

Modul Praktikum Kecerdasan Buatan



Rolly Maulana Awangga
0410118609

Applied Bachelor of Informatics Engineering
Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering
Politeknik Pos Indonesia

Bandung 2019

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

Acknowledgements

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

Abstract

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

Contents

1	Mengenal Kecerdasan Buatan dan Scikit-Learn	1
1.1	Teori	1
1.2	Instalasi	2
1.3	Penanganan Error	2
1.4	Fadila/1164072	2
1.4.1	Teori	2
1.4.2	Instalasi	6
1.4.3	Penanganan Error	19
1.5	Rahmi Roza/1164085	20
1.5.1	Teori	20
1.5.2	Instalasi	24
1.5.3	Penanganan Error	26
1.6	Lusia Violita Aprilian/1164080	27
1.6.1	Artificial Intelegence	27
1.6.2	Supervised Learning dan Data	30
1.6.3	Learning and Predicting	31
1.6.4	Model Persistence	32
1.6.5	Conventions	32
1.6.6	Penanganan Error	33
2	Related Works	35
2.1	Fadila/1164072	35
2.1.1	Teori	35
2.1.2	Praktek Scikit-Learn	43
2.1.3	Penanganan Error	49
2.2	Lusia Violita Aprilian	51
2.2.1	binary classification dilengkapi ilustrasi gambar	51

2.2.2	supervised learning dan unsupervised learning dan clustering dengan ilustrasi gambar	52
2.2.3	evaluasi dan akurasi dari buku dan disertai ilustrasi contoh dengan gambar	53
2.2.4	bagaimana cara membuat dan membaca confusion matrix, buat confusion matrix	54
2.2.5	bagaimana K-fold cross validation bekerja dengan gambar ilustrasi	55
2.2.6	decision tree dengan gambar ilustrasi	55
2.2.7	Information gain dan entropi dengan gambar ilustrasi	56
2.2.8	binary classification dilengkapi ilustrasi gambar	56
2.2.9	supervised learning dan unsupervised learning dan clustering dengan ilustrasi gambar	57
2.2.10	evaluasi dan akurasi dari buku dan disertai ilustrasi contoh dengan gambar	59
2.2.11	bagaimana cara membuat dan membaca confusion matrix, buat confusion matrix	59
2.2.12	bagaimana K-fold cross validation bekerja dengan gambar ilustrasi	60
2.2.13	decision tree dengan gambar ilustrasi	61
2.2.14	Information gain dan entropi dengan gambar ilustrasi	61
2.2.15	scikit-learn	62
2.2.16	Penanganan Error	66
2.3	Rahmi Roza/1164085	67
2.3.1	Teori	67
2.3.2	Scikit-learn	74
2.3.3	Penanganan Error	79
3	Methods	80
3.1	The data	80
3.2	Method 1	80
3.3	Method 2	80
4	Experiment and Result	81
4.1	Experiment	81
4.2	Result	81

5 Conclusion	82
5.1 Conclusion of Problems	82
5.2 Conclusion of Method	82
5.3 Conclusion of Experiment	82
5.4 Conclusion of Result	82
6 Discussion	83
7 Discussion	84
8 Discussion	85
9 Discussion	86
10 Discussion	87
11 Discussion	88
12 Discussion	89
13 Discussion	90
14 Discussion	91
A Form Penilaian Jurnal	92
B FAQ	95
Bibliography	97

List of Figures

1.1	error model persistence	3
1.2	capturing	4
1.3	penanganan error model persistence	6
1.4	install anaconda 1	6
1.5	penanganan error model persistence	7
1.6	install anaconda 2	7
1.7	Pengecekan Anaconda	8
1.8	instalasi pip scikit-learn	8
1.9	instalasi conda scikit-learn	9
1.10	uji coba codingan	9
1.11	pengujian loading an example dataset	10
1.12	pengujian loading an example dataset	10
1.13	hasil print uji cobat	10
1.14	pengujian learning dan predicting	11
1.15	pengujian model persistence 1	12
1.16	pengujian model persistence 2	12
1.17	pengujian type casting 1	14
1.18	pengujian type casting 2	14
1.19	pengujian refting and updating	14
1.20	pengujian multiclass and multiable 1	15
1.21	pengujian multiclass and multiable 2	15
1.22	error model persistence	19
1.23	penanganan error model persistence	20
1.24	penanganan error model persistence	20
1.25	gambar1	24
1.26	gambar2	24
1.27	gambar3	25
1.28	gambar4	25

1.29	gambar5	26
1.30	gambar6	27
1.31	gambar7	27
1.32	Learning and predicting 1	31
1.33	Learning and predicting 2	31
1.34	Learning and predicting 3	32
1.35	Model Persistence	32
1.36	Conventions	33
1.37	skrinsut error	33
1.38	gb 1	34
2.1	binary classification	35
2.2	supervised	36
2.3	unsupervised	36
2.4	clustering	37
2.5	Evaluasi	37
2.6	Akurasi	38
2.7	Contoh Evaluasi Dan Akurasi Secara Bersamaan	38
2.8	confusion matrix	39
2.9	recall	39
2.10	precision	40
2.11	f-measure	40
2.12	k-fold classification 1	41
2.13	k-fold classification 2	41
2.14	decision tree	41
2.15	informaion gain 1	42
2.16	information gain 2	42
2.17	entropi	43
2.18	codingan pertama	44
2.19	codingan kedua	44
2.20	codingan ketiga	45
2.21	codingan keempat	45
2.22	codingan kelima	46
2.23	codingan keenam	46
2.24	codingan ketujuh	46
2.25	codingan kedelapan	47

2.26	codingan kesembilan	47
2.27	codingan ke-10	48
2.28	codingan ke-11	48
2.29	codingan ke-12	49
2.30	error 1	49
2.31	error 2	50
2.32	error 3	50
2.33	error 4	51
2.34	Binary Classification	52
2.35	Supervised Learning	52
2.36	Unsupervised Learning	53
2.37	Cluster	54
2.38	Evaluasi dan Akurasi	54
2.39	K-fold cross validation	55
2.40	Decision Tree	56
2.41	Information gain	56
2.42	Entropi	56
2.43	Binary Classification	57
2.44	Supervised Learning	57
2.45	Unsupervised Learning	58
2.46	Cluster	59
2.47	Evaluasi dan Akurasi	59
2.48	K-fold cross validation	60
2.49	Decision Tree	61
2.50	Information gain	61
2.51	Entropi	62
2.52	load dataset	62
2.53	generate binary label	62
2.54	use one-hot encoding on categorical columns	63
2.55	shuffle rows, split training dan testing data	63
2.56	fit a decision tree	63
2.57	visualize tree	64
2.58	visualize tree	64
2.59	save tree	64
2.60	t.score	65
2.61	show average score	65

2.62	for max depth in range	65
2.63	depth acc	66
2.64	matplotlib	66
2.65	error	67
2.66	penyelesaian	67
2.67	binary classification	68
2.68	supervised	68
2.69	unsupervised	69
2.70	clustering	69
2.71	Evaluasi	70
2.72	Akurasi	70
2.73	Contoh Evaluasi Dan Akurasi Secara Bersamaan	70
2.74	confusion matrix	72
2.75	k-fold classification 1	73
2.76	decision tree	73
2.77	informaion gain 1	74
2.78	entropi	74
2.79	Gambar pertama	75
2.80	Gambar kedua	75
2.81	Gambar Ketiga	75
2.82	Gambar Keempat	76
2.83	Gambar Kelima	76
2.84	Gambar Keenam	76
2.85	Gambar Ketujuh	77
2.86	Gambar Kedelapan	77
2.87	Gambar Kesembilan	77
2.88	Gambar Kesepuluh	78
2.89	Gambar Kesebelas	78
2.90	Gambar Keduabelas	79
2.91	Gambar Ketigabelas	79
A.1	Form nilai bagian 1.	93
A.2	form nilai bagian 2.	94

Chapter 1

Mengenal Kecerdasan Buatan dan Scikit-Learn

Buku umum yang digunakan adalah [2] dan untuk sebelum UTS menggunakan buku *Python Artificial Intelligence Projects for Beginners*[1]. Dengan praktek menggunakan python 3 dan editor anaconda dan library python scikit-learn. Tujuan pembelajaran pada pertemuan pertama antara lain:

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

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

1.1 Teori

Praktek teori penunjang yang dikerjakan :

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

1.2 Instalasi

Membuka <https://scikit-learn.org/stable/tutorial/basic/tutorial.html>. Dengan menggunakan bahasa yang mudah dimengerti dan bebas plagiat. Dan wajib skreensut 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. skreensut 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 Fadila/1164072

1.4.1 Teori

Teori mencakup resume dari beberapa pembahasan. yaitu :

1. Tentang Kecerdasan Buatan

- Definisi Kecerdasan Buatan.

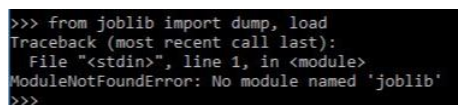
Kecerdasan Buatan biasa disebut dengan istilah AI (Artificial Intelligence) . AI sendiri merupakan suatu cabang dalam bidang sains komputer sains dimana mengkaji tentang bagaimana cara untuk melengkapi sebuah komputer dengan kemampuan atau kepintaran layaknya atau mirip dengan yang dimiliki manusia. Sebagai contoh, sebagaimana komputer dapat berkomunikasi dengan pengguna baik menggunakan kata, suara maupun lain sebagainya . Dengan kemampuan ini, diharapkan komputer mampu mengambil keputusan sendiri untuk berbagai kasus yang ditemuinya kemudian itulah yang disebut dengan kecerdasan buatan.

Kecerdasan buatan makin canggih dengan kemampuan komputer dalam memperbarui pengetahuannya dengan banyaknya testing dan perkembangan target analisa. Untuk kecerdasan buatan ada banyak contoh dan jenisnya. Salah satu contoh yang paling terkenal dari Artificial Intelligence ialah Google Assistant. Google Assistant digunakan untuk kemudahan user dalam menemukan berbagai hal maupun penyettingan langsung terhadap smartphone yang digunakan dan masih banyak lagi.

- Sejarah Kecerdasan Buatan

Artificial intelligence merupakan inovasi baru di bidang ilmu pengetahuan. Mulai terbentuk sejak adanya komputer modern dan kira-kira terjadi sekitaran tahun 1940 dan 1950. Ilmu pengetahuan komputer ini khusus ditujukan dalam perancangan otomatisasi tingkah laku cerdas dalam sistem kecerdasan komputer.

Pada awalnya, kecerdasan buatan hanya ada di universitas-universitas dan laboratorium penelitian, serta hanya sedikit produk yang dihasilkan dan dikembangkan. Menjelang akhir 1970-an dan 1980-an, mulai dikembangkan secara penuh dan hasilnya berangsur-angsur dipublikasikan di khalayak umum.



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

Figure 1.1: error model persistence

2. Penanganan Model Persistence

- Pertama-tama silahkan membuka command prompt
- Kemudian masukkan perintah untuk melakukan instalasi module joblib
perintahnya ialah : `pip install joblib`
- hasilnya akan nampak seperti pada gambar yang ditampilkan



Figure 1.2: capturing

Jika kita berbicara tentang AI atau Artificial Intelligence maka kita tidak bisa melupakan seorang sosok yang sangat terkenal pada bidang tersebut yaitu bapak John McCarthy. McCarthy mendapatkan gelar sarjana matematika dari California Institute of Technology (Caltech) pada September 1948. Dari masa kuliahnya itulah ia mulai mengembangkan ketertarikannya pada mesin yang dapat menirukan cara berpikir manusia. McCarthy kemudian melanjutkan pendidikan ke program doktoral di Princeton University.

McCarthy kemudian mendirikan dua lembaga penelitian kecerdasan buatan. Kedua lembaga AI itu adalah Stanford Artificial Intelligence Laboratory dan MIT Artificial Intelligence Laboratory. Di lembaga-lembaga inilah bermunculan inovasi pengembangan AI yang meliputi bidang human skill, vision, listening, reasoning dan movement of limbs. Bahkan Salah satu lembaga yang didirikan itu, Stanford Artificial Intelligence pernah mendapat bantuan dana dari Pentagon untuk membuat teknologi-teknologi luar angkasa.

- Perkembangan Kecerdasan Buatan

Teknologi Artificial Intelligence semakin ramai dibahas dalam berbagai diskusi teknologi di seluruh dunia. Menurut kebanyakan orang, pekerjaan seperti kasir, operator telepon, pengendara truk, dan lainnya sangat berpeluang besar untuk tergantikan oleh Artificial Intelligence. Mengapa terjadi

hal demikian? dikarenakan memang bahwa AI lebih unggul dalam hal kinerja, fitur dan lain sebagainya. Namun, dalam beberapa aspek memang pekerja manusia masih unggul dibandingkan AI itu sendiri.

Para generasi muda yang ada di dunia terutama di daerah Asia terlihat sudah memahami fungsi dan efek dari AI dalam kehidupan kita sehari-hari. Berdasarkan survei yang dilakukan oleh Microsoft, terdapat 39 persen responden yang mempertimbangkan untuk menggunakan mobil tanpa pengemudi dan 36 persen lainnya setuju bahwa robot masa depan dengan software untuk beroperasi mampu meningkatkan produktivitas. Dari survey tersebut kita sebagai pengguna AI harus lebih bijaksana dalam pengembangan dan penggunaan dari AI sehingga tanpa memberikan efek samping terhadap etos kerja dan keseharian kita sebagai pengguna dalam kehidupan sehari-hari.

3. Tentang Pengertian Terhadap Ilmu Yang Lain

- Supervised Learning adalah pendekatan dimana sudah terdapat data yang dilatih selain itu juga terdapat variable yang ditargetkan sehingga tujuan dari pendekatan ini yaitu mengelompokkan suatu data ke data yang sudah ada.
- Klasifikasi adalah pembagian sesuatu menurut kelas-kelas (class). Menurut Ilmu Pengetahuan, Klasifikasi merupakan proses pengelompokkan benda berdasarkan ciri-ciri persamaan dan juga perbedaan.
- Regresi adalah metode analisis statistik yang digunakan untuk melihat pengaruh antara dua ataupun lebih variabel.
- Unsupervised Learning berbeda dengan Supervised Learning. Perbedaannya ialah unsupervised learning tidak memiliki data latih, sehingga dari data yang ada kita mengelompokkan data tersebut menjadi 2 ataupun 3 bagian dan seterusnya.
- Dataset adalah objek yang merepresentasikan data dan juga relasi yang ada di memory. Strukturnya mirip dengan data di database, namun bedanya dataset berisi koleksi dari data table dan data relation.
- Training Set adalah set digunakan oleh algoritma klasifikasi . Dapat dicontohkan dengan : decision tree, bayesian, neural network dll. Semuanya dapat digunakan untuk membentuk sebuah model classifier.

- Testing Set adalah set yang digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar.

1.4.2 Instalasi

Untuk Instalasinya mencakup i beberapa pembahasan dan tutorial. yaitu :

1. Instalasi Scikit-Learn Dari Anaconda

- Instalasi Anaconda
 - (a) Pertama-tama silahkan pastikan bahwa anda telah melakukan instalasi software Anaconda.
 - (b) Apabila belum, silahkan buka web browser anda untuk melakukan pengunduhan software Anaconda
 - (c) Setelah terunduh, silahkan klik kanan lalu run administrator pada software Anaconda
 - (d) Silahkan lakukan penginstalan dengan menekan tombol install pada tampilan instalasi
 - (e) Kemudian tekan tombol next maka akan sampai pada tampilan diatas

```
C:\WINDOWS\system32>pip install joblib
Collecting joblib
  Downloading https://files.pythonhosted.org/packages/cd/c1/58a758e8247561e58cb87385b1e98b171b8c767b15b12a17340801f41d356/joblib-0.13.2-py2.py3-none-any.whl (278kB)
    100% |#####| 286kB 2.3MB/s
Installing collected packages: joblib
  Successfully installed joblib-0.13.2
You are using pip version 10.0.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\WINDOWS\system32>
```

Figure 1.3: penanganan error model persistence

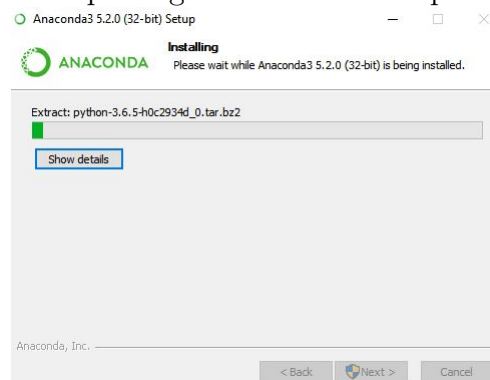


Figure 1.4: install anaconda 1

1. Pengujian Penanganan Model Persistence

- Setelah melakukan penginstalan maka kita harus menguji keberhasilan penginstalan
- Caranya dengan mengecek lewat command prompt bahwa module joblib-nya telah terdefiniskan di python
- Silahkan ketikkan perintah python, lalu masukkan perintah sebagai berikut :
from joblib, import dump, load
- Maka hasilnya akan nampak seperti pada gambar yang ditampilkan

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

Figure 1.5: penanganan error model persistence

1. Selanjutnya apabila instalasi tersebut telah selesai maka silahkan menekan tombol next
2. Tampilan selanjutnya akan seperti ini

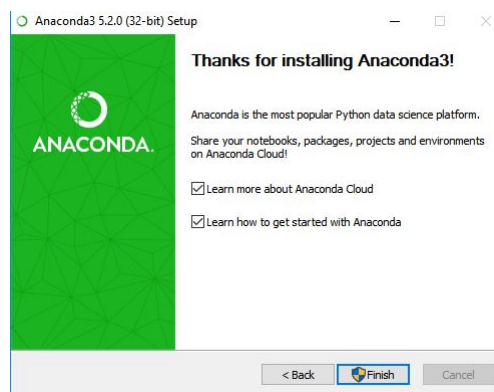


Figure 1.6: install anaconda 2

3. Apabila tampilannya telah sesuai dengan contoh gambar maka instalasi telah selesai
- Instalasi Library Scikit Learn
 1. Silahkan membuka web browser untuk melakukan pengunduhan untuk library scikit dari anaconda.

- <https://anaconda.org/anaconda/scikit-learn>.

- Silahkan masukkan perintah berikut untuk melakukan pengecekan bahwa anaconda anda telah terpasang dengan baik.

conda -version

python -version

5. Tampilannya akan nampak seperti berikut :

```
Microsoft Windows [Version 10.0.17134.500]
(c) 2018 Microsoft Corporation. All rights reserved.

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

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

C:\Users\VASUS>
```

Figure 1.7: Pengecekan Anaconda

- ```
perintahnya : pip install -U scikit-learn
```

7. Tampilannya akan nampak seperti berikut :

[illegible]

Figure 1.8: instalasi pip scikit-learn

- perintahnya : conda install scikit-learn

9. Tampilannya akan nampak seperti berikut :

10. Apabila telah dipraktikkan seperti langkah-langkah dan menghasilkan tampilan seperti contoh diatas, maka instalasi scikit-learn dari anaconda berhasil dilakukan

11. Kemudian untuk pengujian yang lain yaitu pengujian untuk mengecek codingan anaconda

```

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

Package Plan

 environment location: C:\ProgramData\Anaconda

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

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 1.9: instalasi conda scikit-learn

```

In [42]: solok = pd.get_dummies(solok, columns=['sex', 'school', 'address', 'famsize', 'Pstatus', 'Mjob',
'Fjob',
...,
..., 'reason', 'guardian', 'schoolsup', 'famsup', 'paid', 'activities',
..., 'nursery', 'higher', 'internet', 'romantic'])
...: solok.head()
Out[42]:
 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 1.10: uji coba codingan

12. Contoh uji coba codingannya dapat dilihat pada gambar berikut
13. Berdasarkan pengujian tersebut maka dapat dipastikan bahwa anaconda telah ter-include ke dalam python dan dieksekusi dengan script python
14. Setelah pengeksekusiannya berdasarkan scripts python, terdapatlah keluaran yang sesuai
15. Keluaran tersebut yang menandakan bahwa anacondanya berfungsi dengan baik.

- Loading An Example Dataset

- Penerapan Loading An Example Dataset Pada Python Di CMD

1. Pertama-tama silahkan buka command prompt di laptop anda
2. Selanjutnya masuk ke python
3. Setelah masuk kedalam python, silahkan masukkan perintah seperti pada gambar berikut :
4. Secara keseluruhan, hasilnya pada command prompt akan nampak seperti gambar tersebut
5. Apabila tampilanya telah nampak seperti gambar diatas, maka pengujiannya telah selesai dan berhasil.

- Penjelasan Perintah Yang Di Uji

```

In [43]: solok = solok.sample(frac=1)
...: # split training and testing data
...: solok_train = solok[:500]
...: solok_test = solok[500:]
...:
...: solok_train_att = solok_train.drop(['pass'], axis=1)
...: solok_train_pass = solok_train['pass']
...:
...: solok_test_att = solok_test.drop(['pass'], axis=1)
...: solok_test_pass = solok_test['pass']
...:
...: solok_att = solok.drop(['pass'], axis=1)
...: solok_pass = solok['pass']
...:
...: # number of passing students in whole dataset:
...: import numpy as np
...: print("Passing: %d out of %d (%.2f%%)" % (np.sum(solok_pass), len(solok_pass),
100*float(np.sum(solok_pass)) / len(solok_pass)))
Passing: 166 out of 395 (42.03%)

```

Figure 1.11: pengujian loading an example dataset

```

1 from sklearn import datasets
2 iris = datasets.load_iris()
3 digits = datasets.load_digits()

```

Figure 1.12: pengujian loading an example dataset

1. Perhatikan perintah yang telah dieksekusi ini :
2. Penjelasan untuk baris pertama ialah :  
Perintahnya yaitu memasukkan dan memanggil dataset dari sklearn
3. Penjelasan untuk baris kedua ialah :  
Terdapat variabel baru yaitu iris. Dimana variabel iris memanggil datasets dan di dalamnya akan ngeload ( menampilkan ) load iris.
4. Penjelasan untuk baris ketiga ialah :  
Kemudian ada juga variabel baru lainnya yaitu digits yang akan memanggil dataset dan di dalamnya akan ngeload ( menampilkan ) load digits
5. Selanjutnya untuk perintah Print( digits.data ) ditujukan untuk menampilkan output dari pengeksekusian variabel digits dan akan berupa data.
6. Hasilnya printnya sebagai berikut :

```

>>> print(digits.data)
[[0. 0. 5. ... 0. 0. 0.]
 [0. 0. 0. ... 10. 0. 0.]
 [0. 0. 0. ... 16. 9. 0.]
 ...
 [0. 0. 1. ... 6. 0. 0.]
 [0. 0. 2. ... 12. 0. 0.]
 [0. 0. 10. ... 12. 1. 0.]]
>>>

```

Figure 1.13: hasil print uji cobat

7. Untuk penjelasan uji cobanya sudah selesai.

- Learning And Predicting

- Penerapan Learning Dan Predicting Pada Python Di CMD

1. Pertama-tama silahkan buka command prompt di laptop anda
2. Selanjutnya masuk ke python
3. Setelah masuk kedalam python, silahkan masukkan perintah ( script-snya ) sesuai dengan contoh yang akan diberikan

```
In [40]: import pandas as pd
...: solok = pd.read_csv('student-mat.csv', sep=';')
...: len(solok)
Out[40]: 395
```

Figure 1.14: pengujian learning dan predicting

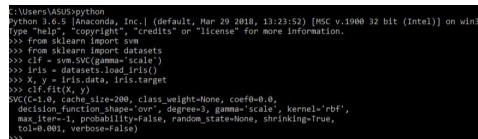
4. Contohnya nampak seperti pada gambar.
  5. Apabila tampilanya telah nampak seperti gambar diatas, maka pengujiannya telah selesai dan berhasil.
- Penjelasan Perintah Yang Di Uji, ( sesuai dengan contoh perintah pada gambar).
  - 1. Penjelasan untuk baris 1 ialah :  
Memanggil dan memasukkan datasets dari sklearn
  - 2. Penjelasan untuk baris 2 ialah :  
membuat variabel iris yang memanggil load data pada datasets tanpa parameter
  - 3. Penjelasan untuk baris 3 ialah :  
membuat variabel digits yang memanggil load digits dari datasets tanpa parameter
  - 4. Penjelasan untuk baris 4 ialah :  
melakukan perintah print data yang akan menampilkan data dari eksekusi variabel digits
  - 5. Penjelasan untuk baris 5 ialah :  
hasil eksekusi
  - 6. Penjelasan untuk baris 6 ialah :  
membuat clf pada module fit metode dengan menggunakan 2 parameter yaitu digits data dan digits target

7. Penjelasan untuk baris 7 ialah :  
svc ini mengimplementasikan yang namanya data berupa klasifikasi dukungan vektor.
8. Untuk penjelasan uji cobanya sudah selesai.

- Model Persistence

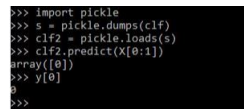
- Penerapan Model Persistence Pada Python Di CMD

1. Pertama-tama silahkan buka command prompt di laptop anda
2. Selanjutnya masuk ke python
3. Setelah masuk kedalam python, silahkan masukkan perintah ( script-snya ) sesuai dengan contoh yang akan diberikan



```
Python 3.6.5 [AMD64, Intel(R) Core(TM) i7-4790K CPU @ 3.60GHz] on win32
>>> from sklearn import svm
>>> from sklearn import datasets
>>> clf = svm.SVC(gamma='scale')
>>> iris = datasets.load_iris()
>>> X, y = iris.data, iris.target
>>> clf.fit(X, y)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
 decision_function_shape='ov', degree=3, gamma='scale', kernel='rbf',
 max_iter=1, probability=False, random_state=None, shrinking=True,
 tol=0.001, verbose=False)
>>>
```

Figure 1.15: pengujian model persistence 1



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

Figure 1.16: pengujian model persistence 2

4. Contohnya nampak seperti pada gambar.
  5. Apabila tampilanya telah nampak seperti gambar diatas, maka pengujiannya telah selesai dan berhasil.
- Penjelasan Perintah Yang Di Uji, ( sesuai dengan contoh perintah pada gambar).
  - Model Persistence 1

1. Penjelasan untuk baris 1 ialah :  
Memanggil ataupun memasukkan datasets dari sklear.
2. Penjelasan untuk baris 2 ialah :  
Memanggil ataupun memasukkan svm dari sklearn
3. Penjelasan untuk baris 3 ialah :  
Membuat variabel baru yaitu clf dimana akan memanggil svm.SVC yang telah mendefinisikan sebuah parameter yaitu gamma.

4. Penjelasan untuk baris 4 ialah :  
Membuat variabel baru lainnya yaitu iris dimana akan memanggil datasets yang didalamnya akan ngeload data iris.
  5. Penjelasan untuk baris 5 ialah :  
Membuat variabel baru lainnya untuk X dan Y dengan mendefinisikan pemanggilan iris data dan iris target.
  6. Penjelasan untuk baris 6 ialah :  
Variabel clf dipasang pada model fit metode dengan parameter X dan Y
  7. Penjelasan untuk baris 7 ialah :  
Kemudian svc ini mengimplementasikan yang namanya data berupa klasifikasi dukungan vektor.
- Model Persistence 2
1. Penjelasan untuk baris 1 ialah :  
Melakukan pemanggilan terhadap library pickle
  2. Penjelasan untuk baris 2 ialah :  
Membuat variabel baru yaitu s dengan pemanggilan pickle dumps dengan pendefinisian variabel clf
  3. Penjelasan untuk baris 3 ialah :  
Membuat variabel clf2 dengan memanggil pickle loads dengan pendefinisian variabel s
  4. Penjelasan untuk baris 4 ialah :  
Prediksi nilai baru dari variabel clf2 dengan parameternya yaitu X
  5. Penjelasan untuk baris 5 ialah :  
set array dari prediksi variabel clf2
  6. Penjelasan untuk baris 6 ialah :  
Parameter X dengan array 0 akan menghasilkan set array 0 juga
  7. Untuk penjelasan uji cobanya sudah selesai.

- Conventions

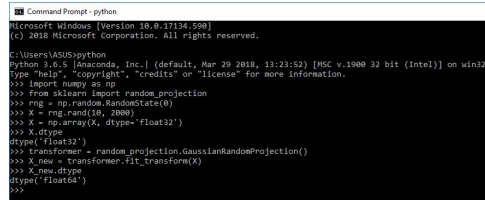
- Penerapan Conventions Pada Python Di CMD

1. Type Casting

- \* Pertama-tama silahkan buka command prompt di laptop anda
- \* Selanjutnya masuk ke python



- \* Setelah masuk kedalam python, silahkan masukkan perintah ( scriptsnya ) sesuai dengan contoh yang akan diberikan



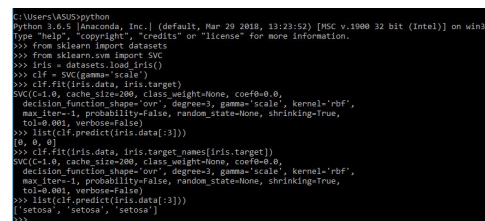
```

Microsoft Windows [Version 10.0.17134.500]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>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.
>>> import numpy as np
>>> from sklearn import random_projection
>>> rng = np.random.RandomState(0)
>>> X = rng.rand(10, 2000)
>>> X = np.array(X, dtype='float32')
>>> X.dtype
dtype('float32')
>>> transformer = random_projection.GaussianRandomProjection()
>>> X_new = transformer.fit_transform(X)
>>> X_new.dtype
dtype('float64')
>>>

```

Figure 1.17: pengujian type casting 1



```

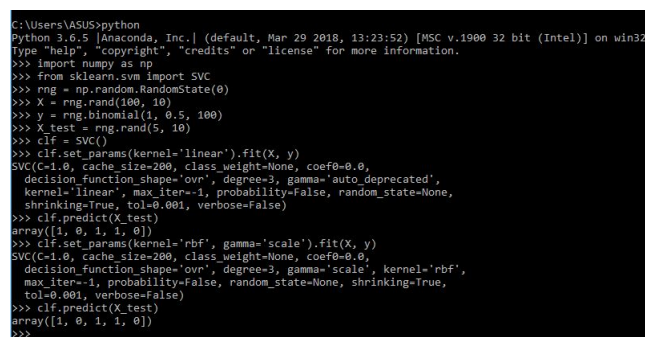
C:\Users\ASUS>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from sklearn import datasets
>>> from sklearn.svm import SVC
>>> iris = datasets.load_iris()
>>> clf = 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: pengujian type casting 2

## 2. Refitting And Updating Parameters

- \* Pertama-tama silahkan buka command prompt di laptop anda
- \* Selanjutnya masuk ke python
- \* Setelah masuk kedalam python, silahkan masukkan perintah ( scriptsnya ) sesuai dengan contoh yang akan diberikan



```

C:\Users\ASUS>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.
>>> 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: pengujian refitting and updating

## 3. Multiclass And Multilable Fitting

- \* Pertama-tama silahkan buka command prompt di laptop anda
- \* Selanjutnya masuk ke python

```

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

```

Figure 1.20: pengujian multiclass and multiable 1

```

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

```

Figure 1.21: pengujian multiclass and multiable 2

- \* Setelah masuk kedalam python, silahkan masukkan perintah ( scriptsnya ) sesuai dengan contoh yang akan diberikan
- 4. Apabila semua proses yang telah dilakukan terlihat seperti contoh-contoh diatas, maka pengujian telah selesai.
- Penjelasan Perintah Yang Di Uji, berdasarkan contoh perintah-perintah diatas :
  1. Type Casting :
    - \* Type Casting 1
      - Penjelasan untuk baris 1 ialah :  
Memasukkan dan memanggil module / library numpy sebagai np
      - Penjelasan untuk baris 2 ialah :  
Memasukkan dan memanggil random projection dari sklearn
      - Penjelasan untuk baris 3 ialah :  
Membuat variabel rng, dimana memanggil np yang akan mengambil dan mengeksekusi random state dengan parameter 0
      - Penjelasan untuk baris 4 ialah :  
Membuat variabel baru lainnya yaitu X dengan memanggil variabel rng dengan 2 parameter yaitu 10 dan 2000
      - Penjelasan untuk baris 5 ialah :  
Membuat variabel X lagi namun dengan pemanggilan yang berbeda yaitu array dari np dengan parameternya x dan dtype='float32'.

- Penjelasan untuk baris 6 ialah :  
Pemanggilan dtype dari variabel X
- Penjelasan untuk baris 7 ialah :  
Hasil dari pemanggilan dtype dari variabel X
- Penjelasan untuk baris 8 ialah :  
Membuat variabel transformer dengan pemanggilan gaussian random projection
- Penjelasan untuk baris 9 ialah :  
Membuat variabel baru yaitu X new dengan memanggil variabel transformer yang berada pada model fit metode dengan parameternya yaitu X
- Penjelasan untuk baris 10 ialah :  
Pemanggilan dtype dari variabel X new
- Penjelasan untuk baris 11 ialah :  
Hasil dari pemanggilan dtype variabel X new
- Untuk penjelasan uji cobanya sudah selesai.

#### \* Type Casting 2

- Penjelasan untuk baris 1 ialah :  
Memasukkan dan memanggil dataset dari sklearn
- Penjelasan untuk baris 2 ialah :  
Memasukkan dan memanggil scv dari sklearn
- Penjelasan untuk baris 3 ialah :  
Membuat variabel iris dengan memanggil load iris dari datasets
- Penjelasan untuk baris 4 ialah :  
Membuat variabel clf dengan memanggil svc dengan parameter gamma
- Penjelasan untuk baris 5 ialah :  
Membuat clf pada module fit metode dengan 2 parameter yaitu iris data dan iris target
- Penjelasan untuk baris 6 ialah :  
membuat variabel list dengan prediksi clf
- Penjelasan untuk baris 7 ialah :  
SVC mengimplementasikan yang namanya data berupa klasifikasi dukungan vektor.

- Penjelasan untuk baris 8 ialah :  
membuat variabel list dengan prediksi iris data
- Penjelasan untuk baris 9 ialah :  
hasil dari prediksi list clf
- Untuk penjelasan uji cobanya sudah selesai.

\* Refting And Updating Parameters :

- Penjelasan untuk baris 1 ialah :  
Memasukkan dan memanggil module / library numphy sebagai np
- Penjelasan untuk baris 2 ialah :  
Memasukkan dan memanggil SVC dari sklearn.svm
- Penjelasan untuk baris 3 ialah :  
Membuat variabel rng, dimana memanggil np yang akan mengam-  
bil dan mengeksekusi random state dengan parameter 0
- Penjelasan untuk baris 4 ialah :  
Membuat variabel baru lainnya yaitu X dengan memanggil vari-  
abel rng dengan 2 parameter yaitu 100 dan 10
- Penjelasan untuk baris 5 ialah :  
Membuat variabel Y dimana memanggil binomal dari rng dengan  
3 parameter yaitu 1, 0.5 dan 100.
- Penjelasan untuk baris 6 ialah :  
Memuat variabel baru lainnya itu X test dimana memanggil rand  
dari rng dengan 2 parameter yaitu 5 dan 10.
- Penjelasan untuk baris 7 ialah :  
Membuat variabel clf dengan mendefinisikan SVC tanpa parame-  
ter
- Penjelasan untuk baris 8 ialah :  
Melakukan parameter set dari clf dengan parameter kernel='linear'.
- Penjelasan untuk baris 9 ialah :  
SVC mengimplementasikan yang namanya data berupa klasifikasi  
dukungan vektor.
- Penjelasan untuk baris 10 ialah :  
Membuat prediksi clf dengan parameternya yaitu variabel X test
- Penjelasan untuk baris 11 ialah :  
set array yang dihasilkan oleh prediksi clf

- Penjelasan untuk baris 12 ialah :  
Melakukan parameter set dari clf dengan parameter kernel='rbf', gamma='scale' dan fit yaitu X dan Y
- Penjelasan untuk baris 13 ialah :  
SVC mengimplementasikan yang namanya data berupa klasifikasi dukungan vektor.
- Penjelasan untuk baris 14 ialah :  
Membuat prediksi clf dengan parameternya yaitu variabel X test
- Penjelasan untuk baris 15 ialah :  
set array yang dihasilkan oleh prediksi clf
- Untuk penjelasan uji cobanya sudah selesai.

\* Multiclass And Multilable Fitting

- Multiclass And Multilable Fitting 1
  1. Penjelasan untuk baris 1 ialah :  
Memasukkan dan memanggil SCV dari sklearn.svm
  2. Penjelasan untuk baris 2 ialah :  
Memasukkan dan memanggil OneVsRestClassifier dari sklearn.svm
  3. Penjelasan untuk baris 3 ialah :  
Memasukkan dan memanggil LabelBinarizer dari sklearn.preprocessing
  4. Penjelasan untuk baris 4 ialah :  
Membuat variabel X dengan beberapa parameter
  5. Penjelasan untuk baris 5 ialah :  
Membuat variabel Y dengan beberapa parameter
  6. Penjelasan untuk baris 6 ialah :  
Membuat variabel classif dengan memanggil OneVsRestClassifier yang didalamnya terdapat 2 parameter yaitu gamma dan random state.
  7. Penjelasan untuk baris 7 ialah :  
Membuat prediksi parameter X dari variabel classif yang berada dalam module fit metode dengan parameter X dan Y
  8. Penjelasan untuk baris 8 ialah :  
set array dari prekdiksi calssif
  9. Penjelasan untuk baris 9 ialah :

Membuat variabel Y baru dengan memanggil LabeBinazier tanpa parameter dan fit transform dengan parameter Y

10. Penjelasan untuk baris 10 ialah :

Membuat prediksi parameter X dari variabel classif yang berada dalam module fit metode dengan parameter X dan Y

11. Penjelasan untuk baris 11 ialah :

set array dari prekdiksi calssif

· Multiclass And Multilable Fitting 2

1. Penjelasan untuk baris 1 ialah :

Memasukkan dan memanggil MultiLabelBinarizer dari sklearn.preprocessing

2. Penjelasan untuk baris 2 ialah :

Membuat variabel Y dengan beberapa parameter / nilai

3. Penjelasan untuk baris 3 ialah :

Membuat variabel Y dengan memanggil MultiLLabeLBinarizer tanpa parameter dan Fit transform dengan parameter y

4. Penjelasan untuk baris 4 ialah :

Membuat prediksi dengan parameter X dari classif pada module fit metode dengan parameter X dan Y

5. Penjelasan untuk baris 5 ialah :

set array dari prediksi classif

6. Untuk penjelasan uji cobanya sudah selesai.

### 1.4.3 Penanganan Error

Terdapat error pada pengujian diatas dan penanganannya, yaitu:

1. Model Persistence

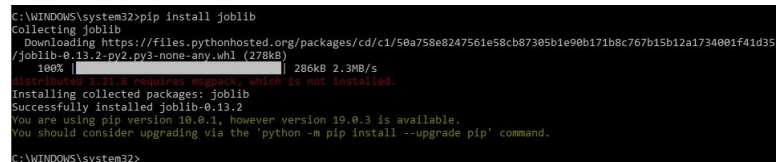
- Errornya ditandai dengan tidak terdefinisinya module joblib pada komputer
- Hal itulah yang menyebabkan tidak terprosesnya perintah terkait

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

Figure 1.22: error model persistence

## 2. Penanganan Model Persistence

- Pertama-tama silahkan membuka command prompt
- Kemudian masukkan perintah untuk melakukan instalasi module joblib  
perintahnya ialah : `pip install joblib`
- hasilnya akan nampak seperti pada gambar yang ditampilkan

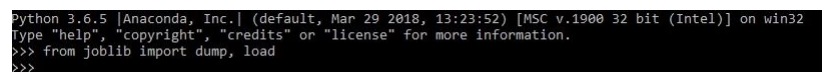


```
C:\WINDOWS\system32>pip install joblib
Collecting joblib
 Downloading https://files.pythonhosted.org/packages/cd/c1/50a758e8247561e58cb87305b1e90b171b8c767b15b12a1734001f41d350
/joblib-0.13.2-py2.py3-none-any.whl (278kB)
 100% |#####| 286kB 2.3MB/s
Installing collected packages: joblib
Successfully installed joblib-0.13.2
You are using pip version 10.0.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\WINDOWS\system32>
```

Figure 1.23: penanganan error model persistence

## 1. Pengujian Penanganan Model Persistence

- Setelah melakukan penginstalan maka kita harus menguji keberhasilan penginstalan
- Caranya dengan mengecek lewat command prompt bahwa module joblibnya telah terdefiniskan di python
- Silahkan ketikkan perintah python, lalu masukkan perintah sebagai berikut :  
`from joblib, import dump, load`
- Maka hasilnya akan nampak seperti pada gambar yang ditampilkan



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

Figure 1.24: penanganan error model persistence

# 1.5 Rahmi Roza/1164085

## 1.5.1 Teori

Teori mencakup resume dari beberapa pembahasan. yaitu :

- Definisi Kecerdasan Buatan.

Kecerdasan Buatan adalah salah satu cabang Ilmu pengetahuan berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi. Hal Ini biasanya dilakukan dengan mengikuti/mencontoh karakteristik dan analogi berpikir dari kecerdasan/Inteligensia manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer.

Agar komputer bisa bertindak seperti dan sebaik manusia, maka komputer juga harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Untuk itu AI akan mencoba untuk memberikan beberapa metoda untuk membekali komputer dengan kedua komponen tersebut agar komputer bisa menjadi mesin pintar.

#### – Sejarah Kecerdasan Buatan

Banyak orang percaya kecerdasan buatan akan memusnahkan kelangsungan hidup manusia saat mereka menyadari kekuatan yang dimilikinya. Semua bermula sejak Turing Machine. Berikut perjalanan kecerdasan buatan hingga akhirnya melahirkan robot dan robot seks.

Tahun 1950. Alan Turing memperkenalkan Turing Test dalam jurnal berjudul *Computing Machinery and Intelligence*. Pada musim panas 1956, Konferensi Dartmouth meluncurkan ide artificial intelligence dan IBM memulai riset tentang AI.

Tahun 1970. Sepanjang 1974 sampai 1980 merupakan gelombang pertama kecerdasan buatan. Pada periode ini pula pengumpulan dana untuk melakukan riset kecerdasan buatan mulai marak.

Tahun 1997. Pada 11 Mei 1997, Deep Blue Computer, kecerdasan buatan buatan IBM berhasil mengalahkan grand master catur asal Rusia, Garry Kasparov

Tahun 2000. Kendaraan buatan tim peneliti dari Universitas Stanford, Amerika Serikat, berhasil menjadi juara dalam DARPA Grand Challenge. Mobil swakemudi ini bisa melaju di gurun pasir sejauh 211 kilometer.

Tahun 2010. Watson, kecerdasan buatan buatan IBM, berhasil mengalahkan mantan juara Brad Rutter dan Ken Jennings dalam acara kuis Jeopardy pada pertengahan 2011. Apple, pada 14 Oktober tahun yang sama, memperkenalkan asistem pribadi berbasis kecerdasan buatan bernama Siri dalam iPhone 4s. Setahun kemudian, tepatnya Juni, tim dari Google Brain melatih komputer agar bisa mengenali seekor kucing



dari jutaan video di YouTube.ChatBot bikin Eugene Goostman mengklaim telah memecahkan tes Turing dalam kompetisi yang digelar di Universitas Reading, Inggris. Imbasnya, pada Agustus tahun yang sama, banyak ilmuwan mengusulkan untuk membuat tes Turing yang baru. Sementara itu, terkesan dengan kemampuan Watson, NASA menggunakannya untuk penelitian bidang kedirgantaraan.

Tahun 2017. Google, Februari lalu, kembali membuat gebrakan soal kecerdasan buatan. Tim dari Google Deep Mind mengungkapkan kecerdasan buatan memiliki tingkat emosi dan kemarahan yang sama dengan manusia. Kecerdasan buatan pun bisa merasakan kalau dirinya ditipu. Tak mau kalah, Microsoft, April lalu, mendeteksi bahwa artificial intelligence ternyata juga bisa rasis. Yang paling fenomenal soal kecerdasan buatan pada tahun ini adalah robot seks bernama Harmony. Tak seperti robot seks pada umumnya, Harmony bisa merasakan cemburu dan mendeteksi penggunaanya saat hendak menuju klimaks.

Jika kita berbicara tentang AI atau Artificial Intelligence maka kita tidak bisa melupakan seorang sosok yang sangat terkenal pada bidang tersebut yaitu bapak John McCarthy. McCarthy mendapatkan gelar sarjana matematika dari California Institute of Technology (Caltech) pada September 1948. Dari masa kuliahnya itulah ia mulai mengembangkan ketertarikannya pada mesin yang dapat menirukan cara berpikir manusia. McCarthy kemudian melanjutkan pendidikan ke program doctoral di Princeton University.

McCarthy kemudian mendirikan dua lembaga penelitian kecerdasan buatan. Kedua lembaga AI itu adalah Stanford Artificial Intelligence Laboratory dan MIT Artificial Intelligence Laboratory. Di lembaga-lembaga inilah bermunculan inovasi pengembangan AI yang meliputi bidang human skill, vision, listening, reasoning dan movement of limbs. Bahkan Salah satu lembaga yang didirikan itu, Stanford Artificial Intelligence pernah mendapat bantuan dana dari Pentagon untuk membuat teknologi-teknologi luar angkasa.

#### – Perkembangan Kecerdasan Buatan

Perkembangan kecerdasan buatan atau artificial intelligence (AI) dinilai tidak bisa dihentikan. Keberadaan AI justru dinilai akan semakin mempermudah kehidupan manusia dalam beraktivitas. Dalam pandangan Johnny

Lie sebagai Vice President of Cheetah Mobile, revolusi teknologi tidak bisa dihentikan. Menurutnya, hal itu sudah terjadi sejak puluhan tahun yang lalu. Cheetah Mobile sendiri merupakan perusahaan teknologi mobile yang fokus pada peranti lunak dan pada hari ini menyematkan unsur AI dalam layanannya.

Sebelumnya, beberapa pakar teknologi kenamaan dunia, seperti Elon Musk dan Stephen Hawking gencar memberikan peringatan bahwa AI yang terus berkembang dapat menjadi ancaman bagi umat manusia. Bahkan, keduanya bersama banyak ilmuwan lainnya membuat surat penegasan kepada PBB untuk mengawasi pertumbuhan AI.

Dengan AI, sebuah produk teknologi dapat bekerja lebih cerdas dan mandiri. Misalnya saja, sebuah aplikasi smartphone yang menggunakan AI dapat mempelajari kebiasaan pengguna. Alhasil, Anda tidak perlu repot lagi dalam melakukan suatu pekerjaan di ponsel.

- Tentang Pengertian Terhadap Ilmu Yang Lain

- Supervised learning adalah sebuah pendekatan dimana sudah terdapat data yang dilatih, dan terdapat variable yang ditargetkan sehingga tujuan dari pendekatan ini adalah mengelompokkan suatu data ke data yang sudah ada
- Klasifikasi adalah pembagian sesuatu menurut kelas-kelas ( class ). Menurut Ilmu Pengetahuan, Klasifikasi merupakan proses pengelompokkan benda berdasarkan ciri-ciri persamaan dan juga perbedaan.
- Regresi adalah metode analisis statistik yang digunakan untuk melihat pengaruh antara dua ataupun lebih variabel.
- Unsupervised Learning berbeda dengan Supervised Learning. Perbedaannya ialah unsupervised learning tidak memiliki data latih, sehingga dari data yang ada kita mengelompokkan data tersebut menjadi 2 ataupun 3 bagian dan seterusnya.
- Dataset adalah objek yang merepresentasikan data dan juga relasi yang ada di memory.
- Training Set adalah set digunakan oleh algoritma klasifikasi . Dapat dicontohkan dengan : decision tree, bayesian, neural network dll. Semuanya dapat digunakan untuk membentuk sebuah model classifier.

- Testing Set adalah set yang digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar.

## 1.5.2 Instalasi



Figure 1.25: gambar1

1. Baris 1 = memasukkan dan memanggil svm dari sklearn
2. Baris 2 = membuat variable



Figure 1.26: gambar2

1. Baris 1 = clf dipasang pada model fit metode
  2. Baris 2 = implementasikan klasifikasi dukungan vektor
- 
1. Baris 1 = prediksi nilai baru



Figure 1.27: gambar3



Figure 1.28: gambar4

2. Baris 2 = set array
1. Baris 1 = memasukkan dan memanggil datasets dari sklearn
2. Baris 2 = memasukkan dan memanggil svm dari sklearn
3. Baris 3 = membuat variable clf
4. Baris 4 = membuat variable iris
5. Baris 5 = membuat variable x, y
6. Baris 6 = clf dipasang pada model fit metode
7. Baris 7 = implementasikan klasifikasi dukungan vektor
8. Baris 8 = memanggil library pickle
9. Baris 9 = membuat variable s
10. Baris 10 = membuat variable clf2
11. Baris 11 = prediksi nilai baru
12. Baris 12 = set array
1. Baris 1 = memasukkan dan memanggil numpy sebagai np



Figure 1.29: gambar5

2. Baris 2 = memasukkan dan memanggil random projection dari sklearn
3. Baris 3 = membuat variable rng
4. Baris 4 = membuat variable rng
5. Baris 5 = membuat variable x
6. Baris 6 = pemanggilan variable x
7. Baris 7 = pemanggilan dtype
8. Baris 8 = membuat variable tranformer
9. Baris 9 = membuat variable x new
10. Baris 10 = pemanggilan x new
11. Baris 11 = pemanggilan dtype

### 1.5.3 Penanganan Error

- Skrintut Error.
- Tuliskan kode error dan jenis errornya
  - Kode error = NameError: name 'roza' is not defined
  - jenis error = NameError
- Solusi pemecahan masalah error



Figure 1.30: gambar6



Figure 1.31: gambar7

## 1.6 Lusita Violita Aprilian/1164080

### 1.6.1 Artificial Intelligence

#### 1. Pengertian AI

Menurut Minsky, Kecerdasan Buatan ialah suatu ilmu yang mempelajari cara membuat komputer melakukan sesuatu seperti yang dilakukan oleh manusia. Lalu menurut Ensiklopedi Britannica, Kecerdasan Buatan ialah cabang ilmu komputer yang merepresentasi pengetahuan lebih banyak menggunakan symbol-simbol daripada bilangan, dan memproses informasi berdasarkan metode heuristik atau berdasarkan jumlah aturan.

Berdasarkan beberapa teori tentang kecerdasan buatan diatas, penulis menyimpulkan bahwa kecerdasan buatan adalah suatu ilmu yang membuat sebuah mesin menjadi cerdas, sehingga kecerdasan mesin tersebut mirip dengan kecer-

dasar manusia serta dapat mengambil keputusan sendiri untuk menyelesaikan sebuah masalah.

## 2. Sejarah dan Perkembangan

- 1941 ( Era Komputer Elektronik )

Pada era ini, telah ditemukan pertama kali yakni alat penyimpanan dan pemrosesan informasi atau disebut komputer elektronik. Ini juga digunakan untuk dasar pengembangan program ke arah AI.

- 1943 – 1956 ( Era Persiapan AI )

Pada tahun 1943, terdapat dua peneliti yakni Warren McCulloch dan Walter Pitts yang berhasil membuat sebuah model tiruan dari tiap neuron seperti on dan off. Mereka membuktikan bahwa setiap fungsi dapat dihitung dengan suatu jaringan sel saraf dan semua hubungan logis bisa diimplementasikan dengan struktur jaringan yang sederhana.

Pada tahun 1950, Norbert Wiener melakukan penelitian tentang prinsip teori feedback. Bentuk implementasi dari penelitian tersebut salah satunya adalah thermostat.

Pada tahun 1956, John McCarthy mencoba meyakinkan Minsky, Claude Shannon, dan Nathaniel Rochester untuk membantunya dalam melakukan penelitian di bidang automata, jaringan saraf, dan pembelajaran intelijen. Mereka mengerjakan proyek ini kurang lebih selama 2 bulan di Universitas Dartmouth. Hasilnya adalah berupa program yang mampu berpikir non-numerik dan menyelesaikan masalah pemikiran, yang disebut Principia Mathematica. Berdasarkan hal ini, telah ditentukan bahwa McCarthy disebut sebagai father of Artificial Intelligence/ Bapak Kecerdasan Buatan.

- 1952 – 1969 ( Awal Perkembangan )

Pada tahun 1958, McCarthy di MIT AI Lab mengeluarkan bahasa pemrograman tingkat tinggi yaitu LISP, dimana sekarang sudah mulai sering digunakan dalam pembuatan program-program AI. Lalu, McCarthy membuat program yang disebut programs with common sense. Di program tersebut, dibuat sebuah rancangan untuk menggunakan pengetahuan dalam mencari solusi dari sebuah masalah.

Pada tahun 1959, Program komputer bernama General Problem Solver

berhasil dibuat oleh Herbert A. Simon, J.C. Shaw, dan Allen Newell. Program tersebut dirancang untuk memulai proses penyelesaian masalah secara manusiawi. Pada tahun yg sama Nathaniel Rochester dari IBM dan para mahasiswanya merilis sebuah program AI yaitu geometry theorem prover. Program ini dapat membuktikan bahwa suatu teorema menggunakan aksioma-aksioma yang ada.

Pada tahun 1963, program yang dibuat oleh James Slagle bisa menyelesaikan masalah integral tertutup untuk mata kuliah Kalkulus.

Pada tahun 1968, program analogi buatan Tom Evan dapat menyelesaikan masalah analogi geometri yang ada pada tes IQ.

- 1966 – 1974 ( Perkembangan AI Lambat )

Perkembangan AI mulai melambat pada tahun 1966 – 1974, yang disebabkan adanya beberapa kesulitan yang di hadapi seperti Program-program AI yang bermunculan hanya mengandung sedikit atau bahkan tidak mengandung sama sekali pengetahuan pada subjeknya, banyak terjadi kegagalan pada pembuatan program AI, serta terdapat beberapa batasan pada struktur dasar yang digunakan untuk menghasilkan perilaku inteligensia.

- 1969 – 1979 (Sistem berbasis pengetahuan )

Pada tahun 1960, Ed Feigenbaum, Bruce Buchanan, dan Joshua Lederberg mulai merintis proyek bernama DENDRAL yaitu program yang digunakan untuk memecahkan masalah struktur molekul dari informasi yang didapatkan dari spectrometer massa. Dari segi diagnosa medis juga terdapat yang menemukan sistem berbasis Ilmu pengetahuan, yaitu Saul Amarel dalam proyek computer ini biomedicine. Proyek ini diawali dari keinginan untuk mendapatkan diagnosa penyakit berdasarkan pengetahuan yang ada pada mekanisme penyebab proses penyakit.

- 1980 – 1988 ( AI menjadi Industri )

Industralisasi AI diawali dengan ditemukannya sebuah sistem pakar yang dinamakan R1 yang mampu mengkonfigurasi sistem-sistem komputer baru. Program tersebut mulai dijalankan di Digital Equipment Corporation (DEC), McDermott, pada tahun 1982. Pada tahun 1986, program ini telah berhasil menghemat biaya sebesar US 40 juta per tahun.

Pada tahun 1988, kelompok AI di DEC menjalankan program sistem pakar sebanyak 40 sistem pakar. Hampir semua perusahaan besar di USA mempunyai divisi Ai sendiri yang menggunakan maupun mempelajari sistem



pakar itu sendiri. Industri AI yang sedang ramai diperbincangkan juga melibatkan perusahaan-perusahaan besar seperti Carnegie Group, Inference, IntelliCorp, dan Technoledge yang menawarkan software tools untuk membangun sistem pakar. Perusahaan bidang hardware seperti LISP Machines Inc., Texas Instruments, Symbolics, dan Xerox dan lain-lain juga ikut berperan dalam membangun sebuah workstation yang dioptimasi untuk pembangunan program LISP. Sehingga, perusahaan yang sudah berdiri sejak tahun 1982 hanya menghasilkan beberapa juta US dollar per tahun meningkat menjadi 2 milyar US dollar per tahun pada tahun 1988.

- 1986 – sekarang ( kembalinya jaringan saraf tiruan ) Meskipun bidang ilmu komputer, telah melakukan penolakan terhadap jaringan saraf tiruan setelah diterbitkannya sebuah buku berjudul ‘Perceptrons’ karangan Minsky dan Papert, tetapi para ilmuwan masih terus mempelajari bidang ilmu tersebut dari sudut pandang yang lain, yaitu bidang fisika. Ahli fisika seperti Hopfield (1982) menggunakan teknik-teknik mekanika statistika untuk menganalisa sifat-sifat penyimpanan dan optimasi pada jaringan saraf. ahli psikolog, David Rumelhart dan Geoff Hinton melanjutkan penelitian tersebut tentang model jaringan saraf pada memori. Pada tahun 1985-an sedikitnya empat kelompok riset menemukan algoritma Back-Propagation. Algoritma inipun akhirnya berhasil diimplementasikan ke dalam ilmu bidang komputer dan psikologi.

### 1.6.2 Supervised Learning dan Data

#### 1. Supervised Learning

Sebuah pendekatan dengan kondisi dimana sudah ada terdapat kumpulan data yang dilatih atau ditraining, dan terdapat beberapa variabel yang sudah ditentukan sehingga tujuan dari pendekatan tersebut mengarah ke data yang sudah ada.

#### 2. Klasifikasi

Sebuah sejenis program yang dapat menentukan objek yang ada termasuk jenis apa berdasarkan variabel-variabel yang sudah ditentukan.

#### 3. Regresi

Sebuah metode yang digunakan untuk menentukan dan memprediksi berdasarkan hubungan sebab – akibat antara satu variabel ke variabel lainnya.

#### 4. Unsupervised Learning

Sebuah pendekatan yang dimana tidak memiliki data yang dilatih, namun ingin di kelompokkan berdasarkan beberapa variabel dengan kemauan sendiri.

#### 5. Data Set

Objek yang merepresentasikan data dan relasi didalam memori.

#### 6. Training Set

Himpunan dari berbagai pasangan objek, kelas yang dapat menunjukkan objek tersebut yang sudah diberi label.

#### 7. Testing Set

Himpunan data yang sudah berlabel lain, yang digunakan untuk mengukur persentase sampel yang diklasifikan dengan benar atau persentase sampel mengalami kesalahan.

### 1.6.3 Learning and Predicting

#### 1. Learning and predicting 1

```
from sklearn import svm
clf = svm.SVC(gamma=0.001, C=100.)
```

Figure 1.32: Learning and predicting 1

- Baris 1 = memasukkan dan memanggil svm dari sklearn
- Baris 2 = membuat variable

#### 2. Learning and predicting 2

```
clf.fit(digits.data[:-1], digits.target[:-1])
SVC(C=100.0, cache_size=200, class_weight=None, coef0=0.0,
 decision_function_shape='ovr', degree=3, gamma=0.001, kernel='rbf',
 max_iter=1, probability=False, random_state=None, shrinking=True,
 tol=0.001, verbose=False)
```

Figure 1.33: Learning and predicting 2

- Baris 1 = clf dipasang pada model fit metode
- Baris 2 = implementasikan klasifikasi dukungan vektor

#### 3. Learning and predicting 3

- Baris 1 = prediksi nilai baru
- Baris 2 = set array

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

Figure 1.34: Learning and predicting 3

### 1.6.4 Model Persistence

1. menjelaskan maksud dari tulisan tersebut dan mengartikan per baris.

```
from sklearn import svm
from sklearn import datasets
clf = svm.SVC(gamma='scale')
iris = datasets.load_iris()
X, y = iris.data, iris.target
clf.fit(X, y)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
 decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
 max_iter=-1, probability=False, random_state=None, shrinking=True,
 tol=0.001, verbose=False)

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

Figure 1.35: Model Persistence

- Baris 1 = memasukkan dan memanggil datasets dari sklearn
- Baris 2 = memasukkan dan memanggil svm dari sklearn
- Baris 3 = membuat variable clf
- Baris 4 = membuat variable iris
- Baris 5 = membuat variable x, y
- Baris 6 = clf dipasang pada model fit metode
- Baris 7 = implementasikan klasifikasi dukungan vektor
- Baris 8 = memanggil library pickle
- Baris 9 = membuat variable s
- Baris 10 = membuat variable clf2
- Baris 11 = prediksi nilai baru
- Baris 12 = set array

### 1.6.5 Conventions

1. menjelaskan maksud dari tulisan tersebut dan mengartikan per baris.

- Baris 1 = memasukkan dan memanggil numpy sebagai np

```
import numpy as np
from sklearn import random_projection

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

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

Figure 1.36: Conventions

- Baris 2 = memasukkan dan memanggil random projection dari sklearn
- Baris 3 = membuat variable rng
- Baris 4 = membuat variable rng
- Baris 5 = membuat variable x
- Baris 6 = pemanggilan variable x
- Baris 7 = pemanggilan dtype
- Baris 8 = membuat variable tranformer
- Baris 9 = membuat variable xnew
- Baris 10 = pemanggilan xnew
- Baris 11 = pemanggilan dtype

### 1.6.6 Penanganan Error

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

1. skrinsut error

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

Figure 1.37: skrinsut error

2. Tuliskan kode error dan jenis errornya

- Kode error = NameError: name 'lusia' is not defined
- jenis error = NameError

3. Solusi pemecahan masalah error

```
>>> print('lusia')
lusia
```

Figure 1.38: gb 1

# Chapter 2

## Related Works

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

### 2.1 Fadila/1164072

#### 2.1.1 Teori

Penyelesaian Tugas Harian 3 ( No. 1-7 )

##### 1. Binary Classification Dan Ilustrasi Gambarnya

- Pengertian Binary Classification / Klasifikasi Biner:

Klasifikasi biner atau binomial merupakan tugas untuk mengklasifikasikan elemen-elemen dari himpunan tertentu ke dalam dua kelompok (mem-prediksi kelompok mana yang masing-masing dimiliki) ber-dasarkan aturan klasifikasi.

- Ilustrasi Gambar Binary Classification :

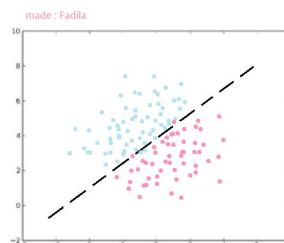


Figure 2.1: binary classification

##### 2. Supervised Learning, Unsupervised Learning, Clustering Dan Ilustrasi Gambar

- Pengertian Supervised Learning :

Sebuah pendekatan dimana terdapat data yang dilatih dan ditargetkan. Lebih singkatnya supervised learning memiliki kategori sehingga tujuan dan outputnya jelas.

- Ilustrasi Gambar Supervised Learning :



Figure 2.2: supervised

Pada contoh gambar diatas dikelompokkan bahwa apabila bentuk dari objek di gambar berbentuk bundar dan berwarna merah maka akan dinamakan atau di sebut sebagai Apel. Dan apabila pada gambar terdapat objek berbentuk panjang dan berwarna kuning maka akan dinamakan atau di sebut sebagai pisang.

- Pengertian Unsupervised Learning :

Tidak memiliki data latih, sehingga dari data yang tersebut kita bisa mengelompokkannya ke berbagai kelompok 2 seterusnya. Dengan lebih singkatnya ialah unsupervised learning tidak memiliki kategori.

- Ilustrasi Gambar Unsupervised Learning :

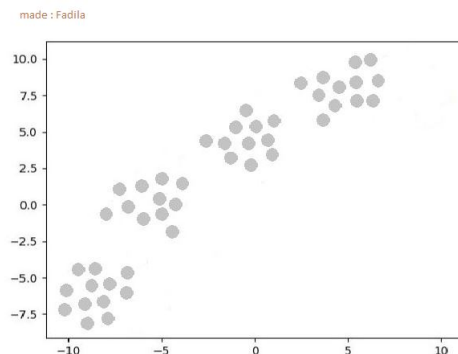


Figure 2.3: unsupervised

Pada gambar dapat dilihat bahwa ada banyak data namun tidak pada pengelompokan yang tepat. Cuman dapat dikelompokkan ke dalam berbagai macam bentuk dan jumlah namun tidak memberikan output yang jelas.

- **Pengertian Clustering :**

Metode pengelompokan data. Clustering juga merupakan proses partisi satu set objek data ke dalam himpunan bagian yang disebut dengan cluster. Objek dalam cluster tersebut memiliki kemiripan karakteristik antar satu sama lain.

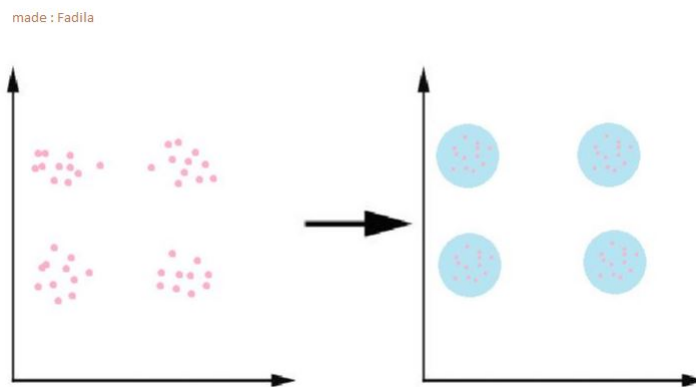


Figure 2.4: clustering

### 3. Evaluasi, Akurasi Dan Ilustrasi Gambar

- **Pengertian Evaluasi**

Evaluasi digunakan untuk memeriksa/memastikan dan mengevaluasi model dalam bekerja ( seberapa baik ) dengan mengukur keakuratannya. Kita juga dapat menanalisis kesalahan yang dibuat pada model yang dijalankan, tingkat kebingungan dan menggunakan matriks kebingungan.

|                | Prediksi "Apel " | Prediksi "Jeruk " |
|----------------|------------------|-------------------|
| Benar "Apel "  | 20               | 5                 |
| Benar "Jeruk " | 3                | 22                |

Figure 2.5: Evaluasi

Pada contoh gambar dapat dilihat bahwa dilakukan evaluasi terhadap kerja dalam penentuan jenis dari objek. Dievaluasi berapa banyak sebuah objek ketika dikelompokkan dan diklasifikasikan kemudian dapat dilihat apakah kerjanya sesuai atau tidak.



- Pengertian Akurasi

Accuracy akan didefinisikan sebagai presentasi kasus yang diklasifikasikan dengan benar. Accuracy lebih jelasnya adalah perbandingan kasus yang diidentifikasi benar dengan jumlah semua kasus

Rumus dari  $accuracy = (a+c)/(a+b+c+d)$

|                                                               |
|---------------------------------------------------------------|
| Made : Fadila                                                 |
| Accuracy / Ketepatan : $((20 + 22) / (20 + 5 + 3 + 22)) = 84$ |

Figure 2.6: Akurasi

Dilakukan perhitungan dengan rumus akurasi terhadap data yang telah diolah pada " Evaluasi ". Kemudian di dapatkan hasil dari pengolahan data tersebut.

Contoh penggabungan Akurasi Dan Evaluasi

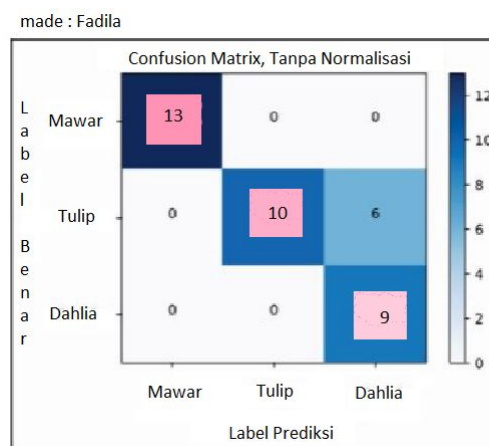


Figure 2.7: Contoh Evaluasi Dan Akurasi Secara Bersamaan

#### 4. Membuat Dan Membaca Confusion Matrix Beserta Contoh

- Pengertian Confusion Matrix

Confusion matrix merupakan suatu metode yang digunakan untuk melakukan perhitungan akurasi pada konsep data mining.

- Pembacaan Confusion Matrix

- Apabila hasil prediksi negatif dan data sebenarnya merupakan negatif.
- Apabila hasil prediksi positif sedangkan nilai sebenarnya merupakan negatif.

- (c) Apabila hasil prediksi negatif sedangkan nilai sebenarnya merupakan positif.
- (d) Apabila hasil prediksi positif dan nilai sebenarnya merupakan positif.
- Pembuatan Confusion Matrix
  - (a) Menentukan 4 proses klasifikasi yang akan digunakan dalam confusion matrix.
  - (b) 4 Istilah tersebut ada True Positive ( TP ), True Negative ( TN ), False Positive ( FP ) dan False Negative ( FN ).
  - (c) Kelompokkan klasifikasi tersebut bisa menggunakan klasifikasi biner
  - (d) Akan menghasilkan keluaran berupa 2 Kelas ( Positif dan Negatif ) dan penentuan TP, FP ( 1 klasifikasi positif ) , FN dan TN ( 1 klasifikasi negatif ).
  - (e) Contoh dasarnya nampak seperti langkah diatas
  - (f) Istilahnya dapat didefinisikan dengan objek lain namun dengan alur yang sama ( sesuai rumus baik klasifikasi dll ).
- Ilustrasi Gambar

| Y | Y Pred | Keluaran Untuk ( threshold ) 0.6 | Recall ( pemanggilan/penarikan ) | Precision ( Presisi ) |
|---|--------|----------------------------------|----------------------------------|-----------------------|
| 0 | 0.5    | 0                                | 1 / 2                            | 4 / 7                 |
| 1 | 0.9    | 1                                |                                  |                       |
| 0 | 0.7    | 1                                |                                  |                       |
| 1 | 0.7    | 1                                |                                  |                       |
| 1 | 0.3    | 0                                |                                  |                       |
| 0 | 0.4    | 0                                |                                  |                       |
| 1 | 0.5    | 0                                |                                  |                       |

Made : Fadila

Figure 2.8: confusion matrix

#### – Penjelasan

##### (a) Recall

Dari semua kelas positif, seberapa banyak yang kami prediksi dengan benar. Itu harus setinggi mungkin.

$$\text{Recall ( Pemanggilan / Penarikan )} = \frac{TP}{TP + FN}$$

# made : Fadila

Figure 2.9: recall

(b) Presisi / Precision

Dari semua kelas, seberapa banyak yang kami prediksi dengan benar. Itu harus setinggi mungkin.

$$\text{Precision / Presisi} = \frac{TP}{TP + FP}$$

# made : Fadila

Figure 2.10: precision

(c) F-Ukur ( measure )

Sulit untuk membandingkan dua model dengan presisi rendah dan daya ingat tinggi atau sebaliknya. Jadi untuk membuatnya sebanding, kami menggunakan F-Score. F-score membantu mengukur Recall dan Precision pada saat yang bersamaan

$$\text{F-Measure ( F-Ukur )} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

# made : Fadila

Figure 2.11: f-measure

## 5. Cara Kerja K-Fold Classification Dan Ilustrasi Gambar

- (a) Pertama-tama untuk total instance dibagi menjadi N bagian.
- (b) Fold ke-1 ( atau pertama ) adalah ketika bagian ke-1 menjadi data uji (testing data) dan sisanya menjadi data latih (training data).
- (c) Hitung akurasi ( berdasarkan porsi data tersebut. Persamaanya sebagai berikut :
  - (sigma) data klasifikasi
  - (sigma) total data uji
  - x 100 persen
- (d) Fold ke-2 ( kedua ) adalah ketika bagian ke-2 menjadi data uji (testing data) dan sisanya menjadi data latih (training data).
- (e) Kemudian dihitunglah akurasi berdasarkan porsi data yang telah ditentukan

- (f) Demikian seterusnya hingga mencapai fold ke-K. Hitung rata-rata akurasi dari K buah akurasi di atas. Rata-rata akurasi ini menjadi akurasi final atau akhir.

- Ilustrasi Gambar

| Kategori | $P(c)$        | $P(w_{ij}   c)$ |                |                |                |                |                |                |                |
|----------|---------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|          |               | hasihnya        | bagus          | pengen         | as             | jelek          | banget         | sih            | waw            |
| Positif  | $\frac{1}{2}$ | $\frac{2}{27}$  | $\frac{2}{27}$ | $\frac{2}{27}$ | $\frac{1}{27}$ | $\frac{1}{27}$ | $\frac{2}{27}$ | $\frac{1}{27}$ | $\frac{4}{27}$ |
| Negatif  | $\frac{1}{2}$ | $\frac{1}{27}$  | $\frac{1}{27}$ | $\frac{2}{27}$ | $\frac{2}{27}$ | $\frac{2}{27}$ | $\frac{2}{27}$ | $\frac{1}{27}$ | $\frac{1}{27}$ |

# made : Fadila

Figure 2.12: k-fold classification 1

| Kategori | $P(c)$        | $P(w_{ij}   c)$ |                |                |                |                |                |                |                |
|----------|---------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|          |               | cakap           | csemang        | maka           | min            | mangrove       | gancempan      | bohong         | Nih            |
| Positif  | $\frac{1}{2}$ | $\frac{2}{27}$  | $\frac{2}{27}$ | $\frac{1}{27}$ | $\frac{1}{27}$ | $\frac{1}{27}$ | $\frac{1}{27}$ | $\frac{1}{27}$ | $\frac{1}{27}$ |
| Negatif  | $\frac{1}{2}$ | $\frac{1}{27}$  | $\frac{1}{27}$ | $\frac{2}{27}$ | $\frac{2}{27}$ | $\frac{2}{27}$ | $\frac{2}{27}$ | $\frac{2}{27}$ | $\frac{2}{27}$ |

# made : Fadila

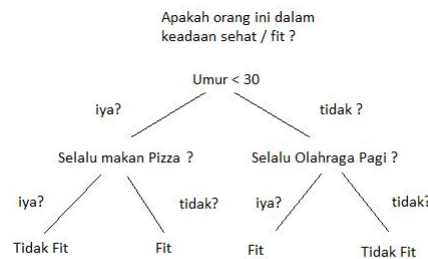
Figure 2.13: k-fold classification 2

## 6. Decision Tree Dan Ilustrasi Gambar

- Pengertian Decision Tree

Decision tree adalah salah satu metode klasifikasi yang paling populer karena mudah diinterpretasikan oleh manusia. Decision tree merupakan metode klasifikasi yang digunakan untuk pengenalan pola dan termasuk dalam pengenalan pola secara statistik. 3 tipe dari decision tree ialah: simpul: simpul root, simpul perantara, dan simpul leaf.

- Ilustrasi Gambar



made : Fadila

Figure 2.14: decision tree

## 7. Information Gain Dan Entropi

- Pengertian Information Gain

Information Gain adalah salah satu attribute selection measure yang digunakan untuk memilih test attribute tiap node pada tree.

Algoritme Information Gain digunakan untuk mengurangi dimensi atribut untuk mendapatkan atribut-atribut yang relevan.

- Ilustrasi Gambar

Tabel 1 Perbandingan Akurasi Kelas Tidak Seimbang dengan 13 fitur dan 6 fitur

| Nilai K | Sebaran Kelas Tidak Seimbang |                            |
|---------|------------------------------|----------------------------|
|         | Akurasi Tanpa Menggunakan    | Akurasi Menggunakan        |
|         | Information Gain (13 fitur)  | Information Gain (6 fitur) |
| 5       | 73,08%                       | 88,46%                     |
| 15      | 80,77%                       | 84,62%                     |
| 25      | 84,62%                       | 88,46%                     |
| 35      | 80,77%                       | 88,46%                     |
| 45      | 80,77%                       | 88,46%                     |
| 55      | 84,62%                       | 84,62%                     |
| 65      | 80,77%                       | 84,62%                     |
| 75      | 84,62%                       | 84,62%                     |
| 85      | 80,77%                       | 84,62%                     |
| 95      | 76,92%                       | 88,46%                     |

Figure 2.15: informaion gain 1

Tabel 2 Perbandingan Akurasi Kelas Seimbang dengan 13 fitur dan 6 fitur

| Nilai K | Sebaran Kelas Seimbang      |                            |
|---------|-----------------------------|----------------------------|
|         | Akurasi Tanpa Menggunakan   | Akurasi Menggunakan        |
|         | Information Gain (13 fitur) | Information Gain (6 fitur) |
| 5       | 61,54%                      | 84,62%                     |
| 15      | 80,77%                      | 84,62%                     |
| 25      | 80,77%                      | 92,31%                     |
| 35      | 80,77%                      | 88,46%                     |
| 45      | 76,92%                      | 84,62%                     |
| 55      | 80,77%                      | 84,62%                     |
| 65      | 80,77%                      | 84,62%                     |
| 75      | 80,77%                      | 84,62%                     |
| 85      | 80,77%                      | 84,62%                     |
| 95      | 80,77%                      | 84,62%                     |

Figure 2.16: information gain 2

- Penjelasan :

Tabel 1 sampai dengan Tabel 2 menunjukkan bahwa penggunaan seleksi fitur Information Gain menghasilkan nilai akurasi yang lebih baik dibandingkan tanpa menggunakan Information Gain.

Pada saat nilai K sama dengan 5 ( K=5) akurasi yang dihasilkan sistem tanpa menggunakan Information Gain menunjukkan hasil yang kurang

baik pada sebaran kelas seimbang maupun tak seimbang yaitu 61,54 persen pada sebaran kelas seimbang dan 73,08 persen pada sebaran kelas tidak seimbang.

- Pengertian Entropi

Entropi pada umumnya merupakan salah satu besaran yang mengukur energi dalam sistem per satuan temperatur yang tak dapat digunakan untuk melakukan usaha.

Namun, secara spesifik untuk " Entropi " sendiri merupakan parameter untuk mengukur tingkat keberagaman (heterogenitas) dari kumpulan data.

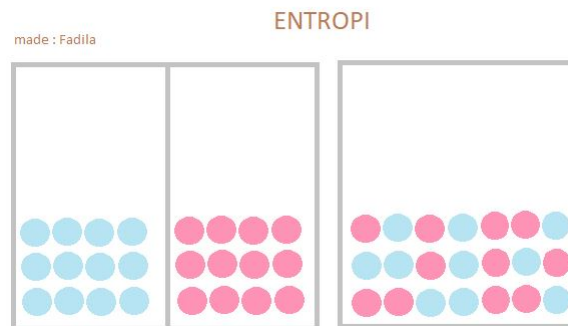


Figure 2.17: entropi

## 2.1.2 Praktek Scikit-Learn

Penyelesaian Tugas Harian 4 ( No. 1-12 )

(a) Pembahasan Codingan Dan Hasilnya

i. Codingan Pertama :

Penjelasan : Codingan pertama ini akan meload ( menampilkan ) data pada file yang ditentukan. Untuk codingan ini file yang dieksekusi ialah " student-mat.csv " . Secara jelasnya, dalam codingan dapat dilihat bahwa variabel bakso didefinisikan untuk pembacaan csv dari " pizza " dimana untuk pemisahannya yaitu separation berupa ; . Setelah itu variabel bakso di "print" dengan perintah menampilkan "len" panjang ataupun jumlah dan hasilnya berupa angka 395 .

- Hasil Codingan Pertama :

```
In [56]: import pandas as pizza
...: bakso = pizza.read_csv('dataset/student-mat.csv', sep=';')
...: len(bakso)
Out[56]: 395
```

Figure 2.18: codingan pertama

ii. Codingan Kedua :

Penjelasan : Codingan kedua ini secara keseluruhan menampilkan baris G1, G2 dan G3 ( berdasarkan kriterianya ) untuk kolom PASS pada variabel bakso. Untuk lebih jelasnya, pada codingan terdapat pendefinisian pembacaan "lamda" ( panjang gelombang ) dari baris G1, G2 dan G3. Apabila row-row tersebut bernilai lebih dari 35 maka akan terdefiniskan angka "1" apabila tidak, maka akan terdefiniskan angka "0" pada kolom PASS ( sesuai permintaan awal ). Selanjutnya variabelnya di "print" sehingga menampilkan keluaran. Tidak lupa terdapat juga jumlah dari baris dan kolom yang berubah sesuai dengan baris yang dieksekusi.

• Hasil Codingan Kedua :

```
In [60]: bakso['pass'] = bakso.apply(lambda row: 1 if (row['G1']+row['G2']
+row['G3']) >= 35 else 0, axis=1)
...: bakso = bakso.drop(['G1', 'G2', 'G3'], axis=1)
...: bakso.head()
Out[60]:
```

|   | 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 2.19: codingan kedua

iii. Codingan Ketiga :

Penjelasan : Secara keseluruhan, codingan ini mendefinisikan pemanggilan get dummies dari pizza dalam variabel bakso. Di dalam get dummies sendiri akan terdefiniskan variabel bakso dengan kolom-kolom yang akan dieksekusi seperti school, address dll. Kemudian variabel tersebut di definisikan untuk mendapatkan kembalian berupa keluaran dari eksekusi perintah variabel bakso beserta dengan jumlah baris dan kolom data yang dieksekusi.

• Hasil Codingan Ketiga :

```

In [61]: bakso = pizza.get_dummies(bakso, columns=['sex', 'school', 'address', 'famsize', 'Pstatus', 'Hjob', 'Fjob',
...: 'reason', 'guardian', 'schoolsup', 'famsup', 'paid', 'activities',
...: 'nursery', 'higher', 'Internet', 'romantic'])
...: bakso.head()
Out[61]:
 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 2.20: codingan ketiga

#### iv. Codingan Keempat :

Penjelasan : Secara keseluruhan codingan ini difungsikan untuk mendefinisikan pembagian data yang berupa training dan testing data. Secara jelasnya pertama-tama variabel bakso akan mendefinisikan sampel yang akan digunakan ( berupa shuffle row ) . Nah kemudian masing2 parameter yaitu bakso train dan bakso test akan berjumlah 500 data ( telah dibagi untuk training dan testing ). Selanjutnya dilakukan pengekseskuan untuk kolom Pass, apabila sesuai dengan "axis=1" maka eksekusi fungsi berhasil. Selain itu juga disertakan jumlah dari peserta yang lolos dari semua nilai data setnya.

- Hasil Codingan Keempat :

```

In [62]: bakso = bakso.sample(frac=1)
...: # split training and testing data
...: bakso_train = bakso[500:]
...: bakso_test = bakso[:500]
...:
...: bakso_train_att = bakso_train.drop(['pass'], axis=1)
...: bakso_train_pass = bakso_train['pass']
...:
...: bakso_test_att = bakso_test.drop(['pass'], axis=1)
...: bakso_test_pass = bakso_test['pass']
...:
...: bakso_att = bakso.drop(['pass'], axis=1)
...: bakso_pass = bakso['pass']
...:
...: # number of passing students in whole dataset:
...: import numpy as np
...: print("Passing: %d out of %d (%.2f%%)" % (np.sum(bakso_pass), len(bakso_pass),
100*float(np.sum(bakso_pass)) / len(bakso_pass)))
Passing: 166 out of 395 (42.03%)

```

Figure 2.21: codingan keempat

#### v. Codingan Kelima :

Penjelasan : Secara keseluruhan, codingan ini hanya membuktikan pengujian dari Klasifikasi Decision Tree yang ada, apakah true atau tidak dan hasilnya true. Apabila dibahas secara lengkap maka pada codingan ini di definisikan library sklearn untuk mengimpor atau menampilkan tree. Variabel sate difungsikan untuk membaca klasifikasi decision tree dari tree itu sendiri dengan 2 parameternya yaitu kriteria="entropy" dan max depth=5. Maka selanjutnya variabel sate akan masuk dan terbaca dalam module fit dengan 2 parameter yaitu bakso trai att dan bakso train pass.

- Hasil Codingan Kelima :



```
In [63]: from sklearn import tree
...: sate = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
...: sate = sate.fit(bakso_train_att, bakso_train_pass)
```

Figure 2.22: codingan kelima

#### vi. Codingan Keenam :

Penjelasan : Codingan ini memberikan gambaran dari klasifikasi decision tree dari pengolahan parameter yang dieksekusi kedalam variabel sate. Tentunya dengan pemanfaatan library graphviz yang telah diimport dan difungsikan.

Untuk gambarnya terlalu besar ketika di Console sehingga hanya sebagian yang bisa di screenshot untuk dijadikan hasil.

- Hasil Codingan Keenam :

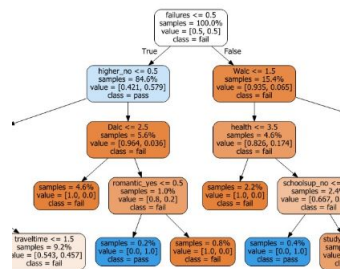


Figure 2.23: codingan keenam

#### vii. Codingan Ketujuh :

Penjelasan : Secara keseluruhan, codingan ini membahas tentang penyimpanan tree dari library graphviz yang dieksekusi bersamaan dengan variabel sate dan parameter lainnya. Dilakukan pengecekan dan pengujian apakah klasifikasi decision treenya dapat berjalan atau tidak. Apabila tidak berjalan, maka akan terjadi error, namun codingan ini berfungsi.

- Hasil Codingan Ketujuh :

```
In [66]: tree.export_graphviz(sate, out_file="student-performance.dot", label="all", impurity=False,
...: proportion=True,
...: feature_names=list(bakso_train_att), class_names=["fail", "pass"],
...: filled=True, rounded=True)
```

Figure 2.24: codingan ketujuh

#### viii. Codingan Kedelapan :

Penjelasan : Secara keseluruhan, codingan ini membaca score dari variabel sate dimana terdapat 2 parameter yang dihitung dan diuji yaitu bakso test att dan bakso test pass. Untuk hasilnya sendiri mengapa berupa angka, dikarenakan pada parameter yang dieksekusi memang memiliki data sehingga dieksekusi dan menghasilkan keluaran dari score tersebut.

- Hasil Codingan Kedelapan :

```
In [33]: sate.score(bakso_test_att, bakso_test_pass)
Out[33]: 0.6308724832214765
```

Figure 2.25: codingan kedelapan

- ix. Codingan Kesembilan :

Penjelasan : Secara keseluruhan, codingan ini membahas mengenai pengkesekusian fungsi dan variabel dari library yang didefinisikan dan yang diimport. Penjelasan lebih jelasnya ialah codingan ini mendefinisikan library sklearn.model.selection kemudian mengimpor cross val score. Kemudian variabel score mendefinisikan cross val score yang telah diimport tadi dengan 4 parameter yaitu sate, bakso att, bakso pass dan cv=5 untuk dieksekusi. Setelah semua pemrosesan tersebut maka hasil yang di "print" ialah rata2 perhitungan dari variabel score dimana dan standar dari (+/-) tentunya dengan ketentuan parameter Accuracy .

- Hasil Codingan Kesembilan :

```
In [69]: from sklearn.model_selection import cross_val_score
...: scores = cross_val_score(sate, bakso_att, bakso_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 2.26: codingan kesembilan

- x. Codingan Ke-10 :

Penjelasan : Codingan ini mendefinisikan max depth dalam jarak angka antara parameter 1 dan 20. Variabel sate mendefinisikan klasi-

fier decision tree dengan 2 parameter. Kemudian variabel score mengeksekusi parameter lainnya yaitu seperti sate, bakso att, bakso pass dan cv=5 ). Hasil yang ditampilkan ialah dari max depth, accuracy dan (+/-) dan akhirnya hasilnya nampak seperti pada gambar.

- Hasil Codingan Ke-10 :

```
In [70]: for max_depth in range(1, 20):
...: sate = tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
...: scores = cross_val_score(sate, bakso_att, bakso_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 (+/- 0.08)
Max depth: 2, Accuracy: 0.58 (+/- 0.10)
Max depth: 3, Accuracy: 0.57 (+/- 0.08)
Max depth: 4, Accuracy: 0.56 (+/- 0.06)
Max depth: 5, Accuracy: 0.57 (+/- 0.10)
Max depth: 6, Accuracy: 0.57 (+/- 0.08)
Max depth: 7, Accuracy: 0.56 (+/- 0.09)
Max depth: 8, Accuracy: 0.56 (+/- 0.09)
Max depth: 9, Accuracy: 0.57 (+/- 0.09)
Max depth: 10, Accuracy: 0.58 (+/- 0.10)
Max depth: 11, Accuracy: 0.57 (+/- 0.08)
Max depth: 12, Accuracy: 0.58 (+/- 0.09)
Max depth: 13, Accuracy: 0.58 (+/- 0.10)
Max depth: 14, Accuracy: 0.56 (+/- 0.09)
Max depth: 15, Accuracy: 0.57 (+/- 0.12)
Max depth: 16, Accuracy: 0.56 (+/- 0.13)
Max depth: 17, Accuracy: 0.56 (+/- 0.09)
Max depth: 18, Accuracy: 0.58 (+/- 0.12)
Max depth: 19, Accuracy: 0.57 (+/- 0.12)
```

Figure 2.27: codingan ke-10

- xi. Codingan Ke-11 :

Penjelasan : Codingan ini mendefinisikan bahwa variabel fadila akan mengeksekusi empty dari importan library numpy yang dinamakan np dengan 2 parameter yaitu 19,3 dan float. i didefinisikan dengan angka 0 kemudian untuk perhitungan jarak max depth diantara parameter 1 dan 20. Variabel sate mendefinisikan klasifikasi decision tree dengan 2 parameter. setelah itu, variabel score mendefinisikan variabel fadila dengan i dan 0, variabel kedua dari fadila dengan i dan 1 serta variabel ketiga dari fadila dengan i dan 2, maka pengeksekusian akhir bahwa variabel i akan ditambah dengan angka 1 untuk hasil akhirnya. Keluarannya akan berupa array dari perhitungan parameter dan variabel yang telah didefinisikan sebelumnya.

- Hasil Codingan Ke-11 :

```
In [55]: fadila = np.empty((19,3), float)
...: i = 0
...: for max_depth in range(1, 20):
...: sate = tree.DecisionTreeClassifier(criterion="entropy",
...: max_depth=max_depth)
...: scores = cross_val_score(sate, bakso_att, bakso_pass, cv=5)
...: fadila[i,0] = max_depth
...: fadila[i,1] = scores.mean()
...: fadila[i,2] = scores.std() * 2
...: i += 1
...:
...: fadila
Out[55]:
array([[1.00000000e+00, 5.7975178e-01, 0.38702390e-01],
 [2.00000000e+00, 5.6844823e-01, 1.1654802e-01],
 [3.00000000e+00, 5.7289509e-01, 9.81795687e-02],
 [4.00000000e+00, 5.7286426e-01, 1.49089399e-01],
 [5.00000000e+00, 5.4418209e-01, 1.46798360e-01],
 [6.00000000e+00, 5.7456264e-01, 8.3623731e-02],
 [7.00000000e+00, 5.6698198e-01, 8.88612754e-02],
 [8.00000000e+00, 5.5181127e-01, 1.88282549e-01],
 [9.00000000e+00, 5.44374391e-01, 1.83876194e-01],
 [1.00000000e+01, 5.44343557e-01, 1.36313639e-01],
 [1.10000000e+01, 5.6162647e-01, 1.35161155e-01],
 [1.20000000e+01, 5.56985224e-01, 1.19212818e-01],
 [1.30000000e+01, 5.70826642e-01, 1.40661789e-01],
 [1.40000000e+01, 5.51985224e-01, 9.18974681e-02],
 [1.50000000e+01, 5.84849856e-01, 7.59423720e-02],
 [1.60000000e+01, 5.79689224e-01, 8.38748467e-02],
 [1.70000000e+01, 5.4424453e-01, 9.81899799e-02],
 [1.80000000e+01, 5.6848931e-01, 1.28899779e-01],
 [1.90000000e+01, 5.64563453e-01, 8.22183928e-02]])
```

Figure 2.28: codingan ke-11

xii. Codingan Ke-12 :

Penjelasan : Codingan ini mendefinisikan pemanggilan dari library matplotlib.pyplot sebagai fadila sehingga nanti hasilnya akan berbentuk gambar grafik/gelombang. Untuk variabel fig dan ax akan mendefinisikan subplots dari fadila. Setelah itu ketentuan dari parameter depth acc = 0, depth acc = 1 dan depth acc 2. Selanjutnya untuk menampilkan gelombang maka panggil variabel fadila dengan perintah show.

- Hasil Codingan Ke-12 :

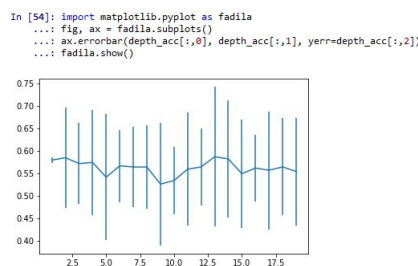


Figure 2.29: codingan ke-12

### 2.1.3 Penanganan Error

#### Pembahasan dan Penyelesaian Error

(a) Error 1 :

```
File "pandas\libs\parsers.pyx", line 384, in
pandas._libs.parsers.TextReader._cinit_
File "pandas\libs\parsers.pyx", line 695, in
pandas._libs.parsers.TextReader._setup_parser_source
FileNotFoundError: File b'student-mat.csv' does not exist
```

Figure 2.30: error 1

- Penjelasan :

- Pada error tersebut dikatakan bahwa untuk file -b”student-mat.csv” tidak ada. Mengapa? karena file codingan yang dieksekusi yaitu student-performance.py tidak berada pada folder yang sama jadi tidak dapat terpanggil dan tereksekusi.
- Nah, untuk penyelesaiannya yaitu dengan menambahkan fungsi yang mendefinisikan folder tempat file ”student-mat.csv” berada.
- Silahkan tambahkan perintah ” dataset/student-mat.csv ” pada codingannya

- Dengan penambahan perintah tersebut maka tidak akan terjadi error lagi.

(b) Error 2 :

```
In [65]: import graphviz
...: dot_data = tree.export_graphviz(sate, out_file=None, label="all",
...: impurity=False, proportion=True,
...: feature_names=list(bakso_train_att),
...: class_names=["fail", "pass"],
...: filled=True, rounded=True)
...: graph = graphviz.Source(dot_data)
...: graph
Traceback (most recent call last):
 File "<ipython-input-65-583fa3dfc87a>", line 1, in <module>
 import graphviz
ModuleNotFoundError: No module named 'graphviz'
```

Figure 2.31: error 2

- Penjelasan :

- Pada error tersebut dikatakan bahwa tidak terdapat module graphviz sehingga codingan tidak dapat dieksekusi.
- Penanganannya yaitu dengan melakukan penginstalan module graphviz itu sendiri
- Penginstalan bisa dilakukan pada Anaconda Prompt ataupun Command Prompt
- Pada prompt tersebut masukkan perintah " conda install graphviz "
- Silahkan tunggu sampai instalasi berhasil
- Setelah selesai, maka cobalah RUN kembali codingan terkait
- Maka tidak akan terjadi error lagi.

(c) Error 3 :

```
In [86]: import graphviz
...: dot_data = tree.export_graphviz(bandung, out_file=None, label="all",
...: impurity=False, proportion=True,
...: feature_names=list(jakarta_train_att),
...: class_names=["fail", "pass"],
...: filled=True, rounded=True)
...: graph = graphviz.Source(dot_data)
...: graphERROR: execution aborted
```

Figure 2.32: error 3

- Penjelasan :

- Pada error tersebut dikatakan bahwa eksekusinya dilarang atau di aborted.
- Penanganannya sebenarnya sederhana
- Tidak terdapat kesalahan dalam pembuatan dan penyesuaian codingan

- Mungkin saja terjadi error tersebut karena pengekseskusion fungsi yang berulang-ulang pada codingan yang berbeda
- Silahkan di Restart atau dijalankan kembali aplikasi spydernya
- Run kembali maka errornya tidak akan muncul lagi dan hasilnya akan muncul seperti pada gambar pengujian no 6

(d) Error 4 :

```
File "C:\ProgramData\Anaconda\lib\site-packages\sklearn\tree\tree.py", line 377, in _validate_X_predict
X = check_array(X, dtype=DTYPE, accept_sparse="csr")
File "C:\ProgramData\Anaconda\lib\site-packages\sklearn\utils\validation.py", line 582, in check_array
context))
ValueError: unsupported format character 'b' (0x62) at index 18
```

Figure 2.33: error 4

- Penjelasan :
  - Pada error tersebut dikatakan terdapat error karakter b nya gak kebaca
  - Penanganannya sebenarnya sederhana, dari codingan tidak ada yang salah
  - Disarankan agar mengulang kembali penamaan untuk variabel terkait dari awal codingan hingga akhir sehingga lebih teratur
  - Silahkan dicoba run codingan tersebut kembali
  - Maka codingannya akan berhasil dan muncul seperti pada contoh codingan no 8 diatas

## 2.2 Lusita Violita Aprilian

### 2.2.1 binary classification dilengkapi ilustrasi gambar

Binary classification yaitu berupa kelas positif dan kelas negatif. Klasifikasi biner adalah dikotomisasi yang diterapkan untuk tujuan praktis, dan dalam banyak masalah klasifikasi biner praktis, kedua kelompok tidak simetris - daripada akurasi keseluruhan, proporsi relatif dari berbagai jenis kesalahan yang menarik. Misalnya, dalam pengujian medis, false positive (mendeteksi penyakit ketika tidak ada) dianggap berbeda dari false negative (tidak mendeteksi penyakit ketika hadir).

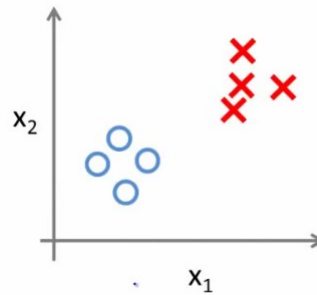


Figure 2.34: Binary Classification

## 2.2.2 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). Tugas paralel dalam psikologi manusia dan hewan sering disebut sebagai pembelajaran konsep.

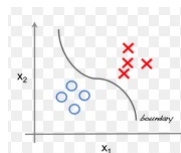


Figure 2.35: Supervised Learning

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 kluster mengidentifikasi kesamaan dalam data dan bereaksi berdasarkan ada tidaknya kesamaan di setiap potongan data baru.

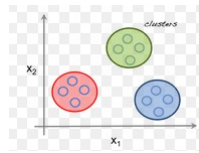


Figure 2.36: Unsupervised Learning

3. Cluster analysis or clustering adalah tugas pengelompokan sekumpulan objek sedemikian rupa sehingga objek dalam kelompok yang sama (disebut kluster) lebih mirip (dalam beberapa hal) satu sama lain daripada pada kelompok lain (kluster). Ini adalah tugas utama penambangan data eksplorasi, dan teknik umum untuk analisis data statistik, yang digunakan di banyak bidang, termasuk pembelajaran mesin, pengenalan pola, analisis gambar, pengambilan informasi, bioinformatika, kompresi data, dan grafik komputer. Analisis Cluster sendiri bukan merupakan salah satu algoritma spesifik, tetapi tugas umum yang harus dipecahkan. Ini dapat dicapai dengan berbagai algoritma yang berbeda secara signifikan dalam pemahaman mereka tentang apa yang merupakan sebuah cluster dan bagaimana cara menemukannya secara efisien. Gagasan populer mengenai cluster termasuk kelompok dengan jarak kecil antara anggota cluster, area padat ruang data, interval atau distribusi statistik tertentu. Clustering karena itu dapat dirumuskan sebagai masalah optimasi multi-objektif. Algoritma pengelompokan dan pengaturan parameter yang sesuai (termasuk parameter seperti fungsi jarak yang akan digunakan, ambang kepadatan atau jumlah cluster yang diharapkan) tergantung pada set data individual dan penggunaan hasil yang dimaksudkan. Analisis kluster bukan merupakan tugas otomatis, tetapi proses berulang penemuan pengetahuan atau optimasi multi-objektif interaktif yang melibatkan percobaan dan kegagalan. Seringkali diperlukan untuk memodifikasi praproses data dan parameter model hingga hasilnya mencapai properti yang diinginkan.

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

Evaluasi adalah tentang bagaimana kita dapat mengevaluasi seberapa baik model bekerja dengan mengukur akurasi. Dan akurasi akan didefinisikan sebagai persen-



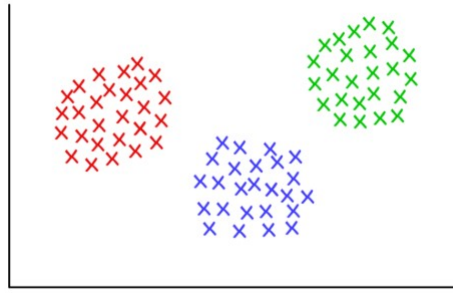


Figure 2.37: Cluster

tase 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.

|               | Predicted "apple" | Predicted "orange" |
|---------------|-------------------|--------------------|
| True "apple"  | 20                | 5                  |
| True "orange" | 3                 | 22                 |

Figure 2.38: Evaluasi dan Akurasi

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

1. Cara membuat dan membaca confusion matrix :
  - (a) Tentukan pokok permasalahan dan atributnya, misal gaji dan listik.
  - (b) Buat pohon keputusan
  - (c) Lalu data testingnya
  - (d) Lalu mencari nilai a, b, c, dan d. Semisal  $a = 5$ ,  $b = 1$ ,  $c = 1$ , dan  $d = 3$ .
  - (e) Selanjutnya mencari nilai recall, precision, accuracy, serta dan error rate.

2. Berikut adalah contoh dari confusion matrix :

- $\text{Recall} = \frac{3}{1+3} = 0,75$
- $\text{Precision} = \frac{3}{1+3} = 0,75$
- $\text{Accuracy} = \frac{5+3}{5+1+1+3} = 0,8$
- $\text{Error Rate} = \frac{1+1}{5+1+1+3} = 0,2$

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

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.

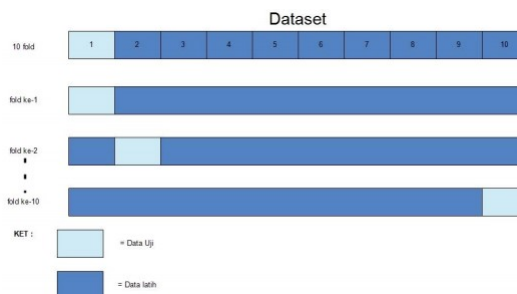


Figure 2.39: K-fold cross validation

### 2.2.6 decision tree dengan gambar ilustrasi

Decision tree adalah model visual yang terdiri dari node dan cabang, seperti Gambar dijelaskan secara rinci nanti dalam artikel ini. Untuk saat ini, amati bahwa ia tumbuh dari kiri ke kanan, dimulai dengan simpul keputusan root (kuadrat, juga disebut simpul pilihan) yang cabang-cabangnya mewakili dua atau lebih opsi bersaing yang tersedia bagi para pembuat keputusan. Pada akhir cabang awal ini, ada simpul akhir (segitiga, juga disebut simpul nilai) atau simpul ketidakpastian (lingkaran, juga disebut simpul peluang). Node akhir mewakili nilai tetap. Cabang lingkaran mewakili

hasil yang mungkin bersama dengan probabilitasnya masing-masing (yang berjumlah 1,0). Di luar cabang-cabang node ketidakpastian awal ini, mungkin ada lebih banyak bujur sangkar dan lebih banyak lingkaran, yang umumnya bergantian sampai setiap jalur berakhir di simpul akhir.

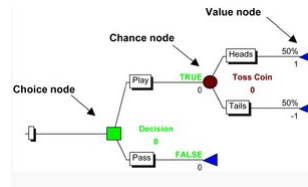


Figure 2.40: Decision Tree

## 2.2.7 Information gain dan entropi dengan gambar ilustrasi

1. Information gain (IG) mengukur seberapa banyak informasi fitur memberi kita tentang kelas. - Fitur yang sempurna mempartisi harus memberikan informasi maksimal. - Fitur yang tidak terkait seharusnya tidak memberikan informasi.

| Feature         | Explanation                          | IG Rank |
|-----------------|--------------------------------------|---------|
| <i>is.obese</i> | The subject is obese (yes, no)       | 0.0203  |
| <i>whr</i>      | Waist hip ratio                      | 0       |
| <i>hc</i>       | Hip circumference (cm)               | 0       |
| <i>bmi</i>      | Body mass index (kg/m <sup>2</sup> ) | 0       |
| <i>wc</i>       | Waist circumference (cm)             | 0       |
| <i>age</i>      | Age (years)                          | 0       |
| <i>id</i>       | Subject ID                           | 0       |

Figure 2.41: Information gain

2. Entropi merupakan kemurnian dalam koleksi contoh yang sewenang-wenang.

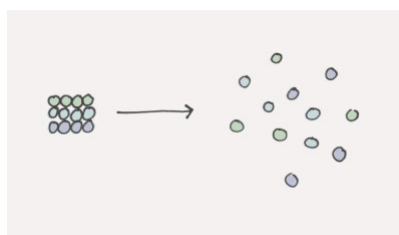


Figure 2.42: Entropi

## 2.2.8 binary classification dilengkapi ilustrasi gambar

Binary classification yaitu berupa kelas positif dan kelas negatif. Klasifikasi biner adalah dikotomisasi yang diterapkan untuk tujuan praktis, dan dalam banyak masalah

klasifikasi biner praktis, kedua kelompok tidak simetris - daripada akurasi keseluruhan, proporsi relatif dari berbagai jenis kesalahan yang menarik. Misalnya, dalam pengujian medis, false positive (mendeteksi penyakit ketika tidak ada) dianggap berbeda dari false negative (tidak mendeteksi penyakit ketika hadir).

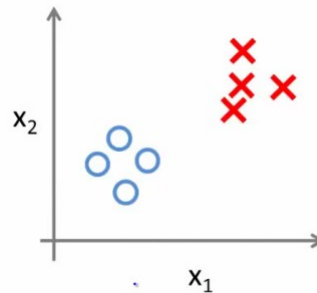


Figure 2.43: Binary Classification

## 2.2.9 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). Tugas paralel dalam psikologi manusia dan hewan sering disebut sebagai pembelajaran konsep.

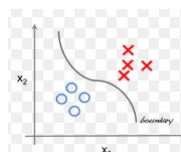


Figure 2.44: Supervised Learning

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 kluster mengidentifikasi kesamaan dalam data dan bereaksi berdasarkan ada tidaknya kesamaan di setiap potongan data baru.

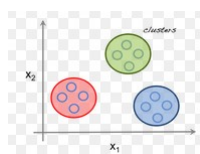


Figure 2.45: Unsupervised Learning

3. Cluster analysis or clustering adalah tugas pengelompokan sekumpulan objek sedemikian rupa sehingga objek dalam kelompok yang sama (disebut kluster) lebih mirip (dalam beberapa hal) satu sama lain daripada pada kelompok lain (kluster). Ini adalah tugas utama penambangan data eksplorasi, dan teknik umum untuk analisis data statistik, yang digunakan di banyak bidang, termasuk pembelajaran mesin, pengenalan pola, analisis gambar, pengambilan informasi, bioinformatika, kompresi data, dan grafik komputer. Analisis Cluster sendiri bukan merupakan salah satu algoritma spesifik, tetapi tugas umum yang harus dipecahkan. Ini dapat dicapai dengan berbagai algoritma yang berbeda secara signifikan dalam pemahaman mereka tentang apa yang merupakan sebuah cluster dan bagaimana cara menemukannya secara efisien. Gagasan populer mengenai cluster termasuk kelompok dengan jarak kecil antara anggota cluster, area padat ruang data, interval atau distribusi statistik tertentu. Clustering karena itu dapat dirumuskan sebagai masalah optimasi multi-objektif. Algoritma pengelompokan dan pengaturan parameter yang sesuai (termasuk parameter seperti fungsi jarak yang akan digunakan, ambang kepadatan atau jumlah cluster yang diharapkan) tergantung pada set data individual dan penggunaan hasil yang dimaksudkan. Analisis kluster bukan merupakan tugas otomatis, tetapi proses berulang penemuan pengetahuan atau optimasi multi-objektif interaktif yang melibatkan percobaan dan kegagalan. Seringkali diperlukan untuk memodifikasi praproses data dan parameter model hingga hasilnya mencapai properti yang diinginkan.

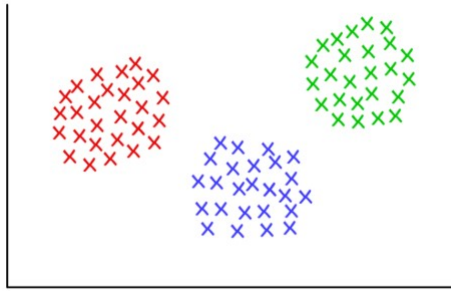


Figure 2.46: Cluster

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

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

|               | Predicted "apple" | Predicted "orange" |
|---------------|-------------------|--------------------|
| True "apple"  | 20                | 5                  |
| True "orange" | 3                 | 22                 |

Figure 2.47: Evaluasi dan Akurasi

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

1. Cara membuat dan membaca confusion matrix :
  - (a) 1) Tentukan pokok permasalahan dan atributnya, misal gaji dan listik.
  - (b) 2) Buat pohon keputusan
  - (c) 3) Lalu data testingnya
  - (d) 4) Lalu mencari nilai a, b, c, dan d. Semisal  $a = 5$ ,  $b = 1$ ,  $c = 1$ , dan  $d = 3$ .

(e) 5) Selanjutnya mencari nilai recall, precision, accuracy, serta dan error rate.

2. Berikut adalah contoh dari confusion matrix :

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

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

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.



Figure 2.48: K-fold cross validation

### 2.2.13 decision tree dengan gambar ilustrasi

Decision tree adalah model visual yang terdiri dari node dan cabang, seperti Gambar dijelaskan secara rinci nanti dalam artikel ini. Untuk saat ini, amati bahwa ia tumbuh dari kiri ke kanan, dimulai dengan simpul keputusan root (kuadrat, juga disebut simpul pilihan) yang cabang-cabangnya mewakili dua atau lebih opsi bersaing yang tersedia bagi para pembuat keputusan. Pada akhir cabang awal ini, ada simpul akhir (segitiga, juga disebut simpul nilai) atau simpul ketidakpastian (lingkaran, juga disebut simpul peluang). Node akhir mewakili nilai tetap. Cabang lingkaran mewakili hasil yang mungkin bersama dengan probabilitasnya masing-masing (yang berjumlah 1,0). Di luar cabang-cabang node ketidakpastian awal ini, mungkin ada lebih banyak bujur sangkar dan lebih banyak lingkaran, yang umumnya bergantian sampai setiap jalur berakhir di simpul akhir.

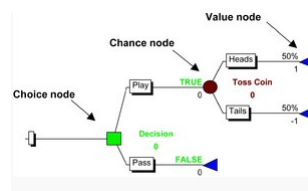


Figure 2.49: Decision Tree

### 2.2.14 Information gain dan entropi dengan gambar ilustrasi

1. Information gain (IG) mengukur seberapa banyak "informasi" fitur memberi kita tentang kelas.
  - Fitur yang sempurna mempartisi harus memberikan informasi maksimal.
  - Fitur yang tidak terkait seharusnya tidak memberikan informasi.

| Feature         | Explanation                          | IG Rank |
|-----------------|--------------------------------------|---------|
| <i>is.obese</i> | The subject is obese (yes, no)       | 0.0203  |
| <i>whr</i>      | Waist hip ratio                      | 0       |
| <i>hc</i>       | Hip circumference (cm)               | 0       |
| <i>bmi</i>      | Body mass index (kg/m <sup>2</sup> ) | 0       |
| <i>wc</i>       | Waist circumference (cm)             | 0       |
| <i>age</i>      | Age (years)                          | 0       |
| <i>id</i>       | Subject ID                           | 0       |

Figure 2.50: Information gain

2. Entropi merupakan kemurnian dalam koleksi contoh yang sewenang-wenang.



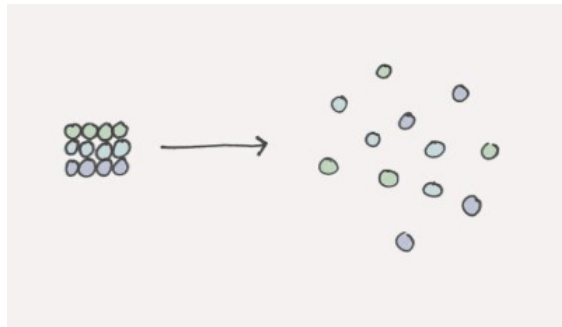


Figure 2.51: Entropi

```
In [5]: import pandas as pd
...: nanas = pd.read_csv('student-mat.csv', sep=';')
...: len(nanas)
Out[5]: 395
```

Figure 2.52: load dataset

## 2.2.15 scikit-learn

### 1. load dataset

Pada gambar tersebut dijelaskan bahwa in dengan mengimport pandas dengan diperumpamakan sebagai pd. Pada baris ke dua, dibuat variable nanas untuk memanggil pd sehingga dapat membaca file csv. Dan pada baris ke 3 variable dipanggil. Sehingga menghasilkan output 395.

### 2. generate binary label (pass/fail) based on G1+G2+G3

Pada gambar tersebut dijelaskan bahwa in pada baris pertama, dibuat variable nanas untuk memanggil nanas dengan apply lamda sehingga baris selanjutnya nanas meng-drop G1, G2, dan G3. Dan pada baris ke 3 variable dipanggil. Sehingga menghasilkan output sebuah data 5 rows x 31 columns.

### 3. use one-hot encoding on categorical columns

```
In [6]: nanas['pass'] = nanas.apply(lambda row: 1 if (row['G1']+row['G2']+row['G3'])
>= 35 else 0, axis=1)
...: nanas = nanas.drop(['G1', 'G2', 'G3'], axis=1)
...: nanas.head()
Out[6]:
 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 2.53: generate binary label

```
In [7]: nanas = pd.get_dummies(nanas, columns=['sex', 'school', 'address', 'famsize',
 ...: 'Pstatus', 'Mjob', 'Fjob',
 ...: 'reason', 'guardian', 'schoolsup', 'famsup',
 ...: 'paid', 'activities',
 ...: 'nursery', 'higher', 'internet', 'romantic'])
 ...: nanas.head()
Out[7]:
```

|   | 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]
In [8]:
```

Figure 2.54: use one-hot encoding on categorical columns

```
In [8]: nanas = nanas.sample(frac=1)
 ...: # split training and testing data
 ...: nanas_train = nanas[:500]
 ...: nanas_test = nanas[500:]
 ...:
 ...: nanas_train_att = nanas_train.drop(['pass'], axis=1)
 ...: nanas_train_pass = nanas_train['pass']
 ...:
 ...: nanas_test_att = nanas_test.drop(['pass'], axis=1)
 ...: nanas_test_pass = nanas_test['pass']
 ...:
 ...: nanas_att = nanas.drop(['pass'], axis=1)
 ...: nanas_pass = nanas['pass']
 ...:
 ...: # number of passing students in whole dataset:
 ...: import numpy as np
 ...: print("Passing: %d out of %d (%.2f%%)" % (np.sum(nanas_pass), len(nanas_pass),
100*float(np.sum(nanas_pass)) / len(nanas_pass)))
Passing: 166 out of 395 (42.03%)
```

Figure 2.55: shuffle rows, split training dan testing data

Pada gambar tersebut dijelaskan bahwa in pada baris pertama, dibuat variable nanas untuk memanggil pd dengan get dummies nanas columns. Dan pada baris selanjutnya variable dipanggil. Sehingga menghasilkan output sebuah data 5 rows x 57 columns.

#### 4. shuffle rows, split training dan testing data

Pada gambar tersebut dijelaskan bahwa in pada baris pertama, dibuat variable nanas untuk memanggil nanas sebagai sample. Lalu pada baris selanjutnya dilakukan split training dan testing data. Setelah data dilakukan split training dan testing, import file numpy sebagai np dan kemudian melakukan perintah print. Sehingga menghasilkan passing: 116 out of 395.

#### 5. fit a decision tree

Pada gambar tersebut dijelaskan bahwa in pada baris pertama, import tree dari sklearn. Pada baris selanjutnya variable duku memanggil tree dengan decision tree classifier dengan criteria entropy dan max dept = 5. Dan pada baris

```
In [11]: from sklearn import tree
 ...: duku = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
 ...: duku = duku.fit(nanas_train_att, nanas_train_pass)
In [12]:
```

Figure 2.56: fit a decision tree

```

In [2]: import graphviz
...: dot_data = tree.export_graphviz(duku, out_file=None, label="all", impurity=False,
...: proportion=True,
...: feature_names=list(nanas_train_att),
...: class_names=["fail", "pass"],
...: filled=True, rounded=True)
...: graph = graphviz.Source(dot_data)
...: graph

```

Figure 2.57: visualize tree

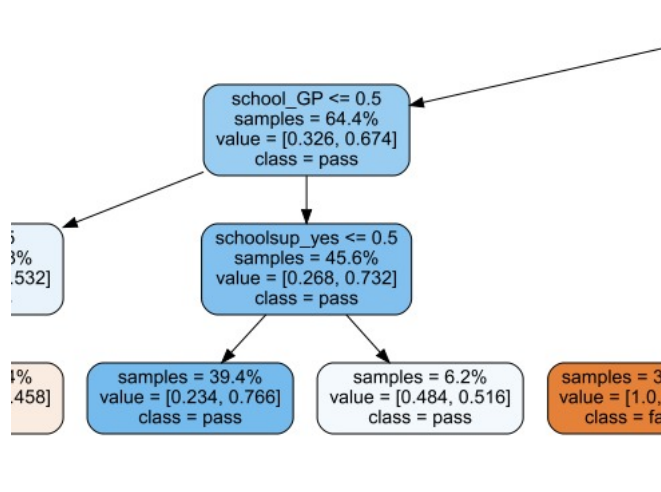


Figure 2.58: visualize tree

terakhir, variable duku memanggil duku dengan fit nanas train att dan nanas train pass.

## 6. visualize tree

Pada gambar tersebut menjelaskan import graphviz dengan variabelm dot data, class name, class name, dan graph. Sehingga menampilkan gambar sebagai berikut :

## 7. save tree

Pada gambar tersebut dijelaskan bahwa in pada baris pertama, Tree export graphviz duku dengan out file student-performance.dot dan label all. pada baris selanjutnya dijelaskan featura names = list dan class name fail atau pas. Apabila bernilai true akan filled dan rounded.

```

In [15]: tree.export_graphviz(duku, out_file="student-performance.dot", label="all",
...: impurity=False, proportion=True,
...: feature_names=list(nanas_train_att),
...: class_names=["fail", "pass"],
...: filled=True, rounded=True)
In [16]:

```

Figure 2.59: save tree

```
In [101]: duku.score(nanas_test_att, nanas_test_pass)
```

Figure 2.60: t.score

```
In [14]: from sklearn.model_selection import cross_val_score
...: scores = cross_val_score(duku, nanas_att, nanas_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: 0.59 (+/- 0.10)

In [15]:
```

Figure 2.61: show average score

## 8. t.score

Pada gambar tersebut dijelaskan bahwa in pada baris pertama, parameter dengan hasilnya adalah duku.score yang dimana terdapat perhitungan nanas test at dan nana test pass.

## 9. show average score and +/- two standard deviations away

Pada gambar tersebut dijelaskan bahwa in pada baris pertama memanggil cross vla score dari sklearn.model selection. Pada baris selanjutnya membuat variabel scores. Dan pada baris terakhir score akan ditampilkan.

## 10. for max depth in range

Pada gambar tersebut dijelaskan bahwa in pada baris pertama mendefinisikan range dari 1 hingga 20. Lalu, variabel duku memdefinisikan tree dengan decision tree clasification kriteria entropy. Dan variable scores mendefinisikan duku, nanas att, nanas pass dan cv = 5. Baris terakhir adalah menampilkan hasil.

## 11. depth acc

```
In [16]: for max_depth in range(1, 20):
...: duku = tree.DecisionTreeClassifier(criterion="entropy",
...: max_depth=max_depth)
...: scores = cross_val_score(duku, nanas_att, nanas_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 (+/- 0.03)
Max depth: 3, Accuracy: 0.61 (+/- 0.11)
Max depth: 4, Accuracy: 0.60 (+/- 0.10)
Max depth: 5, Accuracy: 0.59 (+/- 0.08)
Max depth: 6, Accuracy: 0.60 (+/- 0.10)
Max depth: 7, Accuracy: 0.60 (+/- 0.08)
Max depth: 8, Accuracy: 0.55 (+/- 0.06)
Max depth: 9, Accuracy: 0.56 (+/- 0.04)
Max depth: 10, Accuracy: 0.57 (+/- 0.06)
Max depth: 11, Accuracy: 0.59 (+/- 0.06)
Max depth: 12, Accuracy: 0.57 (+/- 0.09)
Max depth: 13, Accuracy: 0.56 (+/- 0.13)
Max depth: 14, Accuracy: 0.57 (+/- 0.07)
Max depth: 15, Accuracy: 0.58 (+/- 0.09)
Max depth: 16, Accuracy: 0.55 (+/- 0.12)
Max depth: 17, Accuracy: 0.55 (+/- 0.09)
Max depth: 18, Accuracy: 0.56 (+/- 0.06)
Max depth: 19, Accuracy: 0.58 (+/- 0.06)

In [17]:
```

Figure 2.62: for max depth in range

```

In [88]: depth_acc = np.empty((19,3), float)
...: i = 0
...: for max_depth in range(1, 20):
...: duku = tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
...: scores = cross_val_score(duku, nanas_att, nanas_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[88]:
array([[1.00000000e+00, 5.79751704e-01, 6.30768599e-03],
 [2.00000000e+00, 5.79614995e-01, 4.59786402e-02],
 [3.00000000e+00, 5.72091853e-01, 3.22002693e-02],
 [4.00000000e+00, 5.79881532e-01, 8.26691511e-02],
 [5.00000000e+00, 5.87665531e-01, 1.00629211e-01],
 [6.00000000e+00, 5.75006491e-01, 1.06656313e-01],
 [7.00000000e+00, 5.80039760e-01, 8.19527208e-02],
 [8.00000000e+00, 5.90197988e-01, 9.29887851e-02],
 [9.00000000e+00, 5.90037326e-01, 9.46882094e-02],
 [1.00000000e+01, 5.87568160e-01, 1.13990705e-01],
 [1.10000000e+01, 5.77473223e-01, 1.11095161e-01],
 [1.20000000e+01, 5.77507303e-01, 8.03612112e-02],
 [1.30000000e+01, 6.02601428e-01, 4.38647024e-02],
 [1.40000000e+01, 5.95037326e-01, 5.40261640e-02],
 [1.50000000e+01, 5.77757515e-01, 2.74778007e-02],
 [1.60000000e+01, 5.77757515e-01, 2.74778007e-02],
 [1.70000000e+01, 5.77757515e-01, 2.74778007e-02],
 [1.80000000e+01, 5.77757515e-01, 2.74778007e-02],
 [1.90000000e+01, 5.77757515e-01, 2.74778007e-02]
])

```

Figure 2.63: depth acc

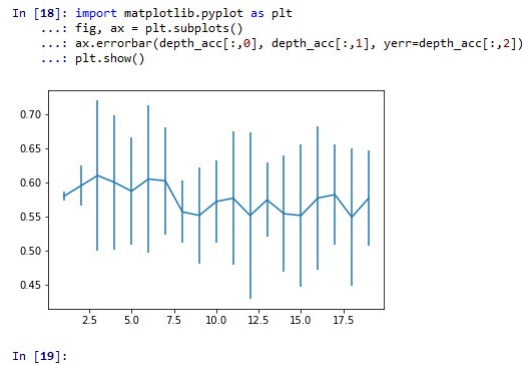


Figure 2.64: matplotlib

Pada gambar tersebut dijelaskan bahwa variabel depth acc mendefinisikan np empty dengan  $i = 0$  dari range 1-20. Dimana pada range tersebut terdapat variabel duku dan score. Variabel duku mendefinisikan tree dengan decision tree clasification kriteria entropy. Dan variable scores mendefinisikan duku, nanas att, nanas pass dan  $cv = 5$ . Dan pada baris selanjutnya adalah pendefinisian depth acc. Sehingga didapatlah hasil berupa array.

## 12. matplotlib

Pada gambar tersebut dijelaskan bahwa in pada baris pertama mendefinisikan pemanggilan matplotlib.pyplot sebagai plt. Lalu pendefinisian figure dari plt serta error bar. Dan pada baris terakhir menampilkan plt. Maka akan muncul sebuah gambar berupa giagram.

## 2.2.16 Penanganan Error

### 1. Skrinsut error

```

File "pandas\libs\parsers.pyx", line 384, in
pandas._libs.parsers.TextReader._cinit_
File "pandas\libs\parsers.pyx", line 695, in
pandas._libs.parsers.TextReader._setup_parser_source
FileNotFoundError: File b'student-por.csv' does not exist

```

Figure 2.65: error

```

In [5]: import pandas as pd
...: nanas = pd.read_csv('student-mat.csv', sep=';')
...: len(nanas)
Out[5]: 395

```

Figure 2.66: penyelesaian

## 2. Tuliskan kode error dan jenis errornya

Kode dan jenis error yang didapatkan adalah :

- Kode error = `FileNotFoundError: File b'student-por.csv' does not exist`
- Jenis error = `FileNotFoundError`

## 3. Solusi pemecahan masalah error tersebut

Berikut adalah solusi dari permasalahan pada no. 1

## 2.3 Rahmi Roza/1164085

### 2.3.1 Teori

Penyelesaian Tugas Harian 3 ( No. 1-7 )

#### 1. Binary Classification Dan Ilustrasi Gambarnya

- Pengertian Binary Classification / Klasifikasi Biner:

Klasifikasi biner atau binomial adalah tugas mengklasifikasikan elemen-elemen dari himpunan yang diberikan ke dalam dua kelompok (memprediksi kelompok mana yang masing-masing dimiliki) berdasarkan aturan klasifikasi.

- Ilustrasi Gambar Binary Classification :
- Supervised Learning, Unsupervised Learning, Clustering Dan Ilustrasi Gambar

- Pengertian Supervised Learning :

Supervised learning adalah sebuah pendekatan dimana sudah terdapat data yang dilatih, dan terdapat variable yang ditargetkan sehingga

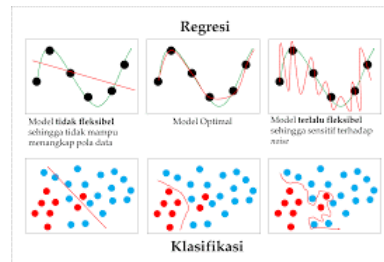


Figure 2.67: binary classification

tujuan dari pendekatan ini adalah mengelompokkan suatu data ke data yang sudah ada.

- Ilustrasi Gambar Supervised Learning :

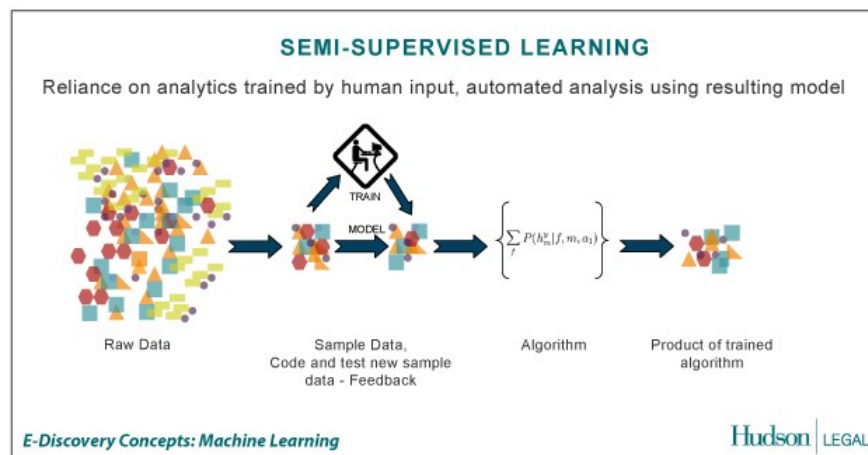


Figure 2.68: supervised

- Pengertian Unsupervised Learning :

unsupervised learning tidak memiliki data latih, sehingga dari data yang ada, kita mengelompokkan data tersebut menjadi 2 bagian atau 3 bagian dan seterusnya.

– Ilustrasi Gambar Unsupervised Learning :

- Pengertian Clustering :

Metode pengelompokan data. Clustering juga merupakan proses partisi satu set objek data ke dalam himpunan bagian yang disebut dengan cluster. Objek dalam cluster tersebut memiliki kemiripan karakteristik antar satu sama lain.

## 2. Evaluasi, Akurasi Dan Ilustrasi Gambar

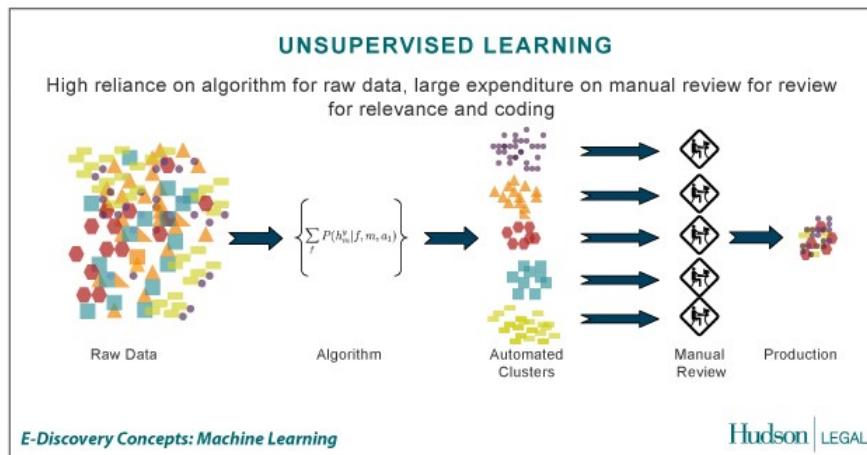


Figure 2.69: unsupervised

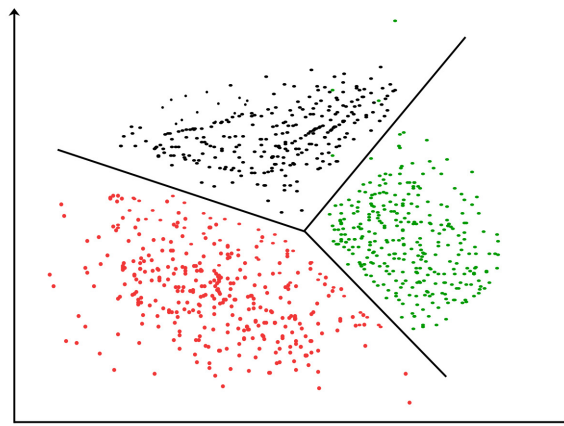


Figure 2.70: clustering

- **Pengertian Evaluasi**

Evaluasi digunakan untuk memeriksa/memastikan dan mengevaluasi model dalam bekerja ( seberapa baik ) dengan mengukur keakuratannya. Kita juga dapat menanalisis kesalahan yang dibuat pada model yang dijalankan, tingkat kebingungan dan menggunakan matriks kebingungan.

- **Pengertian Akurasi**

Accuracy akan didefinisikan sebagai presentasi kasus yang diklasifikasikan dengan benar. Accuracy lebih jelasnya adalah perbandingan kasus yang diidentifikasi benar dengan jumlah semua kasus

Rumus dari accuracy=  $(a+c)/(a+b+c+d)$

Dilakukan perhitungan dengan rumus akurasi terhadap data yang telah diolah pada " Evaluasi ". Kemudian di dapatkan hasil dari pengolahan



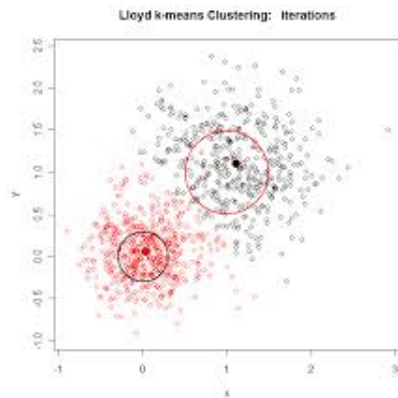


Figure 2.71: Evaluasi

Made : Fadila  
 Accuracy / Ketepatan :  $((20 + 22) / (20 + 5 + 3 + 22)) = 84$

Figure 2.72: Akurasi

data tersebut.

Contoh penggabungan Akurasi Dan Evaluasi

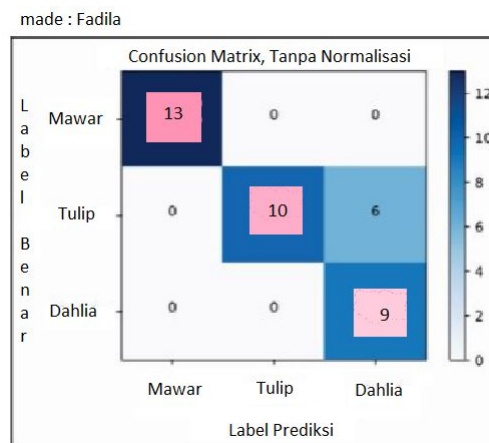


Figure 2.73: Contoh Evaluasi Dan Akurasi Secara Bersamaan

### 3. Membuat Dan Membaca Confusion Matrix Beserta Contoh

- Pengertian Confusion Matrix

Confusion matrix adalah suatu metode yang biasanya digunakan untuk melakukan perhitungan akurasi pada konsep data mining. Rumus ini

melakukan perhitungan dengan 4 keluaran, yaitu: recall, precision, accuracy dan error rate.

- Pembacaan Confusion Matrix

- (a) Apabila hasil prediksi negatif dan data sebenarnya merupakan negatif.
- (b) Apabila hasil prediksi positif sedangkan nilai sebenarnya merupakan negatif.
- (c) Apabila hasil prediksi negatif sedangkan nilai sebenarnya merupakan positif.
- (d) Apabila hasil prediksi positif dan nilai sebenarnya merupakan positif.

- Pembuatan Confusion Matrix

- (a) Menentukan 4 proses klasifikasi yang akan digunakan dalam confusion matrix.
- (b) 4 Istilah tersebut ada True Positive ( TP ), True Negative ( TN ), False Positive ( FP ) dan False Negative ( FN ).
- (c) Kelompokkan klasifikasi tersebut bisa menggunakan klasifikasi biner
- (d) Akan menghasilkan keluaran berupa 2 Kelas ( Positif dan Negatif ) dan penentuan TP, FP ( 1 klasifikasi positif ) , FN dan TN ( 1 klasifikasi negatif ).
- (e) Contoh dasarnya nampak seperti langkah diatas
- (f) Istilahnya dapat didefinisikan dengan objek lain namun dengan alur yang sama ( sesuai rumus baik klasifikasi dll ).

- Ilustrasi Gambar

#### 4. Cara Kerja K-Fold Classification Dan Ilustrasi Gambar

- (a) Pertama-tama untuk total instance dibagi menjadi N bagian.
- (b) Fold ke-1 ( atau pertama ) adalah ketika bagian ke-1 menjadi data uji (testing data) dan sisanya menjadi data latih (training data).
- (c) Hitung akurasi ( berdasarkan porsi data tersebut. Persamaanya sebagai berikut :  
(sigma) data klasifikasi  
(sigma) total data uji  
x 100 persen

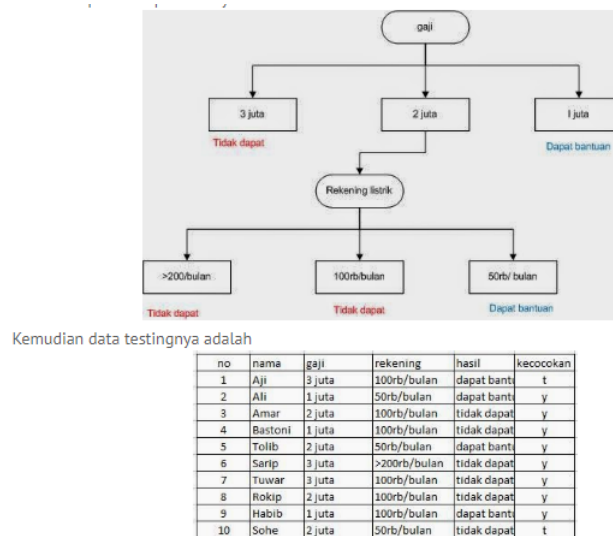


Figure 2.74: confusion matrix

- (d) Fold ke-2 ( kedua ) adalah ketika bagian ke-2 menjadi data uji (testing data) dan sisanya menjadi data latih (training data).
- (e) Kemudian dihitunglah akurasi berdasarkan porsi data yang telah ditentukan
- (f) Demikian seterusnya hingga mencapai fold ke-K. Hitung rata-rata akurasi dari K buah akurasi di atas. Rata-rata akurasi ini menjadi akurasi final atau akhir.

- Ilustrasi Gambar
- Decision Tree Dan Ilustrasi Gambar

#### – Pengertian Decision Tree

Decision Tree (Pohon Keputusan) adalah pohon dimana setiap cabangnyamenunjukkan pilihan diantara sejumlah alternatif pilihan yang ada, dan setiapdaunnya menunjukkan keputusan yang dipilih.Decision tree biasa digunakan untuk mendapatkan informasi untuk tujuan-pengambilan sebuah keputusan. Decision tree dimulai dengan sebuah root node(titik awal) yang dipakai oleh user untuk mengambil tindakan. Dari node root ini,user memecahnya sesuai dengan algoritma decision tree. Hasil akhirnya adalahsebuah decision tree dengan setiap cabangnya menunjukkan kemungkinansekenario dari keputusan yang diambil serta hasilnya.

**Dataset:**

|                        |
|------------------------|
| K1   K2   K3   K4   K5 |
|------------------------|

**Data Eksperimen:**

|                                        |
|----------------------------------------|
| Eksperimen Ke   Data Latih   Data Test |
| 1   K2,K3,K4,K5   K1                   |
| 2   K1,K3,K4,K5   K2                   |
| 3   K1,K2,K4,K5   K3                   |
| 4   K1,K2,K3,K5   K4                   |
| 5   K1,K2,K3,K4   K5                   |

Figure 2.75: k-fold classification 1

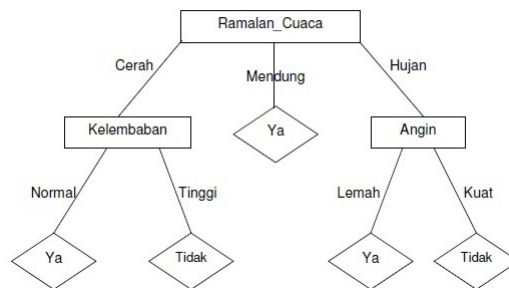


Figure 2.76: decision tree

- Ilustrasi Gambar
- Information Gain Dan Entropi
  - Pengertian Information Gain
 

Information gain adalah salah satu attribute selection measure yang digunakan untuk memilih test attribute tiap node pada tree. Atribut dengan information gain tertinggi dipilih sebagai test atribut dari suatu node. Ada 2 kasus berbeda pada saat penghitungan Information Gain, pertama untuk kasus penghitungan atribut tanpa missing value dan kedua, penghitungan atribut dengan missing value.
- Ilustrasi Gambar
- Pengertian Entropi
 

Adalah suatu parameter untuk mengukur tingkat keberagaman (heterogenitas) dari kumpulan data. Semakin heterogen, nilai entropi semakin

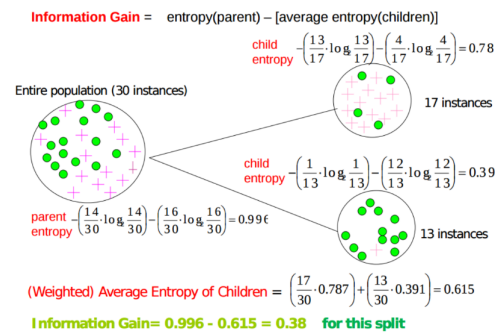


Figure 2.77: informaion gain 1

besar.

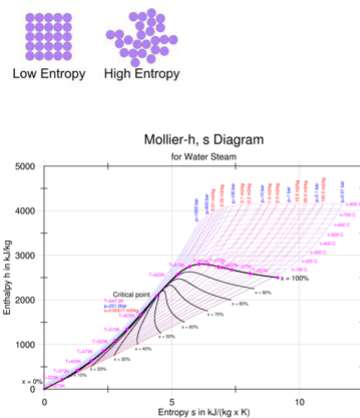


Figure 2.78: entropi

## 2.3.2 Scikit-learn

Penyelesaian Tugas Harian 4 ( No. 1-12 )

- Pembahasan Codingan Dan Hasilnya

(a) Gambar Pertama :

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

– Hasil Gambar Pertama :

(b) Gambar Kedua :

```
In [40]: import pandas as solok
...: solok = solok.read_csv('student-mat.csv', sep=';')
...: len(solok)
Out[40]: 395
```

Figure 2.79: Gambar pertama

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

– Hasil Gambar Kedua :

```
In [41]: solok['pass'] = solok.apply(lambda row: 1 if (row['G1']+row['G2']+row['G3']) >=
35 else 0, axis=1)
...: solok = solok.drop(['G1', 'G2', 'G3'], axis=1)
...: solok.head()
Out[41]:
 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 2.80: Gambar kedua

(c) Gambar Ketiga :

Penjelasan : Variabel solok 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 :

```
In [42]: solok = pd.get_dummies(solok, columns=['sex', 'school', 'address', 'famsize', 'Pstatus', 'Mjob',
'Fjob',
...: 'reason', 'guardian', 'schoolsup', 'famsup', 'paid', 'activities',
...: 'nursery', 'higher', 'internet', 'romantic'])
...: solok.head()
Out[42]:
 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 2.81: Gambar Ketiga

(d) Gambar Keempat :

Penjelasan : Penejelasan pada gambar keempat adalah variabel solok 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.

```

In [43]: solok = solok.sample(frac=1)
...: # split training and testing data
...: solok_train = solok[:500]
...: solok_test = solok[500:]
...:
...: solok_train_att = solok_train.drop(['pass'], axis=1)
...: solok_train_pass = solok_train['pass']
...:
...: solok_test_att = solok_test.drop(['pass'], axis=1)
...: solok_test_pass = solok_test['pass']
...:
...: solok_att = solok.drop(['pass'], axis=1)
...: solok_pass = solok['pass']
...:
...: # number of passing students in whole dataset:
...: import numpy as np
...: print("Passing: %d out of %d (%.2f%%)" % (np.sum(solok_pass), len(solok_pass),
100*float(np.sum(solok_pass)) / len(solok_pass)))
Passing: 166 out of 395 (42.03%)

```

Figure 2.82: Gambar Keempat

– Hasil Gambar Keempat :

(e) Gambar Kelima :

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

– Hasil Gambar Kelima :

```

In [44]: from sklearn import tree
...: solo = tree.DecisionTreeClassifier(criterion="entropy", max_depth=5)
...: solo = solo.fit(solok_train_att, solok_train_pass)

```

Figure 2.83: Gambar Kelima

(f) Gambar Keenam :

Penjelasan : Pada gambar nomor 6, terjadi kesalah error yaitu pada import graphivz.

– Hasil Gambar Keenam :

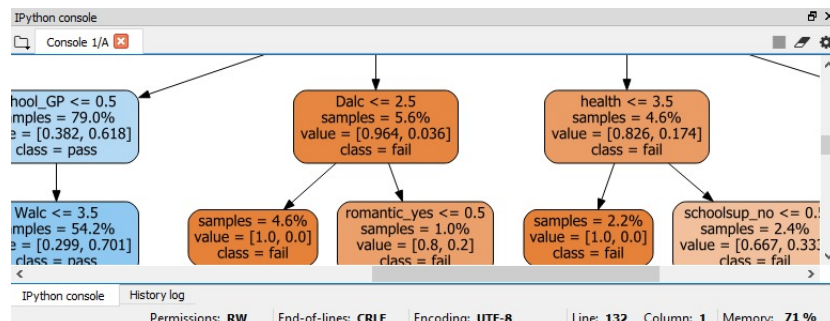


Figure 2.84: Gambar Keenam

(g) 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 :

```
In [47]: tree.export_graphviz(solo, out_file="student-performance.dot", label="all", impurity=False,
...: proportion=True,
...: feature_names=list(solak_train_att), class_names=["fail", "pass"],
...: filled=True, rounded=True)
```

Figure 2.85: Gambar Ketujuh

(h) Gambar Kedelapan :

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

– Hasil Gambar Kedelapan :

```
In [34]: solo.score(solak_test_att, solok_test_pass)
Out[34]: 0.6577181208053692
```

Figure 2.86: Gambar Kedelapan

(i) Gambar Kesembilan:

Penjelasan : Pada gambar 9, kodingan teresbut mnedefinisikan library sklearn model selection dan import cross val score. Dan kemudian variabel scores mengeksekusi fungsi cross val score(solo, solok att, solok pass, cv=5). Kemudian akan menampilkan nilai dari fungsi akurasi.

– Hasil Gambar Kesembilan :

```
In [49]: from sklearn.model_selection import cross_val_score
...: scores = cross_val_score(solo, solok_att, solok_pass, cv=5)
...: # show average score and +/- two standard deviations away (covering 95% of scores)
...: print("Accuracy: %.2f (+/- %.2f)" % (scores.mean(), scores.std() * 2))
Accuracy: 0.53 (+/- 0.11)
```

Figure 2.87: Gambar Kesembilan

(j) 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.



```

In [50]: for max_depth in range(1, 20):
...: solo = tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
...: scores = cross_val_score(solo, solok_att, solok_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 (+/- 0.07)
Max depth: 3, Accuracy: 0.59 (+/- 0.14)
Max depth: 4, Accuracy: 0.54 (+/- 0.02)
Max depth: 5, Accuracy: 0.53 (+/- 0.11)
Max depth: 6, Accuracy: 0.57 (+/- 0.11)
Max depth: 7, Accuracy: 0.56 (+/- 0.11)
Max depth: 8, Accuracy: 0.55 (+/- 0.15)
Max depth: 9, Accuracy: 0.57 (+/- 0.09)
Max depth: 10, Accuracy: 0.56 (+/- 0.11)
Max depth: 11, Accuracy: 0.57 (+/- 0.15)
Max depth: 12, Accuracy: 0.58 (+/- 0.09)
Max depth: 13, Accuracy: 0.57 (+/- 0.09)
Max depth: 14, Accuracy: 0.55 (+/- 0.11)
Max depth: 15, Accuracy: 0.57 (+/- 0.09)
Max depth: 16, Accuracy: 0.57 (+/- 0.09)
Max depth: 17, Accuracy: 0.54 (+/- 0.14)
Max depth: 18, Accuracy: 0.55 (+/- 0.09)
Max depth: 19, Accuracy: 0.55 (+/- 0.09)

```

Figure 2.88: Gambar Kesepuluh

– Hasil Gambar Kesepuluh :

(k) 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 solo, solok att, solok pass, cv=5. Yang mana hasil tampilan dari kodingannya adalah outputan seperti gambar 11.

– Hasil Gambar Kesebelas :

```

In [51]: depth_acc = np.empty((19,3), float)
...: i = 0
...: for max_depth in range(1, 20):
...: solo = tree.DecisionTreeClassifier(criterion="entropy", max_depth=max_depth)
...: scores = cross_val_score(solo, solok_att, solok_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[51]:
array([[1.00000000e+00, 5.79751704e-01, 6.30768599e-03],
 [2.00000000e+00, 5.90073028e-01, 7.19635495e-02],
 [3.00000000e+00, 5.87446446e-01, 1.37192167e-01],
 [4.00000000e+00, 5.44337877e-01, 2.41201036e-02],
 [5.00000000e+00, 5.29500162e-01, 1.05510772e-01],
 [6.00000000e+00, 5.85200422e-01, 1.22237161e-01],
 [7.00000000e+00, 5.64788218e-01, 9.32673367e-02],
 [8.00000000e+00, 5.37129990e-01, 1.21107315e-01],
 [9.00000000e+00, 5.69900526e-01, 1.27250587e-01],
 [1.00000000e+01, 5.52448800e-01, 1.49180519e-01],
 [1.10000000e+01, 5.67672022e-01, 1.59022626e-01],
 [1.20000000e+01, 5.82702045e-01, 1.36814886e-01],
 [1.30000000e+01, 5.49758195e-01, 1.52087945e-01],
 [1.40000000e+01, 5.67417235e-01, 1.37785627e-01],
 [1.50000000e+01, 5.64978903e-01, 1.11136389e-01],

```

Figure 2.89: Gambar Kesebelas

(1) 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 :

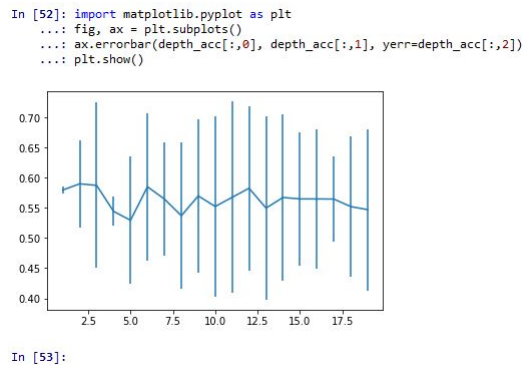


Figure 2.90: Gambar Keduabelas

### 2.3.3 Penanganan Error

(a) Skreensut Error

```
In [65]: import graphviz
...: dot_data = tree.export_graphviz(solo, out_file=None, label="all", impurity=False, proportion=True,
...: feature_names=list(solo.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 2.91: Gambar Ketigabelas

(b) Kode Error dan Jenis Errornya

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

Jenis Error: Pada Grafik

(c) Penanganan

Melakukan install ulang pada graphiz

# Chapter 3

## Methods

### 3.1 The data

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

### 3.2 Method 1

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

### 3.3 Method 2

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

# Chapter 4

## Experiment and Result

brief of experiment and result.

### 4.1 Experiment

Please tell how the experiment conducted from method.

### 4.2 Result

Please provide the result of experiment

# Chapter 5

## Conclusion

brief of conclusion

### **5.1 Conclusion of Problems**

Tell about solving the problem

### **5.2 Conclusion of Method**

Tell about solving using method

### **5.3 Conclusion of Experiment**

Tell about solving in the experiment

### **5.4 Conclusion of Result**

tell about result for purpose of this research.

# Chapter 6

## Discussion

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

# Chapter 7

## Discussion

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

# Chapter 8

## Discussion

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



# Chapter 9

## Discussion

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

# Chapter 10

## Discussion

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

# Chapter 11

## Discussion

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

# Chapter 12

## Discussion

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

# Chapter 13

## Discussion

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

# Chapter 14

## Discussion

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

# Appendix A

## Form Penilaian Jurnal

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

| NO | UNSUR                                        | KETERANGAN                                                                                                                                                                        | MAKS | KETERANGAN                                                                                                                                                                                                                                              |
|----|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Keefektifan Judul Artikel                    | Maksimal 12 (dua belas) kata dalam Bahasa Indonesia atau 10 (sepuluh) kata dalam Bahasa Inggris                                                                                   | 2    | a. Tidak lugas dan tidak ringkas (0)<br>b. Kurang lugas dan kurang ringkas (1)<br>c. Ringkas dan lugas (2)                                                                                                                                              |
| 2  | Pencantuman Nama Penulis dan Lembaga Penulis |                                                                                                                                                                                   | 1    | a. Tidak lengkap dan tidak konsisten (0)<br>b. Lengkap tetapi tidak konsisten (0,5)<br>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)<br>b. Abstrak kurang jelas dan ringkas, atau hanya dalam Bahasa Inggris, atau dalam Bahasa Indonesia saja (1)<br>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)<br>b. Ada tetapi kurang mencerminkan konsep penting dalam artikel (0,5)<br>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)<br>b. Lengkap tetapi tidak sesuai sistematika (0,5)<br>c. Lengkap dan bersistem (1)                                                                                                                                                |
| 6  | Pemanfaatan Instrumen Pendukung              | Pemanfaatan Instrumen Pendukung seperti gambar dan tabel                                                                                                                          | 1    | a. Tidak dimanfaatkan (0)<br>b. Kurang informatif atau komplementer (0,5)<br>c. Informatif dan komplementer (1)                                                                                                                                         |
| 7  | Cara Pengacuan dan Pengutipan                |                                                                                                                                                                                   | 1    | a. Tidak baku (0)<br>b. Kurang baku (0,5)<br>c. Baku (1)                                                                                                                                                                                                |
| 8  | Penyusunan Daftar Pustaka                    | Penyusunan Daftar Pustaka                                                                                                                                                         | 1    | a. Tidak baku (0)<br>b. Kurang baku (0,5)<br>c. Baku (1)                                                                                                                                                                                                |
| 9  | Peristilahan dan Kebahasaan                  |                                                                                                                                                                                   | 2    | a. Buruk (0)<br>b. Baik (1)<br>c. Cukup (2)                                                                                                                                                                                                             |
| 10 | Makna Sumbangan bagi Kemajuan                |                                                                                                                                                                                   | 4    | a. Tidak ada (0)<br>b. Kurang (1)<br>c. Sedang (2)<br>d. Cukup (3)<br>e. Tinggi (4)                                                                                                                                                                     |

Figure A.1: Form nilai bagian 1.



|                                           |                                                      |                                                                                                           |    |                                                                                                                                       |
|-------------------------------------------|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|----|---------------------------------------------------------------------------------------------------------------------------------------|
| 11                                        | Dampak Ilmiah                                        |                                                                                                           | 7  | a. Tidak ada (0)<br>b. Kurang (1)<br>c. Sedang (3)<br>d. Cukup (5)<br>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)<br>b. 40-80% (2)<br>c. > 80% (3)                                                                                         |
| 13                                        | Derajat Kemutakhiran Pustaka Acuan                   | Derajat Kemutakhiran Pustaka Acuan                                                                        | 3  | a. < 40% (1)<br>b. 40-80% (2)<br>c. > 80% (3)                                                                                         |
| 14                                        | Analisis dan Sintesis                                | Analisis dan Sintesis                                                                                     | 4  | a. Sedang (2)<br>b. Cukup (3)<br>c. Baik (4)                                                                                          |
| 15                                        | Penyimpulan                                          | Sangat jelas relevasinya dengan latar belakang dan pembahasan, dirumuskan dengan singkat                  | 3  | a. Kurang (1)<br>b. Cukup (2)<br>c. Baik (3)                                                                                          |
| 16                                        | Unsur Plagiat                                        |                                                                                                           | 0  | a. Tidak mengandung plagiat (0)<br>b. Terdapat bagian-bagian yang merupakan plagiat (-5)<br>c. Keseluruhannya merupakan plagiat (-20) |
| TOTAL                                     |                                                      |                                                                                                           | 36 |                                                                                                                                       |
| Catatan : Nilai minimal untuk diterima 25 |                                                      |                                                                                                           |    |                                                                                                                                       |

Figure A.2: form nilai bagian 2.

# Appendix B

## FAQ

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

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

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

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

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

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

M : Pa saya bingung D : Coba lihat standar alasan

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

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

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

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

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

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

# Bibliography

- [1] Joshua Eckroth. *Python Artificial Intelligence Projects for Beginners: Get up and running with Artificial Intelligence using 8 smart and exciting AI applications*. Packt Publishing Ltd, 2018.
- [2] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.