

# 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 penggerjaan 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 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 Suprapto 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 Eror 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 Suprapto 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 Suprapto 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 dengan 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 ilustrasi . . . . .	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 Methods</b>	<b>50</b>
3.1 The data . . . . .	50
3.2 Method 1 . . . . .	50
3.3 Method 2 . . . . .	50
3.4 Aip Suprapto Munari/1164063 . . . . .	50
3.4.1 Teori . . . . .	50
3.4.2 Praktek . . . . .	55
3.4.3 Penanganan Error . . . . .	60
3.4.4 Praktek . . . . .	63
<b>4 Experiment and Result</b>	<b>83</b>
4.1 Experiment . . . . .	83
4.2 Result . . . . .	83
4.3 Aip Suprapto Munari/1164063 . . . . .	83
4.3.1 Teori . . . . .	83
4.4 BAGIAN PRAKTEK . . . . .	85
4.5 Aip Suprapto Munari /1164063 . . . . .	85
4.5.1 Aplikasi Sederhana Menggunakan Pandas . . . . .	85
4.5.2 Memecah DataFrame Menjadi 2 Dataframe . . . . .	87
4.5.3 Vektorisasi Dan Klasifikasi Dari Data Youtube05-Shakira Dengan Decision Tree . . . . .	87
4.5.4 Vektorisasi Dan Klasifikasi Dari Data Youtube05-Shakira Dengan SVM . . . . .	88
4.5.5 Vektorisasi Dan Klasifikasi Dari Data Youtube05-Shakira Dengan Decision Tree 2 . . . . .	88
4.5.6 Plotting Confusion Matrix . . . . .	89
4.5.7 Menjalankan Program Cross Validation . . . . .	90
4.5.8 Program Pengamatan Komponen Informasi . . . . .	91
4.6 Penanganan Error . . . . .	91
4.6.1 Error Index . . . . .	92
4.7 Andi Muhammad Aslam/1164064 . . . . .	92
4.7.1 Teori . . . . .	92
4.8 Andi Muhammad Aslam/1164064 . . . . .	94
4.8.1 Praktek Program . . . . .	94
4.8.2 Penanganan Eror . . . . .	96

<b>5 Conclusion</b>	<b>103</b>
5.1 Conclusion of Problems . . . . .	103
5.2 Conclusion of Method . . . . .	103
5.3 Conclusion of Experiment . . . . .	103
5.4 Conclusion of Result . . . . .	103
5.5 Andi Muh Aslam/1164064 . . . . .	103
5.5.1 Teori . . . . .	103
5.5.2 Praktek Program . . . . .	106
5.6 Aip Suprapto Munari/1164063 . . . . .	118
5.6.1 Teori . . . . .	118
5.7 PRAKTEK PROGRAM . . . . .	121
5.8 Aip Suprapto Munari/1164063 . . . . .	121
5.8.1 Mencoba Dataset . . . . .	121
5.8.1.1 Vektor . . . . .	121
5.8.1.2 Similariti . . . . .	123
5.8.2 Extract Words dan PermuteSentences . . . . .	123
5.8.2.1 Extract Words . . . . .	123
5.8.2.2 PermuteSentences . . . . .	124
<b>6 Discussion</b>	<b>126</b>
<b>7 Discussion</b>	<b>127</b>
<b>8 Discussion</b>	<b>128</b>
<b>9 Discussion</b>	<b>129</b>
<b>10 Discussion</b>	<b>130</b>
<b>11 Discussion</b>	<b>131</b>
<b>12 Discussion</b>	<b>132</b>
<b>13 Discussion</b>	<b>133</b>
<b>14 Discussion</b>	<b>134</b>
<b>A Form Penilaian Jurnal</b>	<b>135</b>
<b>B FAQ</b>	<b>138</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 Memperbarui Parameter . . . . .	13
1.20	MultiClass Classifier . . . . .	14
1.21	MultiClass Classifier biner 2D . . . . .	14
1.22	MultiLabel Classifier . . . . .	14
1.23	Eror 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 Clustering. . . . .	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 Classification . . . . .	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 Melalukan 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 . . . . .	65
3.27 Melihat isi data frame . . . . .	65
3.28 Membagi data . . . . .	66
3.29 Kelas Random Forest . . . . .	66
3.30 Membangun Random forest . . . . .	67
3.31 Melihat hasil . . . . .	67
3.32 Lihat hasil score . . . . .	68
3.33 Memetakan ke confusion matrix . . . . .	68
3.34 Melihat hasil . . . . .	69
3.35 Melakukan Plot . . . . .	69
3.36 Plotting nama data . . . . .	70
3.37 Melakukan perintah plot . . . . .	70
3.38 Klasifikasi menggunakan decision tree . . . . .	71
3.39 Klasifikasi menggunakan SVM . . . . .	71
3.40 Pengecekan cross validation random forest . . . . .	72
3.41 Pengecekan cross validation decision tree . . . . .	72
3.42 Pengecekan cross validation SVM . . . . .	72
3.43 Pengamatan Komponen . . . . .	72
3.44 Plot informasi . . . . .	73
3.45 Skrinsut Error . . . . .	73
3.46 Penyelesaian . . . . .	73
3.47 Hasil . . . . .	73
3.48 Random Forest. . . . .	74
3.49 Tabel Confusion Matriks . . . . .	74
3.50 Voting . . . . .	75
3.51 Pandas . . . . .	75
3.52 Numpy . . . . .	75
3.53 Matplotlib . . . . .	76
3.54 Gambar1 . . . . .	76
3.55 Gambar2 . . . . .	76

3.56 Gambar3 . . . . .	76
3.57 Gambar 4 . . . . .	76
3.58 Gambar 5 . . . . .	77
3.59 Gambar 6 . . . . .	77
3.60 Gambar 7 . . . . .	77
3.61 Gambar 8 . . . . .	77
3.62 Gambar 9 . . . . .	77
3.63 Gambar 10 . . . . .	77
3.64 Gambar 11 . . . . .	78
3.65 Gambar 12 . . . . .	78
3.66 Gambar 13 . . . . .	78
3.67 Gambar 15 . . . . .	78
3.68 Gambar 16 . . . . .	79
3.69 Gambar 17 . . . . .	79
3.70 Gambar 18 . . . . .	79
3.71 Gambar 20 . . . . .	79
3.72 Gambar 21 . . . . .	80
3.73 Gambar 23 . . . . .	80
3.74 SVM . . . . .	80
3.75 Decission Tree . . . . .	80
3.76 Cross Validation 1 . . . . .	81
3.77 Cross Validation 2 . . . . .	81
3.78 Cross Validation 3 . . . . .	81
3.79 Program Pengamatan Komponen Informasi 1 . . . . .	82
3.80 Program Pengamatan Komponen Informasi 2 . . . . .	82
3.81 Error . . . . .	82
4.1 Aip-Klasifikasi teks . . . . .	83
4.2 Aip-Klasifikasi bunga . . . . .	84
4.3 Aip-Teknik YouTube . . . . .	84
4.4 Aip-Bag of Word . . . . .	85
4.5 Aip-TF IDF . . . . .	85
4.6 Dataset Original Aip . . . . .	86
4.7 Dataset Dummy Aip . . . . .	87
4.8 Split DataFrame Aip . . . . .	87
4.9 Dataset Youtube05-Shakira Aip . . . . .	88

4.10	Dataset Youtube05-Shakira Aip . . . . .	88
4.11	Dataset Youtube05-Shakira SVM Aip . . . . .	88
4.12	Dataset Youtube05-Shakira Aip . . . . .	89
4.13	Confusion Matrix Aip . . . . .	90
4.14	Cross Validation Aip . . . . .	90
4.15	Hasil Cross Validation Aip . . . . .	91
4.16	Program Komponen Informas Aip . . . . .	91
4.17	Error Key Aip . . . . .	92
4.18	Error Key Aip . . . . .	92
4.19	Error Key Aip . . . . .	92
4.20	Error Key Aip . . . . .	93
4.21	Klasifikasi teks . . . . .	98
4.22	Klasifikasi bunga . . . . .	99
4.23	Teknik YouTube . . . . .	99
4.24	Bag of Word . . . . .	99
4.25	TF-IDF . . . . .	100
4.26	Data Dummy 500 Data . . . . .	100
4.27	Membagi 2 Dataframe . . . . .	100
4.28	Vektorisasi dan Klasifikasi Data . . . . .	100
4.29	Data Content . . . . .	101
4.30	DataFrame Kata-kata Pada Content . . . . .	101
4.31	Klasifikasi SVM Dari Data Vektorisasi . . . . .	101
4.32	Klasifikasi Decision Tree Dari Data Vektorisasi . . . . .	101
4.33	Plot Confusion Matrix Menggunakan Matplotlib . . . . .	101
4.34	Program Cross Validation Pada Data Vektorisasi . . . . .	102
4.35	Program Pengamatan Komponen Informasi . . . . .	102
4.36	Eror matplotlib.pyplot . . . . .	102
5.1	Ilustrasi Soal No.1 . . . . .	104
5.2	Ilustrasi Soal No. 2 . . . . .	104
5.3	Ilustrasi Soal No. 3 . . . . .	105
5.4	Ilustrasi Soal No. 4 . . . . .	105
5.5	Ilustrasi Soal No. 5 . . . . .	106
5.6	Ilustrasi Soal No. 6 . . . . .	107
5.7	Love . . . . .	107
5.8	Faith . . . . .	107

5.9 Fall . . . . .	108
5.10 Sick . . . . .	108
5.11 Clear . . . . .	108
5.12 Shine . . . . .	109
5.13 Bag . . . . .	109
5.14 Car . . . . .	109
5.15 Wash . . . . .	110
5.16 Motor . . . . .	110
5.17 Cycle . . . . .	110
5.18 Similariti Pada Kata Motor dan Cycle . . . . .	111
5.19 Similariti Pada Kata Wash dan Motor . . . . .	111
5.20 Extract_Words . . . . .	111
5.21 Permute Sentences . . . . .	112
5.22 Gensim TaggedDocument dan Doc2vec . . . . .	112
5.23 Aclimbdb . . . . .	113
5.24 Aclimbdb . . . . .	114
5.25 Aclimbdb . . . . .	114
5.26 Infer Code . . . . .	116
5.27 Cosine Similarity . . . . .	116
5.28 Cross Validation Metode 1 . . . . .	117
5.29 Cross Validation Metode 2 . . . . .	117
5.30 Cross Validation Metode 3 . . . . .	118
5.31 Vektorisasi Kata Aip . . . . .	118
5.32 Google Dataset Aip . . . . .	119
5.33 Vektorisasi Kata Aip . . . . .	120
5.34 Vektorisasi Dokumen Aip . . . . .	120
5.35 Mean Aip . . . . .	121
5.36 Standar Deviasi Aip . . . . .	121
5.37 Skip-Gram Aip . . . . .	121
5.38 Vektor Love Aip . . . . .	122
5.39 Vektor Faith Aip . . . . .	122
5.40 Vektor Fall Aip . . . . .	122
5.41 Vektor Sick Aip . . . . .	123
5.42 Vektor Clear Aip . . . . .	123
5.43 Vektor Shine Aip . . . . .	124
5.44 Vektor Bag Aip . . . . .	124

5.45 Vektor Car Aip . . . . .	125
5.46 Vektor Wash Aip . . . . .	125
5.47 Vektor cycle Aip . . . . .	125
5.48 Similariti Aip . . . . .	125
5.49 Extract Words Aip . . . . .	125
5.50 PermuteSentencesi Aip . . . . .	125
A.1 Form nilai bagian 1. . . . .	136
A.2 form nilai bagian 2. . . . .	137

# Chapter 1

## Mengenal 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 plagiatis[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)

didefinikasikan sebagai kecerdasan yang ditinjukkan 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 mengelompokan suatu data ke data yang sudah ada. Sedangkan unsupervised learning tidak memiliki data latih, sehingga dari data yang ada, kita mengelompokan 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 Pyhton 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/c89ebdc0d7dbba6e6fd222daabd257da3c28a96...
    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 1.21.3 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 variable yang berfungsi untuk mengambil data pada datasets dengan perintah .load\_iris).
4. digits datasets.load\_digits()(pada baris ketiga ini dimana digits merupakan suatu variable 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 variable digits.data) jjjjjj HEAD

## 1.6 Aip Suprapto 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 mengelompokan suatu data ke data yang sudah ada. Sedangkan unsupervised learning tidak memiliki data latih, sehingga dari data yang ada, kita mengelompokan 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 perinta 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.   0.0053...  0.   0.1) 0.]
 [ 0.   0.0053... 10.   0.   0.] ...
 [ 0.   0.0053... 16.   9.   0.] ...
 ...
 [ 0.   0.0053... 6.   0.   0.] ...
 [ 0.   0.0053... 12.   0.   0.] ...
 [ 0.   0.0053... 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).

```
>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:])

|||||| 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.

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

Figure 1.11: Hasil Pengujian Classifier

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

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 ”clf2.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 merekonstruksi 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 mengimplementasi

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

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)]
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 variian 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 Memperbarui Parameter

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

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

SVCsetparams () setelah estimator dibuat, dan diubah kembali ke rbf untuk

```

>>> 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 unutk 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: Eror 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 diats 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

<b>Filename</b>	<b>Size</b>	<b>Last Modified</b>	<b>MD5</b>
<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	f4d8b10e9a754884fb96e68e0e0b276a
<a href="#">Anaconda2-2018.12-Windows-x86.exe</a>	458.6M	2018-12-21 13:16:27	f123fd0ec8928bb7d55d1ca72c0d784
<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).

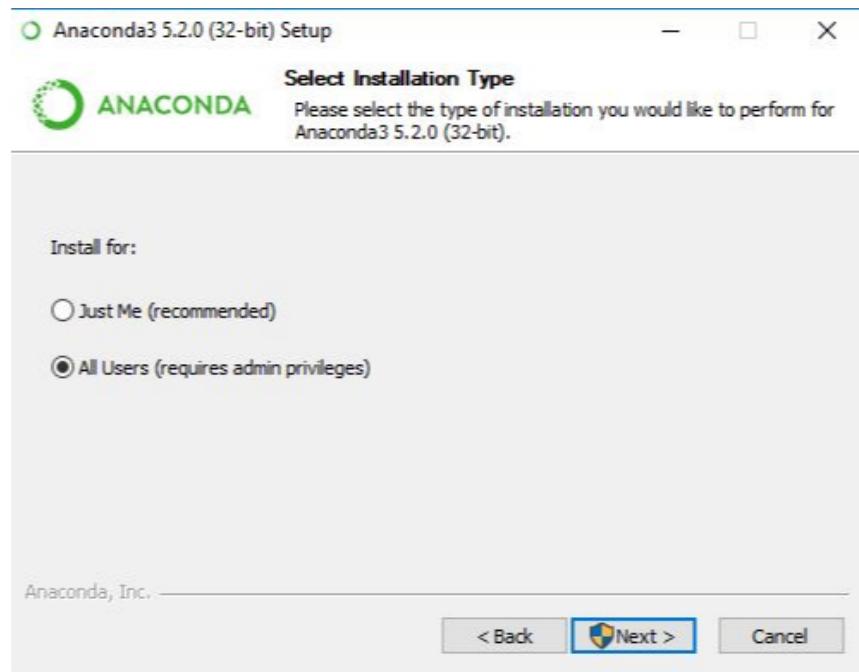


Figure 1.27: Langkah pertama instalasi anaconda.

- `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).

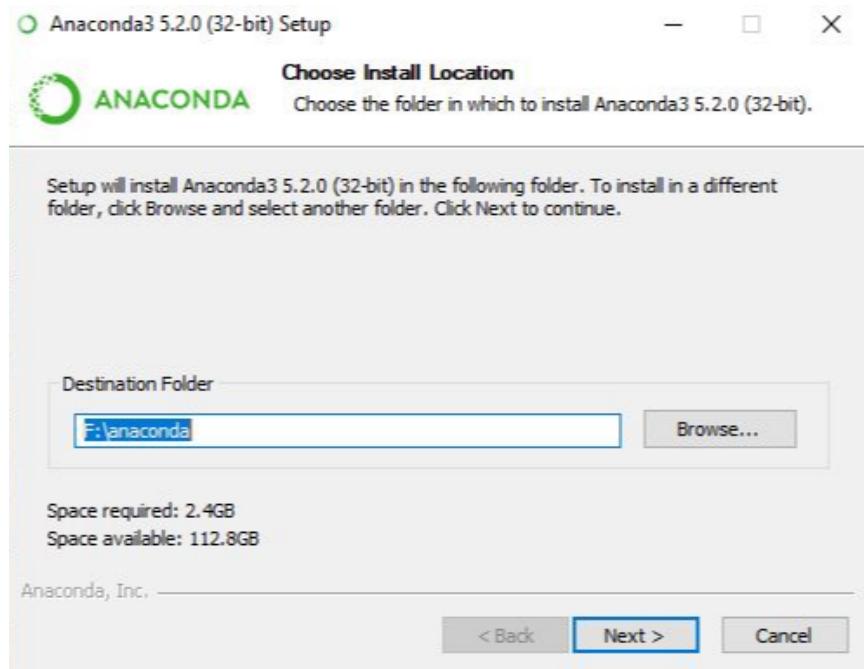


Figure 1.28: Langkah kedua instalasi anaconda.

- `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 atau 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).

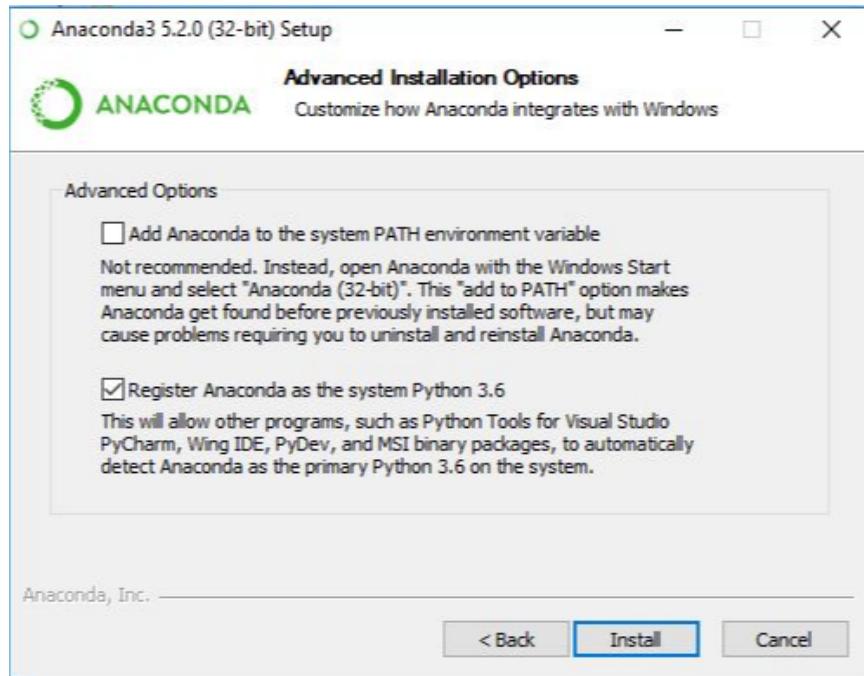


Figure 1.29: Langkah ketiga instalasi anaconda.

- `rng = np.random.RandomState(0)`  
(rng sebagai estimator/parameter dengan nilai suatu itemnya yaitu np.random.RandomState).
- `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`

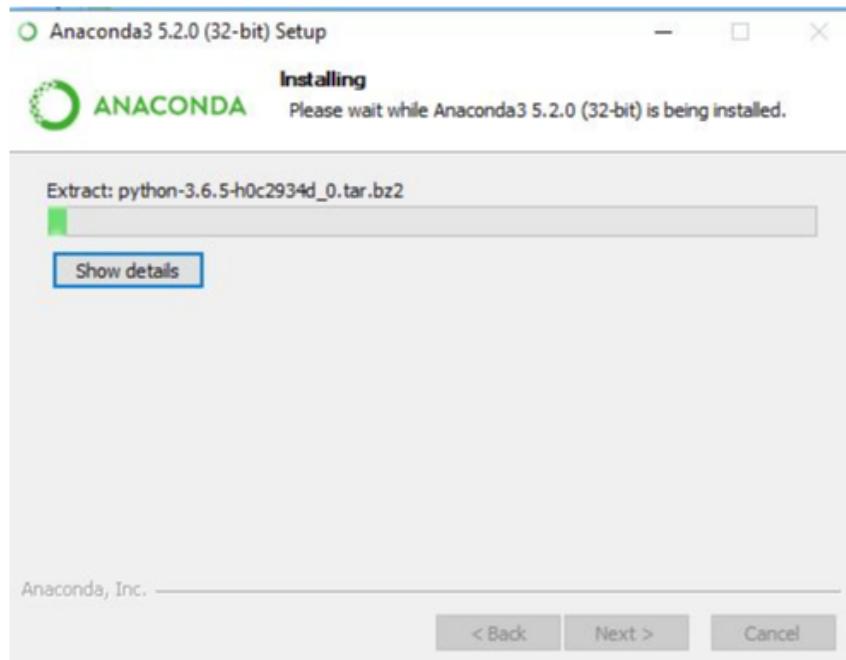


Figure 1.30: Langkah terakhir instalasi anaconda.

```
C:\Users\Asus>PyCpyp install -U scikit-learn
  Downloading https://files.pythonhosted.org/packages/ee/cb/c89ebcd07dbbaf6fd222daahd25/das3c28a967dd7c352d4272b2e1cef6/scikit_learn-0.20.2-cp36-cp36m-win32.whl (4.3MB)
    100% |████████████████████████████████| 4.3MB 600kB/s
Requirement not upgraded as not directly required: numpy>=1.8.2 in c:\programdata\anaconda\lib\site-packages (from scikit-learn) (1.14.3)
Requirement not upgraded as not directly required: scipy>=0.13.3 in c:\programdata\anaconda\lib\site-packages (from scikit-learn) (1.1.0)
distributed 2.2.0 requires msgpack, which is not installed.
  Installing collected packages: scikit-learn, distributed
    Preparing metadata (setup.py) ... done
    Requirement already satisfied: numpy >= 1.8.2 in c:\programdata\anaconda\lib\site-packages (from scikit-learn)
      Uninstalling scikit-learn-0.19.1:
        Successfully uninstalled scikit-learn-0.19.1
  Success: installed distributed-2.2.0-py36h05102b4_0
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 package sklearn).

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

## 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.RandomState(0))

```
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 error

#### 1.7.1.1 ScreenShoot Eror

#### 1.7.1.2 Tuliskan Kode Eror 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.

..... 78cdb6514db9716f252ef2024a7df6097aace611

# **Chapter 2**

## **Related Works**

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

### **2.1 Aip Suprapto 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 mengklarifikasi elemen-elemen dari himpunan yang diberikan kedalam dua kelompok berdasarkan aturan klarifikasi. Pada umumnya 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 Clustering**

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

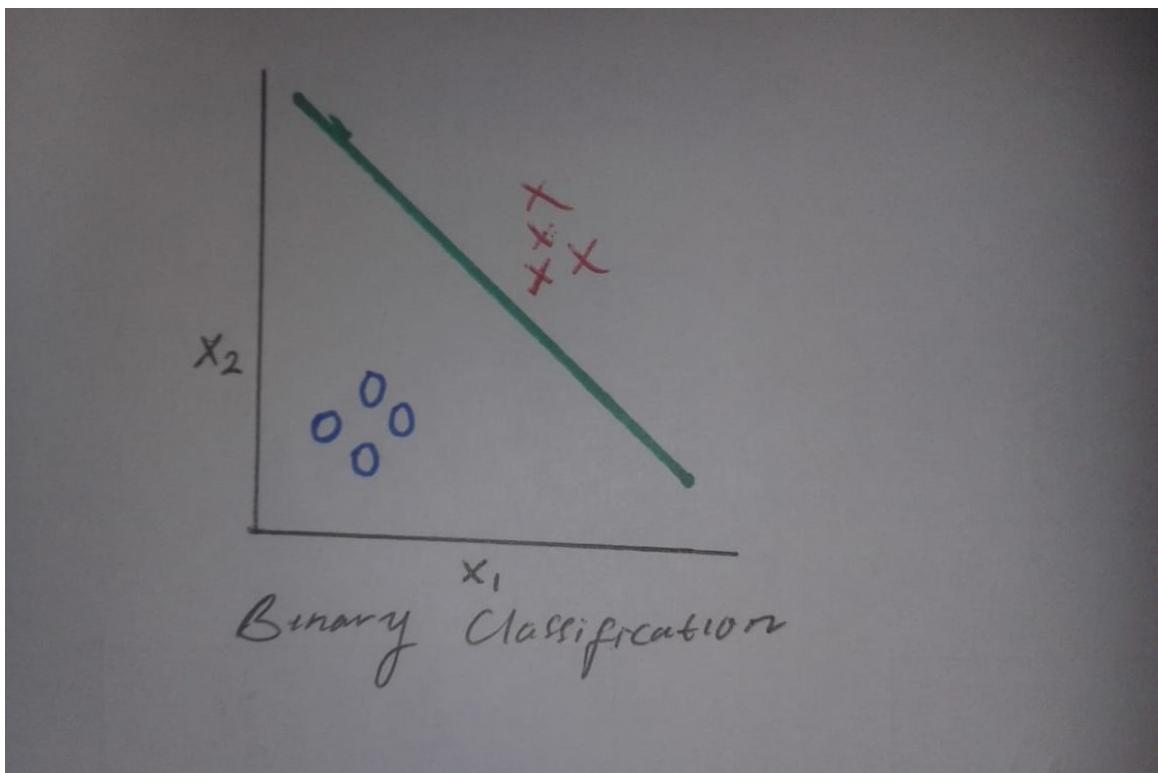


Figure 2.1: Binary Classification.

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 aggap 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 dikategorikan 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.

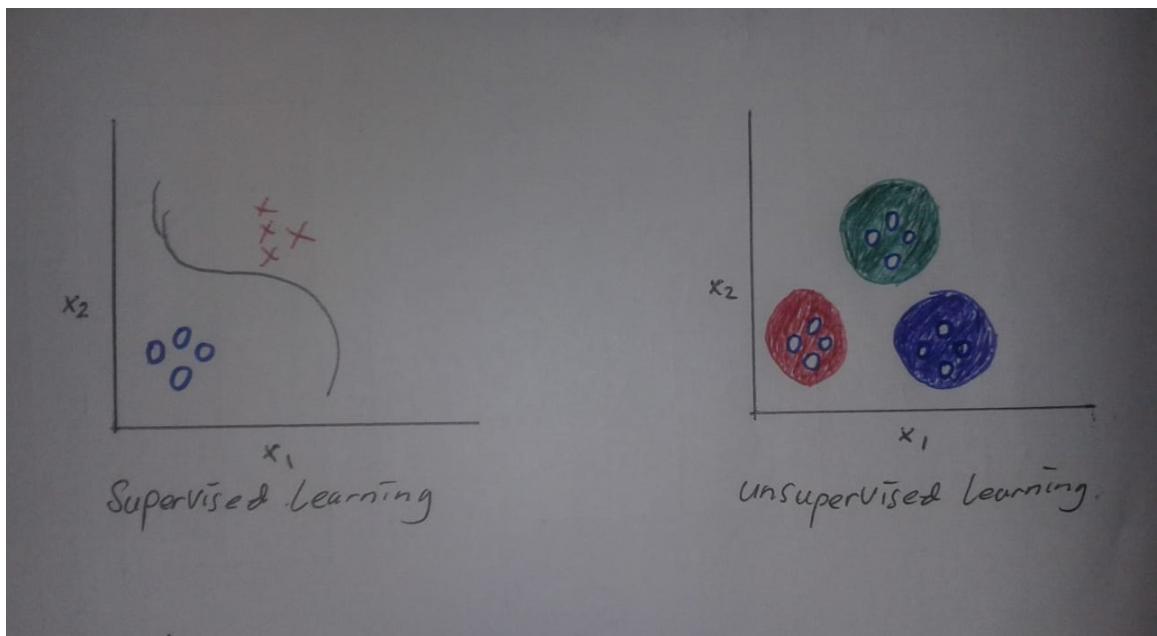


Figure 2.2: Supervised Learning.

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 persentase kasus yang telah diklasifikasikan dengan benar. Kita dapat melakukan analisis kesalahan yang telah dibuat oleh model. Dalam tabel tersebut baris true mangga dan true anggur menunjukkan kasus apakah itu objek mangga atau anggur. Kolom telah diprediksi dan dibuat oleh model. Ada 20 cow yang diprediksi benar dan ada 5 buffalo yang diprediksi 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

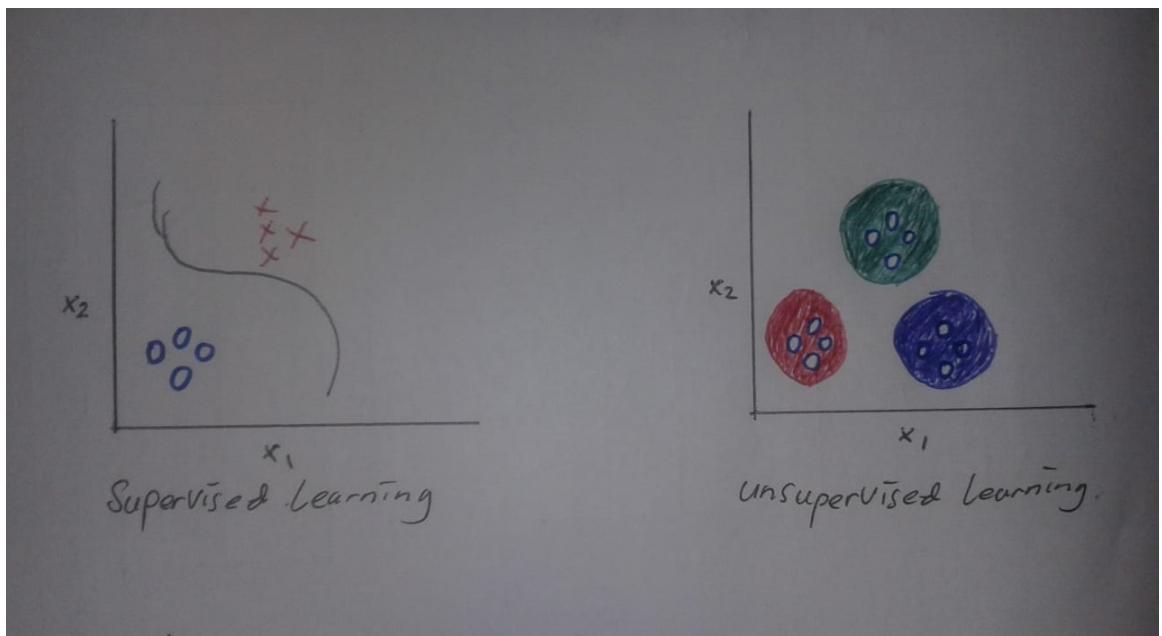


Figure 2.3: Unsupervised Learning.

- 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.

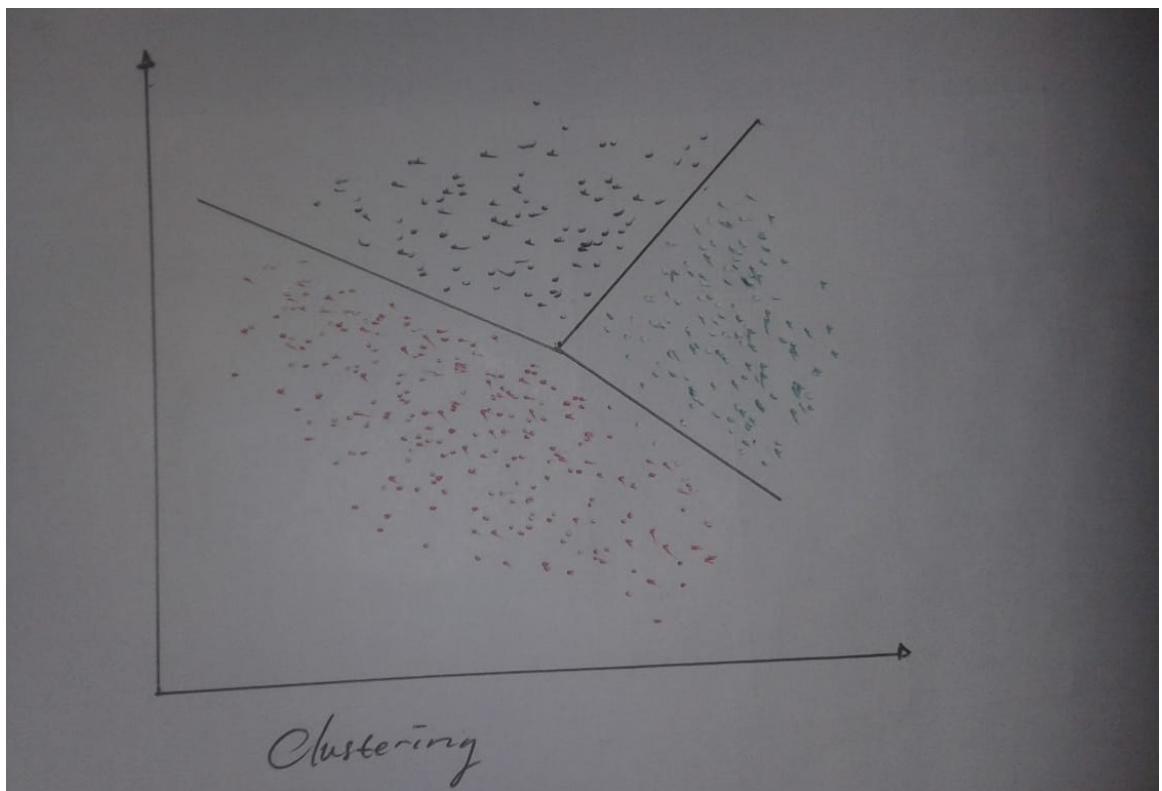


Figure 2.4: Clustering.

- 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.

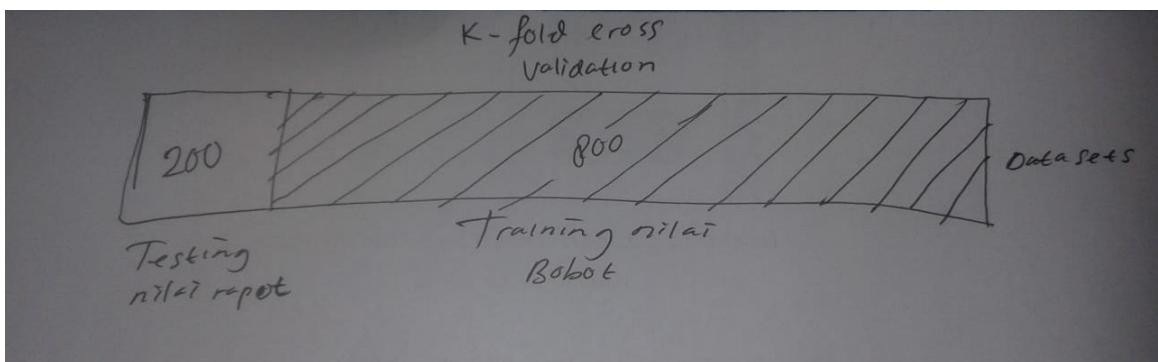


Figure 2.6: K-Fold Cross Validation.

### 2.1.8 Gain Dan Entropi

1. Gain adalah pengurangan yang diharapkan dalam entropy. Dalam machine 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 diartikan 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

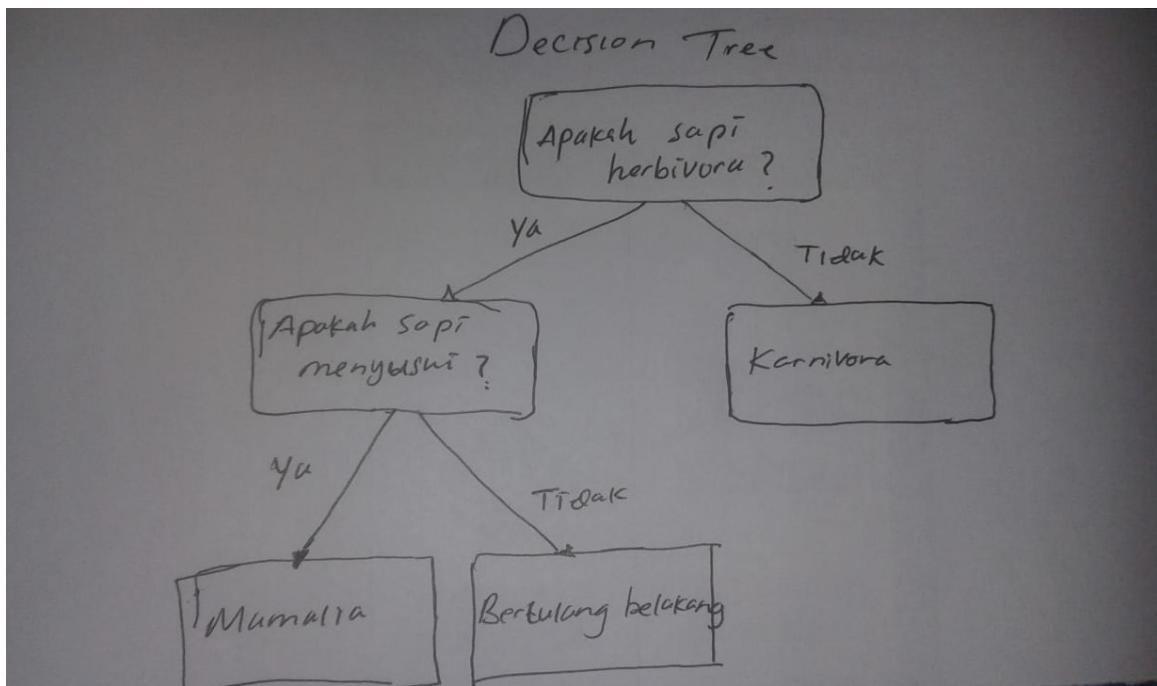


Figure 2.7: Decision Tree.

## 2.2 Aip Suprapto 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 :

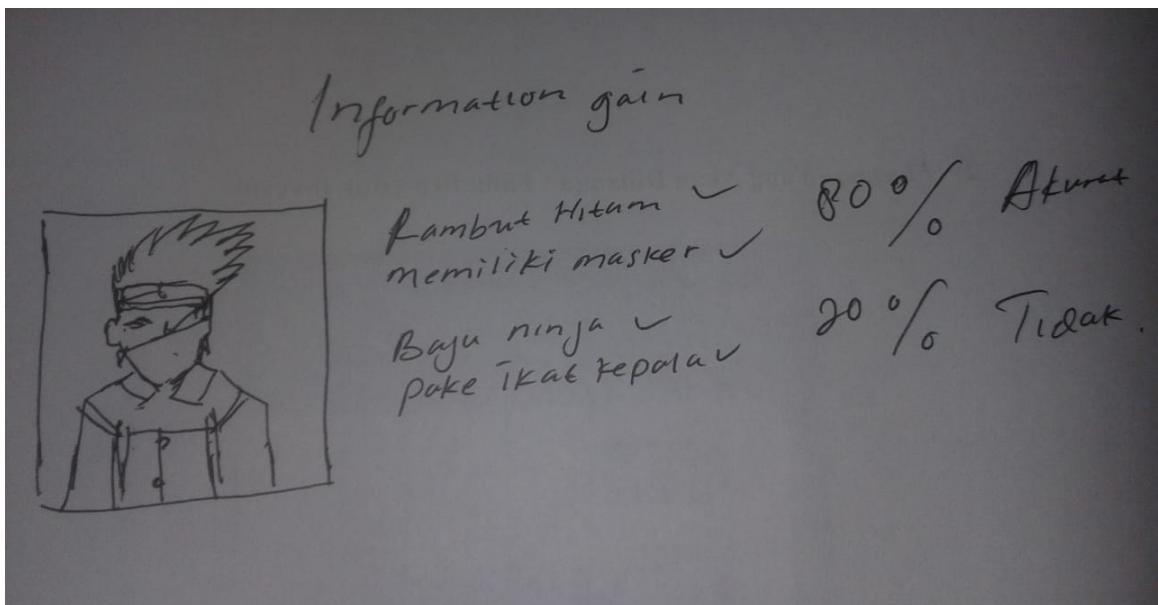


Figure 2.8: Gain Dan Entropi.

```
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 outputan 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, tejadi kesalah 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 teresbut mnedefinisikan 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 akurasinya.

– 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 menmpilkan 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 :

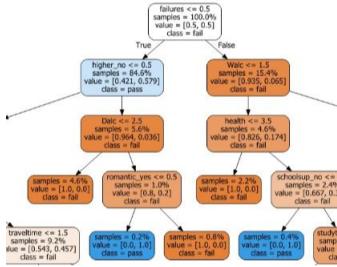


Figure 2.14: Gambar Keenam

```
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"], ...
...:
```

Figure 2.15: Gambar Ketujuh

## 2.2.2 Penanganan Error

1. Skrinsut 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 berasksi 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 akurasinya. 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 atributanya, 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,  0.06117089],
       [ 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.       ,  0.5593038 ,  0.07170369],
      ])

```

Figure 2.19: Gambar Kesebelas

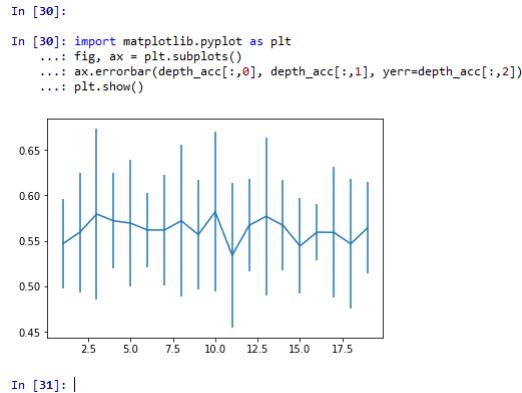


Figure 2.20: Gambar Keduabelas

- 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.

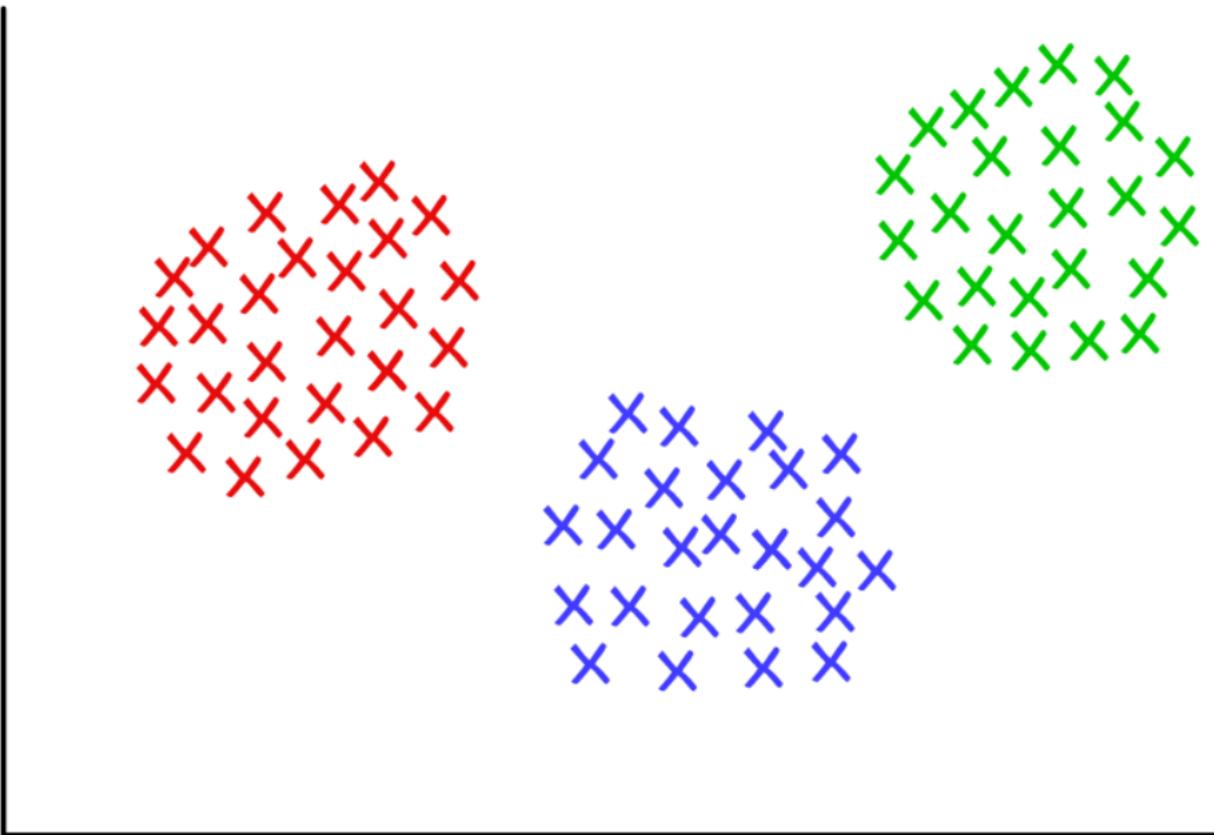


Figure 2.22: Binary Clasification

### 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.

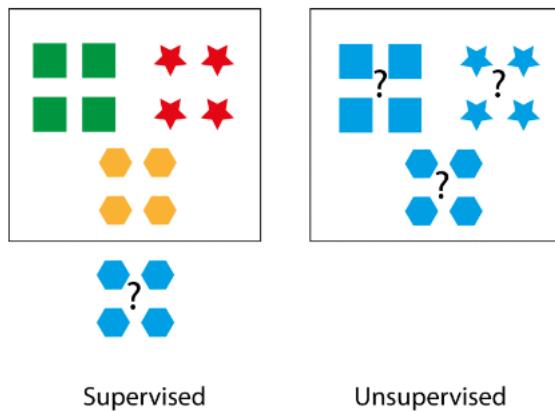


Figure 2.23: Supervised Learning

	<u>Predicted Game</u>	<u>Predicted Movie</u>
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

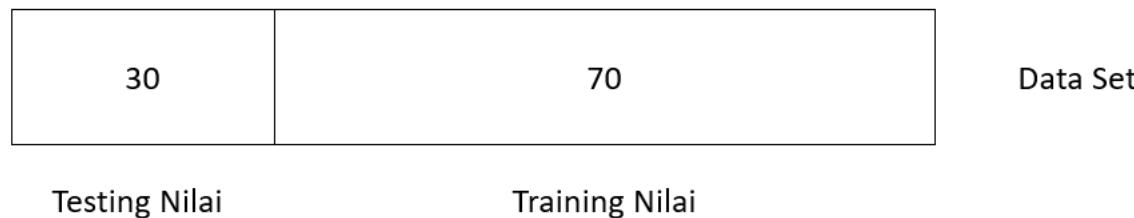


Figure 2.25: K-fold cross validation

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 decission tree.

- Hasil Code 5:

7. Code 6.

Penjelasan:

terjadi error pada import graphviz.

- Hasil Code 6:

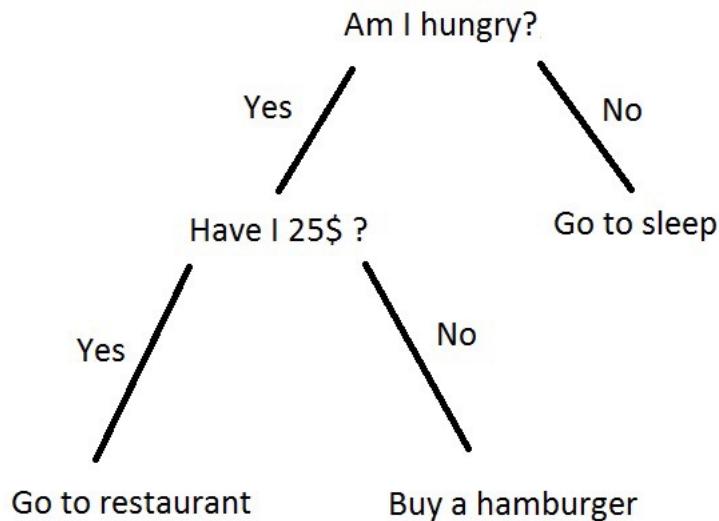


Figure 2.26: Decision Tree

### Information Gain

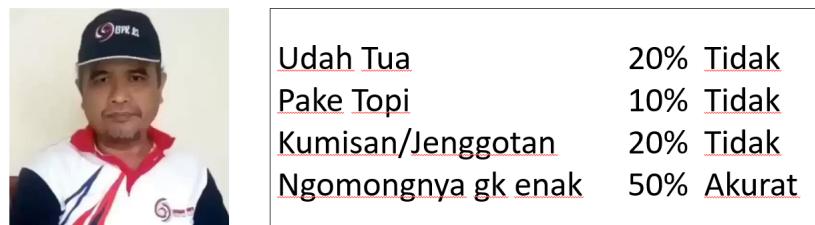


Figure 2.27: Entropi

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 mendefinisikan 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 akurasinya.

- Hasil Code 9:

11. Code 10.

Penjelasan:

berfungsi untuk menampilkan hasil dari fungsi Max Depth dan Accuracy dari dari Decission Tree. Yaitu menampilkan 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   16      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 ponpes
.... print("Passing: %d out of %d (%.2f%%)" % (ponpes.sum(Jatim_pass), len(Jatim_pass), 100*float(ponpes.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(Jatim_train_att, Jatim_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,
...: 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-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="csc")
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: 0.57 (+/- 0.01)
Max depth: 2, Accuracy: 0.57 (+/- 0.09)
Max depth: 3, Accuracy: 0.57 (+/- 0.07)
Max depth: 4, Accuracy: 0.57 (+/- 0.09)
Max depth: 5, Accuracy: 0.57 (+/- 0.11)
Max depth: 6, Accuracy: 0.57 (+/- 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: 0.56 (+/- 0.13)
Max depth: 16, Accuracy: 0.56 (+/- 0.13)
Max depth: 17, Accuracy: 0.57 (+/- 0.07)
Max depth: 18, Accuracy: 0.55 (+/- 0.08)
Max depth: 19, Accuracy: 0.55 (+/- 0.08)
```

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
...:
...: depth_acc
Out[38]:
array([[1.00000000e+00, 5.79751704e-01, 6.30768599e-03],
       [2.00000000e+00, 5.87540571e-01, 5.55999992e-02],
       [3.00000000e+00, 5.72188413e-01, 6.54746741e-02],
       [4.00000000e+00, 5.79718436e-01, 4.96526886e-02],
       [5.00000000e+00, 5.99717624e-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()
```

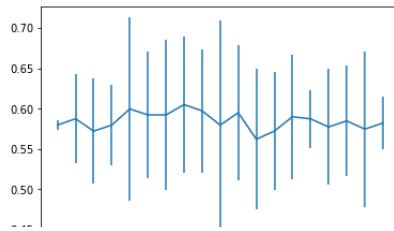


Figure 2.39: Hasil Code 12

---

```
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, algoritm or equation of method 1 and how to apply into your data

### **3.3 Method 2**

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

### **3.4 Aip Suprapto 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 :

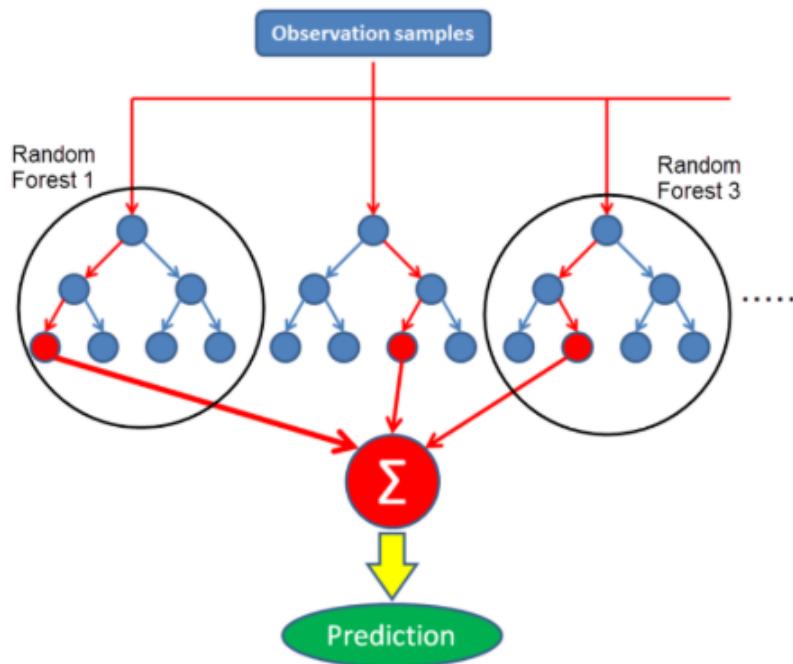


Figure 3.1: Random Forest

## 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 praktik. 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	48000	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 decission tree adalah presentasi hasil dari perhitungan dataset. Dari set tentang burung pipit. Confusion matrix memberi tau hal-hal yang diharapkan, artinya, butrung-burung yang terlihat mirip saling bingung satu sama lain.

- SVM ( 29% )

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

Di sini, akurasinya 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 outputannya.

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.

Definisi	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	Kurang	Tinggi	Baik
P9	Kurang	Sedang	Buruk
P10	Kurang	Rendah	Baik
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 :

- Memanggil library.
- Membaca dari file pandas.
- Menampilkan hasil

Sehingga menghasilkan :

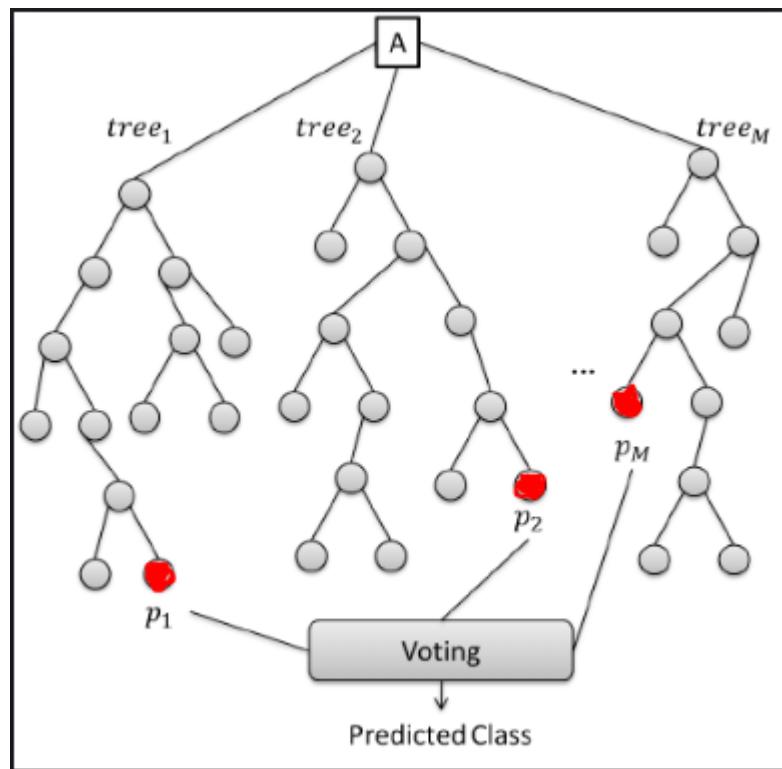


Figure 3.8: Voting Random forest

```

1 import pandas as pd
2 ufo = pd.read_csv('http://bit.ly/uforeports')
3 type(ufo)
4 ufo.head()
5

```

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 = np.arange(1, 12, 2)
...: 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 selanjuknya 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.

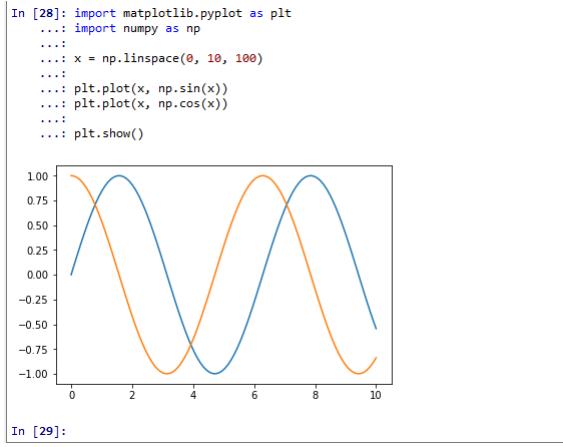


Figure 3.14: Hasil Matplotlib

```

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
```

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. Skrinsut 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 direktoriya 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 dibuat 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 BUAATAN/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 terditeksi benar, True Negatif yang merupakan data negatif akan tetapi terditeksi benar, False Positif merupakan data negatif namun terditeksi sebagai data positif, False Negatif merupakan data positif namun terditeksi 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]:
   label
imgid
1      1
2      1
3      1
4      1
5      1
```

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 dimana 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.

#### 3.4.4 Praktek

1. Aplikasi Sederhana Pandas dan Penjelasan Code

- Pandas:
  - Penjelasan Baris 1 : import pandas as Meong merupakan import dari library pandas dari python menggunakan inisial Meong.
  - Penjelasan Baris 2 : variable a sebagai pendefinisian data yang dibuat pada baris pertama yaitu Meong, sedangkan Series merupakan sebuah array satu dimensi yang memiliki label dan digunakan untuk menyimpan data yang beragam seperti integer,string,float, dan lain sebagainya. angka pada Series yang saya gunakan yaitu ([10,20,30,40,np.nan,50]) untuk menampilkan secara default, pandas secara otomatis memberi index pada setiap baris dari series.

```
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

- Penjelasan Baris 3 : fungsi print(am) untuk mencetak atau menampilkan objek

- Hasil:

- Aplikasi Sederhana Numpy dan Penjelasan Code

- Code Numpy:

- \* Penjelasan Baris 1 : import numpy as np. untuk import library numpy dari python menggunakan inisial Kucing.
    - \* Penjelasan Baris 2 : matrix one np.eye3. matrix dapat dikatakan sebagai list dua dimensi dimana suatu list berisi list lagi. sedangkan eye 3 untuk menghasilkan matriks identitas dengan dimensi yang ditetapkan.
    - \* Penjelasan Baris 3 : matrix one. matrix pada baris ketiga ini untuk menampilkan objek numpy.

- Hasil:

- Aplikasi Sederhana Matplotlib dan Penjelasan Code

- \* Code Matplotlib:

- Penjelasan Baris 1 : import matplotlib.pyplot as plt merupakan modul python untuk menggambarkan plot 2D.
    - Penjelasan Baris 2 : plt.plot([1,2,3,4]) untuk menentukan angka-angka pada gambar plot 2D tersebut
    - Penjelasan Baris 3 : untuk menampilkan matplotlib berupa gambar plot 2D

- \* Hasil:

- Program Klasifikasi Random Forest dan Penjelasan :

```
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   1   0   0
7610    0   0   0   0   0   0   0   ...    0   0   0   0   0   0   1
2327    0   0   0   0   0   0   0   ...    1   0   0   0   0   0   1
2462    0   0   0   0   0   0   0   ...    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

- \* Code Random Forest 1 :
  - Penjelasan Codingan di atas menghasilkan variabel baru yaitu imgatt. Terdapat 3 kolom dan 3677856 baris data.
- \* Code Random Forest 2 :
  - Penjelasan Codingan di atas berfungsi untuk melihat sebagian data awal dari dataset. Hasilnya terdapat pada gambar di atas setelah di eksekusi.
- \* Code Random Forest 3 :
  - Penjelasan Codingan di atas merupakan tampilan untuk menampilkan hasil dari dataset yang telah di run atau di eksekusi. Dimana pada gambar di atas 3677856 merupakan baris dan 3 adalah kolom.
- \* Code Random Forest 4 :
  - Penjelasan Pada gambar di atas menampilkan hasil dari variabel imgatt2. Dimana index nya 'imgid', kolom berisi 'attid' dan values atau nilainya berisi 'present'.
- \* Code Random Forest 5 :
  - Penjelasan Pada gambar di atas menampilkan hasil dari variabel imgatt2.head. Dimana dataset nya ada 5 baris dan 312 kolom.
- \* Code Random Forest 6 :

```

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

- Penjelasan Pada gambar di atas menampilkan jumlah dari baris dan kolom dari variabel imgatt2. Dimana 11788 adalah baris dan 312 adalah kolom.
- \* Code Random Forest 7 :
  - Penjelasan Pada gambar di atas menunjukkan load dari jawabannya yang berisi ” apakah burung tersebut ( subjek pada dataset ) termasuk dalam spesies yang mana ?. Kolom yang digunakan adalah imgid dan label, kemudian melakukan pivot yang mana imgid menjadi index yang artinya unik sehubungan dengan dataset yang telah dieksekusi.
- \* Code Random Forest 8 :
  - Penjelasan Pada gambar di atas menunjukkan hasil dari variabel imglabels. Dimana menampilkan dataset dari imgid dan label. Dan dapat dilihat hasilnya dari gambar di atas.
- \* Code Random Forest 9 :
  - Penjelasan Pada gambar di atas menunjukkan jumlah baris dan kolom dari variabel imglabels. Dimana hasil dari kodingan tersebut dapat dilihat setelah di run.
- \* Code Random Forest 10 :
  - Penjelasan Pada gambar diatas dikarenakan isinya sama, maka bisa melakukan join antara dua data yang diesekusi ( yaitu ada imgatt2 dan imglabels ), sehingga pada hasilnya akan didapatkan data ciri dan data jawaban atau labelnya sehingga bisa dikategorikan/dikelompokkan sebagai supervised learning.

```
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

Jadi perintah untuk menggabungkan kedua data, kemudian dilakukan pemisahan antara data set untuk training dan test pada dataset yang dieksekusi.

- \* Code Random Forest 11 :
  - Penjelasan Pada gambar di atas menghasilkan pemisahan dan pemilihan tabel ( memisahkan dan memilih tabel ).
- \* Code Random Forest 12 :
  - Penjelasan Pada gambar di atas menunjukkan hasil dari variabel dtathead. Dimana data nya dapat dilihat pada gambar diatas. Dan dataset nya terdiri dari 5 baris dan 312 kolom.
- \* Code Random Forest 13 :
  - Penjelasan Pada gambar di atas menunjukkan hasil dari variabel dflabel.head. Dimana berisikan data dari imgid dan label. Dan hasilnya dapat dilihat pada gambar di atas.
- \* Code Random Forest 14 :
  - Penjelasan Pada gambar di atas merupakan pembagian dari data training dan dataset
- \* Code Random Forest 15 :
  - Penjelasan Pada gambar di atas merupakan pemanggilan kelas RandomForestClassifier. max features yang diartikan berapa banyak kolom pada setiap tree.
- \* Code Random Forest 16 :
  - Penjelasan Pada gambar di atas merupakan perintah untuk melakukan fit untuk membangun random forest yang sudah ditentukan dengan maksimum fitur sebanyak 50.

```
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

- \* Code Random Forest 17 :
  - Penjelasan Pada gambar di atas menunjukkan hasil dari cetakan variabel dftrainatt.head.
- \* Code Random Forest 18 :
  - Penjelasan Pada gambar di atas merupakan hasil dari variabel dftestatt da dftsetlabel. Dimana hasilnya dapat dilihat dari pada gambar di atas
- Program Aplikasi Confusion Matrix dan Penjelasan Keluarannya :
  - \* Code Confusion Matrix 1 :
    - Penjelasan Pada gambar di atas merupakan kodingan untuk import confusiion matrik dari random forest. untuk hasilnya dapat dilihat dari gambar.
  - \* Code Confusion Matrix 2 :
    - Penjelasan Pada gambar di atas merupakan tampilan dari variabel cm.
  - \* Code Confusion Matrix 3 :
    - Penjelasan Pada gambar di atas merupakan perintah untuk plot. Dan untuk hasilnya terpadat pada gambar di atas.
  - \* Code Confusion Matrix 5 :
    - Penjelasan Pada gambar di atas merupakan perintah plot dari gambar sebelumnya.
- Program Klasifikasi SVM dan Decision Tree Beserta Penjelasan Keluarannya :
  - \* Code SVM :
    - Penjelasan Pada gambar di atas cara untuk mencoba klasifikasi dengan SVM dengan dataset yang sama.

```
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

\* Code Decision Tree :

- Penjelasan : Pada gambar di atas merupakan cara untuk mencoba klasi kasi dengan decission tree dengan dataset yang sama.

– Program Cross Validation dan Penjelasan Keluarannya :

\* Code Cross Validation 1 :

- Penjelasan : Pada gambar di atas merupakan Hasil dari cross validation random forest.

\* Code Cross Validation 2 :

- Penjelasan : Pada gambar di atas merupakan hasil dari cross validation Decission tree.

\* Code Cross Validation 3 :

- Penjelasan : Pada gambar di atas merupakan hasil dari cross validation SVM.

– Program Pengamatan Komponen Informasi dan Penjelasan Keluarannya :

```
In [22]: birds = pd.read_csv("D:/KULIAH/SEMESTER 6/KECERDASAN BUATAN/CUB_200_2011/
classes.txt",
...: sep='\t', header=None, usecols=[1], names=['birdname'])
...: birds = birds[['birdname']]
...: birds
Out[22]:
0          001.Black_footed_Albatross
1          002.Laysan_Albatross
2          003.Sooty_Albatross
3          004.Groove_billed_Ani
4          005.Crested_Auklet
5          006.Least_Auklet
6          007.Parakeet_Auklet
7          008.Rhinoceros_Auklet
8          009.Brewer_Blackbird
9          010.Red_winged_Blackbird
10         011.Rusty_Blackbird
11         012.Yellow_headed_Blackbird
12         013.Bobolink
13         014.Indigo_Bunting
14         015.Lazuli_Bunting
15         016.Painted_Bunting
16         017.Cardinal
17         018.Spotted_Catbird
18         019.Grey_Catbird
19         020.Yellow_breasted_Chat
20         021.Eastern_Towhee
21         022.Chuckwachay
```

Figure 3.36: Plotting nama data

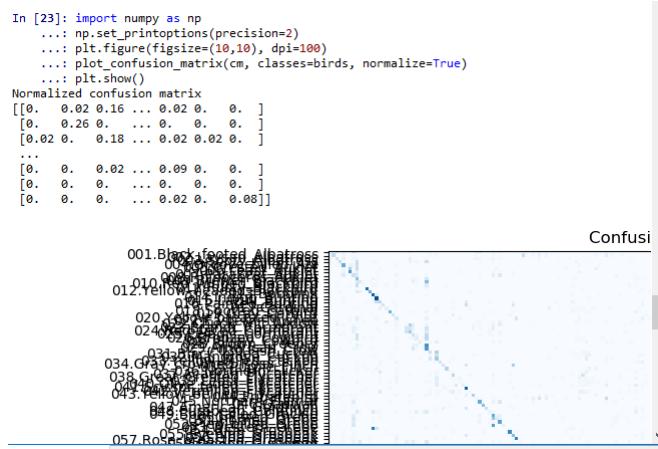


Figure 3.37: Melakukan perintah plot

- \* Code Pengamatan Komponen Informasi 1 :
  - Penjelasan : Pada gambar di atas menunjukkan cara untuk mengetahui berapa banyak tree yang dibuat, berapa banyak atribut yang dipakai dan informasi lainnya menggunakan kode.
- \* Code Pengamatan Komponen Informasi 2 :
  - Penjelasan : Pada gambar di atas merupakan cara untuk melakukan plot informasi ini dengan kode di atas.

## 2. Penanganan Error

- Skrinsut Error

- Kode Error: file b'data/CUB 200 2011/attributes/image attributes labels.txt'

```
In [24]: from sklearn import tree
...: clftree = tree.DecisionTreeClassifier()
...: clftree.fit(df_train_att, df_train_label)
...: clftree.score(df_test_att, df_test_label)
Out[24]: 0.11807293528780836
In [25]: |
```

Figure 3.38: Klasifikasi menggunakan decision tree

```
In [25]: from sklearn import svm
...: clfsvm = svm.SVC()
...: clfsvm.fit(df_train_att, df_train_label)
...: clfsvm.score(df_test_att, df_test_label)
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)
Out[25]: 0.005809796210225242
In [26]: |
```

Figure 3.39: Klasifikasi menggunakan SVM

- Solusi Pemecahan Error : Hapus Direktori data pada kode pastikan satu folder.

```
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: %0.2f (+/- %0.2f)" % (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]: scoretree = cross_val_score(clftree, df_train_att, df_train_label, cv=5)
...: print("Accuracy: %0.2f (+/- %0.2f)" % (scoretree.mean(), scoretree.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: %0.2f (+/- %0.2f)" % (accuracy.mean(), accuracy.std() * 2))
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)
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: %0.2f (+/- %0.2f)" % (accuracy.mean(), accuracy.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)
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

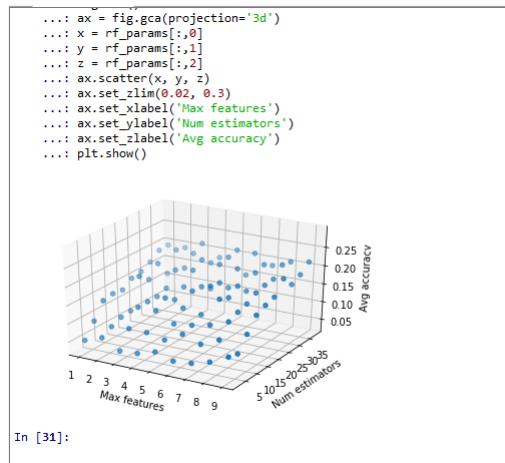


Figure 3.44: Plot informasi

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

Figure 3.45: Skrinsut 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

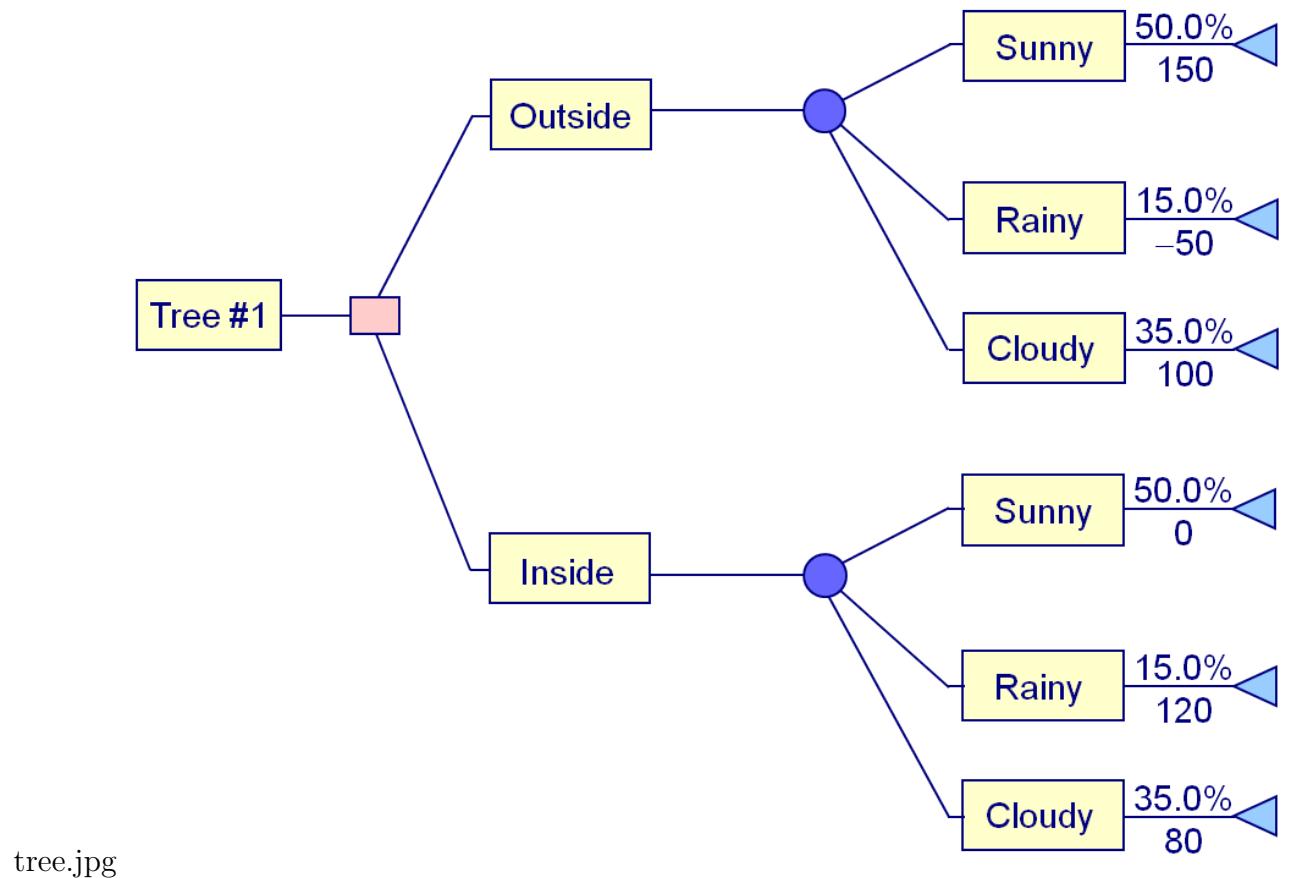


Figure 3.48: Random Forest.

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

Figure 3.49: Tabel Confusion Matriks

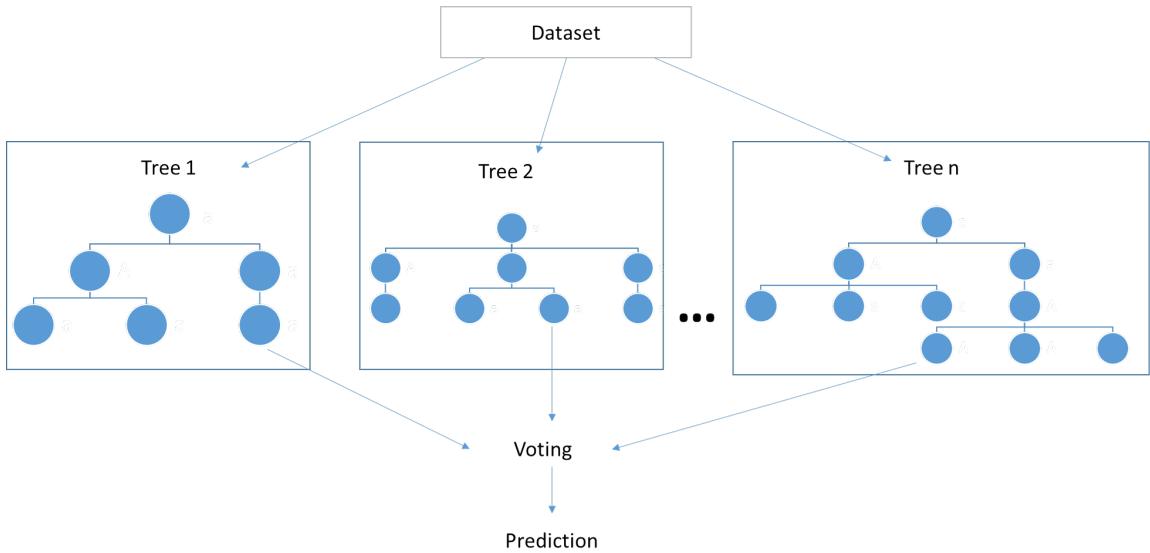


Figure 3.50: Voting

```

In [33]: import pandas as Meong
...: am = Meong.Series([10,20,30,40,np.nan,50])
...: print (am)
0    10.0
1    20.0
2    30.0
3    40.0
4    NaN
5    50.0
dtype: float64

```

---

```

In [34]:

```

Figure 3.51: Pandas

```

In [34]: import numpy as Kucing
...: matrix_one = np.eye(3)
...: matrix_one
Out[34]:
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])

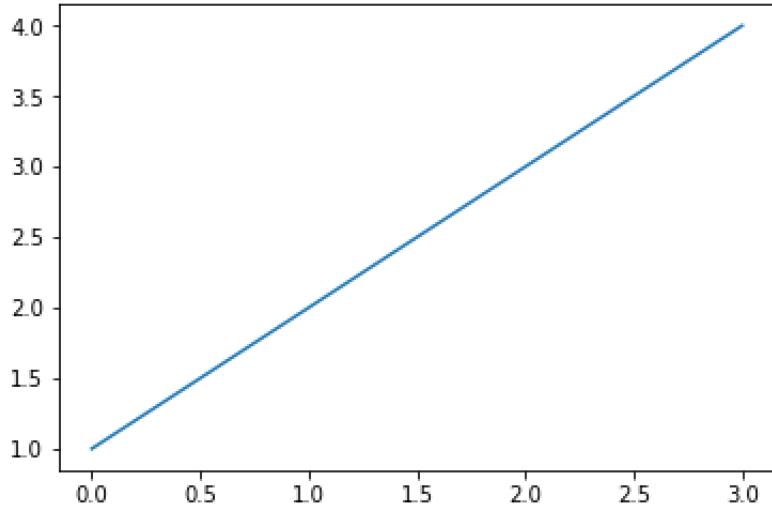
```

```

In [35]:

```

Figure 3.52: Numpy



In [36]:

Figure 3.53: Matplotlib

```
In [45]: import pandas as pd
...
...: # some Lines have too many fields (?), so skip bad Lines
...: imgatt = pd.read_csv("D:/Tugas Kuliah/Semester 6/KECERDASAN BUATAN/git/
hust/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'])
```

Figure 3.54: Gambar1

```
In [46]: imgatt.head()
Out[46]:
      imgid  attid  present
0        1      1        0
1        1      2        0
2        1      3        0
3        1      4        0
4        1      5        1
```

Figure 3.55: Gambar2

```
In [47]: imgatt.shape
Out[47]: (3677856, 3)
```

In [48]:

Figure 3.56: Gambar3

```
In [48]: imgatt2 = imgatt.pivot(index='imgid', columns='attid',
values='present')
```

In [49]:

Figure 3.57: Gambar 4

```
In [49]: imgatt2.head()
Out[49]:
attid  1   2   3   4   5   6   7   ...   306  307  308  309  310  311  312
imgid
1      0   0   0   0   1   0   0 ...
2      0   0   0   0   0   0   0 ...
3      0   0   0   0   1   0   0 ...
4      0   0   0   0   1   0   0 ...
5      0   0   0   0   1   0   0 ...
[5 rows x 312 columns]
```

Figure 3.58: Gambar 5

```
In [50]: imgatt2.shape
Out[50]: (11788, 312)
```

---

In [51]:

Figure 3.59: Gambar 6

```
In [51]: imglabels = pd.read_csv("D:/Tugas Kuliah/Semester 6/KECERDASAN BUATAN/
git/hust/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 [52]:

Figure 3.60: Gambar 7

```
In [52]: imglabels.head()
Out[52]: label
1      1
2      1
3      1
4      1
5      1
In [53]:
```

Figure 3.61: Gambar 8

```
In [53]: imglabels.shape
Out[53]: (11788, 1)
```

---

In [54]:

Figure 3.62: Gambar 9

```
In [54]: df = imgatt2.join(imglabels)
...: df = df.sample(frac=1)
```

---

In [55]:

Figure 3.63: Gambar 10

```
In [55]: df_att = df.iloc[:, :312]
...: df_label = df.iloc[:, 312:]
```

```
In [56]:
```

Figure 3.64: Gambar 11

```
In [56]: df_att.head()
Out[56]:
   1    2    3    4    5    6    7    ...    306   307   308   309   310   311   312
imgid
7270    0    0    0    0    0    0    0    ...
10635   0    0    0    0    0    0    1    ...
2752    0    1    0    0    0    0    0    ...
2416    0    0    0    0    0    0    0    ...
8718    0    0    0    0    0    0    1    ...
[5 rows x 312 columns]
```

Figure 3.65: Gambar 12

```
In [57]: df_label.head()
Out[57]:
      label
imgid
7270    125
10635   181
2752    48
2416    43
8718    149
```

```
In [58]:
```

Figure 3.66: Gambar 13

```
In [58]: df_train_att = df_att[:8000]
...: df_train_label = df_label[:8000]
...: df_test_att = df_att[8000:]
...: df_test_label = df_label[8000:]
...:
...: df_train_label = df_train_label['label']
...: df_test_label = df_test_label['label']
```

```
In [59]:
```

```
In [60]: from sklearn.ensemble import RandomForestClassifier
...: clf = RandomForestClassifier(max_features=50, random_state=0,
n_estimators=100)
```

Figure 3.67: Gambar 15

```

Out[16]:
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                      max_depth=None, max_features=50, 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 [16]: clf.fit(df_train_att, df_train_label)Out[17]:
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                      max_depth=None, max_features=50, 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 3.68: Gambar 16

```

In [17]:
In [18]: print(clf.predict(df_train_att.head()))
[183 57 139 108 28]
In [19]:

```

Figure 3.69: Gambar 17

```

In [19]: clf.score(df_test_att, df_test_label)
Out[19]: 0.44720168954593453

```

**In [20]:**

Figure 3.70: Gambar 18

```

In [20]: from sklearn.metrics import confusion_matrix
...: pred_labels = clf.predict(df_test_att)
...: cm = confusion_matrix(df_test_label, pred_labels)

```

**In [21]:**

```

In [21]: cm
Out[21]:
array([[ 9,  1,  0, ...,  0,  0,  0],
       [ 0, 15,  0, ...,  0,  0,  0],
       [ 3,  0,  7, ...,  0,  0,  0],
       ...,
       [ 0,  0,  0, ...,  2,  0,  0],
       [ 0,  0,  0, ...,  0,  5,  0],
       [ 0,  0,  0, ...,  0,  0, 12]], dtype=int64)

```

**In [22]:**

Figure 3.71: Gambar 20

```

In [22]: 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)

```

Figure 3.72: Gambar 21

```

In [27]: import numpy as np
.... np.set_printoptions(precision=2)
.... plt.figure(figsize=(10,10), dpi=100)
.... plot_confusion_matrix(cm, classes=birds, normalize=True)
.... plt.show()
Normalized confusion matrix
[[0.6  0.07 0.   ... 0.   0.   0.   ]
 [0.   0.58 0.   ... 0.   0.   0.   ]
 [0.18 0.   0.41 ... 0.   0.   0.   ]
 ...
 [0.   0.   0.   ... 0.1  0.   0.   ]
 [0.   0.   0.   ... 0.   0.29 0.   ]
 [0.   0.   0.   ... 0.   0.   0.8  ]]

```

Figure 3.73: Gambar 23

```

In [28]: from sklearn import tree
.... clftree = tree.DecisionTreeClassifier()
.... clftree.fit(df_train_att, df_train_label)
.... clftree.score(df_test_att, df_test_label)
Out[28]: 0.2663674762407603

In [29]:

```

Figure 3.74: SVM

```

In [27]: from sklearn import svm
.... clfsvm = svm.SVC()
.... clfsvm.fit(df_train_att, df_train_label)
.... clfsvm.score(df_test_att, df_test_label)
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)
Out[27]: 0.27613516367476243

In [28]: |

```

Figure 3.75: Decission Tree

```
In [28]: 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: %0.2f (+/- %0.2f)" % (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: 0.25 (+/- 0.12)
```

---

In [29]:

Figure 3.76: Cross Validation 1

```
In [29]: scorestree = cross_val_score(clftree, df_train_att, df_train_label, cv=5)
...: print("Accuracy: %0.2f (+/- %0.2f)" % (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: 0.13 (+/- 0.06)
```

---

In [30]:

Figure 3.77: Cross Validation 2

```
In [30]: scoressvm = cross_val_score(clfsvm, df_train_att, df_train_label, cv=5)
...: print("Accuracy: %0.2f (+/- %0.2f)" % (scoressvm.mean(), scoressvm.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)
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
```

Figure 3.78: Cross Validation 3

```

In [31]: max_features_opts = range(1, 10, 1)
....: n_estimators_opts = range(2, 40, 4)
....: rf_params = np.empty((len(max_features_opts)*len(n_estimators_opts),4), float)
....: i = 0
....: for max_features in max_features_opts:
....:     for n_estimators in n_estimators_opts:
....:         clf = RandomForestClassifier(max_features=max_features,
n_estimators=n_estimators)
....:         scores = cross_val_score(clf, df_train_att, df_train_label, cv=5)
....:         rf_params[i,0] = max_features
....:         rf_params[i,1] = n_estimators
....:         rf_params[i,2] = scores.mean()
....:         rf_params[i,3] = scores.std() * 2
....:         i += 1
....: print("Max features: %d, num estimators: %d, accuracy: %0.2f (+/- %0.2f)" % (max_features, n_estimators, 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)
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

```

Figure 3.79: Program Pengamatan Komponen Informasi 1

```

In [32]: import matplotlib.pyplot as plt
....: from mpl_toolkits.mplot3d import Axes3D
....: from matplotlib import cm
....: fig = plt.figure()
....: fig.clf()
....: ax = fig.gca(projection='3d')
....: x = rf_params[:,0]
....: y = rf_params[:,1]
....: z = rf_params[:,2]
....: ax.scatter(x, y, z)
....: ax.set_zlim(0.02, 0.3)
....: ax.set_xlabel('Max features')
....: ax.set_ylabel('Num estimators')
....: ax.set_zlabel('Avg accuracy')
....: plt.show()

```



Figure 3.80: Program Pengamatan Komponen Informasi 2

```

In [59]: from sklearn.ensemble import RandomForestClassifier
....: clf = RandomForestClassifier(max_features=50, random_state=0,
n_estimators=100)

```

Figure 3.81: Error

# 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

### 4.3 Aip Suprapto Munari/1164063

#### 4.3.1 Teori

##### 1. Klasifikasi teks

Klasifikasi teks atau kategorisasi teks merupakan proses yang secara otomatis menempatkan dokumen teks ke dalam suatu kategori berdasarkan isi dari teks tersebut.

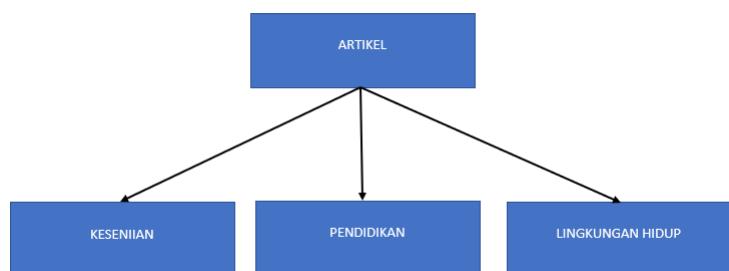


Figure 4.1: Aip-Klasifikasi teks

## 2. Klasifikasi Bunga tidak dapat penggunaan machine learning

Dikarenakan masalah dari input yang serupa namun output yang berbeda noise, yang dimaksud dengan noise adalah contoh pada output yang direkam bukan seperti perkiraan.

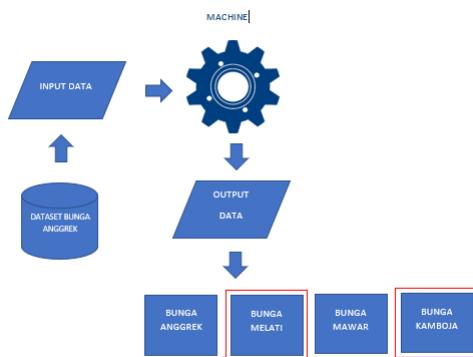


Figure 4.2: Aip-Klasifikasi bunga

## 3. Teknik pembelajaran mesin pada teks YouTube

Teknik Machine Learning pada YouTube memperhatikan apa saja yang menarik perhatian para penggunanya. Ketika kita sedang menonton di YouTube, pada sebelah kanan terdapat 'Up Next' yang menampilkan beberapa video serupa yang sedang ditonton. Dan ketika mengklik salah satu video dari baris tersebut, maka YouTube akan mengingatnya dan menggunakan kata yang tertera sebagai referensi.

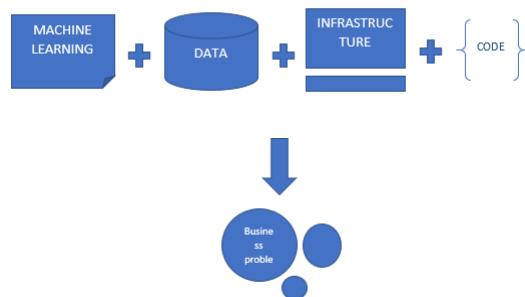


Figure 4.3: Aip-Teknik YouTube

## 4. Vectorisasi Data

- Vectorisasi Data merupakan pemecahan serta pembagian data kemudian dilakukan perhitungan datanya.

## 5. Bag of word

Bag of Words adalah metode untuk mengekstraksi fitur dari dokumen teks.

**1. I love dogs.**

**2. I hate dogs and knitting.**

**3. Knitting is my hobby and my passion.**

Figure 4.4: Aip-Bag of Word

## 6. TF-IDF

TF-IDF merupakan istilah beberapa frekuensi dokumen terbalik, adalah ukuran penilaian yang banyak digunakan dalam pengambilan informasi (IR) atau peringkasan.

Term (t)	df	idf
Akhir	1	$\log(4/1)=0.602$
Awal	1	$\log(4/1)=0.602$
Belajar	2	$\log(4/2)=0.301$
Dokumen	1	$\log(4/1)=0.602$
Frekuensi	1	$\log(4/1)=0.602$
Hitung	3	$\log(4/3)=0.125$
Idf	4	$\log(4/4)=0$
Kita	1	$\log(4/1)=0.602$
Langkah	2	$\log(4/2)=0.301$
Muncul	1	$\log(4/1)=0.602$
Saya	1	$\log(4/1)=0.602$
Term	1	$\log(4/1)=0.602$
Tf	4	$\log(4/4)=0$

Figure 4.5: Aip-TF IDF

## 4.4 BAGIAN PRAKTEK

### 4.5 Aip Suprapto Munari /1164063

#### 4.5.1 Aplikasi Sederhana Menggunakan Pandas

Disini saya akan menggunakan Dataset dari <https://data.world/alexandra/generic-food-database> dan akan mengambil Data Dummy sebanyak 500 records dan membuat Dataframe baru.

```

import pandas as pd
gf = pd.read_csv('D:/KULIAH/SEMESTER 6/KECERDASAN BUATAN/PRAKTEK/Python-Artificial
df = pd.DataFrame(gf, columns = ['FOOD NAME', 'SCIENTIFIC NAME', 'GROUP', 'SUB GR
dummy = pd.get_dummies (df['test preparation course'])
dummy.head()

df = df.join(dummy)

```

Maksud dari kodingan diatas yaitu :

1. Baris pertama impor librari pandas dengan inisiasi pd
2. Definisikan variabel gf (generic-food) untuk membaca file csv dengan pandas
3. variabel df akan menggunakan function pd dataframe untuk membuat datafarme di pandas dari file CSV yang tadi.
4. Mendefinisikan variabel dummy untuk mengubah data categorical menjadi integer. dibawah merupakan data sebelum di Dummy.

Index	FOOD NAME	SCIENTIFIC NAME	GROUP	SUB GROUP
0	nan	nan	nan	nan
1	nan	nan	nan	nan
2	nan	nan	nan	nan
3	nan	nan	nan	nan
4	nan	nan	nan	nan
5	nan	nan	nan	nan
6	nan	nan	nan	nan
7	nan	nan	nan	nan
8	nan	nan	nan	nan
9	nan	nan	nan	nan
10	nan	nan	nan	nan
11	nan	nan	nan	nan
12	nan	nan	nan	nan
13	nan	nan	nan	nan

Figure 4.6: Dataset Original Aip

5. Atribut atau kolom yang ingin di Dummy yaitu test preparation course. Dalam test memunculkan index dan nama keseluruhan kolom.
6. kemudian df akan melakukan join dengan dataframe dummy.

Index	ENTIFIC NAME_GF
0	Angelica,Ang...
1	Savoy cabbage,Bras...
2	Silver linden,Tilia...
3	Kiwi,Actinid...
4	Allium (Onion),Alli...
5	Garden onion,Allium...
6	Leek,Allium porrum,Veget...
7	Garlic,Allium sativum,Herb...
8	Chives,Allium schoenoprasu...
9	Lemon verbena,Aloy...
10	Cashew nut,Anacardi...
11	Pineapple,An...
12	Dill,Anethum graveolens,H...
13	Custard apple,Annona...

Figure 4.7: Dataset Dummy Aip

#### 4.5.2 Memecah DataFrame Menjadi 2 Dataframe

Dari dataframe tersebut dipecah menjadi dua dataframe yaitu 450 row pertama dan 50 row sisanya

```
gf_train= gf[:450]
gf_test= gf[451:]
```

1. gf train akan mendefinisikan dataframe untuk train dengan 450 data pertama
2. gf test mendefinisikan dataframe untuk test untuk data setelah 451. Hasilnya seperti berikut :

gf_test	DataFrame	(50, 1)
gf_train	DataFrame	(450, 1)

Figure 4.8: Split DataFrame Aip

#### 4.5.3 Vektorisasi Dan Klasifikasi Dari Data Youtube05-Shakira Dengan Decision Tree

1. Ini hasil dari impor dataset
2. Ini hasil Setelah di Klasifikasikan dengan Decision Tree
3. Dalam in 13 impor Tree dari Sklearn. Dan mendefinisikan variabel clf untuk memanggil Decision Tree Classifier dan melakukan fit atau pengujian.

```
In [13]: import pandas as pd
...: d = pd.read_csv("D:/KULIAH/SEMESTER 6/KECERDASAN BUATAN/PRAKTEK/Python-Artificial-Intelligence-Projects-for-Beginners-master/Chapter03/Youtube05-Shakira.csv")
In [14]: from sklearn.feature_extraction.text import CountVectorizer
...: vectorizer = CountVectorizer()
```

Figure 4.9: Dataset Youtube05-Shakira Aip

```
In [13]: from sklearn import tree
...: clf = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
...: clf = clf.fit(d_train_att, d_train_label)

In [14]: clf.predict(d_test_att)
Out[14]:
array([1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       0, 0, 0], dtype=int64)

In [15]: clf.score(d_test_att, d_test_label)
Out[15]: 0.8428571428571429
```

Figure 4.10: Dataset Youtube05-Shakira Aip

4. Dalam In 14 menggunakan prediksi untuk clf dengan function predict untuk memprediksi test. Dan hasilnya muncul dalam bentuk array.
5. clf score memunculkan akurasi prediksi yang dilakukan terhadap clf.

#### 4.5.4 Vektorisasi Dan Klasifikasi Dari Data Youtube05-Shakira Dengan SVM

```
In [19]: from sklearn import svm
...: clfsvm = svm.SVC()
...: clfsvm.fit(d_train_att, d_train_label)
...: clfsvm.score(d_test_att, d_test_label)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py
features. Set gamma explicitly to 'auto' or 'scale' to avoid t
"avoid this warning.", FutureWarning)
Out[19]: 0.7571428571428571
```

Figure 4.11: Dataset Youtube05-Shakira SVM Aip

Dari Gambar diatas dapat dijelaskan bahwa :

1. Impor SVM dari sklearn
2. Melakukan fit dari d train att dan d train label atau disebut dengan pengujian
3. Mendefinisikan variabel clf untuk melakukan prediksi dataset Youtube05-Shakira dengan SVM. Dan akan muncul hasil prediksinya

#### 4.5.5 Vektorisasi Dan Klasifikasi Dari Data Youtube05-Shakira Dengan Decision Tree 2

Maksud dari codingan diatas yaitu, mengklasifikasikan Dataset Youtube05-Shakira dengan Decision Tree dengan melakukan prediksi menggunakan function test pada d test att, dan memberikan akurasi prediksi menggunakan prediksi score.

```

In [20]: from sklearn import tree
...: clf = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
...: clf = clf.fit(d_train_att, d_train_label)
...: clf.predict(d_test_att)
...: clf.score(d_test_att,d_test_label)
Out[20]: 0.8428571428571429

```

Figure 4.12: Dataset Youtube05-Shakira Aip

#### 4.5.6 Plotting Confusion Matrix

Berikut adalah skript dari plotting confusion matrix dari contoh yang ada pada bagian teori

```

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",
#               color="white" if cm[i, j] > thresh else "black")

plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')

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

```

Hasilnya adalah sebagai berikut : Dari gambar dapat dijelaskan bahwa data array

```

In [21]: import numpy as np
...: np.set_printoptions(precision=2)
...: plt.figure(figsize=(10,10), dpi=100)
...: plot_confusion_matrix(cm, classes=clf, normalize=True)
...: plt.savefig()
Normalized confusion matrix
[[0.98 0.02]
 [0.18 0.82]]

```

Figure 4.13: Confusion Matrix Aip

merupakan data asli dan data prediksi yang dilakukan dengan Random Forest. Dengan melakukan normalisasi data confusion matrix.

#### 4.5.7 Menjalankan Program Cross Validation

```

In [11]: from sklearn.model_selection import cross_val_score
...: scores = cross_val_score(clf,d_train_att,d_train_label,cv=5)
...:
...: shakirarata2=scores.mean()
...: shakirasd=scores.std()

```

Figure 4.14: Cross Validation Aip

Gambar diatas akan dijelaskan seperti berikut :

1. Dari sklearn mengimpor Cross Validation
2. Variabel scores akan melakukan cross validation pada variabel clf, d train att , dan d train label

3. Variabel shakirarata2 akan menghitung nilai rata rata dari variabel scores tadi menggunakan function mean
4. shakirasd Menghitung standar deviasi dari data yang diberikan. Hasilnya seperti berikut :

```
In [13]: from sklearn import tree
...: clf = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
...: clf = clf.fit(d_train_att, d_train_label)

In [14]: clf.predict(d_test_att)
Out[14]:
array([1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0], dtype=int64)

In [15]: clf.score(d_test_att,d_test_label)
Out[15]: 0.8428571428571429
```

Figure 4.15: Hasil Cross Validation Aip

#### 4.5.8 Program Pengamatan Komponen Informasi

```
In [22]: max_features_opts = range(1, 10, 1)
...: n_estimators_opts = range(2, 40, 4)
...: rf_params = np.empty((len(max_features_opts)*len(n_estimators_opts),4), float)
...: i = 0
...: for max_features in max_features_opts:
...:     for n_estimators in n_estimators_opts:
...:         clf = RandomForestClassifier(max_features=max_features, n_estimators=n_estimators)
...:         scores = cross_val_score(clf, df_train_att, df_train_label, cv=5)
...:         rf_params[i,0] = max_features
...:         rf_params[i,1] = n_estimators
...:         rf_params[i,2] = scores.mean()
...:         rf_params[i,3] = scores.std() * 2
...:         i += 1
```

Figure 4.16: Program Komponen Informas Aip

Dari gambar diatas dapat dijelaskan bahwa :

1. Max featuresnya dari range 1 sampai 10
2. n estimators dengan range 20 sampai 40
3. Variabel rf params berisikan function np empty dimana akan membuat array baru berisikan tipe yang didefinisikan dengan random value
4. Mendefinisika i dimulai dari angka 0 dimana max features dan n estimators menggunakan klasifikasi randomforestclassifier menggunakan data prediksi
5. Mendefinisikan rfparams untuk max features , n estimators, nilai rata dan std

### 4.6 Penanganan Error

AIP SUPRAPTO MUNARI 1164063

#### 4.6.1 Error Index

1. Berikut ini merupakan eror yang didapatkan saat menjalankan program diatas

```
| KeyError: 'generic-food'
```

Figure 4.17: Error Key Aip

2. Pada gambar diatas kode erornya adalah KeyEror. Eror ini terjadi karena keyword yang dimasukan tidak ada.
3. Solusi yang bisa dilakukan untuk mengatasi eror tersebut adalah sebagai berikut

:

```
dummy = pd.get_dummies (df['generic-food'])
dummy.head()
```

Figure 4.18: Error Key Aip

- Pada gambar diatas Dataset Generic-Food tidak terdapat atribut COLOR, maka dari itu kita harus merubahnya dengan atribut yang terdapat di dataset tersebut. Mari kita gunakan atribut SCIENTIFIC NAME. Ubah skrip menjadi seperti berikut

```
dummy = pd.get_dummies (df['COLOR'])
dummy.head()
```

Figure 4.19: Error Key Aip

•

- Maka ketika di run akan muncul data dummy nya seperti berikut

### 4.7 Andi Muhammad Aslam/1164064

#### 4.7.1 Teori

1. Klasifikasi teks

Klasifikasi merupakan kata serapan dari bahasa Belanda, classificatie, yang sendirinya berasal dari bahasa Prancis classification. Istilah ini menunjuk kepada sebuah metode untuk menyusun data secara sistematis atau menurut beberapa aturan atau kaidah yang telah ditetapkan. Di dalam KBBI, klasifikasi adalah

Index
0
1
2
3
4
5
6
7
8
9
10
11
12
13

Figure 4.20: Error Key Aip

penyusunan bersistem dalam kelompok atau golongan menurut kaidah atau standar yang ditetapkan. Secara harafiah bisa pula dikatakan bahwa klasifikasi adalah pembagian sesuatu menurut kelas-kelas. Menurut Ilmu Pengetahuan, Klasifikasi adalah Proses pengelompokan benda berdasarkan ciri-ciri persamaan dan perbedaan.

## 2. Klasifikasi Bunga tidak bisa menggunakan machine learning

Machine Learning tidak dapat mengklasifikasikan bunga, Karena data yang diberikan pada mesin itu akan di algoritmakan untuk mencari sesuatu yang menarik dalam data yang kita berikan, hingga akhirnya sistem AI akan membangun pengetahuan berdasarkan data tersebut. Dengan kata lain, pembelajaran mesin data beradaptasi terhadap suatu masalah dengan mempelajari pola-pola yang ditemukan dalam data Sebagai contoh data pada spesies bunga dari genus Iris dengan melihat ukuran sepal (kelopak) dan petalnya(mahkota) pada algoritma data bunga tersebut akan melatih proses pembelajaran pada mesin dalam menganalisa spesies bunga Iris. Dan algoritma pembelajaran mesin akan mempelajari karakteristik dari masing-masing spesies bunga Iris berdasarkan ukuran sepal dan petal yang diberikan.

## 3. Youtube memungkinkan agar mendapatkan video yang direkomendasikan, karena Machine Learning pada Youtube pasti akan melibatkan data yang sering di lihat oleh penggunanya. Youtube juga akan memberitahukan si pengguna apabila ada video baru yang telah di upload pada chanel yang direkomendasikan untuk

si pengguna. Dan apabila menonton video pada youtube maka youtube dapat mengingat dan menggunakan kata tersebut sebagai referensi.

#### 4. Vectorisasi Data

- proses vektorisasi ini menghasilkan suatu wujud peta yang menggambarkan keadaan permukaan bumi atau bentang alam. Sifat data yang geometris menunjukkan ukuran dimensi yang sesungguhnya.

#### 5. Bag of word

Bag of word merupakan konsep yang diambil dari analisis, kemudian merepresentasikan dokumen berupa kumpulan informasi penting tanpa mengurutkan setiap katanya.

#### 6. TF-IDF

TF-IDF dimaksudkan untuk mencerminkan seberapa relevan suatu istilah dalam dokumen yang diberikan. Intuisi di baliknya adalah bahwa jika sebuah kata muncul beberapa kali dalam sebuah dokumen, kita harus meningkatkan relevansinya karena itu harus lebih bermakna daripada kata-kata lain yang muncul lebih sedikit kali (TF). Pada saat yang sama, jika sebuah kata muncul berkali-kali dalam suatu dokumen tetapi juga di sepanjang banyak dokumen lain, mungkin itu karena kata ini hanya kata yang sering; bukan karena itu relevan atau bermakna (IDF).

## 4.8 Andi Muhammad Aslam/1164064

### 4.8.1 Praktek Program

1. Pada penjelasan kali ini saya akan membuat data dummy dengan menggunakan format csv, data di ambil dari web UCI Machine Learning Repository dengan nama file Youtube02-KatyPerry.csv

Penjelasan pada gambar G1 yaitu mengimport librari pandas dimana kita mengalaskan praktek sebagai pandas. Pandas berguna untuk mengelola dataframe = matrix = tabel kemudian memanggil nama alias untuk membaca format csvnya.

2. Dari dataframe yang sudah ada sebelumnya kita akan membagi menjadi 2 yaitu 450 row pertama dan 50 row

Dapat dilihat pada gambar ??

Penjelasan pada gambar G2 yaitu baris pertama dimana d\_train untuk membagi data training sebanyak 450 row dan pada

baris kedua dimana d\_test untuk data sisa atau data yang baru sebanyak 50 row.

3. Melakukan Vektorisasi dan Klasifikasi Data pada data KetyPerry pada gambar G3 merupakan dataframe keseluruhan dari file csv yang sudah dimasukkan dengan jumlah 448 baris dan 5 kolom. nospam merupakan dataframe yang isinya hanya data yang bukan termasuk spam dengan inisial angka 0. Sedangkan spam merupakan dataframe yang isinya hanya data spam dengan inisial angka 1.

Pada gambar G3 merupakan hasil output content dimana terdapat 448 baris/data yang mempunyai 1602 kata-kata yang digunakan pada komentar yang ada di content tersebut.

Pada gambar G4 maksud dari outputannya merupakan dataframe kata-kata tadi yang berjumlah 1602 kata orang yang komen pada data KetyPerry.

4. Mengklasifikasikan dari data vektorisasi dengan menggunakan klasifikasi svm(support vektorisasi machine) dapat dilihat pada gambar G6.

Jadi pada gambar G6 merupakan hasil dari memprediksi data score vektorisasi dengan svm menggunakan metode fit dimana digunakan untuk data training atau data pelatihannya.

5. Mengklasifikasikan dari data vektorisasi dengan menggunakan klasifikasi Decision Tree dapat dilihat pada gambar G7.

Jadi pada gambar G7 merupakan hasil dari memprediksi data score vektorisasi dengan Decision Tree menggunakan metode fit dimana digunakan untuk data training atau data pelatihannya saja.

6. Mengeplot confusion matrix menggunakan matplotlib dapat dilihat pada gambar G8.

Pada gambar G8 sebelumnya kita harus mengimport matplotlibnya terlebih dahulu setelah itu di sini saya menggunakan numpy untuk mengeluarkan hasil

plot confusion matrix pada matplotlibnya nantinya akan keluar normalisasi dari confusion matrix berupa data baris dan kolom.

7. Menjalankan program cross validation pada data vektorisasi dapat dilihat pada gambar G9.

Pada gambar G9 yang pertama yaitu memunculkan akurasi cross validation dari random forest pada data yang sudah divektorisasi, yang kedua yaitu memunculkan akurasi cross validation dari decision tree pada data yang sudah divektorisasi, dan yang ketiga yaitu memunculkan akurasi cross validation dari svm pada data yang sudah divektorisasi.

8. Membuat program pengamatan komponen informasi pada data yang sudah divektorisasi dapat dilihat pada gambar G10

Pada gambar G10 merupakan hasil outputan yang mana max features di sana terdapat 9 data dari 10 yang sudah kita masukkan ke dalam codingan sebelumnya dan penomoran estimatornya merupakan data per 5 dari angka 40 sedangkan rata-rata akurasinya kita tuliskan datanya mulai dari 0.9 sampai dengan 1. Titik-titik yang didalam tersebut merupakan data vektorisasi dari pengamatan komponen informasinya.

#### 4.8.2 Penanganan Eror

1. Pada gambar G11 merupakan ScreeShootan dari data yang eror.

2. `matplotlib.pyplot`

```
Traceback (most recent call last):
```

```
File "<ipython-input-151-8094433808cc>", line 6, in <module>
    ax = fig.gca(projection='3d')
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\figure.py", line
    return self.add_subplot(1, 1, 1, **kwargs)
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\figure.py", line
    self, *args, **kwargs)
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\projections\__i
    projection_class = get_projection_class(projection)
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\projections\__i
    raise ValueError("Unknown projection '%s'" % projection)

ValueError: Unknown projection '3d'

<Figure size 432x288 with 0 Axes>
```

3. Solusi eror tersebut dengan menambahkan menambahkan codingan tersebut seperti berikut.

```
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
fig = plt.figure()
fig.clf()
```

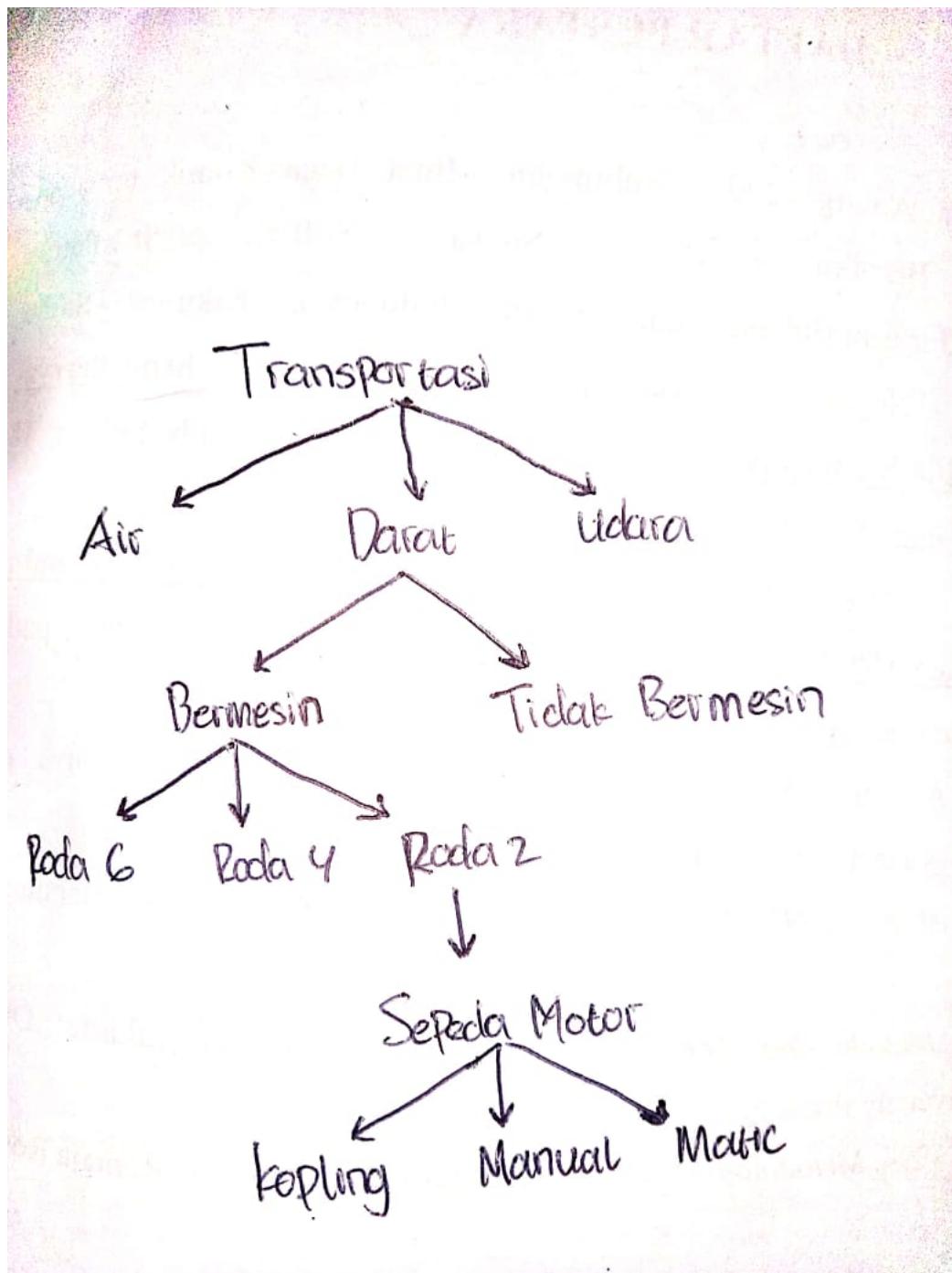


Figure 4.21: Klasifikasi teks



Figure 4.22: Klasifikasi bunga

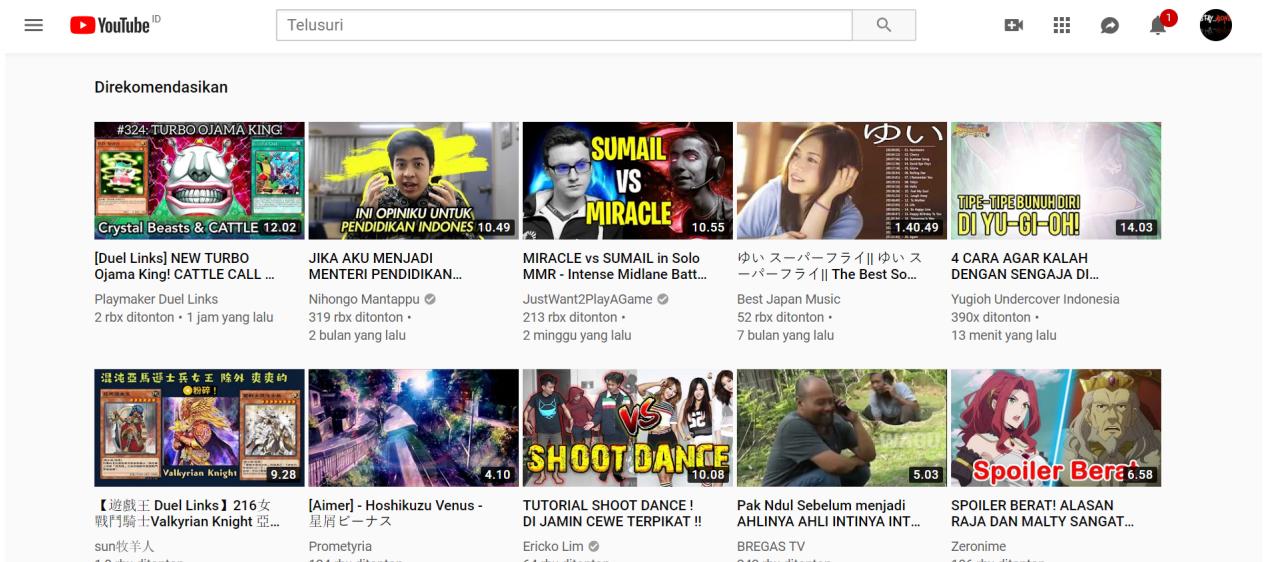


Figure 4.23: Teknik YouTube

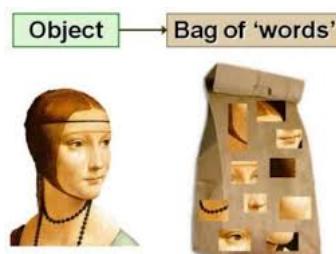


Figure 4.24: Bag of Word

Q	tf			$\frac{df}{D}$	IDF	IDF+1	W = tf * (IDF+1)		
	d1	d2	d3				d1	d2	d3
gold	1	0	1	2	1.5	0.176	1.176	1.176	0
silver	0	2	0	1	3	0.477	1.477	0	2.954
truck	1	1	1	2	1.5	0.176	1.176	0	1.176
						sum(d1)	sum(d2)	sum(d3)	
<b>Nilai Bobot setiap Dokumen =</b>							1.176	4.130	2.352

Figure 4.25: TF-IDF

```
# In[1]: melakukan import pandas dan membaca file csv
import pandas as pd
andi=pd.read_csv('D:\Tugas Kuliah\Semester 6\KECERDASAN BUATAN\Python-Artificial-Intelligence-Projects-for-Beginners\Chapter03\
```

Figure 4.26: Data Dummy 500 Data

nospam	DataFrame	(175, 5)	Column names: COMMENT_ID, AUTHOR, DATE, CONTENT, CLASS
spam	DataFrame	(175, 5)	Column names: COMMENT_ID, AUTHOR, DATE, CONTENT, CLASS
yeay	DataFrame	(350, 5)	Column names: COMMENT_ID, AUTHOR, DATE, CONTENT, CLASS

Variable explorer    File explorer    Help

IPython console

Console 1/A ✖

```
In [118]: yeay = andi.sample(frac=1)

In [119]: andi_train=yeay[:300]
...: andi_test=yeay[300:]

In [120]: |
```

Figure 4.27: Membagi 2 Dataframe

nospam	DataFrame	(175, 5)	Column names: COMMENT_ID, AUTHOR, DATE, CONTENT, CLASS
spam	DataFrame	(175, 5)	Column names: COMMENT_ID, AUTHOR, DATE, CONTENT, CLASS
yeay	DataFrame	(350, 5)	Column names: COMMENT_ID, AUTHOR, DATE, CONTENT, CLASS

Figure 4.28: Vektorisasi dan Klasifikasi Data

```
In [85]: dvec
Out[85]:
<350x1738 sparse matrix of type '<class 'numpy.int64'>
with 5275 stored elements in Compressed Sparse Row format>
```

Figure 4.29: Data Content

dk	list	1738	['00', '000', '002', '018', '04', '053012', '0cb8qfjaa', '0d878a889c', ...]
----	------	------	---

Figure 4.30: DataFrame Kata-kata Pada Content

```
In [141]: from sklearn import svm
....: clfsvm = svm.SVC()
....: clfsvm.fit(andi_train_att, andi_train_label)
....: clfsvm.score(andi_test_att, andi_test_label)
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)
Out[141]: 0.4
```

Figure 4.31: Klasifikasi SVM Dari Data Vektorisasi

```
In [140]: from sklearn import tree
....: clftree = tree.DecisionTreeClassifier()
....: clftree.fit(andi_train_att, andi_train_label)
....: clftree.score(andi_test_att, andi_test_label)
Out[140]: 0.96
```

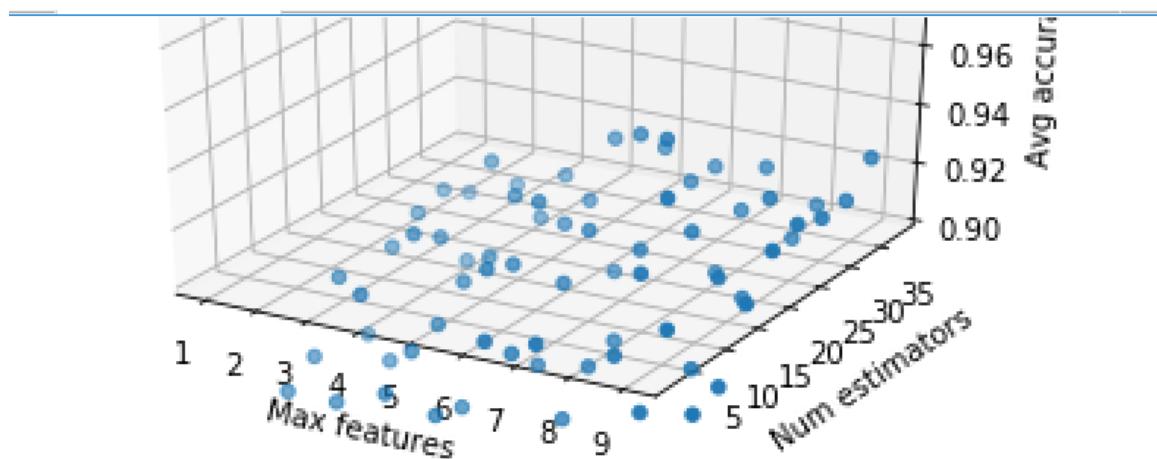
Figure 4.32: Klasifikasi Decision Tree Dari Data Vektorisasi

```
In [144]: import numpy as np
....: np.set_printoptions(precision=2)
....: plot_confusion_matrix(cm, classes=andi, normalize=True)
....: plt.show()
Normalized confusion matrix
[[1.  0. ]
 [0.05 0.95]]
```

Figure 4.33: Plot Confusion Matrix Menggunakan Matplotlib

```
In [145]: from sklearn.model_selection import cross_val_score
....: scores = cross_val_score(clf, andi_train_att, andi_train_label,
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: 0.93 (+/- 0.03)
```

Figure 4.34: Program Cross Validation Pada Data Vektorisasi



```
In [180]: |
```

Figure 4.35: Program Pengamatan Komponen Informasi

```
File "C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\figure.py",
line 1221, in add_subplot
    self, *args, **kwargs)

File "C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\projections
\_init_.py", line 91, in process_projection_requirements
    projection_class = get_projection_class(projection)

File "C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\projections
\_init_.py", line 65, in get_projection_class
    raise ValueError("Unknown projection '%s'" % projection)

ValueError: Unknown projection '3d'
```

Figure 4.36: Eror matplotlib.pyplot

# **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.

### **5.5 Andi Muh Aslam/1164064**

#### **5.5.1 Teori**

1. Jelaskan Kenapa Kata-Kata harus dilakukan vektorisasi lengkapi dengan ilustrasi gambar.

Kata-kata harus dilakukan vektorisasi untuk mengukur nilai kemunculan suatu kata agar dapat di prediksi atau untuk menentukan bobot suatu kata.

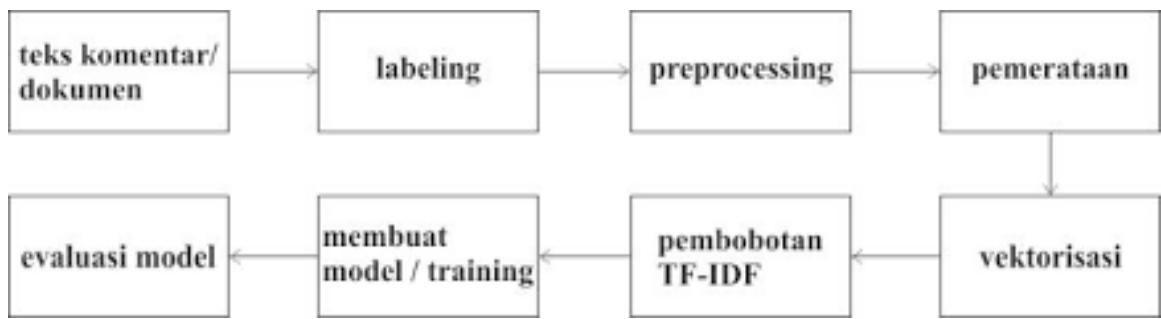


Figure 5.1: Ilustrasi Soal No.1

Untuk ilustrasinya dapat dilihat pada gambar

2. Jelaskan Mengapa dimensi dari vektor dataset google bisa mencapai 300 lengkapi dengan ilustrasi gambar.

Dimensi dari vektor dataset google dapat mencapai 300 karena dimensi pada vektor agar dapat membandingkan bobot dari kata tersebut, misalkan terdapat kata Jaket dan Tas pada data set google setiap kata tersebut dibuat dimensi vektor senilai 300 kata Jaket dan 300 kata Tas, agar dapat membandingkan bobot dari kesamaan kata Jaket dan Tas.

Untuk ilustrasinya dapat dilihat pada gambar

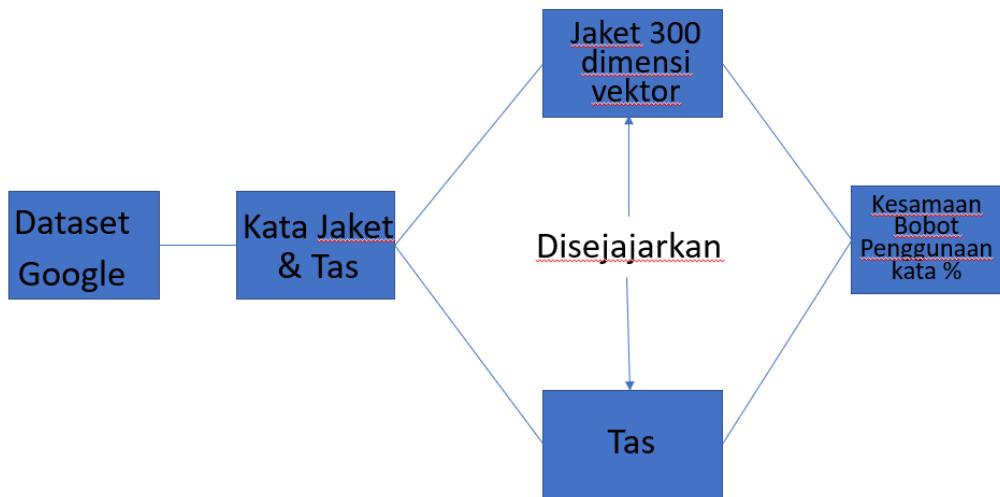


Figure 5.2: Ilustrasi Soal No. 2

3. Jelaskan Konsep vektorisasi untuk kata . dilengkapi dengan ilustrasi atau gambar.

Konsep dari vektorisasi kata yaitu agar dapat mengetahui kata tengah pada kalimat utama, Contoh ( Subscribe channel ini dan Like yah Guys ). Kata tengah dari contoh tersebut merupakan (dan) yang memiliki bobot sebagai kata tengah dari kalimat. Hal ini sangat berkaitan dengan dimensi vektor pada dataset google karena memiliki nilai atau bobot kata tengah.

Untuk ilustrasinya dapat dilihat pada gambar



Figure 5.3: Ilustrasi Soal No. 3

4. Jelaskan Konsep vektorisasi untuk dokumen. dilengkapi dengan ilustrasi atau gambar.

Konsep vektorisasi pada dokumen mesin akan membaca kata-kata terlebih dahulu pada semua kalimat yang ada di dalam dokumen dan nanti kalimat yg ada di dalam dokumen akan dipecah menjadi kata-kata.

Untuk ilustrasinya dapat dilihat pada gambar



Figure 5.4: Ilustrasi Soal No. 4

5. Jelaskan apa mean dan standar deviasi, lengkapi dengan ilustrasi atau gambar.

Mean merupakan nilai rata-rata yang tingkat akurasinya tinggi atau nilai tersebut sering muncul. Standar deviasi mengukur bagaimana nilai-nilai data tersebar. Bisa juga didefinisikan sebagai, rata-rata jarak penyimpangan titik-titik data diukur dari nilai rata-rata data tersebut.

Untuk ilustrasinya dapat dilihat pada gambar

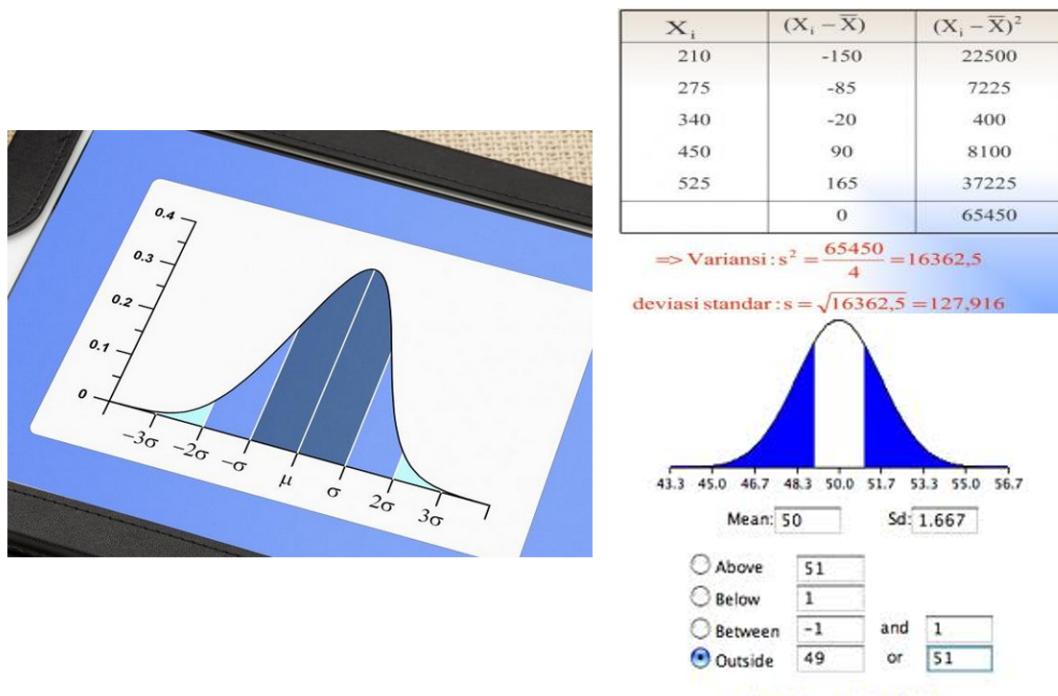


Figure 5.5: Ilustrasi Soal No. 5

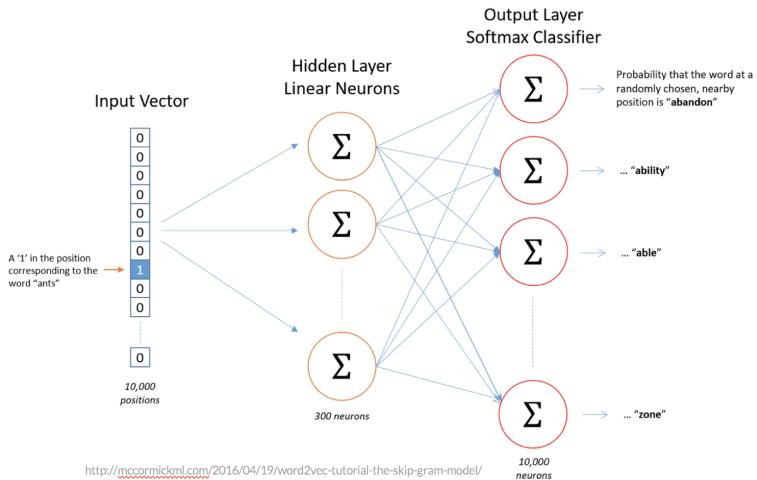
6. Jelaskan Apa itu Skip-Gram sertakan contoh ilustrasi.

Skip-Gram yaitu dimana kata tengah menjadi acuan terhadap kata-kata pelengkap dalam suatu kalimat.

Untuk ilustrasinya dapat dilihat pada gambar

### 5.5.2 Praktek Program

- Cobalah dataset google, dan jelaskan vektor dari kata love, faith, fall, sick, clear, shine, bag, car, wash, motor, cycle dan cobalah untuk melakukan perbandingan similaritas dari masing-masing kata tersebut. Jelaskan arti dari outputan similaritas.



**“Skip-Gram”**  
With one-hot encoded centre word, we can predict context words.

Hidden layer creates embeddings

Figure 5.6: Ilustrasi Soal No. 6

Output source code dibawah akan memunculkan data vektor untuk kata love. bahwa vektor memiliki array sebanyak 300 dimensi. Hasil pada source code tersebut dapat dilihat pada gambar.

```
gmodel['love']
```

```
In [34]: gmodel['love']
Out[34]:
array([ 0.10302734, -0.15234375,  0.02587891,  0.16503906, -0.16503906,
```

Figure 5.7: Love

Output source code dibawah akan memunculkan data vektor untuk kata faith. bahwa vektor memiliki array sebanyak 300 dimensi. Hasil pada source code tersebut dapat dilihat pada gambar 5.8.

```
gmodel['faith']
```

```
In [35]: gmodel['faith']
Out[35]:
array([ 0.26367188, -0.04150391,  0.1953125 ,  0.13476562, -0.14648438,
```

Figure 5.8: Faith

Output source code dibawah akan memunculkan data vektor untuk kata fall. bahwa vektor memiliki array sebanyak 300 dimensi. Hasil pada source code tersebut dapat dilihat pada gambar 5.9.

```
gmodel['fall']
```

```
In [36]: gmodel['fall']
Out[36]:
array([-0.04272461,  0.10742188, -0.09277344,  0.16894531, -0.1328125 ,
```

Figure 5.9: Fall

Output source code dibawah akan memunculkan data vektor untuk kata sick. bahwa vektor memiliki array sebanyak 300 dimensi. Hasil pada source code tersebut dapat dilihat pada gambar 5.10.

```
gmodel['sick']
```

```
In [37]: gmodel['sick']
Out[37]:
array([ 1.82617188e-01,  1.49414062e-01, -4.05273438e-02,  1.64062500e-01,
```

Figure 5.10: Sick

Output source code dibawah akan memunculkan data vektor untuk kata clear. bahwa vektor memiliki array sebanyak 300 dimensi. Hasil pada source code tersebut dapat dilihat pada gambar 5.11.

```
gmodel['clear']
```

```
In [38]: gmodel['clear']
Out[38]:
array([-2.44140625e-04, -1.02050781e-01, -1.49414062e-01, -4.24804688e-02,
```

Figure 5.11: Clear

Output source code dibawah akan memunculkan data vektor untuk kata shine. bahwa vektor memiliki array sebanyak 300 dimensi. Hasil pada source code tersebut dapat dilihat pada gambar 5.12.

```
gmodel['shine']
```

```
In [39]: gmodel['shine']
Out[39]:
array([-0.12402344,  0.25976562, -0.15917969, -0.27734375,  0.30273438,
```

Figure 5.12: Shine

Output source code dibawah akan memunculkan data vektor untuk kata bag. bahwa vektor memiliki array sebanyak 300 dimensi. Hasil pada source code tersebut dapat dilihat pada gambar 5.13.

```
gmodel['bag']
```

```
In [40]: gmodel['bag']
Out[40]:
array([-0.03515625,  0.15234375, -0.12402344,  0.13378906, -0.11328125,
```

Figure 5.13: Bag

Output source code dibawah akan memunculkan data vektor untuk kata car. bahwa vektor memiliki array sebanyak 300 dimensi. Hasil pada source code tersebut dapat dilihat pada gambar 5.14.

```
gmodel['car']
```

```
In [41]: gmodel['car']
Out[41]:
array([ 0.13085938,  0.00842285,  0.03344727, -0.05883789,  0.04003906,
```

Figure 5.14: Car

Output source code dibawah akan memunculkan data vektor untuk kata wash. bahwa vektor memiliki array sebanyak 300 dimensi. Hasil pada source code tersebut dapat dilihat pada gambar 5.15.

```
gmodel['wash']
```

```
In [42]: gmodel['wash']
Out[42]:
array([ 9.46044922e-03,  1.41601562e-01, -5.46875000e-02,  1.34765625e-01,
```

Figure 5.15: Wash

Output source code dibawah akan memunculkan data vektor untuk kata motor. bahwa vektor memiliki array sebanyak 300 dimensi. Hasil pada source code tersebut dapat dilihat pada gambar 5.16.

```
gmodel['motor']
```

```
In [43]: gmodel['motor']
Out[43]:
array([ 5.73730469e-02,  1.50390625e-01, -4.61425781e-02, -1.32812500e-01,
```

Figure 5.16: Motor

Output source code dibawah akan memunculkan data vektor untuk kata cycle. bahwa vektor memiliki array sebanyak 300 dimensi. Hasil pada source code tersebut dapat dilihat pada gambar 5.17.

```
gmodel['cycle']
```

```
In [44]: gmodel['cycle']
Out[44]:
array([ 0.04541016,  0.21679688, -0.02709961,  0.12353516, -0.20703125,
```

Figure 5.17: Cycle

Pada source code dibawah menunjukkan hasil score perbandingan kata apakah kata motor dan cycle memiliki ke samaan atau tidak. Hasil pada source code tersebut dapat dilihat pada gambar 5.18.

```
gmodel.similarity('motor','cycle')
```

Pada source code dibawah menunjukkan hasil score perbandingan kata apakah kata wash dan motor memiliki ke samaan atau tidak. Hasil pada source code tersebut dapat dilihat pada gambar 5.19.

```
In [45]: gmodel.similarity('motor','cycle')
Out[45]: 0.1794929675764453
```

Figure 5.18: Similariti Pada Kata Motor dan Cycle

```
gmodel.similarity('wash','motor')
```

```
In [46]: gmodel.similarity('wash','motor')
Out[46]: 0.10280077965607967
```

Figure 5.19: Similariti Pada Kata Wash dan Motor

Untuk Motor dan Cycle hasilnya adalah 17%

Untuk Wash dan Motor hasilnya adalah 10%

Artinya Motor dan Cyle memang dalam kategori yang sama misalnya dalam kategori kata-kata yang disatukan/berpasangan. Mesin sudah mengetahui bahwa keduanya dapat dikategorikan sebagai sepasang kata.

2. Jelaskan dengan kata dan ilustrasi fungsi dari extract\_words dan PermuteSentences.

Extract\_Words merupakan function untuk menambahkan, menghilangkan atau menghapuskan, hal hal yang tidak penting atau tidak perlu di dalam teks. Pada gambar 5.20 berikut ini menggunakan function extract\_words untuk menghapus komen dengan python style , mencari data yang diinginkan, dan memberikan spasi pada teks.

```
In [48]: import re
...: def extract_words(luarbiasa):
...:     luarbiasa = luarbiasa.lower()
...:     luarbiasa = re.sub(r'<[^>]+>', ' ', luarbiasa) #hapus tag html
...:     luarbiasa = re.sub(r'(\w)\'(\w)', ' ', luarbiasa) #hapus petik satu
...:     luarbiasa = re.sub(r'\W', ' ', luarbiasa) #hapus tanda baca
...:     luarbiasa = re.sub(r'\s+', ' ', luarbiasa) #hapus spasi yang berurutan
...:     return luarbiasa.split()
```

Figure 5.20: Extract\_Words

PermuteSentences berfungsi untuk melakukan pengacakan data supaya memperoleh data yang teratur. Ini merupakan class yang digunakan untuk melakukan pengocokan secara acak pada data yang ada. Digunakan cara ini agar tidak terjadi kelebihan memori pada saat dijalankan. Hasilnya dapat dilihat pada gambar 5.21.

```
In [49]: import random
....: class PermuteSentences(object):
....:     def __init__(self, luarbiasas):
....:         self.luarbiasas = luarbiasas
....:
....:     def __iter__(self):
....:         shuffled = list(self.luarbiasas)
....:         random.shuffle(shuffled)
....:         for luarbiasa in shuffled:
....:             yield luarbiasa
```

Figure 5.21: Permute Sentences

3. Jelaskan fungsi dari librari gensim TaggedDocument dan Doc2Vec disertai praktek pemakaianya.

Fungsi dari library gensim yaitu sebagai pemodelan topik tanpa pengawasan dan pemrosesan bahasa alami, atau bisa kita sebut dengan unsupervised. tagged document itu sendiri untuk memasukan kata-kata pada setiap dokumennya yang akan di vektorisasi. Fungsi dari doc2vec itu sendiri ialah untuk membandingkan bobot data yang terdapat pada dokumen lainnya, apakah kata-kata didalamnya ada yang sama atau tidak. Ketika di running maka tidak terjadi apa-apa, seperti pada gambar 5.22

```
In [10]: from gensim.models.doc2vec import TaggedDocument
....: from gensim.models import Doc2Vec
```

Figure 5.22: Gensim TaggedDocument dan Doc2vec

4. Jelaskan dengan kata dan praktek cara menambahkan data training dari

file yang dimasukkan kepada variabel dalam rangka melatih model doc2vec.

Untuk menambahkan data training yaitu melakukan import library os, library os berfungsi untuk melakukan interaksi antara python dengan os laptop kita masing-masing, setelah itu kita buat variabel unsup\_sentences. Kemudian pilih direktori tempat data kita disimpan. Lalu menyortir data yang terdapat pada folder aclImdb dan membaca file tersebut dengan ekstensi .txt.

```
import os
unsup_sentences = []
for dirname in ["train/pos", "train/neg", "train/unsup", "test/pos", "test/neg"]:
    for fname in sorted(os.listdir("aclImdb/" + dirname)):
        if fname[-4:] == '.txt':
            with open("aclImdb/" + dirname + "/" + fname, encoding='UTF-8') as f:
                sent = f.read()
                words = extract_words(sent)
            unsup_sentences.append(TaggedDocument(words, [dirname + "/" + fname]))
```

Hasil dari code yang diatas ialah terdapatnya data training dengan jumlah 10000 data dari variabel unsup\_sentences hasil running dari folder aclImdb dapat dilihat pada gambar

dirname	str	1	txt_sentoken/neg
fname	str	1	cv999_14636.txt

Figure 5.23: Aclimbdb

```
for dirname in ["review_polarity/txt_sentoken/pos", "review_polarity/txt_sentoken/neg"]:
    for fname in sorted(os.listdir(dirname)):
        if fname[-4:] == '.txt':
            with open(dirname + "/" + fname, encoding='UTF-8') as f:
                for i, sent in enumerate(f):
                    words = extract_words(sent)
            unsup_sentences.append(TaggedDocument(words, ["%s/%s-%d" % (dirname, fname, i)]))
```

Untuk code berikutnya akan menambahkan data training pada variabel unsup\_sentences sekitar 64.720 data, seperti pada gambar

Name	Type	Size	Value
dirname	str	1	txt_sentoken/neg
fname	str	1	cv999_14636.txt
i	int	1	10604
kirim	str	1	after watching _a_night_at_the_roxbury_ , you'll...
sent	str	1	Her fans walked out muttering words like ``horri...
unsup_sentences	list	175325	[TaggedDocument, TaggedDocument, TaggedDocument,...]

Figure 5.24: Aclimbdb

```
with open("stanfordSentimentTreebank/original_rt_snippets.txt",encoding='UTF-8') as f:
    for i, sent in enumerate(f):
        words = extract_words(sent)
        unsup_sentences.append(TaggedDocument(words,["rt-%d" % i]))
```

Untuk code berikutnya akan menambahkan data training pada variabel unsup\_sentences sekitar 10.605 data, seperti pada gambar

dirname	str	1	txt_sentoken/neg
fname	str	1	cv999_14636.txt
i	int	1	10604
kirim	str	1	after watching _a_night_at_the_roxbury_ , you'll b...
sent	str	1	Her fans walked out muttering words like ``horrib...
unsup_sentences	list	175325	[TaggedDocument, TaggedDocument, TaggedDocument, ...]
words	list	29	['her', 'fans', 'walked', 'out', 'muttering', 'wor...

Figure 5.25: Aclimbdb

5. Jelaskan dengan kata dan praktek kenapa harus dilakukan pengocokan dan pembersihan data.

Pengocokan data itu berguna untuk mengacak data supaya pada saat data di running bisa berjalan lebih baik dan hasil presentase akhirnya bisa lebih

bagus. Sedangkan pembersihan data untuk memberikan ruang bagi ram komputer kita setelah melakukan running data sebanyak 3 juta lebih, agar lebih ringan saat proses selanjutnya. Hasil dari pengacakan data tidak ditampilkan pada spyder. Dan sebelumnya memori yang terpakai itu sekitar 87% , setelah dikosongkan jadi 71%. Untuk source codenya dapat dilihat sebagai berikut:

```
# Pengocokan data
mute = PermuteSentences(unsup_sentences)
# Pembersihan data
model.delete_temporary_training_data(keep_inference=True)
```

6. Jelaskan dengan kata dan praktek kenapa model harus di save dan kenapa temporari training harus dihapus.

Save data untuk menyimpan file hasil dari proses training data sebelumnya, model tersebut dilakukan penyimpanan untuk memberikan keringanan pada ram agar saat kita akan melakukan training lagi, model tersebut tinggal di muat saja tanpa harus melakukan training dari awal dan bisa menghemat waktu. Sedangkan untuk delete temporary training data untuk menghapus data training yang sebelumnya sudah dilakukan dan disimpan, tujuannya memberikan keringanan pada ram. Karena setelah melakukan proses training, ram biasanya jadi berat untuk membaca sampai komputer kita jadi lemot. Untuk source codenya dapat dilihat sebagai berikut:

```
# Save data
model.save('ocean.d2v')
# Delete temporary data
model.delete_temporary_training_data(keep_inference=True)
```

7. Jalankan dengan kata dan praktek maksud dari infer code.

Infer vector yaitu membandingkan kata yang tercantum dengan vektor pada dokumen yang sudah di muat sebelumnya. Selain itu infer\_vector juga menghitung vektor dari kata yang dicantumkan pada model yang telah kita buat. Seharusnya kata yang dicantumkan itu lebih panjang lagi agar hasilnya bisa lebih baik lagi.

```
model.infer_vector(extract_words("I will go home"))
```

```
In [14]: model.infer_vector(extract_words("I will go home"))
Out[14]:
array([-0.13122962, -0.2173184 , -0.14073701,  0.47829896,  0.10263892,
       -0.12006506, -0.02554635, -0.06321128,  0.01674332,  0.21312888,
       -0.03759272, -0.21068032,  0.08137176, -0.04404473,  0.23163871,
       -0.11791784, -0.02098055, -0.44984344, -0.21420991,  0.10826829,
       -0.24330835, -0.13386108,  0.05421483, -0.10329439, -0.16347748,
       0.03265174, -0.18829556,  0.14504528, -0.03915659, -0.39758494,
       0.05695166,  0.02784907, -0.15537408,  0.14412752, -0.0224928 ,
       -0.40351996, -0.11055487,  0.18106677,  0.09285883, -0.2614472 ,
       0.24014267,  0.15855208, -0.1287663 ,  0.53696984,  0.08572944,
       0.2602722 ,  0.3341717 ,  0.12752089,  0.15490048,  0.25898734,
      -0.23501003, -0.03285267], dtype=float32)
```

Figure 5.26: Infer Code

Pada source code tersebut menghasilkan keluaran seperti pada gambar.

8. Jelaskan dengan praktik dan kata maksud dari cosine similarity.

Cosine similarity yaitu membandingkan vektorisasi data diantara kedua kata yang di masukkan, Jika hasil presentase dari kedua kata tersebut lebih dari 50 persen kemungkinan kata tersebut terdapat dalam 1 file. Tapi jika kurang dari 50 persen kata tersebut tidak terdapat dalam 1 file. Hasil yang didapatkan pada code tersebut hanya 0.4 persen itu dikarenakan kata pertama dan kedua tidak memiliki kesamaan vektorisasi dan tidak terdapat pada salah satu dokumen.

```
from sklearn.metrics.pairwise import cosine_similarity
cosine_similarity(
    [model.infer_vector(extract_words("she going to school, after wash hand"))],
    [model.infer_vector(extract_words("Services sucks."))])
```

Pada source code tersebut menghasilkan keluaran seperti pada gambar

```
In [15]: from sklearn.metrics.pairwise import cosine_similarity
.... cosine_similarity(
....     [model.infer_vector(extract_words("she going to school, after wash hand"))],
....     [model.infer_vector(extract_words("Services sucks."))])
Out[15]: array([[0.46642035]], dtype=float32)
```

Figure 5.27: Cosine Similarity

9. Jelaskan dengan praktik score dari cross validation masing-masing metode.

score dari cross validation yang mana mengecek data score KNN untuk menghasilkan typedata float64 dimana terdapat 5 data dari score yang type-datanya float64 tersebut. Dan disini menggunakan metode clf

```
scores = cross_val_score(clf, sentvecs, sentiments, cv=5)
np.mean(scores), np.std(scores)
```

Pada source code tersebut menghasilkan keluaran seperti pada gambar

The screenshot shows a Jupyter Notebook environment. At the top, there's a toolbar with tabs for 'Variable explorer', 'File explorer', and 'Help'. Below that is a section labeled 'Python console' with a tab bar showing 'Console 1/A'. The main area displays the following code and its output:

```
In [20]: scores = cross_val_score(clf, sentvecs, sentiments, cv=5)
...: np.mean(scores), np.std(scores)
Out[20]: (0.765, 0.016532795690182997)
```

Figure 5.28: Cross Validation Metode 1

```
scores = cross_val_score(clfrf, sentvecs, sentiments, cv=5)
np.mean(scores), np.std(scores)
```

Pada source code tersebut mengecek data score RF dengan menggunakan metode clfrf yang menghasilkan keluaran seperti pada gambar

```
Out[21]: (0.7150000000000001, 0.02378141197564929)
```

Figure 5.29: Cross Validation Metode 2

```
from sklearn.pipeline import make_pipeline
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
pipeline = make_pipeline(CountVectorizer(), TfidfTransformer(), RandomForestClassifier())
scores = cross_val_score(pipeline, sentences, sentiments, cv=5)
np.mean(scores), np.std(scores)
```

Pada source code tersebut melakukan skoring dari vektorisasi, tfidf, dan rf lalu dibuat perbandingan yang menghasilkan keluaran seperti pada gambar

```
| Out[22]: (0.749, 0.01271918935222596)
```

Figure 5.30: Cross Validation Metode 3

## 5.6 Aip Suprapto Munari/1164063

### 5.6.1 Teori

#### 1. Mengapa Kata-Kata Harus di Vektorisasi

Kata harus divektorisasi karena mesin hanya mampu membaca data dalam tipe/bentuk angka. Oleh sebab itu, diperlukannya vektorisasi kata untuk mampu membaca data tersebut.

- Gambar :

Penjelasan : Berdasarkan pengertian diatas, ada beberapa contoh yang bisa diterapkan. Untuk salah satu contoh dari klasifikasi data sendiri dapat diliat pada gambar berikut ??.

Source Text	Training Samples
The quick brown fox jumps over the lazy dog.	(the, quick) (the, brown)
The quick brown fox jumps over the lazy dog.	(quick, the) (quick, brown) (quick, fox)
The quick brown fox jumps over the lazy dog.	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown fox jumps over the lazy dog.	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

Figure 5.31: Vektorisasi Kata Aip

#### 2. Mengapa Dimensi Dari Vektor Dataset Google Bisa Sampai 300

Setiap nilai dalam vektor 300 dimensi yang terkait dalam sebuah kata "dioptimalkan" dalam beberapa hal untuk menangkap aspek yang berbeda. Dengan kata lain masing-masing dari 300 nilai sesuai dengan beberapa fitur abstrak

kata. Menghapus kombinasi nilai-nilai ini secara acak akan menghasilkan vektor yang mungkin kurang informasi penting tentang kata tersebut dan mungkin tidak lagi berfungsi sebagai representasi yang baik dari kata itu. Atau singkat cerita mungkin ada lebih dari 3 miliar kata-kata dan kalimat atau data yang tidak mungkin disimpan dalam 1 dimensi vektor maka disimpan menjadi 300 dimensi vektor untuk mengurangi kegagalan memori.

- Gambar :

Penjelasan : Berdasarkan pengertian diatas, ada beberapa contoh yang bisa diterapkan. Untuk salah satu contoh dari klasifikasi data sendiri dapat diliat pada gambar berikut ??.

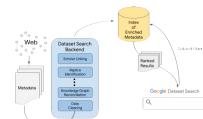


Figure 5.32: Google Dataset Aip

### 3. Konsep Vektorisasi Kata

Konsep vektorisasi data merupakan kata-kata yang di inputkan pada mesin learning. Dan outputnya berupa kara-kata atau keyword dari pencarian yang telah dilakukan sebelumnya. Contohnya pada saat kita melakukan pencarian di youtube maupun pencarian google. Maka akan muncul hasil dari pencarian dari kata-kata yang telah kita cari atau input.

- Gambar :

Penjelasan : Berdasarkan pengertian diatas, ada beberapa contoh yang bisa diterapkan. Untuk salah satu contoh dari klasifikasi data sendiri dapat diliat pada gambar berikut ??.

### 4. Konsep Vektorisasi Dokumen

Konsep vektorisasi dokumen yaitu mesin akan membaca terlebih dahulu semua kalimat yang berada pada dalam dokumen dan nanti kalimat tersebut akan dipecah menjadi kata-kata.

Source Text	Training Samples
The quick brown fox jumps over the lazy dog.	(the, quick) (the, brown)
The quick brown fox jumps over the lazy dog.	(quick, the) (quick, brown) (quick, fox)
The quick brown fox jumps over the lazy dog.	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown fox jumps over the lazy dog.	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

Figure 5.33: Vektorisasi Kata Aip

- Gambar :

Penjelasan : Berdasarkan pengertian diatas, ada beberapa contoh yang bisa diterapkan. Untuk salah satu contoh dari klasifikasi data sendiri dapat diliat pada gambar berikut ??.

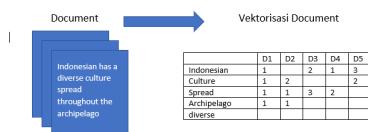


Figure 5.34: Vektorisasi Dokumen Aip

## 5. Mean dan Standar Deviasi

Mean adalah teknik penjelasan kelompok yang didasarkan atas nilai rata-rata dari kelompok tersebut. Rata-Rata (mean) ini didapat dengan menjumlahkan data seluruh individu dalam kelompok itu, kemudian dibagi dengan jumlah individu yang ada pada kelompok tersebut. Standar deviasi adalah nilai statistik yang digunakan untuk menentukan bagaimana sebaran data dalam sampel, dan seberapa dekat titik data individu ke mean atau rata-rata nilai sampel. Sebuah standar deviasi dari kumpulan data sama dengan nol menunjukkan bahwa semua nilai-nilai dalam himpunan tersebut adalah sama. Sebuah nilai deviasi yang lebih besar akan memberikan makna bahwa titik data individu jauh dari nilai rata-rata.

- Gambar :

Penjelasan : Berdasarkan pengertian diatas, ada beberapa contoh yang bisa diterapkan. Untuk salah satu contoh dari klasifikasi data sendiri dapat diliat pada gambar berikut ??.

$$\begin{aligned} \text{Mean} &= \frac{\text{Jumlah data nilai}}{\text{Banyak data nilai}} \\ &= \frac{81}{5} \\ &= 17 \quad \checkmark \end{aligned}$$

Figure 5.35: Mean Aip

$$\begin{aligned} \sigma &= \sqrt{\frac{\sum(X - \mu)^2}{N}}; \text{untuk populasi} \\ s &= \sqrt{\frac{\sum(X - \bar{X})^2}{n-1}}; \text{untuk sampel} \end{aligned}$$

Figure 5.36: Standar Deviasi Aip

## 6. Skip Gram

Skip-Gram mencoba memprediksi vektor kata-kata yang ada di konteks diberikan vektor kata tertentu. Skip-gram membuat sepasang kata target dan konteks sebagai sebuah instance.

- Gambar :

Penjelasan : Berdasarkan pengertian diatas, ada beberapa contoh yang bisa diterapkan. Untuk salah satu contoh dari klasifikasi data sendiri dapat diliat pada gambar berikut ??.

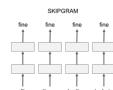


Figure 5.37: Skip-Gram Aip

## 5.7 PRAKTEK PROGRAM

### 5.8 Aip Suprapto Munari/1164063

#### 5.8.1 Mencoba Dataset

##### 5.8.1.1 Vektor

- Pada gambar diatas dapat dilihat bahwa vektor memiliki array sebanyak 300 dimensi. Untuk identitas sektor satu adalah 0.080
- Pada gambar diatas untuk vektor faith dapat dilihat memiliki nilai 0.049 , untuk similaritasnya cukup mendekati vektor love dimana faith dapat dikategorikan dalam satu kategori dengan love.

```
In [37]: gmodel['love']
Out[37]:
array([ 0.10302734, -0.15234375,  0.02587891,  0.16503906, -0.16503906,
       0.06689453,  0.29296875, -0.26367188, -0.140625,   0.20117188,
       -0.02624512, -0.08203125, -0.02770996, -0.04394531, -0.23535156,
       0.16992188,  0.12890625,  0.15722656,  0.00756836, -0.06982422,
       -0.03857422,  0.07958984,  0.22949219, -0.14355469,  0.16796875,
       -0.03515625,  0.055117578,  0.10693359,  0.11181641, -0.16308594,
       -0.11181641,  0.13964844,  0.13556396,  0.12792969,  0.15429688,
       0.07714844,  0.26171875,  0.08642578, -0.02514648,  0.33398438,
       0.18652344, -0.20996894,  0.07880078,  0.02600098, -0.10544531,
       -0.10253906,  0.12304688,  0.04711914,  0.02209473,  0.05834961])
```

Figure 5.38: Vektor Love Aip

```
In [38]: gmodel['faith']
Out[38]:
array([ 0.26367188, -0.04150391,  0.1953125,  0.13476562, -0.14648438,
       0.11962891,  0.04345703,  0.10351562,  0.12287031,  0.13476562,
       0.06646025,  0.18945312, -0.16601562,  0.21679688, -0.27148438,
       0.32803125,  0.18449219,  0.36132812, -0.1953125, -0.18164062,
       0.15332031, -0.18839844,  0.10253906, -0.01367188,  0.23144531,
       -0.05957031, -0.22949219, -0.00604248,  0.26171875,  0.10302734,
       -0.1328125,  0.21484375,  0.01135254,  0.02111816,  0.18554688,
       0.04125977,  0.12011719,  0.17480469, -0.22167969, -0.13476562,
       0.3125,   0.06640625, -0.17675781, -0.01708984, -0.1640625,
       -0.02819824,  0.01257324, -0.09521484, -0.18066406, -0.140625])
```

Figure 5.39: Vektor Faith Aip

- Vektor fall hanya memiliki nilai yaitu 0.046, dimana mesin memahami bahwa fall terdapat dalam satu kategori yang sama dengan faith.

```
In [39]: gmodel['fall']
Out[39]:
array([-0.04272461,  0.10742188, -0.09277344,  0.16894531, -0.1328125,
       -0.10693359,  0.04321289,  0.01094297,  0.14648438,  0.15039062,
       -0.08691406,  0.04492188,  0.0145874,  0.08691406, -0.19824219,
       -0.11035156,  0.01092529, -0.08300781, -0.0189209, -0.1953125,
       -0.1015625,  0.13671875,  0.09228516, -0.12109375,  0.12695312,
       0.03417969,  0.2109375,  0.01977539,  0.125,   0.01544189,
       -0.26953125, -0.0098877, -0.07763672, -0.15527344, -0.03393555,
       0.04199219, -0.29882812, -0.18554688,  0.08496094, -0.02087402])
```

Figure 5.40: Vektor Fall Aip

- Vektor sick memiliki nilai identitas 0.085 mendekati love.
- Vektor clear memiliki nilai identitas -0,025 dan tidak mendekati nilai dari vektor manapun sehingga tidak dapat dijadikan dalam satu kategori.
- Untuk vektor shine 0.35 mendekati vektor Faith dan fall.
- Vektor bag memiliki i=nilai identitas 0.026 yang mendekati dengan vektor fall dan faith. Sehingga mesin memahami bahwa mungkin saja kedua vektor tersebut berada dalam satu kategori.
- Vektor car nilainya 0.48 tidak ada yang mendekati vektor manapun sehingga tidak dapat dikategorikan dalam satu kategori.
- Vektor wash memiliki nilai 0.102 jauh dari vektor vektor lainnya.

```
In [40]: gmodel['sick']
Out[40]:
array([ 1.82617188e-01,  1.49414062e-01, -4.05273438e-02,  1.64062500e-01,
       -2.59765625e-01,  3.22265625e-01,  1.738228125e-01, -1.47460933e-01,
       1.01874219e-01,  5.46875000e-02,  1.66992188e-01, -1.68945312e-01,
       2.24304199e-03,  9.66796875e-02, -1.66015625e-01, -1.12304688e-01,
       1.66015625e-01,  1.79687500e-01,  5.92041016e-03,  2.45117188e-01,
       8.74023438e-02, -2.56347656e-02,  3.41796875e-01,  4.98046875e-02,
       1.78710938e-01, -9.91821289e-04,  8.88671875e-02, -1.95312500e-01,
       1.81640625e-01, -2.65625000e-01, -1.45507812e-01,  1.00585933e-01,
       9.42382812e-02, -3.12500000e-02,  1.98074689e-02, -6.30648438e-02])
```

Figure 5.41: Vektor Sick Aip

```
In [41]: gmodel['clear']
Out[41]:
array([-2.4414805e-04, -1.02050791e-01, -1.49414062e-01, -4.24084688e-02,
       -1.67968750e-01, -1.464643475e-01,  1.76757812e-01,  1.46464375e-01,
       2.26562500e-01,  9.76562500e-02, -2.67578125e-01, -1.29882812e-01,
       1.29882812e-01, -1.29882812e-01, -1.29882812e-01])
```

Figure 5.42: Vektor Clear Aip

- Vektor cycle memiliki nilai identitas 0.179 yang bisa mendekati vektor wash. Dapat dikatakan bahwa motor dapat dicuci jika diarti dalam satu kategori yang sama.

### 5.8.1.2 Similariti

1. Lihat gambar berikut yang merupakan hasil prediksi similariti

Dapat disimpulkan bahwa

- Untuk Motor dan faith hasilnya adalah 0.049
- Untuk Motor dan fall hasilnya adalah 0.046
- Untuk Motor dan clear hasilnya adalah -0.025
- Untuk Motor dan wash hasilnya adalah 0.102
- Untuk Motor dan cycle hasilnya adalah 0.179
- Artinya Motor dan cycle memang dalam kategori yang sama misalnya dalam kategori kendaraan. Mesin sudah mengetahui bahwa keduanya dapat dikategorikan sebagai kendaraan.

## 5.8.2 Extract Words dan PermuteSentences

### 5.8.2.1 Extract Words

ExtractWords merupakan function untuk menambahkan, menghilangkan atau menghapuskan, hal hal yang tidak penting atau tidak perlu di dalam teks. Dalam contoh dibawah ini. menggunakan function extract words untuk menghapus komen dengan python style , mencari data yang diinginkan, dan memberikan spasi pada teks.

```
In [42]: gmodel['shine']
Out[42]:
array([-0.12402344,  0.25976562, -0.15917969, -0.27734375,  0.36273438,
       0.09966938,  0.38257812, -0.22949219, -0.18359375,  0.361875 ,
      -0.08966938,  0.38257812,  0.22949219,  0.18359375,  0.361875 ,
       0.21777544,  0.24823438,  0.5234375 ,  0.12384688, -0.19355938,
```

Figure 5.43: Vektor Shine Aip

```
In [43]: gmodel['bag']
Out[43]:
array([ 0.03151621,  0.15234378, -0.12402344,  0.13378986, -0.11220125,
       -0.0133667 , -0.16112381,  0.14648438, -0.06835938,  0.146825 ,
      -0.06805859, -0.3646175 ,  0.20994694, -0.04345783, -0.2109375 ,
      -0.09345783,  0.20994694, -0.11355156, -0.07910156,  0.06347556,
      0.18847956, -0.07959984, -0.11355156, -0.07910156,  0.06347556,
      -0.15527344, -0.18945312,  0.11132812,  0.27539862, -0.06787189,
      0.08081184,  0.08081184,  0.24121094, -0.21875 ,  0.07568359,
```

Figure 5.44: Vektor Bag Aip

### 5.8.2.2 PermuteSentences

PermuteSentences merupakan class yang digunakan untuk melakukan pengocokan secara acak pada data yang ada. Digunakan cara ini agar tidak terjadi kelebihan memori pada saat dijalankan. Contoh dibawah yaitu fungsi akan memanggil lenght. Yang kemudian mendefinisikan variabel req untuk lenght dan melakukan random choice yaitu pengocokan acak untuk kata motor.

```
In [44]: gmodel['car']
Out[44]:
array([-0.13685938,  0.00842285, -0.03344727, -0.05883789,  0.04003906,
       -0.14257812,  0.04931641, -0.16849531,  0.20808458,  0.11962891,
       0.18000186, -0.25,  0.10404521, -0.19892819, -0.05919513,
       0.00010001,  0.166757,  0.03545331,  0.14453128,  0.04521072,
       0.16113281, -0.0161128, -0.03088379,  0.08447266,  0.16210935,
       0.04447777, -0.15577344,  0.75000074,  0.74042475,  0.04753305]
```

Figure 5.45: Vektor Car Aip

```
In [45]: gmodel['wash']
Out[45]:
array([ 9.4684932e-05,  1.41601562e-01, -5.46574600e-02,  1.34756625e-01,
       2.38201398e-01,  3.44218798e-01, -8.4473552e-02, -1.29883212e-01,
       1.07910156e-01,  2.53906259e-01,  1.13525391e-02, -1.66992188e-01,
       -2.79541816e-02,  2.40887812e-01, -4.11544258e-02,  1.85492150e-01,
       7.40254341e-01,  1.13944602e-01,  1.13944602e-01,  0.11944602e-01,
       2.67579125e-01,  2.12890625e-01,  1.74560547e-02,  2.02941895e-02,
       0.29882812e-01,  1.62109375e-01,  1.93559375e-01,  2.17205150e-02,
```

Figure 5.46: Vektor Wash Aip

```
In [47]: gmodel['cycle']
Out[47]:
array( 0.04541016,  0.21679688, -0.02709961,  0.12353516, -0.28703125,
       -0.1328125,  0.26367188, -0.12808625, -0.125,  0.15320301,
       -0.1328125, -0.13808625,  0.076758, -0.15320301,
       0.02563454,  0.08383838,  0.0825,  0.0464258,  0.11944602,
       -0.13378906, -0.11669922, -0.3359375,  0.07825,  0.08447266,
       0.0722754,  0.0217285,  0.11669922,  0.3359375,  0.18645351,
       0.10382754,  0.0217285, -0.10903359,  0.024598254,  0.18645351,
       -0.05541992, -0.29429288, -0.48039862,  0.06347656, -0.08447266,
```

Figure 5.47: Vektor cycle Aip

```
In [52]: gmodel.similarity('motor', 'love')
Out[52]: 0.0887447004184623
In [53]: gmodel.similarity('motor', 'faith')
Out[53]: 0.049242012249245404
In [54]: gmodel.similarity('motor', 'fall')
Out[54]: 0.046621998318158265
In [55]: gmodel.similarity('motor', 'sick')
Out[55]: 0.085940412798331
In [56]: gmodel.similarity('motor', 'clean')
Out[56]: -0.02343404021318162
In [57]: gmodel.similarity('motor', 'shine')
Out[57]: 0.03908095156368616
In [58]: gmodel.similarity('motor', 'bag')
Out[58]: 0.02609421187493132
In [59]: gmodel.similarity('motor', 'pac')  
In [59]: gmodel.similarity('motor', 'pac')
```

Figure 5.48: Similariti Aip

```
In [59]: import re
...: def extract_words(sku):
...:     sku = sku.lower():
...:     sku = re.sub('<[^>]+>', ' ', sku) #hapus tag html
...:     sku = re.sub('<[^>]+>', ' ', sku) #hapus petik satu
...:     sku = re.sub('<[^>]+>', ' ', sku) #hapus petik dua
...:     sku = re.sub('<[^>]+>', ' ', sku) #hapus spasi yang berurutan
...:     return sku.split()
```

Figure 5.49: Extract Words Aip

```
import random
class PermuteSentences(object):
    def __init__(self, sku):
        self.sku = sku
    def __iter__(self):
        res = list(self.length)
        random.shuffle(res)
```

Figure 5.50: PermuteSentences Aip

# **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 sistem (0,5) c. Lengkap dan bersistem (1)
6	Pemanfaatan Instrumen Pendukung	Pemanfaatan Instrumen Pendukung seperti gambar dan tabel	1	a. Takermanfaatkan (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)
<b>TOTAL</b>			<b>36</b>	
Catatan : Nilai minimal untuk diterima <b>25</b>				

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.