . . . . .	4
1.5.1	instalasi Library Scikit dari Anaconda . . . . .	4
1.5.1.1	Mencoba Loading an example dataset . . . . .	4
1.6	Aip Suprpto Munari/1164063 . . . . .	5
1.6.1	Teori . . . . .	5
1.6.2	Instalasi . . . . .	7
1.6.2.1	Instalasi Library Scikit dari Anaconda . . . . .	7
1.6.2.2	Mencoba Loading an example Dataset . . . . .	8
1.6.2.3	Learning and Predicting . . . . .	8
1.6.3	Mencoba Model Persistance, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris . . . . .	9
1.6.4	Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris . . . . .	10
1.7	Penanganan Error . . . . .	14
1.7.0.1	Model Presistence . . . . .	16
1.7.0.2	Conventions . . . . .	18
1.7.1	Penanganan eror . . . . .	24
1.7.1.1	ScreenShoot Eror . . . . .	24
1.7.1.2	Tuliskan Kode Error dan Jenis Erornya . . . . .	24
1.7.1.3	Solusi Pemecahan Masalah Error . . . . .	25

<b>2</b>	<b>Related Works</b>	<b>26</b>
2.1	Aip Suprpto Munari/1164063 . . . . .	26
2.1.1	Teori . . . . .	26
2.1.2	Binary Classification . . . . .	26
2.1.3	Supervised Learning, Unsupervised Learning, Dan Classtering	26
2.1.4	Evaluasi Dan Akurasi . . . . .	28
2.1.5	Confusion Matrix . . . . .	28
2.1.6	Cara Kerja K-Fold Cross Validation . . . . .	29
2.1.7	Decision Tree . . . . .	30
2.1.8	Gain Dan Entropi . . . . .	31
2.2	Aip Suprpto Munari/1164063 . . . . .	32
2.2.1	Scikit-learn . . . . .	32
2.2.2	Penanganan Error . . . . .	36
2.3	Andi Aslam/1164064 . . . . .	36
2.3.1	Binary Clasification beserta gambar . . . . .	36
2.3.2	supervised learning dan unsupervised learning dan clustering dengan ilustrasi gambar . . . . .	37
2.3.3	evaluasi dan akurasi dari buku dan disertai ilustrasi contoh den- gan gambar . . . . .	38
2.3.4	bagaimana cara membuat dan membaca confusion matrix, buat confusion matrix . . . . .	38
2.3.5	bagaimana K-fold cross validation bekerja dengan gambar ilus- trasi . . . . .	40
2.3.6	decision tree dengan gambar ilustrasi . . . . .	40
2.3.7	Information Gain dan entropi dengan gambar ilustrasi . . . .	41
2.4	Andiaslam/1164064 . . . . .	42
2.4.1	Scikit-learn . . . . .	42
2.4.2	Praktek Penanganan Error . . . . .	46
2.5	Same Topics . . . . .	46
2.5.1	Topic 1 . . . . .	47
2.5.2	Topic 2 . . . . .	47
2.6	Same Method . . . . .	47
2.6.1	Method 1 . . . . .	47
2.6.2	Method 2 . . . . .	47

<b>3</b>	<b>Methods</b>	<b>50</b>
3.1	The data . . . . .	50
3.2	Method 1 . . . . .	50
3.3	Method 2 . . . . .	50
3.4	Aip Suprpto Munari/1164063 . . . . .	50
3.4.1	Teori . . . . .	50
3.4.2	Praktek . . . . .	55
3.4.3	Penanganan Error . . . . .	60
<b>4</b>	<b>Experiment and Result</b>	<b>71</b>
4.1	Experiment . . . . .	71
4.2	Result . . . . .	71
<b>5</b>	<b>Conclusion</b>	<b>72</b>
5.1	Conclusion of Problems . . . . .	72
5.2	Conclusion of Method . . . . .	72
5.3	Conclusion of Experiment . . . . .	72
5.4	Conclusion of Result . . . . .	72
<b>6</b>	<b>Discussion</b>	<b>73</b>
<b>7</b>	<b>Discussion</b>	<b>74</b>
<b>8</b>	<b>Discussion</b>	<b>75</b>
<b>9</b>	<b>Discussion</b>	<b>76</b>
<b>10</b>	<b>Discussion</b>	<b>77</b>
<b>11</b>	<b>Discussion</b>	<b>78</b>
<b>12</b>	<b>Discussion</b>	<b>79</b>
<b>13</b>	<b>Discussion</b>	<b>80</b>
<b>14</b>	<b>Discussion</b>	<b>81</b>
<b>A</b>	<b>Form Penilaian Jurnal</b>	<b>82</b>
<b>B</b>	<b>FAQ</b>	<b>85</b>





# List of Figures

1.1	Langkah 1 instalasi anaconda.. . . . .	4
1.2	Langkah 2 instalasi anaconda. . . . .	4
1.3	Langkah 3 instalasi anaconda. . . . .	5
1.4	Langkah 4 instalasi anaconda. . . . .	5
1.5	Langkah 1 dataset. . . . .	6
1.6	Langkah 2 dataset. . . . .	6
1.7	Langkah 3 dataset. . . . .	7
1.8	Langkah 4 dataset. . . . .	7
1.9	Langkah 5 dataset. . . . .	8
1.10	Hasil Pengujian Classifier . . . . .	9
1.11	Hasil Pengujian Classifier . . . . .	10
1.12	Pickle Pada Python . . . . .	10
1.13	Pengujian Classifier Pickle . . . . .	10
1.14	Penggunaan Joblib . . . . .	11
1.15	Deklarasi Numpy . . . . .	11
1.16	Contoh Type Casting . . . . .	11
1.17	Menggunakan FitTransform . . . . .	12
1.18	Regresi Yang Dilempar . . . . .	12
1.19	Refitting dan Memperbaharui Parameter . . . . .	13
1.20	MultiClass Classifier . . . . .	14
1.21	MultiClass Classifier biner 2D . . . . .	14
1.22	MultiLabel Classifier . . . . .	14
1.23	Error Import . . . . .	15
1.24	Instal Library Joblib . . . . .	15
1.25	Berhasil Import Library Joblib . . . . .	15
1.26	Download Anaconda. . . . .	16
1.27	Langkah pertama instalasi anaconda. . . . .	17
1.28	Langkah kedua instalasi anaconda. . . . .	18

1.29	Langkah ketiga instalasi anaconda. . . . .	19
1.30	Langkah terakhir instalasi anaconda. . . . .	20
1.31	Langkah pertama instalasi scikit pada CMD. . . . .	20
1.32	Langkah ketiga instalasi conda scikit pada CMD. . . . .	21
1.33	Langkah kedua pilih y. . . . .	21
1.34	Langkah cek version yang diinstall. . . . .	22
1.35	Hasil Tampilan 1. . . . .	22
1.36	Hasil Tampilan 2. . . . .	23
1.37	Hasil Tampilan 3. . . . .	23
1.38	Hasil Tampilan Error. . . . .	24
1.39	Hasil Tampilan Uji coba perintah joblib. . . . .	25
2.1	Binary Classification. . . . .	27
2.2	Supervised Learning. . . . .	28
2.3	Unsupervised Learning. . . . .	29
2.4	Clusterring. . . . .	30
2.5	Evaluasi Dan Akurasi. . . . .	31
2.6	K-Fold Cross Validation. . . . .	31
2.7	Decision Tree. . . . .	32
2.8	Gain Dan Entropi. . . . .	33
2.9	Gambar pertama . . . . .	33
2.10	Gambar kedua . . . . .	34
2.11	Gambar Ketiga . . . . .	34
2.12	Gambar Keempat . . . . .	35
2.13	Gambar Kelima . . . . .	35
2.14	Gambar Keenam . . . . .	36
2.15	Gambar Ketujuh . . . . .	36
2.16	Gambar Kedelapan . . . . .	37
2.17	Gambar Kesembilan . . . . .	37
2.18	Gambar Kesepuluh . . . . .	38
2.19	Gambar Kesebelas . . . . .	39
2.20	Gambar Keduabelas . . . . .	39
2.21	Gambar Ketigabelas . . . . .	40
2.22	Binary Clasification . . . . .	41
2.23	Supervised Learning . . . . .	42
2.24	Evaluasi dan Akurasi . . . . .	42

2.25	K-fold cross validation . . . . .	43
2.26	Decision Tree . . . . .	44
2.27	Entropi . . . . .	44
2.28	Hasil Code 1 . . . . .	45
2.29	Hasil Code 2 . . . . .	45
2.30	Hasil Code 3 . . . . .	46
2.31	Hasil Code 4 . . . . .	46
2.32	Hasil Code 5 . . . . .	47
2.33	Hasil Code 6 . . . . .	47
2.34	Hasil Code 7 . . . . .	48
2.35	Hasil Code 8 . . . . .	48
2.36	Hasil Code 9 . . . . .	48
2.37	Hasil Code 10 . . . . .	48
2.38	Hasil Code 11 . . . . .	48
2.39	Hasil Code 12 . . . . .	49
2.40	HASIL YANG MASIH ERROR . . . . .	49
3.1	Random Forest . . . . .	51
3.2	(b) . . . . .	51
3.3	(c) . . . . .	52
3.4	(d) . . . . .	52
3.5	(e) . . . . .	53
3.6	(h) . . . . .	53
3.7	Confussion Matrik . . . . .	55
3.8	Voting Random forest . . . . .	56
3.9	Aplikasi Pandas . . . . .	56
3.10	Hasil Pandas . . . . .	57
3.11	Aplikasi Numpy . . . . .	57
3.12	Hasil Numpy . . . . .	58
3.13	Aplikasi Matplotlib . . . . .	58
3.14	Hasil Matplotlib . . . . .	59
3.15	Membaca Data File . . . . .	59
3.16	Melihat Data Sebagian . . . . .	60
3.17	Melihat Jumlah Data . . . . .	60
3.18	Mengubah menjadi kolom . . . . .	61
3.19	Lihat sebagian data awal . . . . .	61

3.20	Melihat jumlah data . . . . .	62
3.21	Mengelompokkan burung . . . . .	62
3.22	Melakukan pivot . . . . .	63
3.23	Melihat data awal imgid . . . . .	63
3.24	Melihat jumlah data imgid . . . . .	64
3.25	Data ciri label dari join . . . . .	64
3.26	Mengubah menjadi kolom . . . . .	64
3.27	Melihat isi data frame . . . . .	64
3.28	Membagi data . . . . .	64
3.29	Kelas Random Forest . . . . .	64
3.30	Membangun Random forest . . . . .	65
3.31	Melihat hasil . . . . .	65
3.32	Lihat hasil score . . . . .	65
3.33	Memetakan ke confusion matrix . . . . .	65
3.34	Melihat hasil . . . . .	65
3.35	Melakukan Plot . . . . .	65
3.36	Plotting nama data . . . . .	66
3.37	Melakukan perintah plot . . . . .	66
3.38	Klasifikasi menggunakan decision tree . . . . .	66
3.39	Klasifikasi menggunakan SVM . . . . .	66
3.40	Pengecekan cross validation random forest . . . . .	67
3.41	Pengecekan cross validation decision tree . . . . .	67
3.42	Pengecekan cross validation SVM . . . . .	67
3.43	Pengamatan Komponen . . . . .	67
3.44	Plot informasi . . . . .	68
3.45	Skrinsut Error . . . . .	68
3.46	Penyelesaian . . . . .	68
3.47	Hasil . . . . .	68
3.48	Random Forest. . . . .	69
3.49	Tabel Confusion Matriks . . . . .	69
3.50	Voting . . . . .	70
A.1	Form nilai bagian 1. . . . .	83
A.2	form nilai bagian 2. . . . .	84

# 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*[2]. 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 andi muh aslam/1164064

### 1.4.1 sejarah dan perkembangan kecerdasan buatan

1. 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.

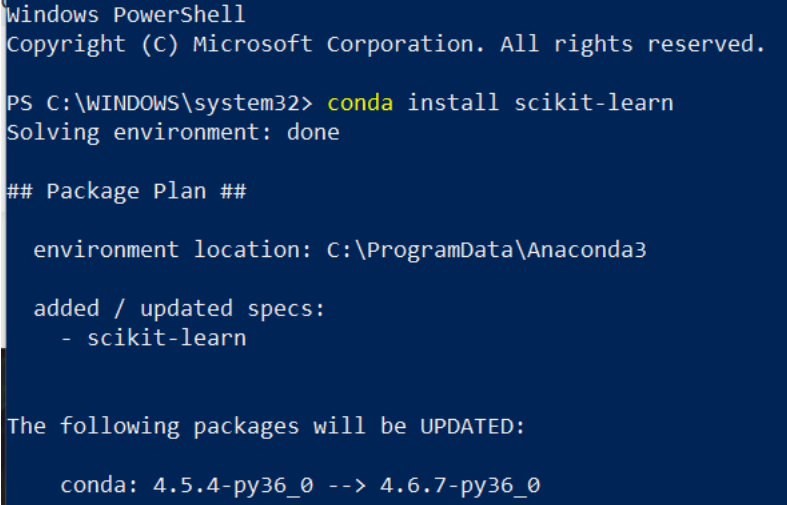
2. Sejarah dan perkembangan kecerdasan buatan terjadi pada musim panas tahun 1956 tercatat adanya seminar mengenai AI di Darmouth College. Seminar pada waktu itu dihadiri oleh sejumlah pakar komputer dan membahas potensi komputer dalam meniru kepandaian manusia. Akan tetapi perkembangan yang sering terjadi 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[3].
3. 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.
4. 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.
5. 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.
6. 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.
7. 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].

## 1.5 Instalasi

### 1.5.1 instalasi Library Scikit dari Anaconda

1. Download aplikasi Anaconda terlebih dahulu



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

## Package Plan ##

  environment location: C:\ProgramData\Anaconda3

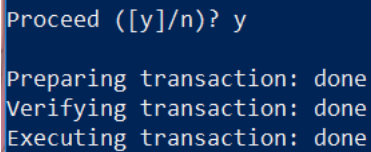
  added / updated specs:
    - scikit-learn

The following packages will be UPDATED:

  conda: 4.5.4-py36_0 --> 4.6.7-py36_0
```

Figure 1.1: Langkah 1 instalasi anaconda..

2. Proceed install anaconda



```
Proceed ([y]/n)? y

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

Figure 1.2: Langkah 2 instalasi anaconda.

3. install scikit-learn

4. perintah print

#### 1.5.1.1 Mencoba Loading an example dataset

1. Masuk Python terlebih dahulu
2. `from sklearn import datasets`(pada baris ini merupakan sebuah perintah untuk mengimport sebuah datasets dari file sklearn).



```

PS C:\WINDOWS\system32> pip install -U scikit-learn
Collecting scikit-learn
  Downloading https://files.pythonhosted.org/packages/ee/c8/c89ebdc0d7dbba6e6fd222daabd257da3c28a967
cp36-cp36m-win32.whl (4.3MB)
    100% |████████████████████████████████████████| 4.3MB 463kB/s
Requirement not upgraded as not directly required: numpy>=1.8.2 in c:\programdata\anaconda3\lib\site
Requirement not upgraded as not directly required: scipy>=0.13.3 in c:\programdata\anaconda3\lib\site
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 1.3: Langkah 3 instalasi anaconda.

```

PS 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.
>>> print('Andi')
Andi

```

Figure 1.4: Langkah 4 instalasi anaconda.

3. iris datasets.load\_iris()(pada baris kedua ini dimana iris merupakan suatu variabel yang berfungsi untuk mengambil data pada datasets dengan perintah .load\_iris).
4. digits datasets.load\_digits()(pada baris ketiga ini dimana digits merupakan suatu variabel yang berfungsi untuk mengambil data pada datasets dengan perintah .load\_digits)
5. print(digits.data)(pada baris keempat ini merupakan perintah yang berfungsi untuk memanggil atau menampilkan variabel digits.data) `##### HEAD`

## 1.6 Aip Suprpto Munari/1164063

### 1.6.1 Teori

- (a) Definisi, sejarah, dan perkembangan kecerdasan buatan.

Definisi kecerdasan buatan adalah ilmu pengetahuan yang dapat membuat komputer untuk meniru kecerdasan manusia yang berhubungan dengan penangkapan, pemodelan, dan penyimpanan kecerdasan manusia dalam

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

Figure 1.5: Langkah 1 dataset.

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

Figure 1.6: Langkah 2 dataset.

sebuah sistem teknologi. Contohnya yaitu melakukan analisa penalaran untuk mengambil suatu kesimpulan atau penerjemahan atau keputusan dari satu bahasa satu ke bahasa lain.

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[?].

- (b) Definisi supervised learning, klasifikasi, regresi, dan unsupervised learning. Data set, training set dan testing set.

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 merupakan salah satu topik utama dalam data mining atau machine learning. Klasifikasi yaitu suatu pengelompokan data dimana data tersebut digunakan untuk 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.

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

Figure 1.7: Langkah 3 dataset.

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

Figure 1.8: Langkah 4 dataset.

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.

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[?].

## 1.6.2 Instalasi

### 1.6.2.1 Instalasi Library Scikit dari Anaconda

- (a) Download aplikasi Anaconda terlebih dahulu. Lihat pada gambar 1.1
- (b) Install aplikasi Anaconda yang sudah di download tadi. Lihat pada gambar 1.2
- (c) Centang Keduanya lalu tekan tombol install. Lihat pada gambar 1.3
- (d) Setelah itu tunggu sampai proses instalasi selesai lalu jika sudah tekan tombol finish. Lihat pada gambar 1.4
- (e) Lalu buka command prompt anda dan tuliskan perintah berikut ini untuk mengecek apakah aplikasinya sudah terinstall. Lihat pada gambar 1.5
- (f) Kemudian ketikkan perintah pip install -U scikit-learn seperti gambar berikut. Lihat pada gambar 1.6
- (g) Lalu jika sudah ketikkan juga perintah conda install scikit-learn. Lihat pada gambar 1.7
- (h) dan setelah itu pilih y. Lihat pada gambar 1.8
- (i) Hasil version yang diinstall. Lihat pada gambar 1.9

```

>>> 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 1.9: Langkah 5 dataset.

### 1.6.2.2 Mencoba Loading an example Dataset

- `from sklearn import datasets`(pada baris ini merupakan sebuah perintah untuk mengimport class datasets dari packaged sklearn).
- `iris = datasets.load_iris()`(pada baris kedua ini dimana iris merupakan suatu estimator/parameter yang berfungsi untuk mengambil data pada item datasets.load\_iris).
- `digits = datasets.load_digits()`(pada baris ketiga ini dimana digits merupakan suatu estimator/parameter yang berfungsi untuk mengambil data pada item datasets.load\_digits).
- `print(digits.data)`(pada baris keempat ini merupakan perintah yang berfungsi untuk menampilkan estimator/parameter yang dipanggil pada item digits.data dan menampilkan outputannya) Lihat gambar 1.10.
- `digits.target`(barisan ini untuk mengambil target pada estimator/parameter digits dan menampilkan outputannya) Lihat gambar 1.11.
- `digits.images[0]`(barisan ini untuk mengambil images[0] pada estimator/parameter digits dan menampilkan outputannya) Lihat gambar 1.12.

=====

78cdb6514db9716f252ef2024a7df6097aace611

### 1.6.2.3 Learning and Predicting

- `from sklearn import svm`(pada baris ini merupakan sebuah perintah untuk mengimport class svm dari packaged sklearn).

```
svm.SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
```

Figure 1.10: Hasil Pengujian Classifier

- `clf = svm.SVC(gamma=0.001, C=100.)`(pada baris kedua ini `clf` sebagai estimator/parameter, `svm.SVC` sebagai class, `gamma` sebagai parameter untuk menetapkan nilai secara manual)
- `clf.fit(digits.data[:-1], digits.target[:-1])`(pada baris ketiga ini `clf` sebagai estimator/parameter, `fit` sebagai metode, `digits.data` sebagai item, `[:-1]` sebagai syntax pythonnya dan menampilkan outputannya) Lihat gambar 1.13.
- `clf.predict(digits.data[-1:])`

iiiiiii HEAD

### 1.6.3 Mencoba Model Persistance, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

1. Pada Python Shell ketikan "from sklearn import svm" artinya akan mengimport sebuah Support Vector Machine(SVM) yang merupakan algoritma classification yang akan diambil dari Scikit-Learn.
2. Kemudian, lanjutkan dengan "from sklearn import datasets" yang 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`. Parameter `gamma` menentukan seberapa jauh pengaruh dari satu contoh training.
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.

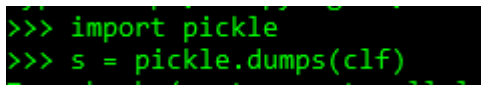
6. 

Figure 1.11: Hasil Pengujian Classifier

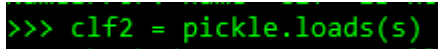


Figure 1.12: Pickle Pada Python

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 "`cf2.predict(X[0:1])`" dengan target asumsinya (0,1) hasilnya berbentuk array.

9. Dalam kasus khusus scikit-learn, mungkin lebih menarik untuk menggunakan `joblib` (dump dan load) untuk menggantikan `Pickle`, yang lebih efisien pada data besar tetapi hanya bisa di `Pickle` ke disk dan tidak ke string. untuk menggunakan `Joblib` pertama ketikan "`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`

#### 1.6.4 Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris

1. `Import numpy as np`, digunakan untuk mengimport Numpy sebagai np.  
`From sklearn import randomprojection` artinya modul yang mengimplemen-

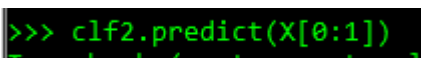
8. 

Figure 1.13: Pengujian Classifier Pickle

```
C:\Users\Asus-PC>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
>>> dump(clf, 'filename.joblib')
```

Figure 1.14: Penggunaan Joblib

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

Figure 1.15: Deklarasi Numpy

tasikan 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 target-

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

2.

Figure 1.16: Contoh Type Casting

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

Figure 1.17: Menggunakan FitTransform

```
C:\Users\Asus-PC>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
>>> 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 1.18: Regresi Yang Dilempar

nya yaitu irisnya sendiri.

`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)` mengubahn 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



```

>>> 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 1.19: Refitting dan Memperbaharui Parameter

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).

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.

```

>>> 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, 11]]
... y = [0, 0, 1, 1, 2]
      File "<stdin>", line 2
        y = [0, 0, 1, 1, 2]
          ^
SyntaxError: invalid syntax
>>> X = [[1, 2], [2, 4], [4, 5], [3, 2], [3, 11]]
>>> 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 1.20: MultiClass Classifier

```

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

```

Figure 1.21: MultiClass Classifier biner 2D

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.

## 1.7 Penanganan Error

HARI KEDUA

```

>>> 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, 0, 1, 0, 0],
       [1, 0, 1, 0, 0],
       [0, 0, 1, 1, 0],
       [1, 0, 1, 0, 0],
       [0, 0, 1, 0, 1]])

```

Figure 1.22: MultiLabel Classifier

```

ameError: name 'clf' is not defined
>> from joblib import dump, load
>>

```

Figure 1.23: Error Import

```

C:\Users\Asus-PC>pip install joblib
Collecting joblib
  Downloading https://files.pythonhosted.org/packages/cd/c1/50a758e8247561e58cb8730
  100% | 286kB 1.8MB/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

```

Figure 1.24: Instal Library Joblib

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.
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

=====

```

C:\Users\Asus-PC>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (
Type "help", "copyright", "credits" or "license" for more information.
>>> from joblib import dump, load
>>> dump(clf, 'filename.joblib')

```

Figure 1.25: Berhasil Import Library Joblib

Anaconda installer archive			
Filename	Size	Last Modified	MD5
<a href="#">Anaconda2-2018.12-Linux-ppc64le.sh</a>	289.7M	2018-12-21 13:14:33	d50ce6eb037f72edfe8f94f90d61aca6
<a href="#">Anaconda2-2018.12-Linux-x86.sh</a>	518.6M	2018-12-21 13:13:15	7d26c7551af6802eb83ecd34282056d7
<a href="#">Anaconda2-2018.12-Linux-x86_64.sh</a>	628.2M	2018-12-21 13:13:10	84f39388da2c747477cf14cb02721b93
<a href="#">Anaconda2-2018.12-MacOSX-x86_64.pkg</a>	640.7M	2018-12-21 13:14:30	c2bfeef310714501a59fd58166e6393d
<a href="#">Anaconda2-2018.12-MacOSX-x86_64.sh</a>	547.1M	2018-12-21 13:14:31	f4d8b10e9a754884fb96e68e0b276a
<a href="#">Anaconda2-2018.12-Windows-x86.exe</a>	458.6M	2018-12-21 13:16:27	f123fda0ec8928bb7d55d1ca72c0d784
<a href="#">Anaconda2-2018.12-Windows-x86_64.exe</a>	560.6M	2018-12-21 13:16:17	10ff4176a94fcff86e6253b0cc82c782
<a href="#">Anaconda3-2018.12-Linux-ppc64le.sh</a>	313.6M	2018-12-21 13:13:03	a775fb6d6c441b899ff2327bd9dadcd6d
<a href="#">Anaconda3-2018.12-Linux-x86.sh</a>	542.7M	2018-12-21 13:13:14	4c9922d1547128b866c6b9cf750c03c7
<a href="#">Anaconda3-2018.12-Linux-x86_64.sh</a>	652.5M	2018-12-21 13:13:06	c9af603d89656bc89680889ef1f92623

Figure 1.26: Download Anaconda.

### 1.7.0.1 Model Persistence

- `from sklearn import svm`(pada baris ini merupakan sebuah perintah untuk mengimport class svm dari packaged sklearn).

- `from sklearn import datasets`(pada baris ini merupakan sebuah perintah untuk mengimport class datasets dari packaged sklearn).

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

(pada baris ketiga ini clf sebagai estimator/parameter, svm.SVC sebagai class, gamma sebagai parameter untuk menetapkan nilai secara manual dengan nilai scale).

- `iris = datasets.load_iris()`

(pada baris keempat ini iris sebagai estimator/parameter, datasets.load\_iris() sebagai item dari suatu nilai).

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

(pada baris kelima ini X, y sebagai estimator/parameter, iris.data, iris.target sebagai item dari 2 nilai yang ada).

- `clf.fit(X, y)`

(pada baris keenam ini clf sebagai estimator/parameter dengan menggunakan metode fit untuk memanggil estimator X, y dengan outputannya)

- `import pickle`

(pickle merupakan sebuah class yang di import).

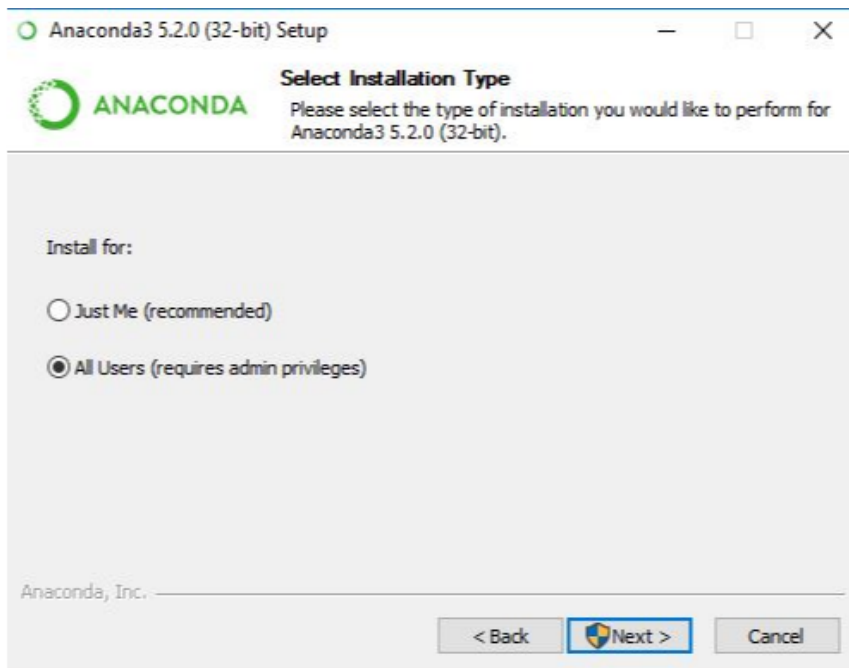


Figure 1.27: Langkah pertama instalasi anaconda.

- `s = pickle.dumps(clf)`  
(pada baris ini `s` sebagai estimator/parameter dengan `pickle.dumps` merupakan suatu nilai/item dari estimator/parameter `clf`)
- `clf2 = pickle.loads(s)`  
(pada baris ini `clf2` sebagai estimator/parameter, `pickle.loads` sebagai suatu item, dan `s` sebagai estimator/parameter yang dipanggil)
- `clf2.predict(X[0:1])`  
(pada baris ini `clf2.predict` sebagai suatu item dengan menggunakan metode `predict` untuk menentukan suatu nilai dari (`X[0:1]`))
- `y[0]`  
(pada estimator/parameter `y` berapapun angka yang diganti nilainya akan selalu konstan yaitu 0)
- `from joblib import dump, load`  
(pada baris berikut ini merupakan sebuah perintah untuk mengimport class `dump`, `load` dari packaged `joblib`).

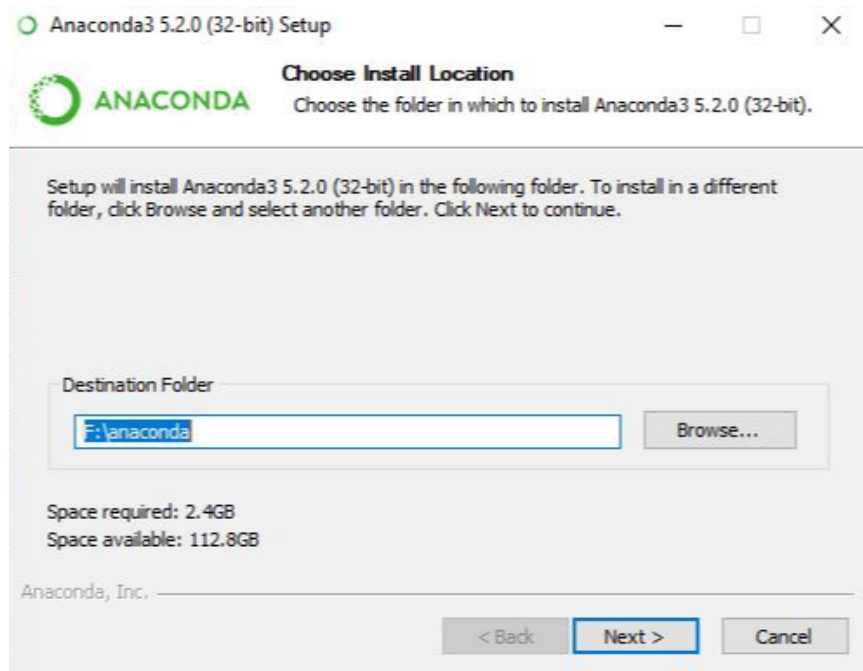


Figure 1.28: Langkah kedua instalasi anaconda.

- `dump(clf, 'filename.joblib')`  
(pada baris berikutnya dump di sini sebagai class yang didalamnya terdapat nilai dari suatu item clf dan data joblib).
- `clf = load('filename.joblib')`  
(pada baris terakhir clf sebagai estimato/parameter dengan suatu nilai load berfungsi untuk mengulang data sebelumnya)
- dari ketiga baris akhir tersebut jika di jalankan aau dituliskan perintah seperti itu maka akan menampilkan tampilan eror

### 1.7.0.2 Conventions

#### 1. Type Casting

- `from sklearn import svm`  
(pada baris ini merupakan sebuah perintah untuk mengimport class svm dari packaged sklearn).
- `from sklearn import random_projection`  
(pada baris ini merupakan sebuah perintah untuk mengimport class random\_projection dari packaged sklearn).

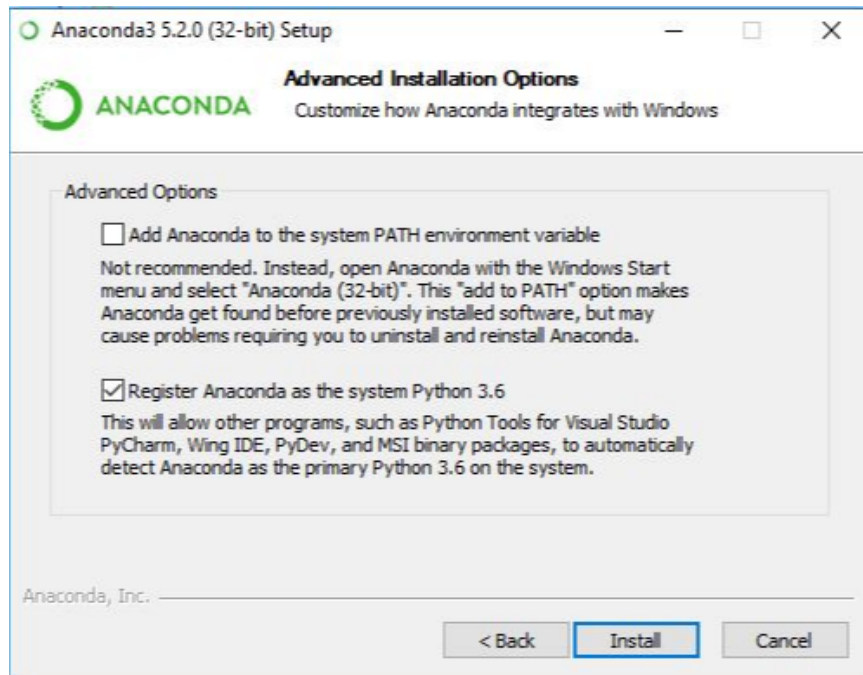


Figure 1.29: Langkah ketiga instalasi anaconda.

- `rng = np.random.RandomState(0)`  
(rng sebagai estimator/parameter dengan nilai suatu itemnya yaitu `np.random.RandomState(0)`)
- `X = rng.rand(10, 2000)`  
(X sebagai estimator/parameter dengan nilai item `rng.rand()`)
- `X = np.array(X, dtype='float32')`  
(X sebagai estimator/parameter dengan nilai item `np.array()`)
- `X.dtype`  
(`X.dtype` sebagai item pemanggil)
- `transformer = random_projection.GaussianRandomProjection()`  
(transformer sebagai estimator/parameter dengan memanggil class `random_projection`)
- `X_new = transformer.fit_transform(X)`  
(`X_new` di sini sebagai estimator/parameter dan menggunakan metode `fit`)
- `X_new.dtype`  
(`X_new.dtype` sebagai item)
- `from sklearn import datasets`

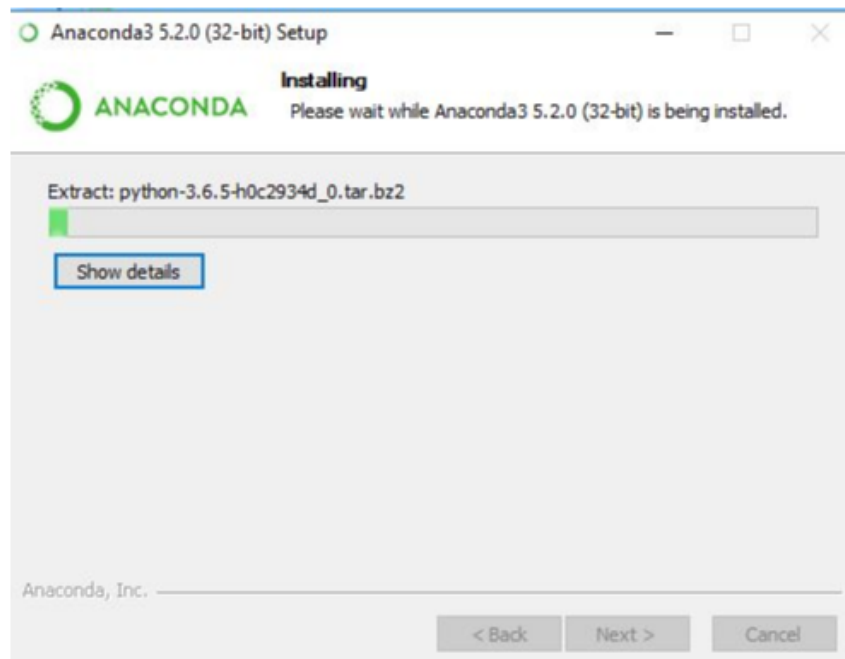


Figure 1.30: Langkah terakhir instalasi anaconda.

```
C:\Users\Asus-PC>pip install -U scikit-learn
Collecting scikit-learn
  Downloading https://files.pythonhosted.org/packages/ee/c8/c89ebdc8d7dbba6e6fd222daab257da3c28a967d07c35284272b2e1cefd/scikit_learn-0.20.2-cp36-cp36m-win32.whl (4.3MB)
    100% |#####| 4.3MB 660kB/s
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)
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 1.31: Langkah pertama instalasi scikit pada CMD.

(pada baris ini merupakan sebuah perintah untuk mengimport class datasets dari packaged sklearn).

- `from sklearn.svm import SVC`

(pada baris ini merupakan sebuah perintah untuk mengimport class SVC dari packaged sklearn.svm).

- `iris = datasets.load_iris()`

(iris sebagai estimator/parameter dengan item datasets.load\_iris()).

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

(clf sebagai estimator/parameter dengan nilai class SVC pada parameter gamma sebagai set penilaian).

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

(estimator/parameter clf menggunakan metode fit dengan itemnya)



```

C:\Users\Asus-PC>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 |
  -----|-----|
  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

```

Figure 1.32: Langkah ketiga instalasi conda scikit pada CMD.

```

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 1.33: Langkah kedua pilih y.

- `list(clf.predict(iris.data[:3]))`  
(menambahkan item list dengan metode predict)
- `clf.fit(iris.data, iris.target_names[iris.target])`  
(estimator/parameter clf menggunakan metode fit dengan itemnya)
- `list(clf.predict(iris.data[:3]))`(menambahkan item list dengan metode pr

## 2. Refitting and Updating Parameters

- `import numpy as np`  
(pada baris ini merupakan sebuah perintah untuk mengimport class svm dari np).
- `from sklearn.svm import SVC`  
(pada baris ini merupakan sebuah perintah untuk mengimport class SVC dari packaged sklearn.svm).
- `rng = np.random.RandomState(0)`  
(rng sebagai estimator/parameter dengan nilai suatu itemnya yaitu np.random.RandomS

```

C:\Users\Asus-PC>conda --version
conda 4.6.7

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

C:\Users\Asus-PC>

```

Figure 1.34: Langkah cek version yang diinstall.

```

C:\Users\Asus-PC>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.metrics import confusion_matrix
>>> y_true = [2, 0, 2, 2, 0, 1]
>>> y_pred = [0, 0, 2, 2, 0, 2]
>>> confusion_matrix(y_true, y_pred)
array([[2, 0, 0],
       [0, 0, 1],
       [1, 0, 2]], dtype=int64)
>>> y_true = ["cow", "goat", "cow", "goat", "horse"]
>>> y_pred = ["goat", "goat", "cow", "goat", "cow"]
>>> confusion_matrix(y_true, y_pred, labels=["cow", "goat", "horse"])
array([[1, 1, 0],
       [0, 2, 0],
       [1, 0, 0]], dtype=int64)
>>> ab, fa, fb, aa = confusion_matrix([0, 1, 0, 1], [1, 1, 1, 0]).ravel()
>>> (ab, fa, fb, aa)
(0, 2, 1, 1)
>>>

```

Figure 1.35: Hasil Tampilan 1.

- `X = rng.rand(100, 10)`  
(X sebagai estimator/parameter dengan nilai item `rng.rand`).
- `y = rng.binomial(1, 0.5, 100)`  
(y sebagai estimator/parameter dengan nilai item `rng.binomial`).
- `X_test = rng.rand(5, 10)`  
(X\_test sebagai estimator/parameter dengan nilai item `rng.rand`).
- `clf = SVC()`  
(clf sebagai estimator/parameter dan class SVC)
- `clf.set_params(kernel='linear').fit(X, y)`  
(set\_params sebagai item)
- `clf.predict(X_test)`  
(menggunakan metode predict)
- `clf.set_params(kernel='rbf', gamma='scale').fit(X, y)`
- `clf.predict(X_test)`

```
>>> y_true = ["cow", "goat", "cow", "goat", "horse"]
>>> y_pred = ["goat", "goat", "cow", "goat", "cow"]
>>> confusion_matrix(y_true, y_pred, labels=["cow", "goat", "horse"])
array([[1, 1, 0],
       [0, 2, 0],
       [1, 0, 0]], dtype=int64)
```

Figure 1.36: Hasil Tampilan 2.

```
>>> ab, fa, fb, aa = confusion_matrix([0, 1, 0, 1], [1, 1, 1, 0]).ravel()
>>> (ab, fa, fb, aa)
(0, 2, 1, 1)
```

Figure 1.37: Hasil Tampilan 3.

### 3. Multiclass vs. Multilabel Fitting

- `from sklearn.svm import SVC`  
(pada baris ini merupakan sebuah perintah untuk mengimport class SVC dari packaged sklearn.svm).
- `from sklearn.multiclass import OneVsRestClassifier`  
(pada baris ini merupakan sebuah perintah untuk mengimport class OneVsRestClassifier dari packaged sklearn.multiclass).
- `from sklearn.preprocessing import LabelBinarizer`  
(pada baris ini merupakan sebuah perintah untuk mengimport class LabelBinarizer dari packaged sklearn.preprocessing).
- `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)`
- `y = LabelBinarizer().fit_transform(y)`
- `classif.fit(X, y).predict(X)`
- `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)`

```

>>> 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
>>> clf = load('filename.joblib')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'load' is not defined

```

Figure 1.38: Hasil Tampilan Error.

## 1.7.1 Penanganan eror

### 1.7.1.1 ScreenShoot Error

#### 1.7.1.2 Tuliskan Kode Error dan Jenis Erornya

- from joblib import dump, load

(Kode baris pertama)

```

Traceback(most recent call last):
  File "<stdin>", line 1, in<module>
ModuleNotFoundError: No module named 'joblib'

```

(Errornya)

- dump(clf, 'filename.joblib')

(Kode baris kedua)

```

Traceback(most recent call last):
  File "<stdin>", line 1, in<module>
NameError: name 'dump' is not defined

```

(Errornya)

- `clf = load('filename.joblib')`

(Kode baris ketiga)

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'load' is not defined
```

(Errornya)

### 1.7.1.3 Solusi Pemecahan Masalah Error

1. Pada masalah error sebelumnya itu dikarenakan kita belum mempunyai packaged joblib. Jadi solusinya yaitu dengan cara menginstall terlebih dahulu packaged joblibnya setelah itu baru perintah tersebut dapat dijalankan

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

Figure 1.39: Hasil Tampilan Uji coba perintah joblib.

lllllll 78cdb6514db9716f252ef2024a7df6097aace611

# Chapter 2

## Related Works

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

### 2.1 Aip Suprpto Munari/1164063

#### 2.1.1 Teori

#### 2.1.2 Binary Classification

1. Binary Classification atau diartikan kedalam bahasa indonesia yaitu Klasifikasi Biner adalah tugas dalam mengkalrifikasikan elemen-elemen dari himpunan yang diberikan kedalam dua kelompok berdasarkan aturan klarifikasi. Pada ummnya klarifikasi biner akan jatuh ke dalam domain Supervised Learning dan dimana kasus khusus hanya memiliki dua kelas. Beberapa contoh yang meliputi Binary Classification adalah

- Deteksi Transaksi Penipuan Kartu Kredit
- Diagnosa medis
- Deteksi Spam

Untuk contoh Binary Classification dapat dilihat pada gambar 2.9

#### 2.1.3 Supervised Learning, Unsupervised Learning, Dan Classtering

1. Supervised Learning merupakan sebuah pendekatan yang dimana sudah adanya sdata yang dilatih dan telah terdapat variabel yang telah ditargetkan sehingga

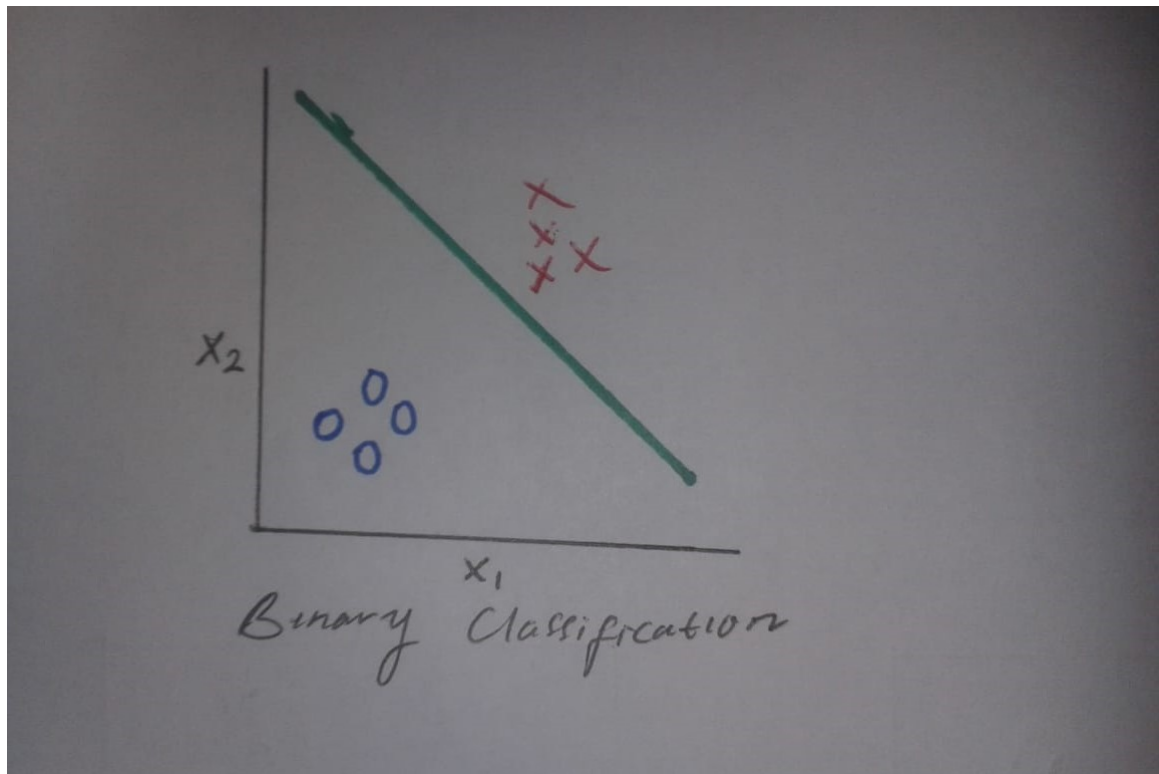


Figure 2.1: Binary Classification.

bertujuan untuk mengelompokkan suatu data ke data yang sudah ada. Contoh dalam Supervised Learning yaitu ketika anda memiliki sejumlah buku yang telah dilabel dengan urutan kategori tertentu. Ketika anda akan membeli sebuah buku baru, maka harus diidentifikasi isi dari buku tersebut dan memasukkannya kedalam kategori tertentu. Ketika anda membeli sebuah buku tersebut maka anda telah menerapkan sebuah logika fuzzy. Ilustrasi Supervised Learning dapat dilihat pada gambar 2.10.

2. Unsupervised Learning merupakan sebuah data yang belum ditentukan variabelnya jadi hanya berupa data saja. Dalam sebuah kasus Unsupervised Learning adalah anggap saja anda belum pernah membeli buku sama sekali dan pada suatu hari anda telah membeli buku dengan sangat banyak dalam kategori yang berbeda. Sehingga buku tersebut belum di kategorikan dan hanya berupa data buku saja. Ilustrasi Unsupervised Learning dapat dilihat pada gambar 2.10.
3. Clustering merupakan sebuah proses untuk mengklasifikasikan sebuah data dalam satu parameter. Dalam kasus ini dapat dijelaskan ada beberapa orang yang memiliki kekuatan tubuh yang sehat dan kekuatan tubuh yang lemah.

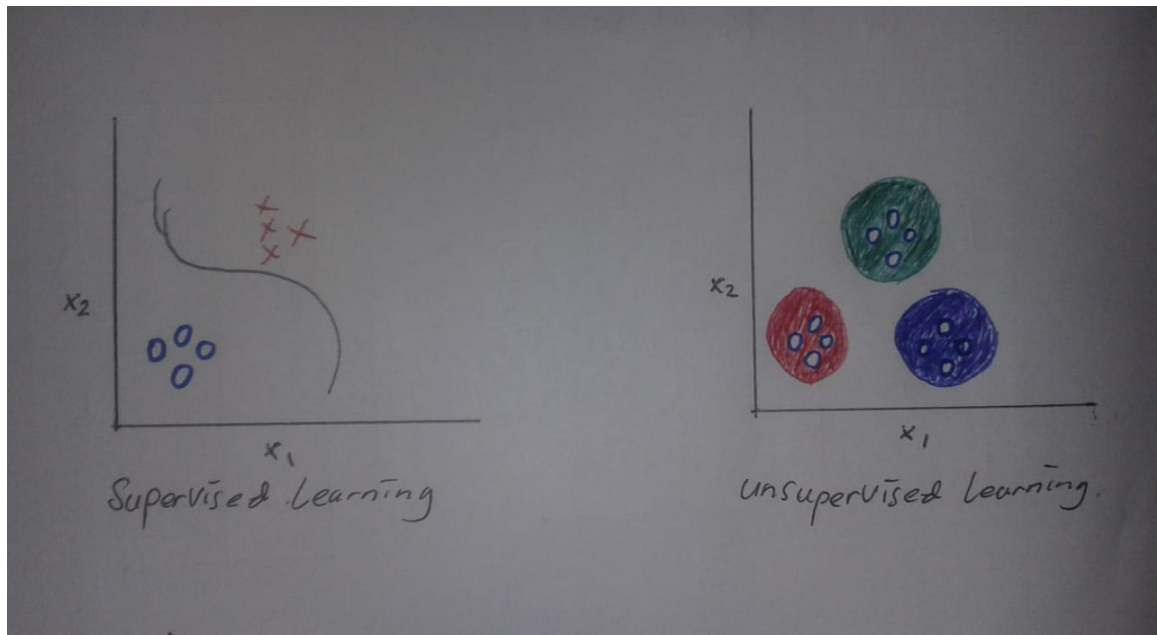


Figure 2.2: Supervised Learning.

Parameter bagi orang yang memiliki tubuh yang kuat adalah orang yang terlihat bugar dan sehat maka dengan orang yang memiliki parameter adalah orang yang memiliki kekuatan tubuh yang kuat dan untuk kekuatan tubuh yang lemah adalah sebaliknya. Ilustrasi gambar dapat di lihat di gambar 2.11

#### 2.1.4 Evaluasi Dan Akurasi

1. Evaluasi dan akurasi adalah bagaimana cara kita dapat mengevaluasi seberapa baik model melakukan pekerjaannya dengan cara mengukur akurasinya. Akurasi akan didefinisikan sebagai presentase kasus yang telah diklasifikasikan dengan benar. Kita dapat melakukan analisis kesalahan yang telah di buat oleh model. Dalam tabel tersebut baris true mangga dan true anggur menunjukkan kasus apakah itu objek mangga atau anggur. Kolom telah di prediksi dan dibuat oleh model. Ada 20 cow yang di prediksi benar dan ada 5 buffalo yang di prediksi salah. Ilustrasi dapat di lihat pada gambar 2.12

#### 2.1.5 Confusion Matrix

1. Ada beberapa cara untuk membuat dan membaca confusion matrix antara lain
  - Tentukan pokok permasalahan serta atributnya
  - Buat Decision Tree



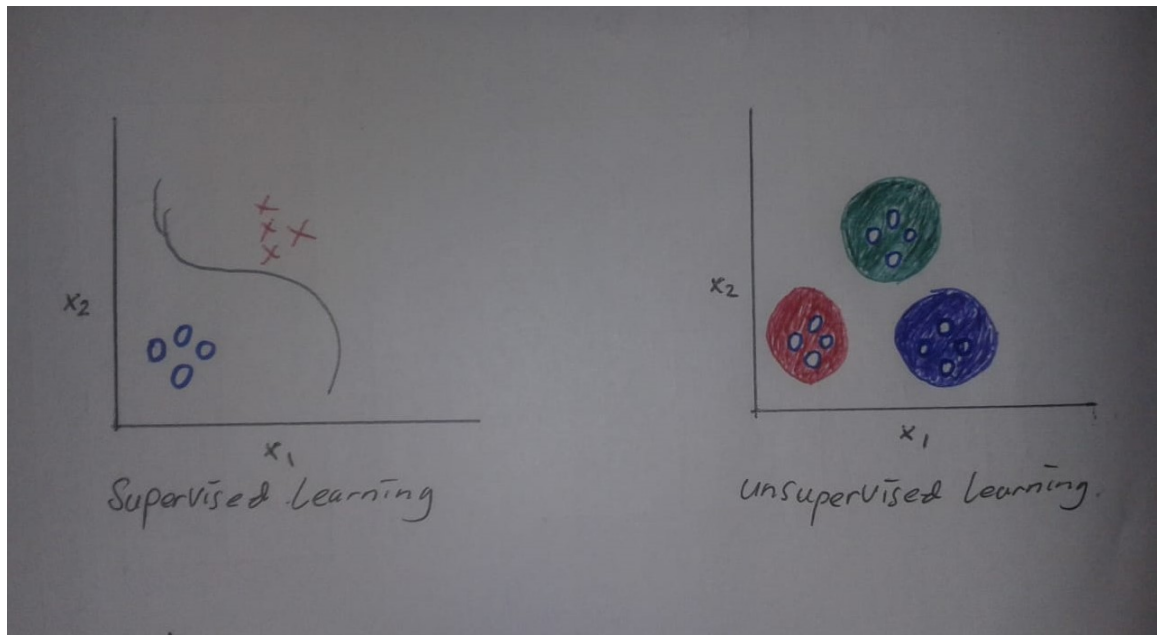


Figure 2.3: Unsupervised Learning.

- Buat Data Testing
- Mencari nilai variabelnya misal a,b,c, dan d
- Mencari nilai recall, precision, accuracy, dan error rate

Di bawah ini adalah contoh dari confusion matrix

$$\text{Recall} = 3/(1+3) = 0,75$$

$$\text{Precision} = 3/(1+3) = 0,75$$

$$\text{Accuracy} = (5+3)/(5+1+1+3) = 0,8$$

$$\text{Error Rate} = (1+1)/(5+1+1+3) = 0,2$$

### 2.1.6 Cara Kerja K-Fold Cross Validation

1. Untuk cara kerja K-Fold Cross Validation adalah sebagai berikut
  - Total instance dibagi menjadi N bagian.
  - Fold yang pertama adalah bagian pertama menjadi testing data dan sisanya menjadi training data.
  - Hitung akurasi berdasarkan porsi data tersebut dengan menggunakan persamaan.

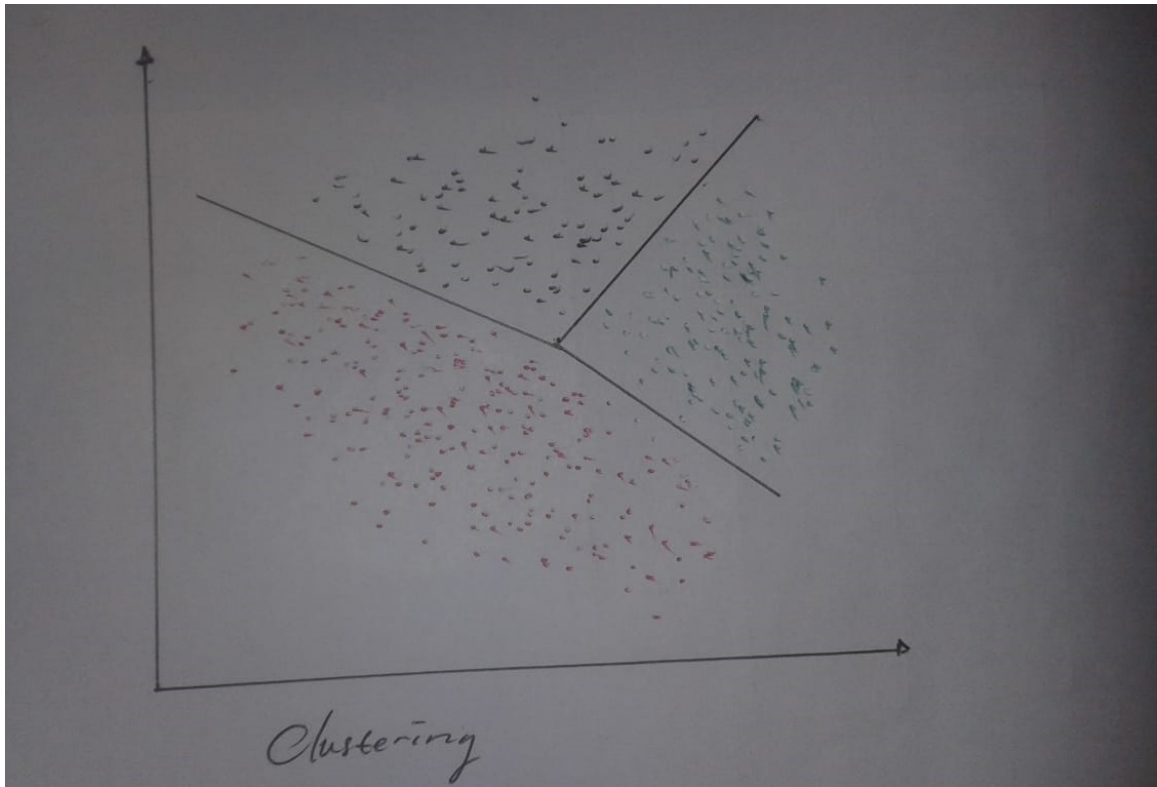


Figure 2.4: Clusterring.

- Fold yang ke dua adalah bagian ke dua menjadi testing data dan sisanya training data.
- Hitung akurasi berdasarkan porsi data tersebut.
- Lakukan step secara berulang hingga habis mencapai fold ke-K.
- Terakhir hitung rata-rata akurasi K buah.

Untuk ilustrasi K-Fold Cross Validation data di lihat pada gambar 2.13

### 2.1.7 Decision Tree

1. Decision Tree adalah sebuah metode pembelajaran yang digunakan untuk melakukan klarifikasi dan regresi. Decision Tree digunakan untuk membuat sebuah model yang dapat memprediksi sebuah nilai variabel target dengan cara mempelajari aturan keputusan dari fitur data. Contoh Decision Tree adalah untuk melakukan prediksi apakah Sapi termasuk hewan herbivora atau bukan, lihat pada gambar 2.21.

	<u>predicted cow</u>	<u>predicted buffalo</u>
<u>True cow</u>	15	5
<u>True buffalo</u>	7	13

Figure 2.5: Evaluasi Dan Akurasi.

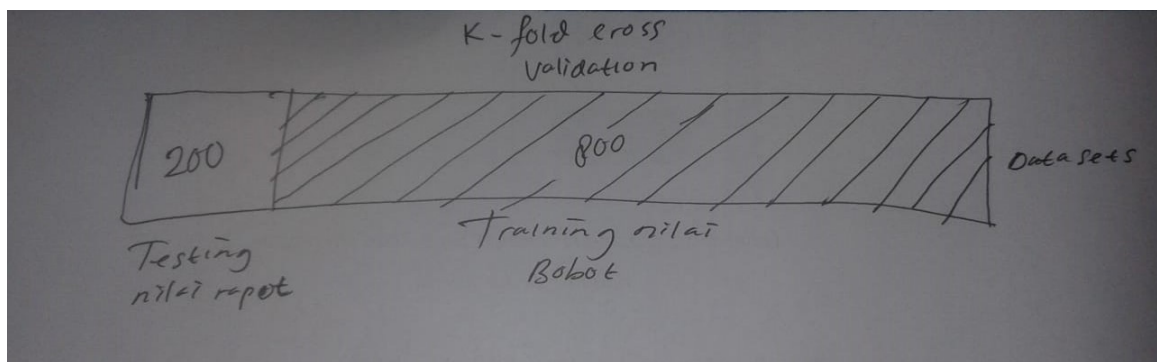


Figure 2.6: K-Fold Cross Validation.

### 2.1.8 Gain Dan Entropi

1. Gain adalah pengurangan yang diharapkan dalam entrophy. Dalam mechine learning, gain dapat digunakan untuk menentukan sebuah urutan atribut atau memperkecil atribut yang telah dipilih. Urutan ini akan membentuk decision tree. atribut gain dipilih yang paling besar.
2. Entropi adalah ukuran ketidakpastian sebuah variabel acak sehingga dapat di artikan entropi adalah ukuran ketidakpastian dari sebuah atribut.

Ilustrasi dari gain dan entropi adalah bagaimana kita memprediksi jenis kelamin berdasarkan atributnya, perhatikan pada gambar 2.15

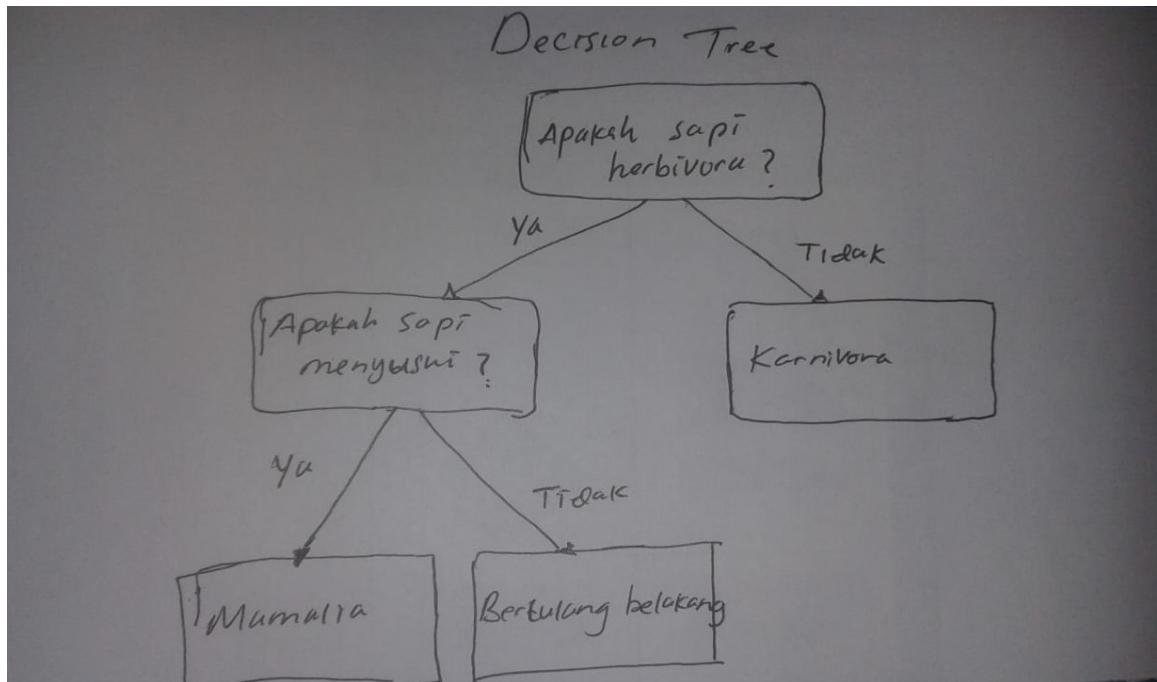


Figure 2.7: Decision Tree.

## 2.2 Aip Suprpto Munari/1164063

### 2.2.1 Scikit-learn

#### Penyelesaian Tugas Harian 4

- Pembahasan Codingan Dan Hasilnya

1. Gambar Pertama :

Penjelasan : Pada baris pertama itu merupakan import library sebagai variabel siomay Dan pada baris kedua variabel siomay membaca file csv nya. Dan pada baris ketiga merupakan hasilnya yaitu 395.

– Hasil Gambar Pertama :

2. Gambar Kedua :

Penjelasan : Variabel Siomay mengimplementasikan baris 1, dari baris G1, G2, G3. Dan variabel siomay akan ngedrop kolom G1, G2,G3. Dan hasilnya akan seperti gambar di out nya.

– Hasil Gambar Kedua :

3. Gambar Ketiga :

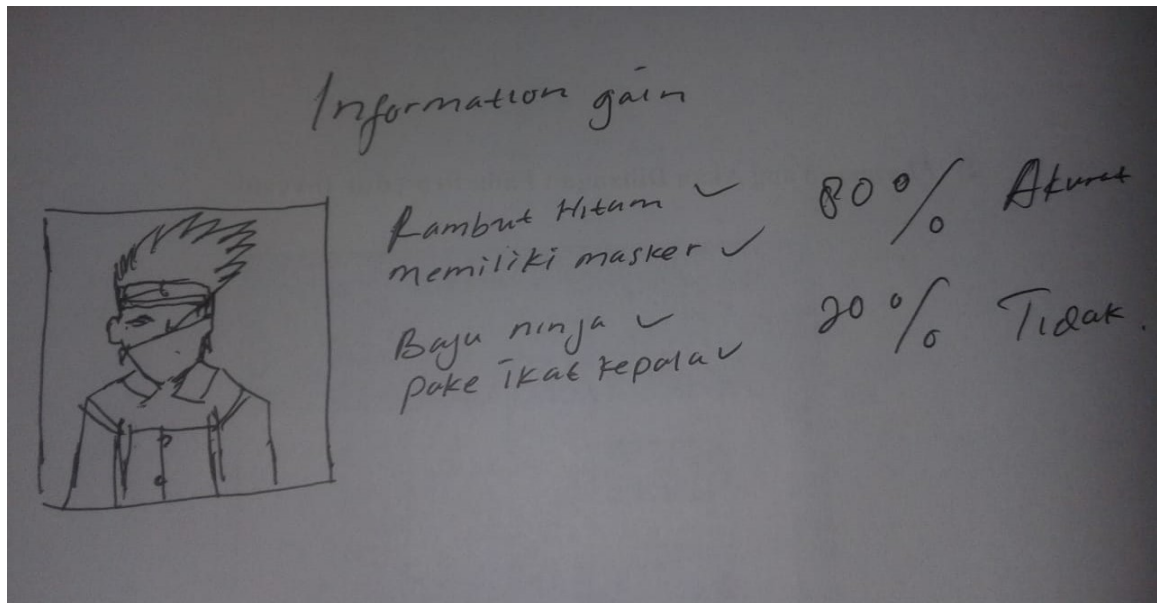


Figure 2.8: Gain Dan Entropi.

```
In [6]: import pandas as Batagor
...: Siomay = Batagor.read_csv('D:\KULIAH\SEMESTER 6\KECERDASAN BUATAN\PRAKTEK
\Python-Artificial-Intelligence-Projects-for-Beginners-master\Python-Artificial-
Intelligence-Projects-for-Beginners-master\Chapter01\dataset\student-mat.csv',
sep=';')
...: len(Siomay)
Out[6]: 395
```

Figure 2.9: Gambar pertama

Penjelasan : Variabel Siomay mengambil atau get data dari dalam kolom. Atau yang tulisan berwarna hijau. Dan kemudian ditampilkan pada output yang dibawah atau menampilkan hasilnya.

– Hasil Gambar Ketiga :

#### 4. Gambar Keempat :

Penjelasan : Penejelasan pada gambar keempat adalah variabel Siomay akan menampilkan sampel data dari 500 training data dan 500 tetsing data. Kemudia data akan dicetak atau di print dari training data dan testing data.

– Hasil Gambar Keempat :

#### 5. Gambar Kelima :

Penjelasan : Pada gambar tersebut variabel hanya melakukan pengetesan/pengecekan terhadap decission tree. Apabila decission tree nya benar maka kodingan tidak eror tapi jika tidak benar maka kodingan akan error.

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

[5 rows x 31 columns]
```

Figure 2.10: Gambar kedua

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

[5 rows x 57 columns]
```

Figure 2.11: Gambar Ketiga

– Hasil Gambar Kelima :

#### 6. Gambar Keenam :

Penjelasan : Pada gambar nomor 6, terjadi kesalahan error yaitu pada import graphviz.

– Hasil Gambar Keenam :

#### 7. Gambar Ketujuh :

Penjelasan : Pada gambar 7 akan menampilkan yang terdapat pada Library Graphviz, apabila benar akan menampilkan hasil output seperti yang terdapat pada gambar atau kalau pengujian gagal akan terdapat error.

– Hasil Gambar Ketujuh :

#### 8. Gambar Kedelapan :

Penjelasan : Pada gambar 8 menampilkan hasil perhitungan dari kedua parameter yang terdapat pada code tersebut.

– Hasil Gambar Kedelapan :

#### 9. Gambar Kesembilan:

Penjelasan : Pada gambar 9, kodingan tersebut mendefinisikan library sklearn model selection dan import cross val score. Dan kemudian variabel

```

In [10]: Siomay = Siomay.sample(frac=1)
...: # split training and testing data
...: Siomay_train = Siomay[:500]
...: Siomay_test = Siomay[500:]
...:
...: Siomay_train_att = Siomay_train.drop(['pass'], axis=1)
...: Siomay_train_pass = Siomay_train['pass']
...:
...: Siomay_test_att = Siomay_test.drop(['pass'], axis=1)
...: Siomay_test_pass = Siomay_test['pass']
...:
...: Siomay_att = Siomay.drop(['pass'], axis=1)
...: Siomay_pass = Siomay['pass']
...:
...: # number of passing students in whole dataset:
...: import numpy as np
...: print("Passing: %d out of %d (%.2f%%)" % (np.sum(Siomay_pass), len(Siomay_pass),
...: 100*float(np.sum(Siomay_pass)) / len(Siomay_pass)))
Passing: 166 out of 395 (42.03%)

```

Figure 2.12: Gambar Keempat

```

In [12]: from sklearn import tree
...: lepet = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
...: lepet = lepet.fit(Siomay_train_att, Siomay_train_pass)

```

Figure 2.13: Gambar Kelima

scores mengeksekusi fungsi cross val score(Batagor, Siomay att, Siomay pass, cv=5). Kemudian akan menampilkan nilai dari fungsi akurasi.

– Hasil Gambar Kesembilan :

#### 10. Gambar Kesepuluh :

Penjelasan : Pada gambar di atas kodingan nya berfungsi untuk menampilkan hasil dari fungsi Max Depth dan Accuraccy dari dari Decission Tree. Yaitu menampilkan data dari angka 1-20.

– Hasil Gambar Kesepuluh :

#### 11. Gambar Kesebelas :

Penjelasan : Pada gambar 11 dijelaskan bahwa variable scores akan menampilkan atau mendefinisikan nilai dari variabel score yang mana isi dari variable score yaitu Batagor, Siomay att, Siomay pass, cv=5. Yang mana hasil tampilan dari kodingannya adalah outputan seperti gambar 11.

– Hasil Gambar Kesebelas :

#### 12. Gambar Keduabelas :

Penjelasan : Pada gambar di atas dijelaskan bahwa pada library matplotlib akan menampilkan gambar grafik pada gambar 12 dari eksekusi fungsi ax.errorbar.

– Hasil Gambar Keduabelas :

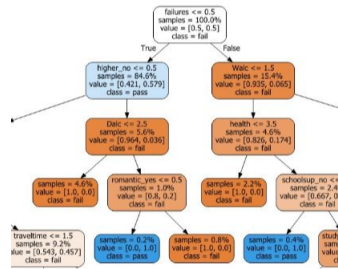


Figure 2.14: Gambar Keenam

```
In [22]:
In [22]: from sklearn import tree
...: tree.export_graphviz(Batagor, out_file="student-performance.dot", label="all",
...: impurity=False, proportion=True,
...: feature_names=list(Siomay_train_att), class_names=["fail",
...: "pass"],
...: filled=True, rounded=True)
```

Figure 2.15: Gambar Ketujuh

## 2.2.2 Penanganan Error

1. Skrintsut Error
2. Kode Error dan Jenis Errornya

Kode Error: "Import Graphiz" dan "ModulNotFoundError".

Jenis Error: Pada Grafik

3. Penanganan

Melakukan install ulang pada graphiz

## 2.3 Andi Aslam/1164064

### 2.3.1 Binary Clasification beserta gambar

1. Output aktual dari banyak algoritma klasifikasi biner adalah skor prediksi. Skor menunjukkan kepastian sistem bahwa pengamatan yang diberikan adalah milik kelas positif. Untuk membuat keputusan tentang apakah pengamatan harus diklasifikasikan sebagai positif atau negatif, sebagai konsumen skor ini, Anda akan menginterpretasikan skor dengan memilih ambang klasifikasi (cut-off) dan membandingkan skor dengan itu. Setiap pengamatan dengan skor lebih tinggi dari ambang kemudian diprediksi sebagai kelas positif dan skor lebih rendah dari ambang diprediksi sebagai kelas negatif.



```
In [34]: Batagor.score(Siomay_test_att, Siomay_test_pass)
Out[34]: 0.8583579357035
```

Figure 2.16: Gambar Kedelapan

```
In [26]: from sklearn.model_selection import cross_val_score
...: scores = cross_val_score(Batagor, Siomay_att, Siomay_pass, cv=5)
...: # show average score and +/- two standard deviations away (covering 95% of
...: scores)
...: print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
Accuracy: 0.56 (+/- 0.06)
```

Figure 2.17: Gambar Kesembilan

### 2.3.2 supervised learning dan unsupervised learning dan clustering dengan ilustrasi gambar

1. Supervised learning adalah tugas pembelajaran mesin untuk mempelajari suatu fungsi yang memetakan input ke output berdasarkan contoh pasangan input-output. Ini menyimpulkan fungsi dari data pelatihan berlabel yang terdiri dari serangkaian contoh pelatihan. Dalam pembelajaran yang diawasi, setiap contoh adalah pasangan yang terdiri dari objek input (biasanya vektor) dan nilai output yang diinginkan (juga disebut sinyal pengawas). Algoritma pembelajaran yang diawasi menganalisis data pelatihan dan menghasilkan fungsi yang disimpulkan, yang dapat digunakan untuk memetakan contoh-contoh baru. Skenario optimal akan memungkinkan algoritma menentukan label kelas dengan benar untuk instance yang tidak terlihat. Ini membutuhkan algoritma pembelajaran untuk menggeneralisasi dari data pelatihan untuk situasi yang tidak terlihat dengan cara yang "masuk akal" (lihat bias induktif).
2. Unsupervised learning adalah istilah yang digunakan untuk pembelajaran bahasa Ibrani, yang terkait dengan pembelajaran tanpa guru, juga dikenal sebagai organisasi mandiri dan metode pemodelan kepadatan probabilitas input. Analisis cluster sebagai cabang pembelajaran mesin yang mengelompokkan data yang belum diberi label, diklasifikasikan atau dikategorikan. Alih-alih menanggapi umpan balik, analisis klaster mengidentifikasi kesamaan dalam data dan bereaksi berdasarkan ada tidaknya kesamaan di setiap potongan data baru. Berikut merupakan contoh Unsupervised Learning dengan Gaussian mixture models.

```

In [27]: for max_depth in range(1, 20):
...:     Batagor = tree.DecisionTreeClassifier(criterion="entropy",
max_depth=max_depth)
...:     scores = cross_val_score(Batagor, Siomay_att, Siomay_pass, cv=5)
...:     print("Max depth: %d, Accuracy: %0.2f (+/- %0.2f)" % (max_depth,
scores.mean(), scores.std() * 2))
Max depth: 1, Accuracy: 0.58 (+/- 0.01)
Max depth: 2, Accuracy: 0.59 (+/- 0.08)
Max depth: 3, Accuracy: 0.53 (+/- 0.05)
Max depth: 4, Accuracy: 0.54 (+/- 0.05)
Max depth: 5, Accuracy: 0.56 (+/- 0.07)
Max depth: 6, Accuracy: 0.58 (+/- 0.09)
Max depth: 7, Accuracy: 0.57 (+/- 0.06)
Max depth: 8, Accuracy: 0.57 (+/- 0.08)
Max depth: 9, Accuracy: 0.56 (+/- 0.06)
Max depth: 10, Accuracy: 0.56 (+/- 0.06)
Max depth: 11, Accuracy: 0.55 (+/- 0.05)
Max depth: 12, Accuracy: 0.56 (+/- 0.06)
Max depth: 13, Accuracy: 0.55 (+/- 0.05)
Max depth: 14, Accuracy: 0.57 (+/- 0.09)
Max depth: 15, Accuracy: 0.56 (+/- 0.06)
Max depth: 16, Accuracy: 0.57 (+/- 0.07)
Max depth: 17, Accuracy: 0.55 (+/- 0.07)
Max depth: 18, Accuracy: 0.57 (+/- 0.06)
Max depth: 19, Accuracy: 0.56 (+/- 0.07)

In [28]: |

```

Figure 2.18: Gambar Kesepuluh

### 2.3.3 evaluasi dan akurasi dari buku dan disertai ilustrasi contoh dengan gambar

1. Evaluasi adalah tentang bagaimana kita dapat mengevaluasi seberapa baik model bekerja dengan mengukur akurasi. Dan akurasi akan didefinisikan sebagai persentase kasus yang diklasifikasikan dengan benar. Kita dapat menganalisis kesalahan yang dibuat oleh model, atau tingkat kebingungannya, menggunakan matriks kebingungan. Matriks kebingungan mengacu pada kebingungan dalam model, tetapi matriks kebingungan ini bisa menjadi sedikit sulit untuk dipahami ketika mereka menjadi sangat besar.

### 2.3.4 bagaimana cara membuat dan membaca confusion matrix, buat confusion matrix

1. Confusion matrix :
  - 1) Tentukan pokok permasalahan dan atributnya, misal gaji dan listik.
  - 2) Buat pohon keputusan
  - 3) Lalu data testingnya

```

In [28]: depth_acc = np.empty((19,3), float)
...: i = 0
...: for max_depth in range(1, 20):
...:     t = tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
...:     scores = cross_val_score(Batagor, Siomay_att, Siomay_pass, cv=5)
...:     depth_acc[i,0] = max_depth
...:     depth_acc[i,1] = scores.mean()
...:     depth_acc[i,2] = scores.std() * 2
...:     i += 1
...:
...:
...: depth_acc
Out[28]:
array([[ 1., 0.54674294, 0.04923122],
       [ 2., 0.55936952, 0.06579881],
       [ 3., 0.57949692, 0.09362112],
       [ 4., 0.57202856, 0.05271394],
       [ 5., 0.56953019, 0.06961587],
       [ 6., 0.56193363, 0.04121254],
       [ 7., 0.56186871, 0.0611709 ],
       [ 8., 0.57209185, 0.08326673],
       [ 9., 0.55687033, 0.05993239],
       [10., 0.58196284, 0.08779916],
       [11., 0.53408309, 0.07935713],
       [12., 0.56702856, 0.0507212 ],
       [13., 0.57693119, 0.08667513],
       [14., 0.56712512, 0.04993824],
       [15., 0.54430785, 0.05251297],
       [16., 0.55953019, 0.03094885],
       [17., 0.55930308, 0.07170369],

```

Figure 2.19: Gambar Kesebelas

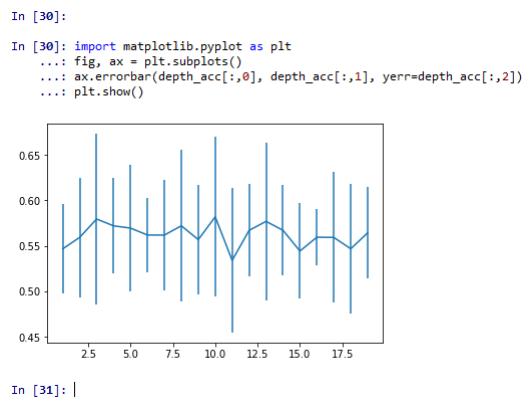


Figure 2.20: Gambar Keduabelas

- 4) Lalu mencari nilai a, b, c, dan d. Semisal a = 5, b = 1, c = 1, dan d = 3.
- 5) Selanjutnya mencari nilai recall, precision, accuracy, serta dan error rate.

2. Berikut adalah contoh dari confusion matrix :

- Recall =  $3/(1+3) = 0,75$
- Precision =  $3/(1+3) = 0,75$
- Accuracy =  $(5+3)/(5+1+1+3) = 0,8$
- Error Rate =  $(1+1)/(5+1+1+3) = 0,2$

```

In [65]: import graphviz
...: dot_data = tree.export_graphviz(solo, out_file=None, label="all", impurity=False, proportion=True,
...:                                feature_names=list(solok_train_att), class_names=["fail", "pass"],
...:                                filled=True, rounded=True)
...: graph = graphviz.Source(dot_data)
...: graph
Traceback (most recent call last):
  File "<ipython-input-65-47940b5e4aa1>", line 1, in <module>
    import graphviz
ModuleNotFoundError: No module named 'graphviz'

```

Figure 2.21: Gambar Ketigabelas

### 2.3.5 bagaimana K-fold cross validation bekerja dengan gambar ilustrasi

1. Cara kerja K-fold cross validation :

- 1) Total instance dibagi menjadi N bagian.
- 2) Fold yang pertama adalah bagian pertama menjadi data uji (testing data) dan sisanya menjadi training data.
- 3) Lalu hitung akurasi berdasarkan porsi data tersebut dengan menggunakan persamaan.
- 4) Fold yang ke dua adalah bagian ke dua menjadi data uji (testing data) dan sisanya training data.
- 5) Kemudian hitung akurasi berdasarkan porsi data tersebut.
- 6) Dan seterusnya hingga habis mencapai fold ke-K.
- 7) Terakhir hitung rata-rata akurasi K buah.

### 2.3.6 decision tree dengan gambar ilustrasi

1. Decision Tree adalah metode pembelajaran yang diawasi non-parametrik yang digunakan untuk klasifikasi dan regresi. Tujuannya adalah untuk membuat model yang memprediksi nilai variabel target dengan mempelajari aturan keputusan sederhana yang disimpulkan dari fitur data.

Misalnya, dalam contoh di bawah ini, decision tree belajar dari data untuk memperkirakan kurva sinus dengan seperangkat aturan keputusan if-then-else. Semakin dalam pohon, semakin rumit aturan keputusan dan semakin bugar modelnya.

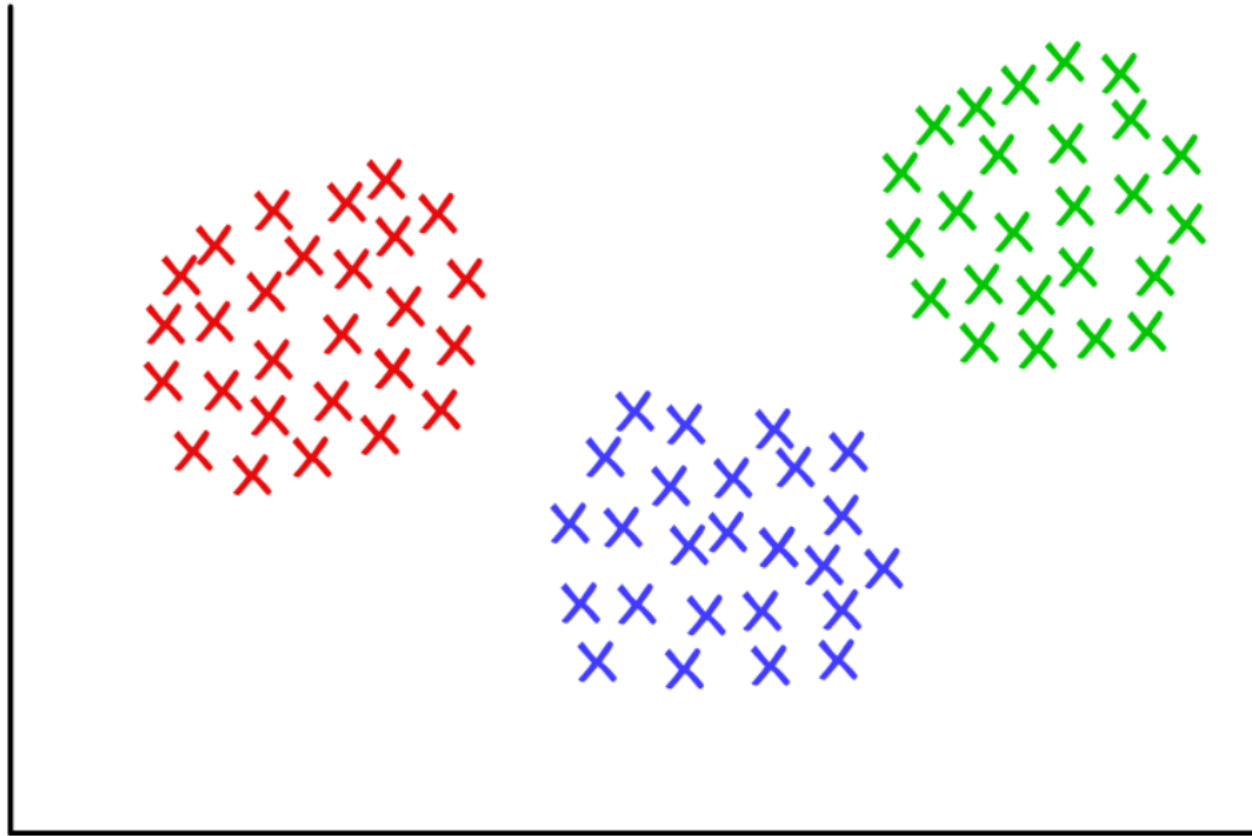


Figure 2.22: Binary Clasification

### 2.3.7 Information Gain dan entropi dengan gambar ilustrasi

1. Information gain didasarkan pada penurunan entropi setelah dataset dibagi pada atribut. Membangun decision tree adalah semua tentang menemukan atribut yang mengembalikan perolehan informasi tertinggi (mis., Cabang yang paling homogen).
2. Entropi adalah ukuran keacakan dalam informasi yang sedang diproses. Semakin tinggi entropi, semakin sulit untuk menarik kesimpulan dari informasi itu. Membalik koin adalah contoh tindakan yang memberikan informasi yang acak. Untuk koin yang tidak memiliki afinitas untuk kepala atau ekor, hasil dari sejumlah lemparan sulit diprediksi. Mengapa? Karena tidak ada hubungan antara membalik dan hasilnya. Inilah inti dari entropi.

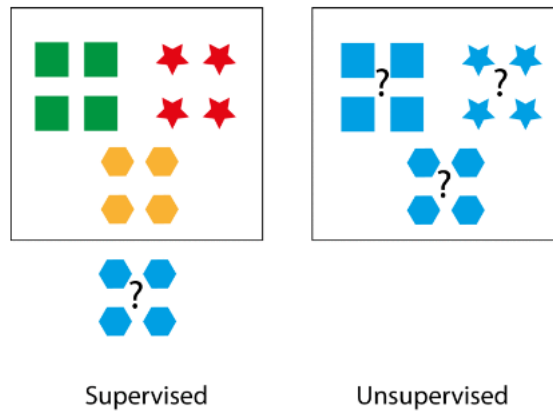


Figure 2.23: Supervised Learning

	<u>Pradicted</u> Game	<u>Pradicted</u> Movie
True Game	99%	56%
True Movie	66%	73%

Figure 2.24: Evaluasi dan Akurasi

## 2.4 Andiaslam/1164064

### 2.4.1 Scikit-learn

1. Pembahasan Dan Hasil
2. variabel Dan pada baris kedua variabel solok membaca file csv nya. Dan pada baris ketiga merupakan hasilnya yaitu 395.
  - Hasil Code 1:
3. variable dapat mengimplementasi bari ke 1 dari baris G1, G2, dan G3.
  - Hasil 2 :
4. Code 3.

## K-Fold Cross Validation

30	70
----	----

Data Set

Testing Nilai

Training Nilai

Figure 2.25: K-fold cross validation

Penjelasan:

variabel mengambil data dari kolom. kemudian akan di tampilkan pada output

- Hasil Code 3:

5. Code 4.

Penjelasan:

Variabel akan menampilkan sampel data dari 500 data training dan 500 data testing.

- Hasil Code 4:

6. Code 5.

Penjelasan:

variabel hanya melakukan pengetesan/pengecekan terhadap decision tree.

- Hasil Code 5:

7. Code 6.

Penjelasan:

terjadi error pada import graphivz.

- Hasil Code 6:

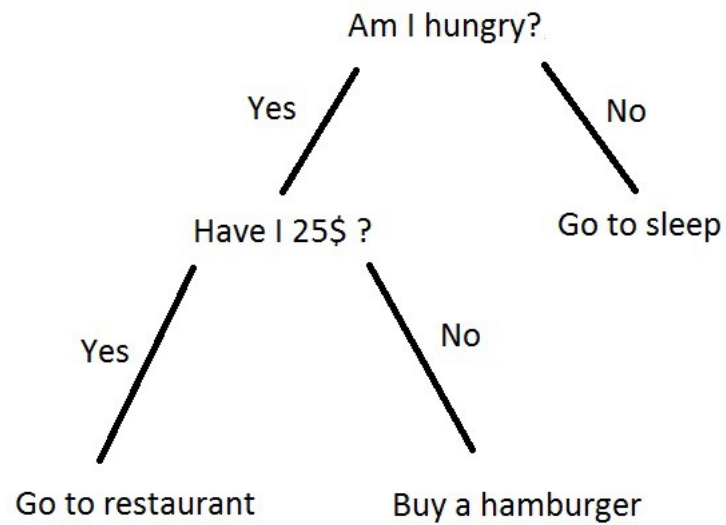


Figure 2.26: Decision Tree

### Information Gain



Udah Tua	20%	Tidak
Pake Topi	10%	Tidak
Kumisan/Jenggotan	20%	Tidak
Ngomongnya gk enak	50%	Akurat

Figure 2.27: Entropi

8. Code 7.

Penjelasan:

menampilkan yang terdapat pada Library Graphviz, apabila benar akan menampilkan hasil output seperti yang terdapat pada gambar atau kalau pengujian gagal akan terdapat error.

- Hasil Code 7:

9. Code 8.

Penjelasan:



```
In [6]: import pandas as Surabaya
...: Jatim = Surabaya.read_csv('D:\Tugas Kuliah\Semester 6\KECERDASAN BUATAN\Python-Artificial-Intelligence-Projects-for-Beginners\Chapter01\dataset\student-mat.csv', sep=';')
...: len(Jatim)
Out[6]: 395
```

Figure 2.28: Hasil Code 1

```
In [7]: Jatim['pass'] = Jatim.apply(lambda row: 1 if (row['G1']+row['G2']+row['G3'])
...: >= 35 else 0, axis=1)
...: Jatim = Jatim.drop(['G1', 'G2', 'G3'], axis=1)
...: Jatim.head()
Out[7]:
  school sex  age address famsize ... Dalc  Walc  health absences pass
0    GP   F   18     U    GT3 ...    1    1    3         6      0
1    GP   F   17     U    GT3 ...    1    1    3         4      0
2    GP   F   15     U    LE3 ...    2    3    3        10      0
3    GP   F   15     U    GT3 ...    1    1    5         2      1
4    GP   F   16     U    GT3 ...    1    2    5         4      0

[5 rows x 31 columns]
```

Figure 2.29: Hasil Code 2

menampilkan hasil perhitungan dari kedua parameter yang terdapat pada code tersebut.

- Hasil Code 8:

10. Code 9.

Penjelasan:

kodingan tersebut mnedefinisikan library sklearn model selection dan import cross val score. Dan kemudian variabel scores mengeksekusi fungsi cross val score(solo, solok att, solok pass, cv=5). Kemudian akan menampilkan nilai dari fungsi akurasi.

- Hasil Code 9:

11. Code 10.

Penjelasan:

berfungsi untuk menampilkan hasil dari fungsi Max Depth dan Accuraccy dari dari Decission Tree. Yaitu menmpilkan data dari angka 1-20.

- Hasil Code 10:

12. Code 11.

Penjelasan:

variable scores akan menampilkan atau mendefinisikan nilai dari variabel score yang mana isi dari variable score Yang mana hasil tampilan dari kodingannya adalah output

- Hasil Code 11:

```
In [8]: Jatim = Surabaya.get_dummies(Jatim, columns=['sex', 'school', 'address', 'famsize', 'Pstatus', 'Mjob', 'Fjob',
...:                                     'reason', 'guardian', 'schoolsup', 'famsup', 'paid', 'activities',
...:                                     'nursery', 'higher', 'internet', 'romantic'])
...: Jatim.head()
Out[8]:
```

	age	Medu	Fedu	...	internet_yes	romantic_no	romantic_yes
0	18	4	4	...	0	1	0
1	17	1	1	...	1	1	0
2	15	1	1	...	1	1	0
3	15	4	2	...	1	0	1
4	16	3	3	...	0	1	0

```
[5 rows x 57 columns]
```

Figure 2.30: Hasil Code 3

```
...: Jatim_train = Jatim[:500]
...: Jatim_test = Jatim[500:]
...:
...: Jatim_train_att = Jatim_train.drop(['pass'], axis=1)
...: Jatim_train_pass = Jatim_train['pass']
...:
...: Jatim_test_att = Jatim_test.drop(['pass'], axis=1)
...: Jatim_test_pass = Jatim_test['pass']
...:
...: Jatim_att = Jatim.drop(['pass'], axis=1)
...: Jatim_pass = Jatim['pass']
...:
...: # number of passing students in whole dataset:
...: import numpy as np
...: print("Passing: %d out of %d (%.2f%%)" % (np.sum(Jatim_pass), len(Jatim_pass), 100*float(np.sum(Jatim_pass)) /
len(Jatim_pass)))
Passing: 166 out of 395 (42.03%)
```

Figure 2.31: Hasil Code 4

### 13. Code 12.

Penjelasan:

pada library matplotlib akan menampilkan gambar grafik pada gambar 12 dari eksekusi fungsi `ax.errorbar`.

- Hasil Code 12:

## 2.4.2 Praktek Penanganan Error

Traceback (most recent call last):

```
File "<ipython-input-33-b4843d06cfa2>", line 1, in <module>
import graphviz
```

ModuleNotFoundError: No module named 'graphviz'

Solusi dari error tersebut adalah memasukkan path dari graphviz ke environment variabels, maka akan menampilkan hasil yaitu sebuah decision tree.

## 2.5 Same Topics

Cite every latest journal with same topic

```
In [33]: from sklearn import tree
...: amanatul = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
...: amanatul = amanatul.fit(Jatin_train_att, Jatin_train_pass)
```

Figure 2.32: Hasil Code 5

```
In [42]: import graphviz
...: dot_data = tree.export_graphviz(amanatul, out_file=None, label="all", impurity=False,
proportions=True,
...: feature_names=list(Jatin_train_att), class_names=["fail", "pass"],
...: filled=True, rounded=True)
...: graph = graphviz.Source(dot_data)
...: graph
Traceback (most recent call last):
  File "<ipython-input-42-554c8002ecf8>", line 1, in <module>
    import graphviz
ModuleNotFoundError: No module named 'graphviz'
```

Figure 2.33: Hasil Code 6

### 2.5.1 Topic 1

cite for first topic

### 2.5.2 Topic 2

if you have two topics you can include here to

## 2.6 Same Method

write and cite latest journal with same method

### 2.6.1 Method 1

cite and paraphrase method 1

### 2.6.2 Method 2

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

```
In [34]: tree.export_graphviz(amanatul, out_file="student-performance.dot", label="all",
...:     impurity=False, proportion=True,
...:     feature_names=list(Jatim_train_att), class_names=["fail",
...:     "pass"],
...:     filled=True, rounded=True)
```

Figure 2.34: Hasil Code 7

```
In [43]: amanatul.score(Jatim_test_att, Jatim_test_pass)
Traceback (most recent call last):

File "<ipython-input-43-8e7ca9451ca2>", line 1, in <module>
  amanatul.score(Jatim_test_att, Jatim_test_pass)

File "C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py", line 288, in score
  return accuracy_score(y, self.predict(X), sample_weight=sample_weight)

File "C:\ProgramData\Anaconda3\lib\site-packages\sklearn\tree\tree.py", line 416, in predict
  X = self._validate_X_predict(X, check_input)

File "C:\ProgramData\Anaconda3\lib\site-packages\sklearn\tree\tree.py", line 377, in
_validate_X_predict
  X = check_array(X, dtype=DTYPE, accept_sparse="csr")

File "C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py", line 582, in
check_array
  context))
```

Figure 2.35: Hasil Code 8

```
In [13]: from sklearn.model_selection import cross_val_score
...: scores = cross_val_score(amanatul, Jatim_att, Jatim_pass, cv=5)
...: # show average score and +/- two standard deviations away (covering 95% of scores)
...: print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
Accuracy: 0.57 (+/- 0.11)
```

Figure 2.36: Hasil Code 9

```
In [15]: for max_depth in range(1, 20):
...:     amanatul = tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
...:     scores = cross_val_score(amanatul, Jatim_att, Jatim_pass, cv=5)
...:     print("Max depth: %d, Accuracy: %0.2f (+/- %0.2f)" % (max_depth, scores.mean(), scores.std() * 2))
Max depth: 1, Accuracy: 0.58 (+/- 0.01)
Max depth: 2, Accuracy: 0.55 (+/- 0.09)
Max depth: 3, Accuracy: 0.56 (+/- 0.07)
Max depth: 4, Accuracy: 0.58 (+/- 0.08)
Max depth: 5, Accuracy: 0.57 (+/- 0.11)
Max depth: 6, Accuracy: 0.58 (+/- 0.05)
Max depth: 7, Accuracy: 0.57 (+/- 0.08)
Max depth: 8, Accuracy: 0.57 (+/- 0.09)
Max depth: 9, Accuracy: 0.57 (+/- 0.05)
Max depth: 10, Accuracy: 0.58 (+/- 0.07)
Max depth: 11, Accuracy: 0.59 (+/- 0.09)
Max depth: 12, Accuracy: 0.57 (+/- 0.09)
Max depth: 13, Accuracy: 0.57 (+/- 0.10)
Max depth: 14, Accuracy: 0.55 (+/- 0.12)
Max depth: 15, Accuracy: 0.57 (+/- 0.07)
Max depth: 16, Accuracy: 0.56 (+/- 0.15)
Max depth: 17, Accuracy: 0.57 (+/- 0.07)
Max depth: 18, Accuracy: 0.55 (+/- 0.08)
Max depth: 19, Accuracy: 0.55 (+/- 0.10)
```

Figure 2.37: Hasil Code 10

```
In [38]:
In [38]: depth_acc = np.zeros((19,3), float)
...: i = 0
...: for max_depth in range(1, 20):
...:     amanatul = tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
...:     scores = cross_val_score(amanatul, Jatim_att, Jatim_pass, cv=5)
...:     depth_acc[i,0] = max_depth
...:     depth_acc[i,1] = scores.mean()
...:     depth_acc[i,2] = scores.std() * 2
...:     i += 1
...:
...:
Out[38]:
array([[1.00000000e+00, 5.79751704e-01, 6.30768599e-01],
       [2.00000000e+00, 5.87540571e-01, 5.55999992e-02],
       [3.00000000e+00, 5.72188413e-01, 6.54746741e-02],
       [4.00000000e+00, 5.79718436e-01, 4.96526806e-02],
       [5.00000000e+00, 5.99717524e-01, 1.13283469e-01],
       [6.00000000e+00, 5.92316618e-01, 7.88112865e-02],
       [7.00000000e+00, 5.92314184e-01, 9.25897095e-02],
       [8.00000000e+00, 6.05004869e-01, 8.47416799e-02],
```

Figure 2.38: Hasil Code 11

```
In [39]: import matplotlib.pyplot as plt
...: fig, ax = plt.subplots()
...: ax.errorbar(depth_acc[:,0], depth_acc[:,1], yerr=depth_acc[:,2])
...: plt.show()
```

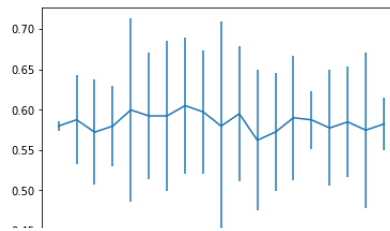


Figure 2.39: Hasil Code 12

```
IPython 6.4.0 -- An enhanced Interactive Python.

In [1]: import graphviz
...: dot_data = tree.export_graphviz(amanatul, out_file=None, label="all", impurity=False, proportion=True,
...: feature_names=list(jatim_train_att), class_names=["fail", "pass"],
...: filled=True, rounded=True)
...: graph = graphviz.Source(dot_data)
...: graph
Traceback (most recent call last):
  File "<ipython-input-1-554c8002ecf8>", line 1, in <module>
    import graphviz
ModuleNotFoundError: No module named 'graphviz'
```

Figure 2.40: HASIL YANG MASIH ERROR

# 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

### 3.4 Aip Suprpto Munari/1164063

#### 3.4.1 Teori

Tugas Harian 5

#### 1. Random Forest Dan Ilustrasi Gambarnya

- Pengertian Random Forest:

Random Forest adalah suatu algoritma yang digunakan pada klasifikasi data dalam jumlah yang besar. Klasifikasi random forest dilakukan melalui penggabungan pohon dengan melakukan training pada sampel data yang dimiliki. Penggunaan pohon (tree) yang semakin banyak akan mempengaruhi akurasi yang akan didapatkan menjadi lebih baik. Penentuan klasifikasi dengan random forest diambil berdasarkan hasil voting dari pohon

yang terbentuk. Pemenang dari pohon yang terbentuk ditentukan dengan vote terbanyak.

Pembangunan pohon pada random forest sampai dengan mencapai ukuran maksimum dari pohon data. Akan tetapi, pembangunan pohon random forest tidak dilakukan pemangkasan yang merupakan sebuah metode untuk mengurangi kompleksitas ruang.

- Ilustrasi Gambar Random Forest :

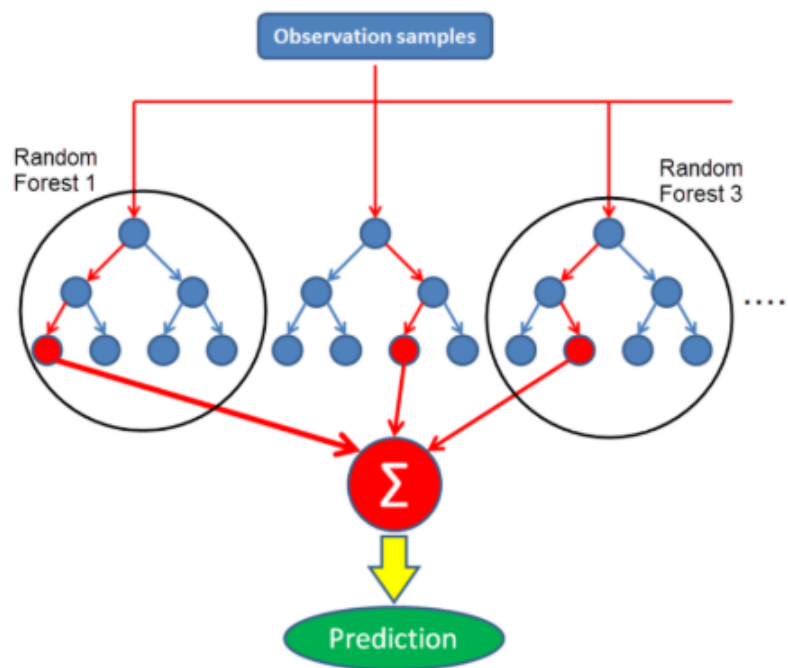


Figure 3.1: Random Forest

## 2. Cara Membaca Dataset

Berikut adalah cara membaca dataset :

- Buka Anaconda Navigator lalu jalankan Syder, kemudian import libraries yang dibutuhkan.
- Masukkan kode python untuk membaca file csv, lalu jalankan

```
dataset = pd.read_csv('Data.csv')
```

Figure 3.2: (b)

(c) Maka pada window console akan menampilkan pesan berikut :

```
In [6]: dataset = pd.read_csv('Data.csv')
```

Figure 3.3: (c)

Name ▲	Type	Size	Value
dataset	DataFrame	(10, 4)	Column names: Country, Age, Salary, Purchased

Figure 3.4: (d)

- (d) Dari explorer dapat terlihat dataset yang terimport.
- (e) Lalu klik dataset cell, maka akan muncul seperti berikut :
- (f) Seperti yang terlihat pada gambar tersebut dataset ini memiliki Kolom Country, Age, dan Salary sebagai independent variable-nya dan kolom Purchased sebagai dependent variable-nya.
- (g) Selanjutnya buat 2 matrix of features yang berisi values dari independent variable dan dependent variable.
- (h) Lalu tuliskan perintah berikut :
- (i) Perintah yang telah dibuat di atas akan membuat sebuah global environment baru dan muncul dataset.
- (j) Klik dataset tersebut maka muncul tabel berisi dataset.

### 3. Cross Validation

- Pengertian Cross Validation :

Cross Validation adalah sebuah teknik validasi model yang digunakan untuk menilai bagaimana hasil analisis statistik akan digeneralisasi ke kumpulan data independen. Cross validation digunakan dengan tujuan prediksi, dan bila kita ingin memperkirakan seberapa akurat model prediksi yang dilakukan dalam sebuah praktek. Tujuan dari cross validation yaitu untuk mendefinisikan dataset guna menguji dalam fase pelatihan untuk membatasi masalah seperti overfitting dan underfitting serta mendapatkan wawasan tentang bagaimana model akan digeneralisasikan ke set data independen.

### 4. Penjelasan / Maksud Dari Score pada :



Index	Country	Age	Salary	Purchased
0	France	44	72000	No
1	Spain	27	46000	Yes
2	Germany	30	54000	No
3	Spain	38	61000	No
4	Germany	40	nan	Yes
5	France	35	58000	Yes
6	Spain	nan	52000	No
7	France	48	79000	Yes
8	Germany	50	83000	No
9	France	37	67000	Yes

Figure 3.5: (e)

```
dataset = read.csv('Data.csv')
```

Figure 3.6: (h)

- Random forest ( 44% )

Maksud arti score 44% pada random forest adalah hasil dari akurasi. Yang menggunakan 5 buah atribut yaitu dari 5 baris pertama dari set pelatihan yang akan memprediksi spesies 10, 28, 156, 10 dan 43.

- Decision Tree ( 27% )

Maksud arti score 27% pada decision tree adalah presentasi hasil dari perhitungan dataset. Dari set tentang burung pipit. Confusion matrix memberi tau hal-hal yang diharapkan, artinya, burung-burung yang terlihat mirip saling bingung satu sama lain.

- SVM ( 29% )

Maksud arti score 29% dari SVM adalah hasil pendekatan jaringan saraf.

Di sini, akurasi adalah 27%, yang kurang dari akurasi 44% sebelumnya. Oleh karena itu, decision tree menjadi lebih buruk. Jika kita menggunakan Support Vector Machine (SVM), yang merupakan neural pendekatan jaringan, outputnya 29%. Jadi 29% pada SVM merupakan hasil outputnya.

Hasil tersebut didapat dari hasil validasi silang untuk memastikan bahwa membagi training test dengan cara yang berbeda. Sehingga didapat outputnya 44% untuk hutan acak, 27% untuk pohon keputusan, dan 29% untuk SVM.

## 5. Confusion Matrix Dan Ilustrasinya

- Cara Membaca Confusion Matrix :

Perhitungan confusion matrix adalah sebagai berikut, akan saya beri contoh sederhana yaitu pengambilan keputusan untuk mendapatkan bantuan beasiswa. Saya menggunakan dua atribut, yaitu rekening listrik dan gaji. Yang pertama kita lakukan yaitu mencari 4 nilai yaitu a,b,c, dan d:

$$a = 4$$

$$b = 1$$

$$c = 1$$

$$d = 2$$

Kemudian kita dapat mencari nilai Recall, Precision, accuracy dan Error Rate

$$\text{Recall} = 2/(1+2) = 0,66$$

$$\text{Precision} = 2/(1+2) = 0,66$$

$$\text{Accuracy} = (4+2)/(4+1+1+2) = 0,75$$

$$\text{Error Rate} = (1+1)/(4+1+1+4) = 0,2$$

Ilustrasi Confusion Matrix :

## 6. Voting Random Forest Dan Ilustrasi Gambarnya.

- Pengertian Voting pada Random Forest :

Metode ensemble dapat mencapai akurasi tinggi dengan membangun beberapa pengklasifikasi dan menjalankan masing-masing secara mandiri. Ketika classifier membuat sebuah keputusan, kamu dapat memanfaatkan yang terbaik keputusan umum dan rata-rata. Jika kita menggunakan metode yang paling umum, itu disebut voting.

Rekan	IPK	Psikologi	Wawancara
P1	Bagus	Tinggi	Baik
P2	Bagus	Sedang	Baik
P3	Bagus	Sedang	Buruk
P4	Bagus	Rendah	Buruk
P5	Cukup	Tinggi	Baik
P6	Cukup	Sedang	Baik
P7	Cukup	Sedang	Buruk
P8	Cukup	Rendah	Buruk
P9	Kurang	Tinggi	Baik
P10	Kurang	Sedang	Buruk
P11	Kurang	Rendah	Baik

Figure 3.7: Confussion Matrik

- Ilustrasi Gambar Voting Random Forest :

### 3.4.2 Praktek

1. Aplikasi Sederhana Menggunakan Pandas

Penjelasan kodingan :

- (a) Memanggil library.
- (b) Membaca dari file pandas.
- (c) Menampilkan hasil

Sehingga menghasilkan :

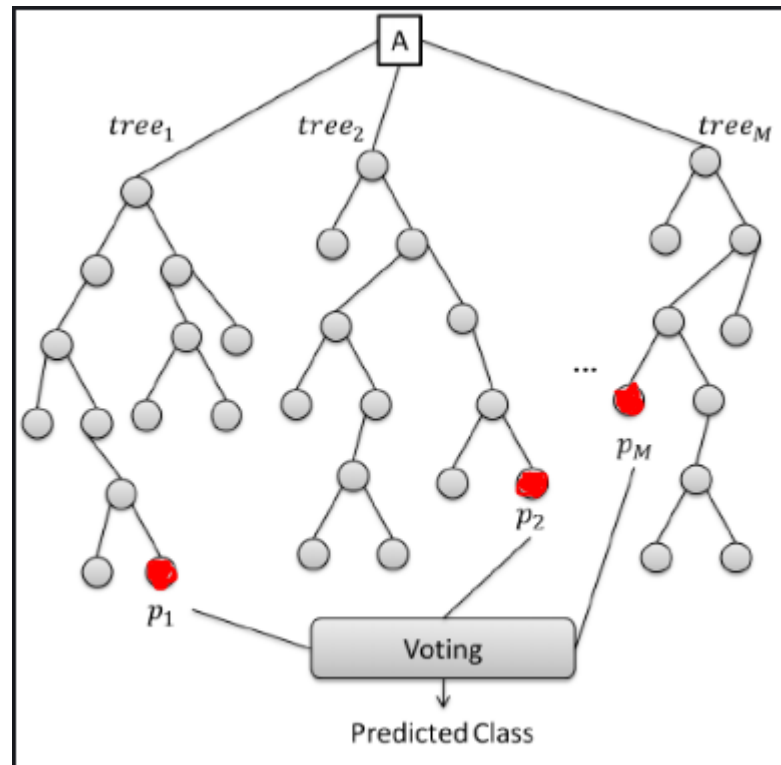


Figure 3.8: Voting Random forest

```
import pandas as pd
ufo = pd.read_csv('http://bit.ly/uforeports')
type(ufo)
ufo.head()
```

Figure 3.9: Aplikasi Pandas

## 2. Aplikasi Sederhana Menggunakan Numpy

Penjelasan kodingan :

- (a) Memanggil library
- (b) Membuat variable dengan value linspace
- (c) Menampilkan hasil value

Sehingga menghasilkan :

## 3. Aplikasi Sederhana Menggunakan Matplotlib

Penjelasan kodingan :

- (a) Memanggil library
- (b) Membuat variable yang berisi bahasa pemrograman

```

In [24]: import pandas as pd
...: ufo = pd.read_csv('http://bit.ly/uforeports')
...: type(ufo)
...: ufo.head()
Out[24]:
   City Colors Reported  ... State      Time
0   Ithaca           NaN  ...  NY  6/1/1930  22:00
1  Willingboro        NaN  ...  NJ  6/30/1930  20:00
2   Holyoke          NaN  ...  CO  2/15/1931  14:00
3   Abilene          NaN  ...  KS  6/1/1931  13:00
4 New York Worlds Fair  NaN  ...  NY  4/18/1933  19:00

[5 rows x 5 columns]
In [25]:

```

Figure 3.10: Hasil Pandas

```

import numpy as np
a = np.array([2,3,4])
a
a = np.arange(1, 12, 2)
a
a = np.linspace(1, 12, 6)
a

```

Figure 3.11: Aplikasi Numpy

- (c) Membuat variable dengan value
- (d) Membuat variable untuk menentukan garis
- (e) Membuat garis koordinat
- (f) Menampilkan hasil

Sehingga menghasilkan :

#### 4. Program Klasifikasi Random Fores

- Yang pertama dataset akan dibaca.
- Selanjutnya sebagian data awal akan dilihat dengan menggunakan listing.
- Selanjutnya jumlah data dilihat dengan menggunakan listing.
- Lalu atribut diubah menjadi kolom dengan menggunakan perintah pivot.
- Selanjutnya atribut yang telah diubah, sebagian data awalnya akan dilihat dengan menggunakan listing kembali.
- Selanjutnya atribut yang telah diubah, jumlah data dilihat dengan menggunakan listing kembali.
- Lalu mengelompokkan burung kedalam spesies yang sama dengan dua kolom imgid dan label.
- Lalu melakukan pivot dimana imgid menjadi index.
- Selanjutnya imgid, sebagian data awalnya akan dilihat dengan menggunakan listing untuk mengecek data.

```
In [27]: import numpy as np
...: a = np.array([2,3,4])
...: a
...: a = np.arange(1, 12, 2)
...: a
...: a = np.linspace(1, 12, 6)
...: a
Out[27]: array([ 1. ,  3.2,  5.4,  7.6,  9.8, 12. ])
```

Figure 3.12: Hasil Numpy

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 10, 100)

plt.plot(x, np.sin(x))
plt.plot(x, np.cos(x))

plt.show()
```

Figure 3.13: Aplikasi Matplotlib

- Selanjutnya imgid, jumlah data dilihat dengan menggunakan listing untuk mengecek data.
- Lalu melakukan join karena isi datanya adalah sama di antara dua data. Sehingga mendapatkan data ciri labelnya sehingga bisa dikategorikan.
- Kemudian label yang didepan di drop dan berikan label pada data yang telah dilakukan join dengan perintah listing.
- Lalu cek kembali isinya dengan perintah listing.
- Kemudian data dibagi menjadi dua bagian, dimana 8000 row pertama merupakan data training dan sisanya adalah data testing.
- Kelas random forest selanjutnya dipanggil dengan RandomForestClassifier, dengan banyak kolom yang telah ditentukan oleh max feature.
- Kemudian untuk membangun random forest dilakukan perintah fitting dengan maksimum fitur sebanyak 50.
- Kemudian lihat hasilnya dengan perintah predict.
- Lalu akan terlihat hasil score dari klasifikasi.

## 5. Program Klasifikasi Confusion Matrix

- Setelah melakukan random forest kemudian dipetakan ke dalam confusion matrix.
- Lalu melihat hasilnya.
- Kemudian dilakukan perintah plot.
- Selanjutnya nama data akan di set agar plot sumbunya sesuai.

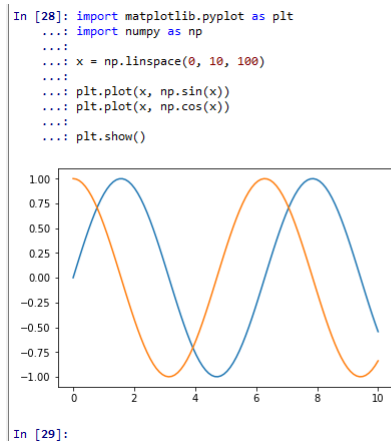


Figure 3.14: Hasil Matplotlib

```
In [1]: import pandas as pd
...:
...: # some lines have too many fields (?), so skip bad lines
...: imgatt = pd.read_csv('D:/KULIAH/SEMESTER 6/KECERDASAN BUATAN/CUB_200_2011/
attributes/image_attribute_labels.txt',
...: sep='\s+', header=None, error_bad_lines=False,
warn_bad_lines=False,
...: usecols=[0,1,2], names=['imgid', 'attid', 'present'])
...:
...: # description from dataset README:
...: #
...: # The set of attribute labels as perceived by MTurkers for each image
...: # is contained in the file attributes/image_attribute_labels.txt, with
...: # each line corresponding to one image/attribute/worker triplet:
...: #
...: # <image_id> <attribute_id> <is_present> <certainty_id> <time>
...: #
...: # where <image_id>, <attribute_id>, <certainty_id> correspond to the IDs
...: # in images.txt, attributes/attributes.txt, and attributes/certainties.txt
...: # respectively. <is_present> is 0 or 1 (1 denotes that the attribute is
...: # present). <time> denotes the time spent by the MTurker in seconds.
```

```
In [2]:
```

Figure 3.15: Membaca Data File

- Setelah label berubah, maka dilakukan perintah plot.

## 6. Program Klasifikasi SVM dan Decision Tree

### (a) Program Decision Tree

Mengklasifikasikan dataset yang sama menggunakan decision tree.

### (b) Program Klasifikasi SVM

Mengklasifikasikan dataset yang sama menggunakan SVM.

## 7. Program Cross Validation

- Melakukan pengecekan cross validation untuk random forest.
- Melakukan pengecekan cross validation untuk decision tree.
- Melakukan pengecekan cross validation untuk SVM.

## 8. Program Pengamatan Komponen Informasi

```
In [2]: imgatt.head()
Out[2]:
```

	imgid	attid	present
0	1	1	0
1	1	2	0
2	1	3	0
3	1	4	0
4	1	5	1

```
In [3]: |
```

Figure 3.16: Melihat Data Sebagian

```
In [3]: imgatt.shape
Out[3]: (3677856, 3)
In [4]:
```

Figure 3.17: Melihat Jumlah Data

- Melakukan pengamatan komponen informasi untuk mengetahui berapa banyak tree yang dibuat, atribut yang dipakai, dan informasi lainnya.
- Melakukan plot informasi agar bisa dibaca.

### 3.4.3 Penanganan Error

1. Skrinshot Error
2. Tuliskan kode eror dan jenis errornya

Kode Error = FileNotFoundError: File b'data/CUB 200 2011/attributes/image attribute labels . txt' does not exist

Jenis Error = File not found

3. Solusi Pemecahan Masalah Error

Solusi dari error yang terjadi pada nomor 1 adalah perbaiki alamat direktorinya sebagai berikut :

Sehingga didapat hasil seperti berikut :

sectionAndi Muhammad Aslam/1164064

1. Random Forest merupakan algoritma yang digunakan terhadap klasifikasi data dalam jumlah yang besar. Klasifikasi pada random forest dilakukan dengan penggabungan decision tree dengan melakukan training terhadap sampel data yang dimiliki. Pembentukan decision tree menggunakan sample data berupa variable secara acak lalu menjalankan klasifikasi pada semua tree yang terbentuk. Random forest berupa Decision Tree agar dapat melakukan proses seleksi. Decision tree yang di buat dibagi secara strategis dari data pada kelas yang



```
In [4]: imgatt2 = imgatt.pivot(index='imgid', columns='attid', values='present')
In [5]:
```

Figure 3.18: Mengubah menjadi kolom

```
In [5]: imgatt2.head()
Out[5]:
attid 1 2 3 4 5 6 7 ... 306 307 308 309 310 311 312
imgid
1      0 0 0 0 1 0 0 ... 0 0 1 0 0 0 0
2      0 0 0 0 0 0 0 ... 0 0 0 0 0 0 0
3      0 0 0 0 1 0 0 ... 0 0 1 0 0 1 0
4      0 0 0 0 1 0 0 ... 1 0 0 1 0 0 0
5      0 0 0 0 1 0 0 ... 0 0 0 0 0 0 0

[5 rows x 312 columns]
In [6]: |
```

Figure 3.19: Lihat sebagian data awal

sama. Pemecahan digunakan untuk membagi data berdasarkan jenis atribut yang digunakan.. ??

2. Download dataset terdahulu kemudian buka software spyder untuk melihat isi dataset. Data yang di download berupa extensi file bernama .txt yang terdapat class dari field. Contohnya pada data jenis burung memiliki file index dan angka, dimana index berisi angka yang memiliki makna berupa jenis burung atau bahkan nama burung sedangkan field memiliki isi nilai berupa 0 dan 1 yang dimana sifatnya boolean, Ya dan Tidak. Hal ini dikarenakan komputer hanya dapat membaca bilangan biner maka dari itu field yang di isikan berupa angka. Artinya angka 0 berarti tidak dan angka 1 berarti Ya.
3. Cross validation adalah metode statistik yang digunakan untuk memperkirakan keterampilan model pembelajaran mesin. Ini biasanya digunakan dalam pembelajaran mesin yang diterapkan untuk membandingkan dan memilih model untuk masalah pemodelan prediktif yang diberikan karena mudah dipahami, mudah diimplementasikan, dan menghasilkan estimasi keterampilan yang umumnya memiliki bias lebih rendah daripada metode lainnya.
4. Penjelasan Score
  - Pada score 44% pada random forest berupa hasil akurasi.
  - Pada score 27% pada decision tree adalah presentasi hasil dari perhitungan dataset.
  - Pada score 29% dari SVM adalah hasil pendekatan neural network.

```
In [6]: imgatt2.shape
Out[6]: (11788, 312)
In [7]: |
```

Figure 3.20: Melihat jumlah data

```
In [7]: imglabels = pd.read_csv("D:/KULIAH/SEMESTER 6/KECERDASAN BUATAN/CUB_200_2011/
image_class_labels.txt",
...:                               sep=' ', header=None, names=['imgid', 'label'])
...:
...: imglabels = imglabels.set_index('imgid')
...:
...: # description from dataset README:
...: #
...: # The ground truth class labels (bird species labels) for each image are contained
...: # in the file image_class_labels.txt, with each line corresponding to one image:
...: #
...: # <image_id> <class_id>
...: #
...: # where <image_id> and <class_id> correspond to the IDs in images.txt and
...: # classes.txt,
...: # respectively.
In [8]: |
```

Figure 3.21: Mengelompokkan burung

- Hasil tersebut didapat dari hasil validasi silang untuk memastikan bahwa membagi training test dengan cara yang berbeda. Sehingga dapat diketahui hasil output yaitu 44% untuk hutan, 27% untuk pohon keputusan, dan 29% untuk SVM.

5. Untuk membaca confusion matriks dapat menggunakan source code berikut :

```
import numpy as np
np.set_printoptions(precision=2)
plt.figure(figsize=(60,60), dpi=300)
plot_confusion_matrix(cm, classes=birds, normalize=True)
plt.show()
```

Dimana numpy dapat mengelola data yang berhubungan pada matrix. Pada perintah code tersebut digunakan dalam melakukan read pada dataset burung dengan menggunakan metode confusion matrix. Dalam confusion matrix memiliki 4 istilah yaitu True Positive yang merupakan data positif yang terdeteksi benar, True Negatif yang merupakan data negatif akan tetapi terdeteksi benar, False Positif merupakan data negatif namun terdeteksi sebagai data positif, False Negatif merupakan data positif namun terdeteksi sebagai data negatif. Adapun contoh hasil read dataset menggunakan confusion matrix dapat dilihat pada figure ??

6. Untuk mengetahui confusion matriks kita dapat melihat contoh klasifikasi dari biner berikut ini :

```

In [8]: imglabels.head()
Out[8]:
  imgid  label
1      1      1
2      2      1
3      3      1
4      4      1
5      5      1
In [9]: |

```

Figure 3.22: Melakukan pivot

```

In [9]: imglabels.shape
Out[9]: (11788, 1)
In [10]:

```

Figure 3.23: Melihat data awal imgid

7. Voting merupakan proses pemilihan dari tree yang dimana akan dimunculkan hasilnya dan disimpulkan menjadi informasi yang pasti.

Pada figure Voting terdapat Decision Tree yang terbagi menjadi 3 Branch yaitu tree 1, tree 2, dan tree 3. Pada tree tersebut akan dilakukan proses voting. Pada masing-masing tree tersebut memiliki data-data yang berbeda, yang di mana data tersebut akan di pilih dengan cara voting. Hasil voting dari setiap tree tersebut menunjukkan data pada setiap tree, Di sini kita dapat menghitung akurasi dengan menambahkan angka secara diagonal, sehingga ini semua adalah contoh yang diklasifikasikan dengan benar, dan membagi jumlah tersebut dengan jumlah semua angka dalam matriks.

```
In [10]: df = imgatt2.join(imglabels)
...: df = df.sample(frac=1)

In [11]: |
```

Figure 3.24: Melihat jumlah data imgid

```
In [11]: df_att = df.iloc[:, :312]
...: df_label = df.iloc[:, 312:]

In [12]: |
```

Figure 3.25: Data ciri label dari join

```
In [12]: df_att.head()
Out[12]:
```

	1	2	3	4	5	6	7	...	306	307	308	309	310	311	312
imgid								...							
3218	0	0	0	0	0	0	0	...	1	1	0	0	0	1	0
4781	0	1	0	0	0	0	0	...	0	0	0	0	0	1	0
7610	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1
2327	0	0	0	0	0	0	1	...	1	0	0	0	0	0	1
2462	0	0	0	0	0	0	1	...	0	0	0	1	0	0	0

```
[5 rows x 312 columns]

In [13]: |
```

Figure 3.26: Mengubah menjadi kolom

```
In [13]: df_label.head()
Out[13]:
```

	label
imgid	
3218	56
4781	82
7610	130
2327	41
2462	43

```
In [14]: |
```

Figure 3.27: Melihat isi data frame

```
In [14]: df_train_att = df_att[:600]
...: df_train_label = df_label[:600]
...: df_test_att = df_att[600:]
...: df_test_label = df_label[600:]
...:
...: df_train_label = df_train_label['label']
...: df_test_label = df_test_label['label']

In [15]: |
```

Figure 3.28: Membagi data

```
In [15]: from sklearn.ensemble import RandomForestClassifier
...: clf = RandomForestClassifier(max_features=20, random_state=0, n_estimators=100)

In [16]: |
```

Figure 3.29: Kelas Random Forest

```
In [16]: clf.fit(df_train_att, df_train_label)
Out[16]:
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                        max_depth=None, max_features=20, max_leaf_nodes=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=None,
                        oob_score=False, random_state=0, verbose=0, warm_start=False)

In [17]: |
```

Figure 3.30: Membangun Random forest

```
In [17]: print(clf.predict(df_train_att.head()))
[ 56  82 130  41  43]
```

Figure 3.31: Melihat hasil

```
In [18]: clf.score(df_test_att, df_test_label)
Out[18]: 0.21290668573471577
In [19]: |
```

Figure 3.32: Lihat hasil score

```
In [19]: from sklearn.metrics import confusion_matrix
...: pred_labels = clf.predict(df_test_att)
...: cm = confusion_matrix(df_test_label, pred_labels)

In [20]: |
```

Figure 3.33: Memetakan ke confusion matrix

```
In [20]: cm
Out[20]:
array([[ 7,  3,  0, ...,  0,  0,  0],
       [ 4, 26,  0, ...,  0,  0,  0],
       [ 1,  2,  0, ...,  0,  0,  0],
       ...,
       [ 0,  0,  0, ...,  5,  0,  0],
       [ 0,  0,  0, ...,  1,  0,  0],
       [ 0,  0,  0, ...,  0,  0, 19]], dtype=int64)

In [21]: |
```

Figure 3.34: Melihat hasil

```
In [21]: import matplotlib.pyplot as plt
...: import itertools
...: def plot_confusion_matrix(cm, classes,
...:                           normalize=False,
...:                           title='Confusion matrix',
...:                           cmap=plt.cm.Blues):
...:     """
...:     This function prints and plots the confusion matrix.
...:     Normalization can be applied by setting 'normalize=True'.
...:     """
...:     if normalize:
...:         cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
...:         print("Normalized confusion matrix")
...:     else:
...:         print('Confusion matrix, without normalization')
...:
...:     print(cm)
...:
...:     plt.imshow(cm, interpolation='nearest', cmap=cmap)
...:     plt.title(title)
...:     #plt.colorbar()
...:     tick_marks = np.arange(len(classes))
...:     plt.xticks(tick_marks, classes, rotation=90)
...:     plt.yticks(tick_marks, classes)
...:
...:     fmt = '.2f' if normalize else 'd'
...:     thresh = cm.max() / 2.
...:     #for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
...:     #    plt.text(j, i, format(cm[i, j], fmt),
...:                 #    horizontalalignment="center"
```

Figure 3.35: Melakukan Plot



```
In [26]: from sklearn.model_selection import cross_val_score
...: scores = cross_val_score(clf, df_train_att, df_train_label, cv=5)
...: # show average score and +/- two standard deviations away (covering 95% of
scores)
...: print("Accuracy: %.2f (+/- %.2f)" % (scores.mean(), scores.std() * 2))
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:652: Warning:
The least populated class in y has only 1 members, which is too few. The minimum number of
members in any class cannot be less than n_splits=5.
% (min_groups, self.n_splits)), Warning)
Accuracy: 0.27 (+/- 0.13)
```

Figure 3.40: Pengecekan cross validation random forest

```
In [27]: scorestree = cross_val_score(clftree, df_train_att, df_train_label, cv=5)
...: print("Accuracy: %.2f (+/- %.2f)" % (scorestree.mean(), scorestree.std() * 2))
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:652: Warning:
The least populated class in y has only 1 members, which is too few. The minimum number of
members in any class cannot be less than n_splits=5.
% (min_groups, self.n_splits)), Warning)
Accuracy: 0.11 (+/- 0.09)
```

Figure 3.41: Pengecekan cross validation decision tree

```
...: print("Accuracy: %.2f (+/- %.2f)" % (scorestree.mean(), scorestree.std() * 2))
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:652: Warni
The least populated class in y has only 1 members, which is too few. The minimum number
members in any class cannot be less than n_splits=5.
% (min_groups, self.n_splits)), Warning)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The
default value of gamma will change from 'auto' to 'scale' in version 0.22 to account
better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this
warning.
"avoid this warning.", FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The
default value of gamma will change from 'auto' to 'scale' in version 0.22 to account
better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this
warning.
"avoid this warning.", FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The
default value of gamma will change from 'auto' to 'scale' in version 0.22 to account
better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this
warning.
"avoid this warning.", FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The
default value of gamma will change from 'auto' to 'scale' in version 0.22 to account
better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this
warning.
"avoid this warning.", FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The
default value of gamma will change from 'auto' to 'scale' in version 0.22 to account
better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this
warning.
"avoid this warning.", FutureWarning)
Accuracy: 0.03 (+/- 0.04)
```

Figure 3.42: Pengecekan cross validation SVM

```
...: print("Accuracy: %.2f (+/- %.2f)" % (scorestree.mean(), scorestree.std() * 2))
The least populated class in y has only 1 members, which is too few. The minimum number of
members in any class cannot be less than n_splits=5.
% (min_groups, self.n_splits)), Warning)
Max features: 9, num estimators: 18, accuracy: 0.17 (+/- 0.10)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:652: Warning:
The least populated class in y has only 1 members, which is too few. The minimum number of
members in any class cannot be less than n_splits=5.
% (min_groups, self.n_splits)), Warning)
Max features: 9, num estimators: 22, accuracy: 0.17 (+/- 0.08)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:652: Warning:
The least populated class in y has only 1 members, which is too few. The minimum number of
members in any class cannot be less than n_splits=5.
% (min_groups, self.n_splits)), Warning)
Max features: 9, num estimators: 26, accuracy: 0.21 (+/- 0.14)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:652: Warning:
The least populated class in y has only 1 members, which is too few. The minimum number of
members in any class cannot be less than n_splits=5.
% (min_groups, self.n_splits)), Warning)
Max features: 9, num estimators: 30, accuracy: 0.18 (+/- 0.10)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:652: Warning:
The least populated class in y has only 1 members, which is too few. The minimum number of
members in any class cannot be less than n_splits=5.
% (min_groups, self.n_splits)), Warning)
Max features: 9, num estimators: 34, accuracy: 0.19 (+/- 0.12)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:652: Warning:
The least populated class in y has only 1 members, which is too few. The minimum number of
members in any class cannot be less than n_splits=5.
% (min_groups, self.n_splits)), Warning)
Max features: 9, num estimators: 38, accuracy: 0.21 (+/- 0.12)

In [30]:
```

Figure 3.43: Pengamatan Komponen

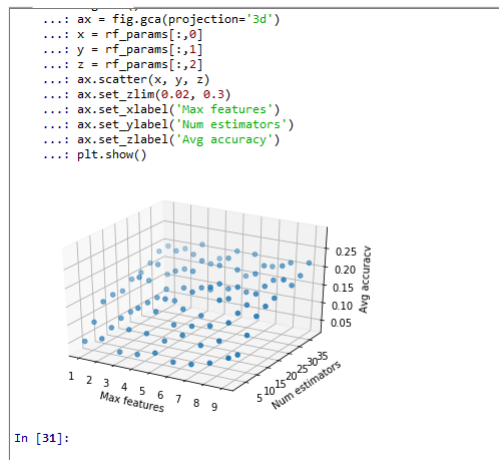


Figure 3.44: Plot informasi

```

FileNotFoundError: File b'data/CUB_200_2011/attributes/image_attribute_labels.txt' does not exist

```

Figure 3.45: Skrintut Error

```

# some lines have too many fields (?), so skip bad lines
imgatt = pd.read_csv('D:/KULIAH/SEMESTER 6/KECERDASAN BUATAN/CUB_200_2011/attributes/image_attribute_labels.txt',
                    sep='\\s+', header=None, error_bad_lines=False, warn_bad_lines=False,
                    usecols=[0,1,2], names=['imgid', 'attid', 'present'])

# description from dataset README:
#
# The set of attribute labels as perceived by MTurkers for each image
# is contained in the file attributes/image_attribute_labels.txt, with
# each line corresponding to one image/attribute/worker triplet:
#
# <image_id> <attribute_id> <is_present> <certainty_id> <time>
#
# where <image_id>, <attribute_id>, <certainty_id> correspond to the IDs
# in images.txt, attributes/attributes.txt, and attributes/certainties.txt
# respectively. <is_present> is 0 or 1 (1 denotes that the attribute is
# present). <time> denotes the time spent by the MTurker in seconds.

```

Figure 3.46: Penyelesaian

```

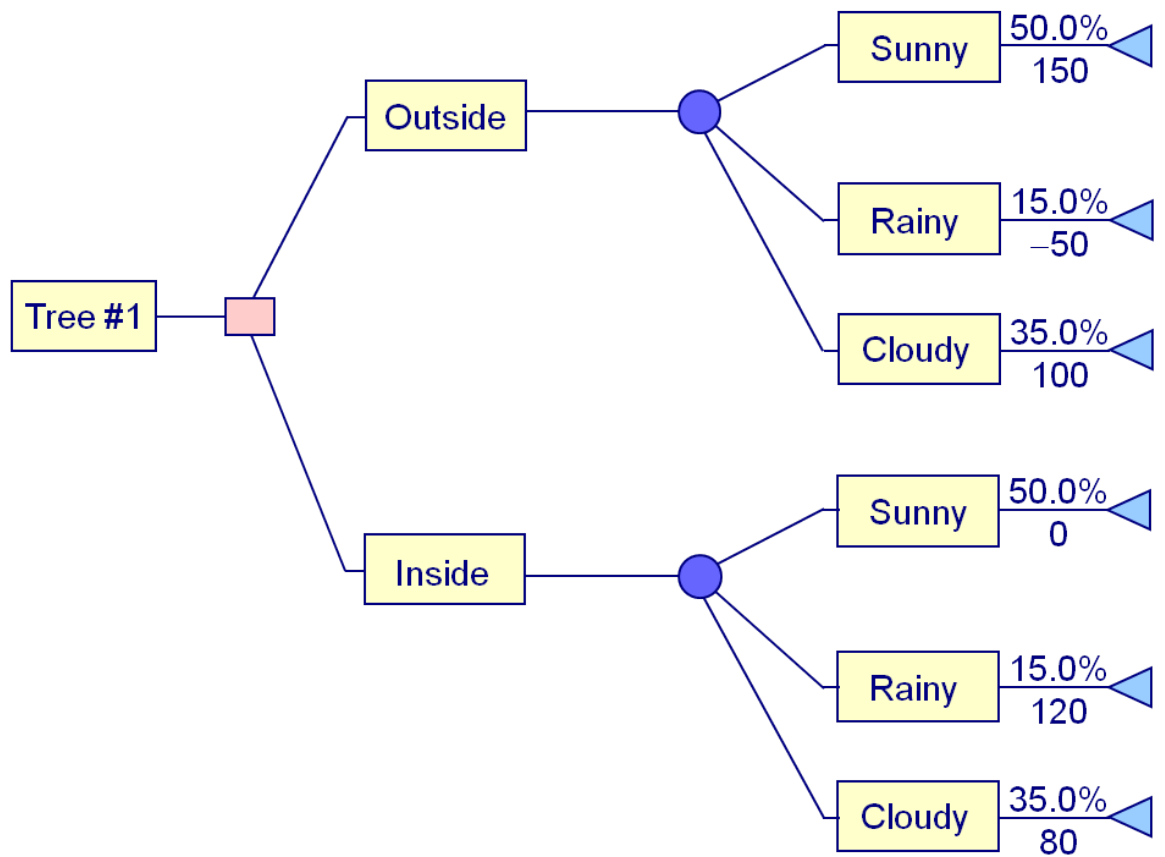
In [1]: import pandas as pd
...:
...: # some lines have too many fields (?), so skip bad lines
...: imgatt = pd.read_csv('D:/KULIAH/SEMESTER 6/KECERDASAN BUATAN/CUB_200_2011/
...: attributes/image_attribute_labels.txt',
...:                      sep='\\s+', header=None, error_bad_lines=False,
warn_bad_lines=False,
...:                      usecols=[0,1,2], names=['imgid', 'attid', 'present'])
...:
...: # description from dataset README:
...: #
...: # The set of attribute labels as perceived by MTurkers for each image
...: # is contained in the file attributes/image_attribute_labels.txt, with
...: # each line corresponding to one image/attribute/worker triplet:
...: #
...: # <image_id> <attribute_id> <is_present> <certainty_id> <time>
...: #
...: # where <image_id>, <attribute_id>, <certainty_id> correspond to the IDs
...: # in images.txt, attributes/attributes.txt, and attributes/certainties.txt
...: # respectively. <is_present> is 0 or 1 (1 denotes that the attribute is
...: # present). <time> denotes the time spent by the MTurker in seconds.

In [2]:

```

Figure 3.47: Hasil





tree.jpg

Figure 3.48: Random Forest.

	Predicted: NO	Predicted: YES
Actual: NO	TN = ??	FP = ??
Actual: YES	FN = ??	TP = ??

Figure 3.49: Tabel Confusion Matriks

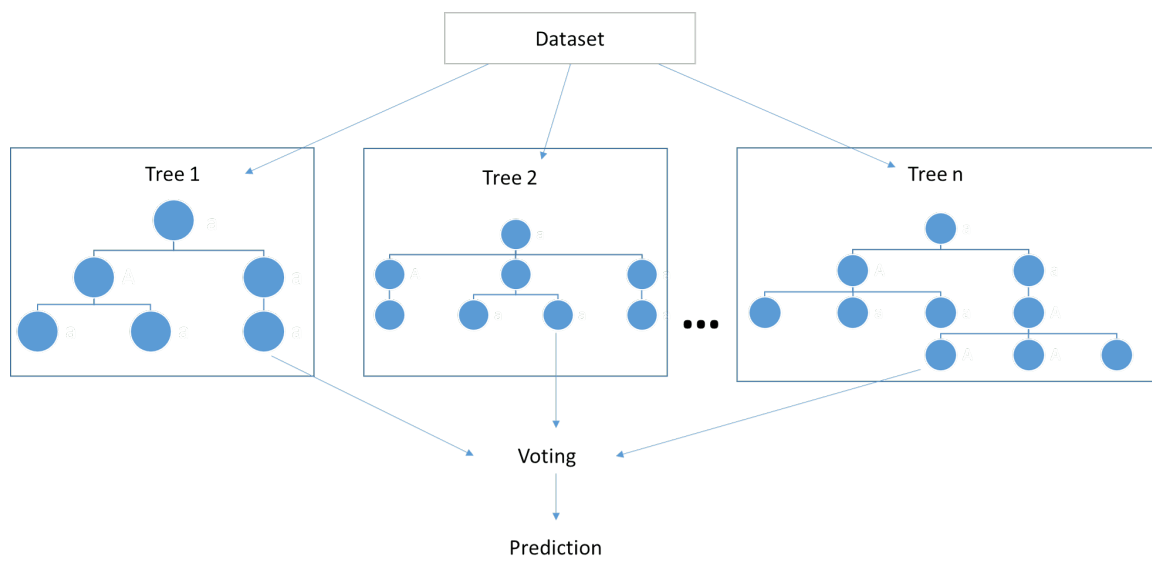


Figure 3.50: Voting

# 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] Cahyo Darujati and Agustinus Bimo Gumelar. Pemanfaatan teknik supervised untuk klasifikasi teks bahasa indonesia. *Jurnal Bandung Text Mining*, 16(1):5–1, 2012.
- [2] 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.
- [3] Helfi Nasution. Implementasi logika fuzzy pada sistem kecerdasan buatan. *ELKHA*, 4(2), 2012.
- [4] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.