

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	Ahmad Syafrizal Huda/1164062	2
1.4.1	Teori	2
1.4.2	Instalasi	4
1.4.2.1	Instalasi Library Scikit dari Anaconda	4
1.4.2.2	Mencoba Loading an example Dataset	4
1.4.2.3	Learning and Predicting	5
1.4.2.4	Model Presistence	5
1.4.2.5	Conventions	7
1.4.3	Penanganan eror	10
1.4.3.1	ScreenShoot Error	10
1.4.3.2	Tuliskan Kode Error dan Jenis Erornya	11
1.4.3.3	Solusi Pemecahan Masalah Error	11
1.5	Cokro Edi Prawiro/1164069	12
1.5.1	Praktek teori penunjang	12
1.5.2	Instalisasi	18
1.5.3	Penanganan Error	38
2	Related Works	44
2.1	Same Topics	44
2.1.1	Topic 1	44
2.1.2	Topic 2	44
2.2	Same Method	44
2.2.1	Method 1	44
2.2.2	Method 2	44

3	Methods	45
3.1	The data	45
3.2	Method 1	45
3.3	Method 2	45
4	Experiment and Result	46
4.1	Experiment	46
4.2	Result	46
5	Conclusion	47
5.1	Conclusion of Problems	47
5.2	Conclusion of Method	47
5.3	Conclusion of Experiment	47
5.4	Conclusion of Result	47
6	Discussion	48
7	Discussion	49
8	Discussion	50
9	Discussion	51
10	Discussion	52
11	Discussion	53
12	Discussion	54
13	Discussion	55
14	Discussion	56
A	Form Penilaian Jurnal	57
B	FAQ	60
	Bibliography	62

List of Figures

1.1	Hasil Tampilan Error.	10
1.2	Hasil Tampilan Install joblib.	12
1.3	Hasil Tampilan Uji coba perintah joblib.	12
1.4	Download Anaconda.	13
1.5	Langkah pertama instalasi anaconda.	13
1.6	Langkah kedua instalasi anaconda.	14
1.7	Langkah ketiga instalasi anaconda.	15
1.8	Langkah terakhir instalasi anaconda.	16
1.9	Langkah pertama instalasi scikit pada CMD.	16
1.10	Langkah kedua instalasi scikit pada CMD.	17
1.11	Langkah ketiga instalasi scikit pada CMD.	17
1.12	Langkah compile code pada python anaconda.	18
1.13	Hasil Tampilan 1.	18
1.14	Hasil Tampilan 2.	19
1.15	Hasil Tampilan 3.	19
1.16	Hasil Tampilan 4.	20
1.17	Hasil Tampilan 5.	20
1.18	Hasil Tampilan 6.	21
1.19	Hasil Tampilan 7.	21
1.20	Hasil Tampilan 8.	22
1.21	Hasil Tampilan 9.	22
1.22	Hasil Tampilan 10.	23
1.23	Hasil Tampilan 11.	23
1.24	Hasil Tampilan 12.	24
1.25	Hasil Tampilan 13.	24
1.26	Hasil Tampilan 14.	25
1.27	Hasil Tampilan 15.	25
1.28	Hasil Tampilan 16.	26

1.29 Hasil Tampilan 17.	26
1.30 Hasil Tampilan 18.	27
1.31 Hasil Tampilan 19.	27
1.32 Hasil Tampilan 20.	28
1.33 Hasil Tampilan 21.	28
1.34 Hasil Tampilan 22.	29
1.35 Tampilan website Scikit 1.	29
1.36 Tampilan website Scikit 2.	30
1.37 Tampilan Versi Python dan Anaconda	39
1.38 Instalasi Library Sikic.	40
1.39 Instalasi Library Sikic Melalui Conda	40
1.40 Console Python Include Anaconda	41
1.41 Contoh Codingan Dataset	41
1.42 Error Coding 1	42
1.43 Error Coding 2	42
1.44 Codingan Solusi Untuk Error digits	42
1.45 Codingan Solusi Untuk Error Joblib	42
1.46 Hasil Solusi Error Joblib	43
1.47 Error Type data, yang harus digunakan number sedangkan isinya 'SCALE' pada gamma	43
1.48 Error no Module found, modul yang dicari tidak ditemukan atau tidak ada 'JOBLIB'	43
A.1 Form nilai bagian 1.	58
A.2 form nilai bagian 2.	59

Chapter 1

Mengenai Kecerdasan Buatan dan Scikit-Learn

Buku umum yang digunakan adalah [5] dan untuk sebelum UTS menggunakan buku *Python Artificial Intelligence Projects for Beginners*[2]. Dengan praktek menggunakan python 3 dan editor anaconda dan library python scikit-learn. Tujuan pembelajaran pada pertemuan pertama antara lain:

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

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

1.1 Teori

Praktek teori penunjang yang dikerjakan :

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

1.2 Instalasi

Membuka <https://scikit-learn.org/stable/tutorial/basic/tutorial.html>. Dengan menggunakan bahasa yang mudah dimengerti dan bebas plagiat. Dan wajib skrinsut dari komputer sendiri.

1. Instalasi library scikit dari anaconda, mencoba kompilasi dan uji coba ambil contoh kode dan lihat variabel explorer[hari ke 1](10)
2. Mencoba Loading an example dataset, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 1](10)
3. Mencoba Learning and predicting, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)
4. mencoba Model persistence, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)
5. Mencoba Conventions, menjelaskan maksud dari tulisan tersebut dan mengartikan per baris[hari ke 2](10)

1.3 Penanganan Error

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

1. skrinsut error[hari ke 2](10)
2. Tuliskan kode eror dan jenis errornya [hari ke 2](10)
3. Solusi pemecahan masalah error tersebut[hari ke 2](10)

iiiiiii HEAD

1.4 Ahmad Syafrizal Huda/1164062

1.4.1 Teori

1. Definisi, sejarah, dan perkembangan kecerdasan buatan.

Definisi kecerdasan buatan adalah suatu pengetahuan yang dapat membuat komputer untuk meniru kecerdasan manusia yang berhubungan dengan

penangkapan, pemodelan, dan penyimpanan kecerdasan manusia dalam sebuah sistem teknologi. Contohnya yaitu melakukan analisa penalaran untuk mengambil suatu kesimpulan atau penerjemahan atau keputusan dari satu bahasa satu ke bahasa lain.

Sejarah dan perkembangan kecerdasan buatan terjadi pada musim panas tahun 1956 tercatat adanya seminar mengenai AI di Darmouth College. Seminar pada waktu itu dihadiri oleh sejumlah pakar komputer dan membahas potensi komputer dalam meniru kepandaian manusia. Akan tetapi perkembangan yang sering terjadi semenjak diciptakannya LISP, yaitu bahasa kecerdasan buatan yang dibuat tahun 1960 oleh John McCarthy. Istilah pada kecerdasan buatan atau Artificial Intelligence diambil dari Marvin Minsky dari MIT. Dia menulis karya ilmiah berjudul Step towards Artificial Intelligence, The Institute of radio Engineers Proceedings 49, January 1961[1].

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

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

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

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

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

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

Testing set digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar[6].

1.4.2 Instalasi

1.4.2.1 Instalasi Library Scikit dari Anaconda

1. Download aplikasi Anaconda terlebih dahulu. Lihat pada gambar 1.4.
2. Install aplikasi Anaconda yang sudah di download tadi. Lihat pada gambar 1.5.
3. Simpan aplikasi sesuai folder yang kita pilih lalu next. Lihat pada gambar 1.6.
4. Centang Keduanya lalu tekan tombol install. Lihat pada gambar 1.7.
5. Setelah itu tunggu sampai proses instalasi selesai lalu jika sudah tekan tombol finish. Lihat pada gambar 1.8.
6. Lalu buka command prompt anda dan tuliskan perintah berikut ini untuk mengecek apakah aplikasinya sudah terinstall. Lihat pada gambar 1.9.
7. Kemudian ketikkan perintah `pip install -U scikit-learn` seperti gambar berikut. Lihat pada gambar 1.10.
8. Lalu jika sudah ketikkan juga perintah `conda install scikit-learn`. Lihat pada gambar 1.11.
9. Hasil compile dari beberapa code yang mempunyai variable explorer. Lihat pada gambar 1.12.

1.4.2.2 Mencoba Loading an example Dataset

- `from sklearn import datasets`

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

- `iris = datasets.load_iris()`

(pada baris kedua ini dimana iris merupakan suatu estimator/parameter yang berfungsi untuk mengambil data pada item datasets.load_iris).

- `digits = datasets.load_digits()`

(pada baris ketiga ini dimana digits merupakan suatu estimator/parameter yang berfungsi untuk mengambil data pada item datasets.load_digits).

- `print(digits.data)`

(pada baris keempat ini merupakan perintah yang berfungsi untuk menampilkan estimator/parameter yang dipanggil pada item `digits.data` dan menampilkan outputannya) Lihat gambar 1.13.

- `digits.target`

(barisan ini untuk mengambil target pada estimator/parameter `digits` dan menampilkan outputannya) Lihat gambar 1.14.

- `digits.images[0]`

(barisan ini untuk mengambil `images[0]` pada estimator/parameter `digits` dan menampilkan outputannya) Lihat gambar 1.15.

1.4.2.3 Learning and Predicting

- `from sklearn import svm`

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

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

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

- `clf.fit(digits.data[:-1], digits.target[:-1])`

(pada baris ketiga ini `clf` sebagai estimator/parameter, `fit` sebagai metode, `digits.data` sebagai item, `[:-1]` sebagai syntax pythonnya dan menampilkan outputannya) Lihat gambar 1.16.

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

(pada baris terakhir ini `clf` sebagai estimator/parameter, `predict` sebagai metode lainnya, `digits.data` sebagai item dan menampilkan outputannya) Lihat gambar 1.17.

1.4.2.4 Model Persistence

- `from sklearn import svm`

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

- `from sklearn import datasets`

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

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

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

- `iris = datasets.load_iris()`

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

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

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

- `clf.fit(X, y)`

(pada baris keenam ini clf sebagai estimator/parameter dengan menggunakan metode fit untuk memanggil estimator X, y dengan outputannya) Lihat gambar 1.18.

- `import pickle`

(pickle merupakan sebuah class yang di import).

- `s = pickle.dumps(clf)`

(pada baris ini s sebagai estimator/parameter dengan pickle.dumps merupakan suatu nilai/item dari estimator/parameter clf)

- `clf2 = pickle.loads(s)`

(pada baris ini clf2 sebagai estimator/parameter, pickle.loads sebagai suatu item, dan s sebagai estimator/parameter yang dipanggil)

- `clf2.predict(X[0:1])`

(pada baris ini clf2.predict sebagai suatu item dengan menggunakan metode predict untuk menentukan suatu nilai dari (X[0:1])) Lihat gambar 1.19.

- `y[0]`

(pada estimator/parameter `y` berapapun angka yang diganti nilainya akan selalu konstan yaitu 0) Lihat gambar 1.20.

- `from joblib import dump, load`

(pada baris berikut ini merupakan sebuah perintah untuk mengimport class `dump`, `load` dari packaged `joblib`).

- `dump(clf, 'filename.joblib')`

(pada baris berikutnya `dump` di sini sebagai class yang didalamnya terdapat nilai dari suatu item `clf` dan data `joblib`).

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

(pada baris terakhir `clf` sebagai estimator/parameter dengan suatu nilai `load` berfungsi untuk mengulang data sebelumnya)

- dari ketiga baris akhir tersebut jika di jalankan aau dituliskan perintah seperti itu maka akan menampilkan tampilan eror terlihat pada gambar 1.21.

1.4.2.5 Conventions

1. Type Casting

- `from sklearn import svm`

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

- `from sklearn import random_projection`

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

- `rng = np.random.RandomState(0)`

(`rng` sebagai estimator/parameter dengan nilai suatu itemnya yaitu `np.random.RandomS`

- `X = rng.rand(10, 2000)`

(`X` sebagai estimator/parameter dengan nilai item `rng.rand`).

- `X = np.array(X, dtype='float32')`

(`X` sebagai estimator/parameter dengan nilai item `np.array`).

- `X.dtype`
(`X.dtype` sebagai item pemanggil) Lihat gambar 1.22.
- `transformer = random_projection.GaussianRandomProjection()`
(`transformer` sebagai estimator/parameter dengan memanggil class `random_projection`).
- `X_new = transformer.fit_transform(X)`
(`X_new` di sini sebagai estimator/parameter dan menggunakan metode `fit`)
- `X_new.dtype`
(`X_new.dtype` sebagai item) Lihat gambar 1.23.
- `from sklearn import datasets`
(pada baris ini merupakan sebuah perintah untuk mengimport class `datasets` dari packaged `sklearn`).
- `from sklearn.svm import SVC`
(pada baris ini merupakan sebuah perintah untuk mengimport class `SVC` dari packaged `sklearn.svm`).
- `iris = datasets.load_iris()`
(`iris` sebagai estimator/parameter dengan item `datasets.load_iris()`).
- `clf = SVC(gamma='scale')`
(`clf` sebagai estimator/parameter dengan nilai class `SVC` pada parameter `gamma` sebagai set penilaian).
- `clf.fit(iris.data, iris.target)`
(estimator/parameter `clf` menggunakan metode `fit` dengan itemnya) Lihat gambar 1.24.
- `list(clf.predict(iris.data[:3]))`
(menambahkan item `list` dengan metode `predict`) Lihat gambar 1.25.
- `clf.fit(iris.data, iris.target_names[iris.target])`
(estimator/parameter `clf` menggunakan metode `fit` dengan itemnya) Lihat gambar 1.26.
- `list(clf.predict(iris.data[:3]))` (menambahkan item `list` dengan metode `predict`)
Lihat gambar 1.27.

2. Refitting and Updating Parameters

- `import numpy as np`
(pada baris ini merupakan sebuah perintah untuk mengimport class svm dari np).
- `from sklearn.svm import SVC`
(pada baris ini merupakan sebuah perintah untuk mengimport class SVC dari packaged sklearn.svm).
- `rng = np.random.RandomState(0)`
(rng sebagai estimator/parameter dengan nilai suatu itemnya yaitu np.random.RandomS
- `X = rng.rand(100, 10)`
(X sebagai estimator/parameter dengan nilai item rng.rand).
- `y = rng.binomial(1, 0.5, 100)`
(y sebagai estimator/parameter dengan nilai item rng.binomial).
- `X_test = rng.rand(5, 10)`
(X_test sebagai estimator/parameter dengan nilai item rng.rand).
- `clf = SVC()`
(clf sebagai estimator/parameter dan class SVC)
- `clf.set_params(kernel='linear').fit(X, y)`
(set_params sebagai item) Lihat gambar 1.28.
- `clf.predict(X_test)`
(menggunakan metode predict) Lihat gambar 1.29.
- `clf.set_params(kernel='rbf', gamma='scale').fit(X, y)`
Lihat gambar 1.30.
- `clf.predict(X_test)`
Lihat gambar 1.31.

3. Multiclass vs. Multilabel Fitting

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

- `from sklearn.preprocessing import LabelBinarizer`
(pada baris ini merupakan sebuah perintah untuk mengimport class LabelBinarizer dari packaged sklearn.preprocessing).
- `X = [[1, 2], [2, 4], [4, 5], [3, 2], [3, 1]]`
- `y = [0, 0, 1, 1, 2]`
- `classif = OneVsRestClassifier(estimator=SVC(gamma='scale', random_state=0))`
- `classif.fit(X, y).predict(X)`

Lihat gambar 1.32.

- `y = LabelBinarizer().fit_transform(y)`
- `classif.fit(X, y).predict(X)`

Lihat gambar 1.33.

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

Lihat gambar 1.34.

1.4.3 Penanganan error

1.4.3.1 ScreenShoot Error

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

Figure 1.1: Hasil Tampilan Error.

1.4.3.2 Tuliskan Kode Error dan Jenis Erornya

- `from joblib import dump, load`

(Kode baris pertama)

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

(Errornya)

- `dump(clf, 'filename.joblib')`

(Kode baris kedua)

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

(Errornya)

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

(Kode baris ketiga)

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

(Errornya)

1.4.3.3 Solusi Pemecahan Masalah Error

1. Pada masalah error sebelumnya itu dikarenakan kita belum mempunyai packaged joblib. Jadi solusinya yaitu dengan cara menginstall terlebih dahulu packaged joblibnya setelah itu baru perintah tersebut dapat dijalankan seperti pada gambar 1.2 dan 1.3

===== iiiiii HEAD

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

C:\Users\HUDA>pip install joblib
Requirement already satisfied: joblib in f:\anaconda\lib\site-packages (0.13.2)
distributed 1.21.8 requires msgpack, which is not installed.
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:\Users\HUDA>
```

Figure 1.2: Hasil Tampilan Install joblib.

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

Figure 1.3: Hasil Tampilan Uji coba perintah joblib.

1.5 Cokro Edi Prawiro/1164069

1.5.1 Praktek teori penunjang

1. Kecerdasan Buatan *Artificial Intelligence* adalah suatu cabang dalam bidang sains komputer yang mengkaji bagaimana untuk melengkapi sebuah komputer dengan kemampuan atau kepintaran seperti manusia. Komputer tersebut diharapkan dapat belajar sendiri dengan cara mengumpulkan data-data yang diterimanya, yang berguna sebagai parameter untuk memecahkan masalah. Jadi kecerdasan buatan merupakan kecerdasan yang di program dalam komputer untuk memecahkan masalah secara tepat dan cepat atau untuk memberikan kemungkinan keberhasilan dan kegagalan pada solusi dari suatu masalah.

Adapun kecerdasan buatan menurut para ahli adalah sebagai berikut :

- Kecerdasan Buatan merupakan Kawasan penelitian, aplikasi dan intruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas (H. A. Simon[1997]).
- Kecerdasan buatan adalah bidang studi yang berhubungan dengan penangkapan, pemodelan, dan penyimpanan kecerdasan manusia dalam sebuah sistem teknologi sehingga sistem tersebut dapat memfasilitasi proses pengambilan keputusan yang biasanya dilakukan oleh manusia (Haag dan Keen[1996]).

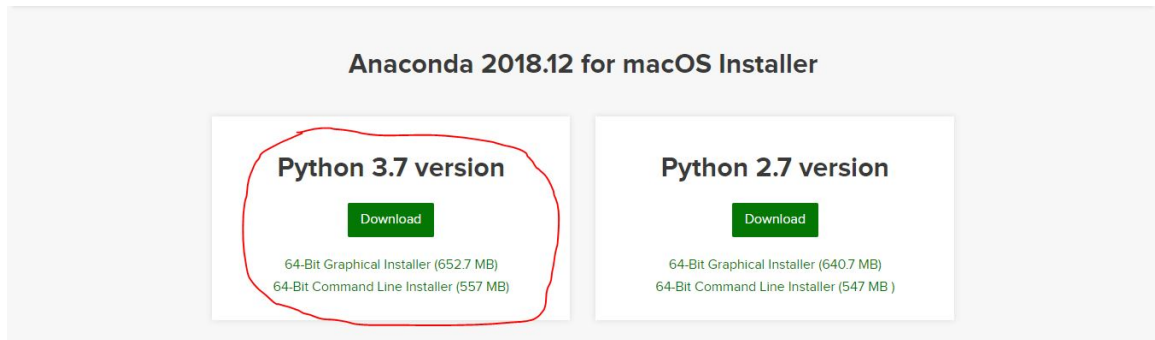


Figure 1.4: Download Anaconda.

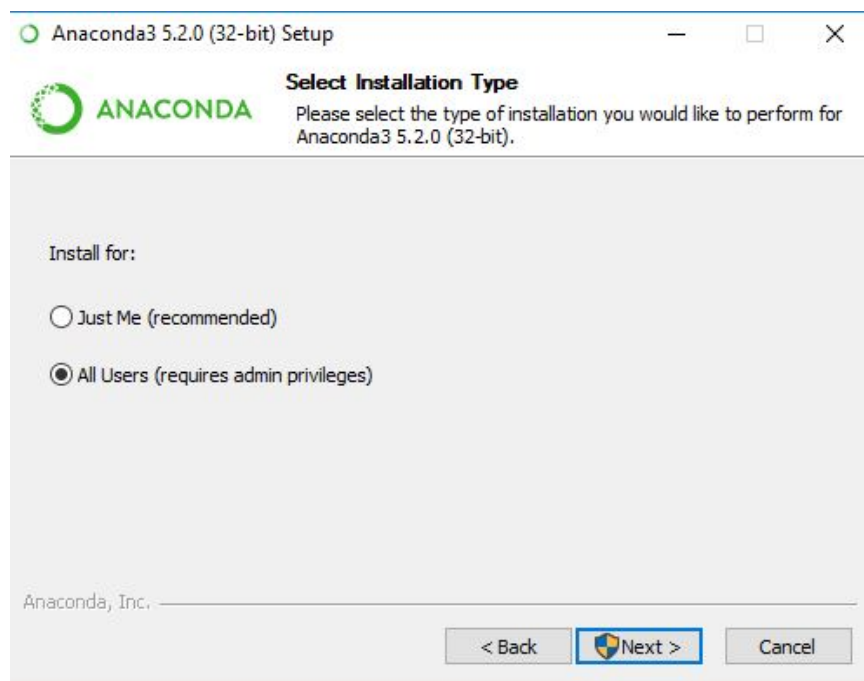


Figure 1.5: Langkah pertama instalasi anaconda.

Sejarah dan Perkembangan Kecerdasan Buatan.

ketika *Rene Descartes* mengemukakan gagasan yang menjadi cikal bakal kecerdasan buatan pada abad 17 mengemukakan bahwa hewan bukan apa-apa melainkan hanya mesin yang rumit yang dilanjutkan oleh *Belaise Pascal* yang telah menciptakan mesin penghitung digital mekanis pertama pada tahun 1642. Lalu pada abad 19 *Charles Babbage* dan *Ada lovelace* bekerja sama membuat mesin penghitung mekanis yang dapat di program.

Perkembangan kecerdasan buatan inipun terus berlanjut, *Bertrand Russell* dan *Alferd North Whithead* menerbitkan *mathematica*, yang merombak logika formal. Setelah itu dilanjutkan dengan penemuan oleh *Warren McCulloch* dan

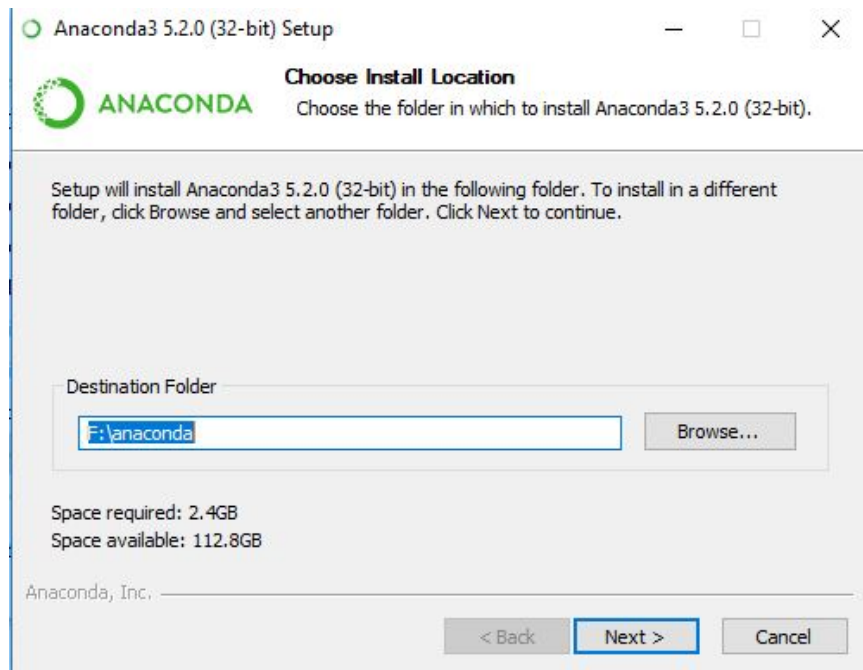


Figure 1.6: Langkah kedua instalasi anaconda.

Walter Pitts menerbitkan “Kalkulus Logis Gagasan yang tetap ada dalam Aktivitas” pada 1943 yang meletakkan pondasi awal berupa jaringan syaraf. Kemudian pada tahun 1950-an adalah periode awal usaha aktif kecerdasan buatan. Program Kecerdasan buatan pertama yang bekerja diciptakan pada tahun 1951 untuk menjalankan mesin Ferranti Mark I di University of Manchester (UK) yaitu sebuah program permainan naskah yang ditulis oleh *Christopher Strachey* dan program permainan catur yang ditulis oleh *Dietric Prinz*. Kemudian pada konferensi pertama tahun 1956 *John McCarthy* mengemukakan istilah “kecerdasan buatan” kemudian dia juga menemukan bahasa pemrograman lisp. *Joseph Weizenbaum* menciptakan ELIZA, sebuah chatterbot yang menerapkan psikoterapi Rogerian.

Selama rentang waktu tahun 1960-an dan 1970-an, *Joel Moses* mendemonstrasikan kekuatan pertimbangan simbolis untuk mengintegrasikan masalah di dalam program Macsyma, yang merupakan program yang pertamakali sukses dalam bidang matematika. Kemudian pada tahun 1980-an industri kecerdasan buatan ini berkembang walau sudah di mulai pada tahun 1970-an. Evolusi kecerdasan buatan berjalan dalam dua jalur yang berbeda yaitu meniru proses berpikir manusia untuk menyelesaikan masalah umum. Kedua mengkombinasikan pemikiran terbaik para ahli pada sepotong software yang dirancang

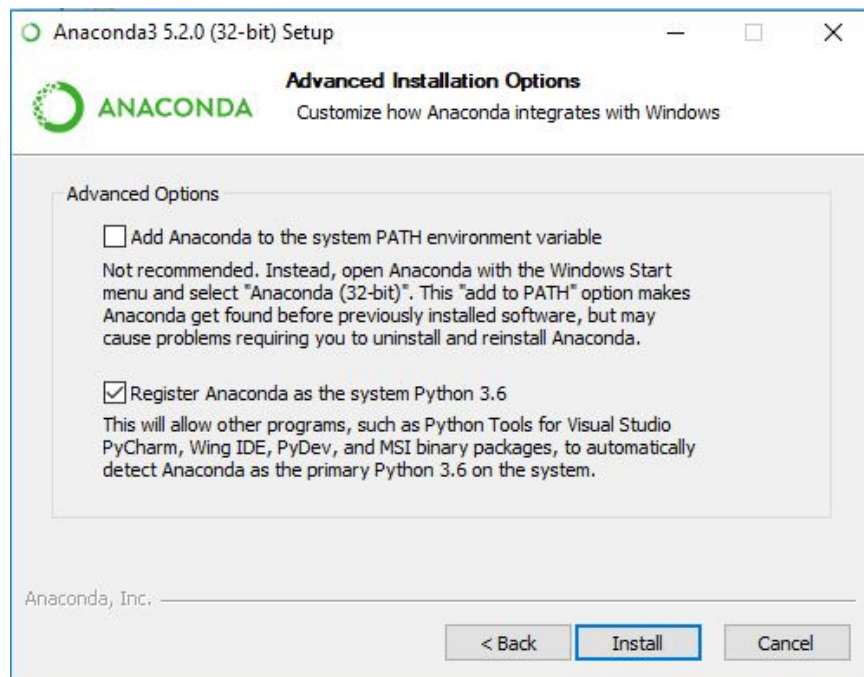


Figure 1.7: Langkah ketiga instalasi anaconda.

untuk memecahkan persolalan yang spesifik.

2. Supervised learning adalah sebuah pendekatan dengan syarat sudah terdapat data yang dilatih kemudian harus terdapat variable yang ditargetkan sehingga tujuan dari pendekatan ini adalah pengelompokan data terhadap data yang telah ada. Ciri khas dari Supervised learning yaitu terdapat label atau nama kelas pada data latih (supervisi) dan data baru di klasifikasikan berdasarkan data latih. Data latih dikelompokkan berdasarkan ukuran kemiripan pada suatu kelas. Berdasarkan keluaran dari fungsi, Supervised learning dibagi menjadi 2, regresi dan klasifikasi. Regresi terjadi jika output dari fungsi merupakan nilai yang kontinyu, sedangkan klasifikasi terjadi jika keluaran dari fungsi adalah nilai tertentu dari suatu atribut (tidak kontinyu). Tujuan dari Supervised learning adalah untuk memprediksi nilai dari fungsi untuk sebuah data masukanyang sah setelah melihat sejumlah data latih[3].

Adapun pengertian klasifikasi dan regresi adalah sebagai berikut :

- Klasifikasi merupakan pengelompokan berdasarkan parameter tertentu yang tidak konstan contoh pada mahluk hidup yaitu persamaan ciri cara hidup dan tempat hidup.

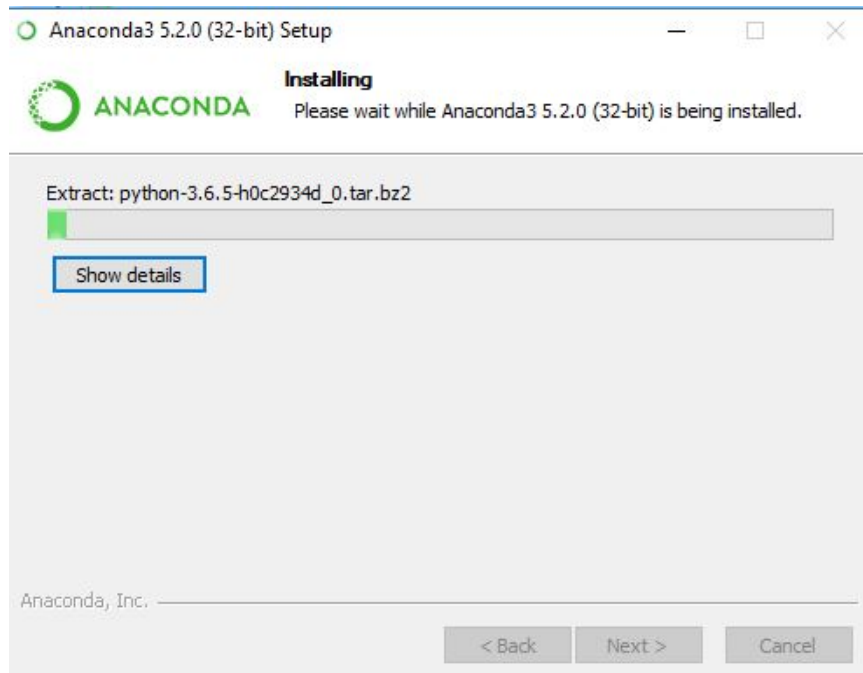


Figure 1.8: Langkah terakhir instalasi anaconda.

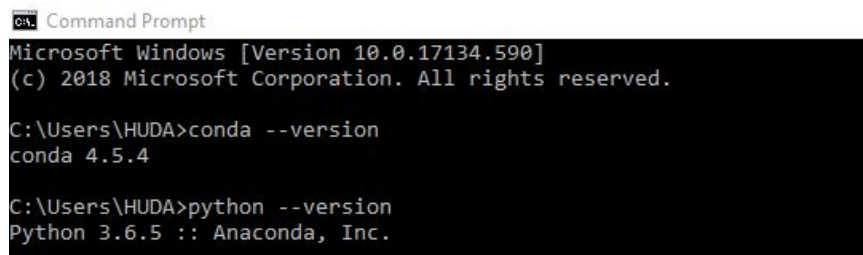


Figure 1.9: Langkah pertama instalasi scikit pada CMD.

- Regresi yaitu pengeluaran nilai output yang konstan jika dipicu dengan parameter tertentu biasanya regresi disini berbentuk regresi linier. Regresi linier yaitu metode statistika yang digunakan untuk membentuk model hubungan antara variabel terikat(dependen,respon,Y) dengan satu atau lebih variabel bebas(independent, prdikator, X). Apabila banyaknya variabel bebas hanya ada satu, disebut sebagai regresi linier sederhana, sedangkan apabila terdapat lebih dari satu variabel bebas, disebut sebagai regresi linier berganda [4].

unsupervised learning adalah pendekatan yang tidak memerlukan data latih atau data training untuk melakukan prediksi atau klasifikasi. Berdasarkan model secara matematisnya, algoritma ini tidak memiliki target variabel. Tu-


```
C:\Users\HUDA>pip install -U scikit-learn
Collecting scikit-learn
  Downloading https://files.pythonhosted.org/packages/ee/c8/c89ebdc0d7dbbafefcd222daabd257da3c28a967dd7c352d4272b2e1cef6/scikit_learn-0.20.2-cp36-cp36m-win32.whl (4.3MB)
    100% |#####| 4.3MB 425KB/s
Requirement not upgraded as not directly required: numpy>=1.8.2 in f:\anaconda\lib\site-packages (from scikit-learn) (1.14.3)
Requirement not upgraded as not directly required: scipy>=0.13.3 in f:\anaconda\lib\site-packages (from scikit-learn) (1.1.0)
Installing collected packages: scikit-learn
  Found existing installation: scikit-learn 0.19.1
    Uninstalling scikit-learn-0.19.1:
      Successfully uninstalled scikit-learn-0.19.1
Successfully installed scikit-learn-0.20.2
You are using pip version 18.0.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Figure 1.10: Langkah kedua instalasi scikit pada CMD.

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

## Package Plan ##

  environment location: F:\anaconda

  added / updated specs:
    - scikit-learn

The following packages will be downloaded:

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

The following packages will be UPDATED:

  conda: 4.5.4-py36_0 --> 4.6.7-py36_0

Proceed ([y]/n)? y

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

Figure 1.11: Langkah ketiga instalasi scikit pada CMD.

juan dari algoritma ini yaitu pengelompokan objek yang memiliki kesamaan atau hampir sama dalam satu cakupan wilayah tertentu. Kemudian pada unsupervised learning tidak terdapat label atau nama kelas pada data latih. Kemudian dataset merupakan objek yang menggambarkan data itu sendiri dan relasinya di memory. Struktur datanya mirip dengan struktur data di basis data. Jadi strikturnya terdiri atas baris kolom dan juga ada sejenis relasi data. Pada dataset terdiri bagian bagian yaitu training set dan Testing set. Adapun pengertian dari training set dan Testing set adalah sebagai berikut :

- Training set adalah bagian dari dataset itu sendiri yang dilatih untuk membuat prediksi atau algoritma mesin learning lainnya sesuai keinginan atau tujuan data itu dibuat.
- Testing set adalah bagian dari dataset yang di tes atau diujicoba untuk melihat keakuratannya dengan katalain melihat peformanya.

```

C:\Users\HUDA>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from sklearn.metrics import confusion_matrix
>>> y_true = [2, 0, 2, 2, 0, 1]
>>> y_pred = [0, 0, 2, 2, 0, 2]
>>> confusion_matrix(y_true, y_pred)
array([[2, 0, 0],
       [0, 0, 1],
       [1, 0, 2]], dtype=int64)
>>> y_true = ["cat", "ant", "cat", "cat", "ant", "bird"]
>>> y_pred = ["ant", "ant", "cat", "cat", "ant", "cat"]
>>> confusion_matrix(y_true, y_pred, labels=["ant", "bird", "cat"])
array([[2, 0, 0],
       [0, 0, 1],
       [1, 0, 2]], dtype=int64)
>>> tn, fp, fn, tp = confusion_matrix([0, 1, 0, 1], [1, 1, 1, 0]).ravel()
>>> (tn, fp, fn, tp)
(0, 2, 1, 1)
>>>

```

Figure 1.12: Langkah compile code pada python anaconda.

```

[[ 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 Tampilan 1.

1.5.2 Instalikasi

Pada proses instalikasi ini langkah pertama yaitu mengakses website scikit dengan mengakses link berikut <https://scikit-learn.org/stable/tutorial/basic/tutorial.html> maka hasilnya dapat dilihat pada gambar 1.35 kemudian setelah itu klik button installation maka akan muncul tampilan yang dapat dilihat pada gambar 1.36.

1. cara instalikasi Instalasi library scikit dari anaconda langkah pertama instal terlebih dahulu anacondanya dikarenakan anaconda sudah include dengan python maka codingan python dapat digunakan di anaconda dan ketika diperikas versinya maka akan muncul tampilan seperti Gambar 1.37

kemudian pada cmd administrator install library scikit dengan cara memasukan codingan `pip install -U scikit-learn` maka hasilnya seperti seperti pada gambar 1.38 berikut.

setelah itu masukan kembali perintah berikut di cmd `conda install scikit-learn` jika hasilnya seperti pada gambar 1.39 maka librari scikit telah terinstal dan siap untuk di gunakan.

```
array([0, 1, 2, ..., 8, 9, 8])
```

Figure 1.14: Hasil Tampilan 2.

```
array([[ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.],
       [ 0.,  0., 13., 15., 10., 15.,  5.,  0.],
       [ 0.,  3., 15.,  2.,  0., 11.,  8.,  0.],
       [ 0.,  4., 12.,  0.,  0.,  8.,  8.,  0.],
       [ 0.,  5.,  8.,  0.,  0.,  9.,  8.,  0.],
       [ 0.,  4., 11.,  0.,  1., 12.,  7.,  0.],
       [ 0.,  2., 14.,  5., 10., 12.,  0.,  0.],
       [ 0.,  0.,  6., 13., 10.,  0.,  0.,  0.]])
```

Figure 1.15: Hasil Tampilan 3.

kemudian untuk mencobanya tuliskan perintah python pada cmd lalu masukan codingan `print("Hello Anaconda!")` maka hasilnya terlihat seperti gambar 1.40 codingan `print("Hello Anaconda!")` yaitu berfungsi mencetak nilai yang ada di dalam kurung dan di antara kutip.

2. cara mencoba dataset yaitu dengan cara memasukan perintah berikut pada cmd seperti pada gambar 1.41

- pada codingan `from sklearn import datasets` menjelaskan inport librari dataset dari librari sikic pada gambar 1.41
- pada codingan `iris = datasets.load_iris` iris berarti parameter atau acuan bernama iris kemudian di load ke dalam dataset sebagai perbandingan kalau dalam diagram iris bisa disebut X nya pada gambar 1.41
- pada codingan `digits = datasets.load_digits` digits berarti parameter hampir mirip seperti iris tadi namun digits merupakan kebalikannya kalau di dalam diagram dia bernilai Y pada gambar 1.41
- kemudian pada codingan `print(digits.data)` yaitu mencetak data digits yang di bandingkan dengan data iris pada gambar 1.41
- pada codingan `print(iris.data)` yaitu mencetak data iris yang dibandingkan dengan data digits pada gambar 1.41

3. Mencoba Learning and Predicting

Pada kasus ini dataset digit digunakan untuk memprediksi yang mana telah di berikan gambar untuk mewakili sampel masing-masing dari 10 kelas yang dimulai dari digit nol hingga sembilan yang digunakan untuk memprediksi sampel

```
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 1.16: Hasil Tampilan 4.

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

Figure 1.17: Hasil Tampilan 5.

yang tidak terlihat untuk lebih jelasnya dapat di praktikan codingan berikut ini.

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

pada baris ini dapat diartikan bahwa librari sklearn mengimport package dataset

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

pada baris ini dimasukan parameter iris yang di sandingkan dengan dataset load sehingga iris berisi nilai dataset

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

pada baris ini dimasukan parameter digits yang di sandingkan dengan dataset load sehingga digits berisi nilai dataset

```
>>> from sklearn import svm
```

pada baris ini librari sklearn mengimport package svm

```
>>> clf = svm.SVC(gamma=0.0001, C=100.)
```

pada codingan diatas dibuat variabel clf yang di isi dengan nilai svm dengan nilai gama 0.0001 dan 100.

```
>>> clf.fit(digits.data[:-1], digits.target[:-1])
```

pada codingan clf di implementasikan dengan perintah fit

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Figure 1.18: Hasil Tampilan 6.

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

Figure 1.19: Hasil Tampilan 7.

```
SVC(C=100.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.0001, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

yang hasilnya seperti codingan diatas yang menjabarkan isi dari SVC itu sendiri.

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

kemudain pada codingan diatas digunakan perintah predic yang merupakan perintah untuk mengimplementasikan method digits.

4. Mencoba model pertistence

model pertistence yaitu model yang digunakan untuk mengolah data sehingga data tersebut konstan atau konsisten terhadap parameter tertentu contoh pada codingan di bagawah nilai y akan konstan di nol walaupun telah di isi nilai lebih dari nol.

```
>>> from sklearn import svm
```

pada baris ini librari sklearn mengimport package svm.

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

pada baris ini librari sklearn mengimport package datasets.

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

Figure 1.20: Hasil Tampilan 8.

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

Figure 1.21: Hasil Tampilan 9.

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

pada codingan diatas dibuat variabel clf yang di isi dengan nilai svm dengan gamma sama dengan scale.

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

pada baris ini dimasukan parameter iris yang di sandingkan dengan dataset load sehingga iris berisi nilai dataset.

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

pada codingan diatas X berisi nilai iris.data dan y berisi nilai iris.target.

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

method clf di implementasikan dengan perintah fit dengan X, y sebagai nilai untuk implementasinya.

```
>>> X.dtype
dtype('float32')
```

Figure 1.22: Hasil Tampilan 10.

```
>>> X_new.dtype
dtype('float64')
```

Figure 1.23: Hasil Tampilan 11.

```
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)
```

maka hasilnya penjabaran dari SVC seperti codingan diatas.

```
>>> import pickle
```

mengimport library atau package pickle.

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

kemudian di buat variabel s yang di load oleh package pickle dengan di isi nilai clf.

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

setelah itu pada codingan diatas dibuat lagi variabel clf2 kemudian di load pickle.

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

kemudian variabel clf2 di implementasikan dengan parameter X dengan nilai 0 berbanding 1 maka hasilnya nilainya array bernilai nol dan y bernilai nol.

```
>>> 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)
```

Figure 1.24: Hasil Tampilan 12.

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

Figure 1.25: Hasil Tampilan 13.

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

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

```
0
```

5. mencoba conventions conventions merupakan aturan aturan dasar atau kesepakatan kesepakatan dalam pemrograman sikit python dan anaconda berikut merupakan jenis-jenis codingan conventions :

- Type casting yaitu tipe pelemparan parameter atau variabel kedalam variabel baru.

```
>>> import numpy as np
```

codingan diatas yaitu import librari numpy yang di inisialisasi menjadi np

```
>>> from sklearn import random_projection
```

import librari random_projection

```
>>> rng = np.random.RandomState(0)
```

membuat variabel baru dengan nama rng dengan nilai random

```
>>> X = rng.rand(10, 2000)
```

memasukan nilai rng kedalam variabel X dengan rad nilai 10 sampai 2000

```
>>> X = np.array(X, dtype='float32')
```

menambahkan nilai np berupa array yaitu X dan float 32

```
>>> X.dtype
```

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

X.dtype di running menghasilkan nilai dtype float


```
>>> 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)
```

Figure 1.26: Hasil Tampilan 14.

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

Figure 1.27: Hasil Tampilan 15.

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

membuat variabel transformer dengan nilai random

```
>>> X_new = transformer.fit_transform(X)
```

membuat variabel X_new dan di isi nilai transformer kemudian di implementasikan

```
>>> X_new.dtype
```

merunning variabel X_new

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

hasil running X_new

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

mengimport library dataset

```
>>> from sklearn.svm import SVC
```

mengimport library SVC

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

membuat variabel iris dengan nilai load dataset

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

membuat variabel clf bernilai SVC dengan gamma menggunakan nilai sekala

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

```
>>> 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)
```

Figure 1.28: Hasil Tampilan 16.

```
>>> clf.predict(X_test)
array([1, 0, 1, 1, 0])
```

Figure 1.29: Hasil Tampilan 17.

merunning variabel atau method clf dengan isian nilai iris data dan 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)
```

detail hasil runing clf

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

memunculkan detail atau lis sebanyak tiga nilai

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

merunning kembali clf dengan nilai iris data, iris target name

```
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)
```

hasil dari running clf

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

memunculkan tiga nilai yang telah dilempar dari SVC

```
>>> 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)
```

Figure 1.30: Hasil Tampilan 18.

```
>>> clf.predict(X_test)
array([1, 0, 1, 1, 0])
```

Figure 1.31: Hasil Tampilan 19.

- Refitting and updating parameters atau pengisian ulang atau memperbarui parameter merupakan cara untuk merubah nilai dari sebuah parameter contoh nilai x adalah 10 jika di perbaharui bisa menjadi 15 begitu juga dalam codiangan berikut hal ini dapat dilakukan untuk lebih jelasnya dapat dilihat codingan dibawah ini :

```
>>> import numpy as np
```

codingan diatas yaitu import librari numpy yang di inisialisasi menjadi np

```
>>> from sklearn.svm import SVC
```

mengimport library SVC

```
>>> rng = np.random.RandomState(0).
```

membuat variabel baru dengan nama rng dengan nilai random.

```
>>> X = rng.rand(100, 10)
```

parameter X dengan nilai dari variabel rng dan rad dari 100 sampai 10.

```
>>> y = rng.binomial(1, 0.5, 100)
```

parameter y dengan nilai rng binominal dari 1 0,5 sampai 100.

```
>>> X_test = rng.rand(5, 10)
```

parameter X_test dengan nilai rng dan rad dari 5 ke 10

```
>>> clf = SVC()
```

parameter clf bernilai SVC

```
>>> clf.set_params(kernel='linear').fit(X, y)
```

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

Figure 1.32: Hasil Tampilan 20.

```
>>> clf.fit(X, y).predict(X)
array([[1, 0, 0],
       [1, 0, 0],
       [0, 1, 0],
       [0, 0, 0],
       [0, 0, 0]])
```

Figure 1.33: Hasil Tampilan 21.

parameter clf di set dengan mengkompilasi atau mengekstrak nilai X dan y dengan kernel linear.

```
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)
```

penjabaran nilai SVC hasil running CLF

```
>>> clf.predict(X_test)
array([1, 0, 1, 1, 0])
```

merunning clf dengan nilai X_test

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

parameter clf di set dengan kernel rbf dan gamma skala dan mengkompilasi nilai X dan 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)
```

penjabaran nilai SVC hasil running CLF

```
>>> 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 1.34: Hasil Tampilan 22.

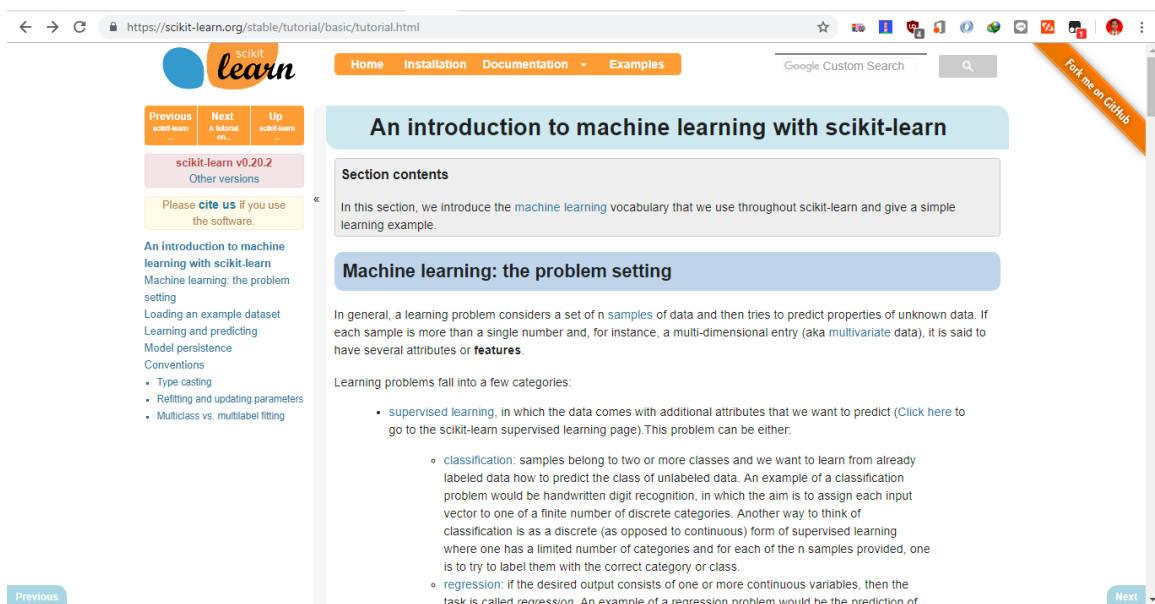


Figure 1.35: Tampilan website Scikit 1.

```
>>> clf.predict(X_test)
array([1, 0, 1, 1, 0])
```

Hasil dari running clf.

- Multiclass vs. multilabel fitting perbandingan antara banyak kelas dan pelabelan yang tepat berikut merupakan codingannya.

```
>>> from sklearn.svm import SVC
```

mengimport library SVC

```
>>> from sklearn.multiclass import OneVsRestClassifier
```

memasukan librari OneVsRestClassifier dengan kondisi multi class.

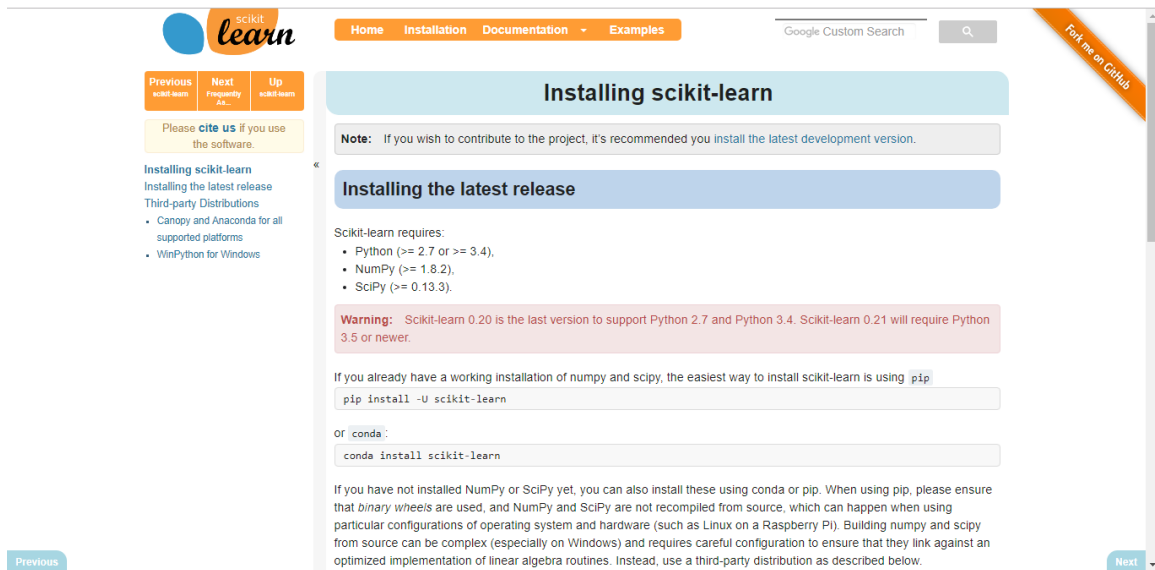


Figure 1.36: Tampilan website Scikit 2.

```
>>> from sklearn.preprocessing import LabelBinarizer
```

memasukan librari LabelBinarizer

```
>>> X = [[1, 2], [2, 4], [4, 5], [3, 2], [3, 1]]
```

pemberian nilai pada parameter X

```
>>> y = [0, 0, 1, 1, 2]
```

pemberian nilai pada parameter y

```
>>> classif = OneVsRestClassifier(estimator=SVC(gamma='scale',
... random_state=0))
```

opsi untuk class dengan ketentuan estimator SVC gamma berbentuk skala dan random (acak).

```
>>> classif.fit(X, y).predict(X)
```

```
array([0, 0, 1, 1, 2])
```

hasil array dari running classif

```
>>> y = LabelBinarizer().fit_transform(y)
```

memberikan nilai pada parameter y

```
>>> classif.fit(X, y).predict(X)
```

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

```
[1, 0, 0],
[0, 1, 0],
[0, 0, 0],
[0, 0, 0]])
```

hasil running classif

```
>>> from sklearn.preprocessing import MultiLabelBinarizer
```

import librari multi label

```
>>> y = [[0, 1], [0, 2], [1, 3], [0, 2, 3], [2, 4]]
```

memberikan nilai pada parameter y

```
>>> y = MultiLabelBinarizer().fit_transform(y)
```

membuat parameter y menjadi multi label.

```
>>> classif.fit(X, y).predict(X)
```

```
=====
```

```
\section{Fathi Rabbani / 1164074}
```

```
\subsection{Teori}
```

```
\begin{enumerate}
```

```
\item
```

Sejarah dan Perkembangan Kecerdasan Buatan

```
\subitem
```

Sejarah dari sebuah Artificial Intelligence atau dalam Bahasa indonesian

```
\begin{itemize}
```

```
\item
```

Symbol Manipulating AI

```
\item
```

Nueral AI

```
\item
```

Neural Network

```
\end{itemize}
```

```
\subitem
```

Peneliti yang selalu disebutkan sebagai Bapak AI adalah Jhon McCharty me

```
\subitem
```

Sedangkan perkembangan kecerdasan buatan saat ini sudah mencapai tahap di

- \item Definisi Supervised, Unsupervised Learning, Klasifikasi, Regresi serta Data Set
- \begin{itemize}
- \item Supervised learning merupakan sebuah pendekatan AI dengan latihan yang supervised
- \item Unsupervised learning merupakan sebuah pendekatan AI tanpa menggunakan data
- \item Klasifikasi merupakan sebuah pengelompokkan suatu objek ke dalam kategori
- \item Regresi merupakan pendekatan model matematika untuk mendeskripsikan hubungan
- \item Data Set, merupakan sebuah objek yang merepresentasikan data dan hubungan
- \item Training Set, subset untuk melatih model.
- \item Testing Set, subset untuk menguji model yang sudah dilatih.
- \end{itemize}

\subsection{Praktikum}

- \item Instalasi Library Scikit dari Anaconda

\subitem

Pertama Download terlebih dahulu anaconda-nya di <https://www.anaconda.com>

- \begin{itemize}
- \item Proses Instalasi Anaconda pada gambar \ref{proses2} hingga proses \ref{proses3}
- \item Proses Instalasi Scikit-Learn dengan menggunakan Conda pada gambar \ref{proses4}
- \item contoh dari Variable Explorer yang digunakan ada pada gambar \ref{proses5}
- \end{itemize}

\begin{figure}[ht]


```

\centerline{\includegraphics[width=1\textwidth]{figures/fathi/2.PNG}}
\caption{setelah membuka data instalasi klik next}
\label{proses2}

```

```

\centerline{\includegraphics[width=1\textwidth]{figures/fathi/3.PNG}}
\caption{pilih i agree}
\label{proses3}
\end{figure}

```

```

\begin{figure}
\centerline{\includegraphics[width=1\textwidth]{figures/fathi/4.PNG}}
\caption{pilih instalasi Just Me}
\label{proses4}

```

```

\centerline{\includegraphics[width=1\textwidth]{figures/fathi/5.PNG}}
\caption{langsung saja next}
\label{proses5}
\end{figure}

```

```

\begin{figure}
\centerline{\includegraphics[width=1\textwidth]{figures/fathi/6.PNG}}
\caption{cek kedua pilihan tersebut}
\label{proses6}

```

```

\centerline{\includegraphics[width=1\textwidth]{figures/fathi/7.PNG}}
\caption{proses Instalasi}
\label{proses7}
\end{figure}

```

```

\begin{figure}
\centerline{\includegraphics[width=1\textwidth]{figures/fathi/8.PNG}}
\caption{klik next}
\label{proses8}

```

```

\centerline{\includegraphics[width=1\textwidth]{figures/fathi/9.PNG}}
\caption{selesai instalasi anaconda}
\label{proses9}
\end{figure}
\begin{figure}

```

```

\centerline{\includegraphics[width=1\textwidth]{figures/fathi/10.PNG}}
\caption{Instalasi SCIKIT dengan menggunakan anaconda}
\label{proses10}

```

```

\centerline{\includegraphics[width=1\textwidth]{figures/fathi/11.PNG}}
\caption{Konfirmasi Instalasi}
\label{proses11}
\end{figure}

```

```

\begin{figure}
\centerline{\includegraphics[width=1\textwidth]{figures/fathi/12.PNG}}
\caption{hasil dari instalasi SCIKIT}
\label{proses12}
\centerline{\includegraphics[width=1\textwidth]{figures/fathi/14.PNG}}
\caption{data variable explorer}
\label{proses14}
\end{figure}

```

```

\item

```

Load Example Dataset dan Menjeleaskan kegunaan barisan Code

```

\subitem

```

berikut ini adalah contoh dataset yang digunakan untuk melakukan compile

```

\begin{figure}
\centerline{\includegraphics[width=1\textwidth]{figures/fathi/16.PNG}}
\caption{code example dataset yang digunakan}
\label{proses16}

```

```

\centerline{\includegraphics[width=1\textwidth]{figures/fathi/15.PNG}}
\caption{data hasil dari code example dataset yang digunakan}
\label{proses15}

```

```

\end{figure}

```

```

\begin{itemize}

```

```

\item

```

dari code yang dicoba diketahui bahwa data set yang digunakan adalah data

```

\end{itemize}

```

```

\begin{itemize}

```

\item

Learning and Predicting

\subitem

Dalam scikit-learn estimator untuk klasifikasi adalah sebuah objek data

\begin{verbatim}

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

mengambil nilai data svm ada pada class sklearn, lalu set nilai data dengan `clf = sv.SVC(gamma=0.001, C=100.)`. variable `clf` digunakan dengan method `fit` yang di set nilainya `[:-1]` yang memproduksi array baru dari data `digits.data`, dengan menggunakan `digits.data` sebagai acuan, sekarang tinggal melakukan prediksinya.

- Model Persistence

```
from sklearn import svm
from sklearn import datasets
clf = svm.SVC(gamma=0.001)
iris = datasets.load_iris()
X, y = iris.data, iris.target
clf.fit(X, y)
```

mengambil nilai data svm ada pada class sklearn dan mengambil nilai data `datasets` ada pada class sklearn, lalu buat variable `clf` dengan nilai data dan buat variable yang berisi nilai `load_iris`. variable `X` dan `y` yang berisi nilai iris data dan iris target, lalu memanggil nilai variable `X` dan `y` dengan data variable `clf` dan method `fit`.

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

mengambil nilai data pickle dan membuat variable `s` dengan data nilai `pickle.dumps` yang berisi data variable `clf`, membuat variable `clf2` dengan

data pickle.loads yang menggunakan variable s. menggunakan data variable clf2 dengan method predict dengan data variable X dan data variable y.

- Conventions

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

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

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

mengambil data numpy dan dialiaskan sebagai np. dari data sklearn mengambil data random_projection. lalu buat variable rng yang berisi nilai data np dengan random yang berawal 0. lalu variable X dengan data rng yang memiliki type rand berisi data 10 dan 2000. lalu dibuatkan arraynya dengan format `X = np.array(X, dtype='float32')`. dan nilai variable transform yang digunakan untuk menampilkan hasil random, dengan format Gaussian random projection. format penampilannya `X_new = transformer.fit_transform(X)` dan `X_new.dtype`

```
– from sklearn import datasets
  from sklearn.svm import SVC
```

```
iris = datasets.load_iris()
clf = SVC(gamma=0.001)
clf.fit(iris.data, iris.target)
```

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

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

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

dari data sklearn mengambil datasets dan SVC dari svm, selanjutnya membuat format data variable dengan nilai load_iris, clf dengan format `SVCgamma = 0.001`, lalu di jalankan dengan moethod `clf.fit` dengan `iris.data` dan `iris.target` sebagai nilainya.

buatkan tampilan datanya dengan list menampilkan data clf dengan `predict` pada `iris.data`, dan dilakukan selanjutnya dengan `clf.fit` dengan nilai data `iris.data` dan `iris.target_names[iris.target]`. tampilkan lagi dalam bentuk list data tersebut.

```
- 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)

  clf.predict(X_test)

  clf.set_params(kernel='rbf', gamma=0.001).fit(X, y)

  clf.predict(X_test)
```

mengambil data numpy dan dialiaskan sebagai np dan dari sklearn.svm mengambil data SVC, lalu buat variable rng yang berisi nilai data np dengan random yang berawal 0, lalu variable X dengan data rng yang memiliki type rand berisi data(100, 10) dan y yang memiliki data (1, 0.5, 100) dengan type data binomial lalu buat X_test untuk variable test.

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

  X = [[1, 2], [2, 4], [4, 5], [3, 2], [3, 1]]
  y = [0, 0, 1, 1, 2]
```

```
classif = OneVsRestClassifier(estimator=SVC(gamma=1, random_state=0))
```

```
classif.fit(X, y).predict(X)
```

```
y = LabelBinarizer().fit_transform(y)
```

```
classif.fit(X, y).predict(X)
```

mengambil data sklearn SVC dari svm, OneVsRestClassifier dari multiclass, dan LabelBinarizer dari preprocessing. lalu buat variable X dan y yang berisi data nilai dan buat data variable classif untuk digunakan sebagai parameter bagi OneVsRestClassifier yang menghitung data estimator SVC. lalu jalankan dengan method fit dan predict, lalu variable y digunakan sebagai parameter yang menjalankan method fit_transform yang berisi data LabelBinarizer.

```
– from sklearn.preprocessing import MultiLabelBinarizer
```

```
y = [[0, 1], [0, 2], [1, 3], [0, 2, 3], [2, 4]]
```

```
y = MultiLabelBinarizer().fit_transform(y)
```

```
classif.fit(X, y).predict(X)
```

mengambil data dari sklearn datanya adalah MultiLabelBinarizer dari preprocessing, buat variable y yang berisikan nilai integer untuk di proses agar menghasilkan data seperti berikut

```
>>>>>> c6dbbab89a86daa2bf691a75d9a15c4b56fb15d7
```

```
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]])
```

|||||| HEAD hasil running classif setelah nilai parameter y telah di ganti.

1.5.3 Penanganan Error

- Screenshot Error untuk lebih jelasnya Screenshot codingan dapat dilihat pada gambar 1.42 dan 1.43
- Kode error pada screenshot untuk kode error pada gambar 1.42 yaitu pada source code berikut

```
clf.fit(digits.data[:-1], digits.target[:-1])
```

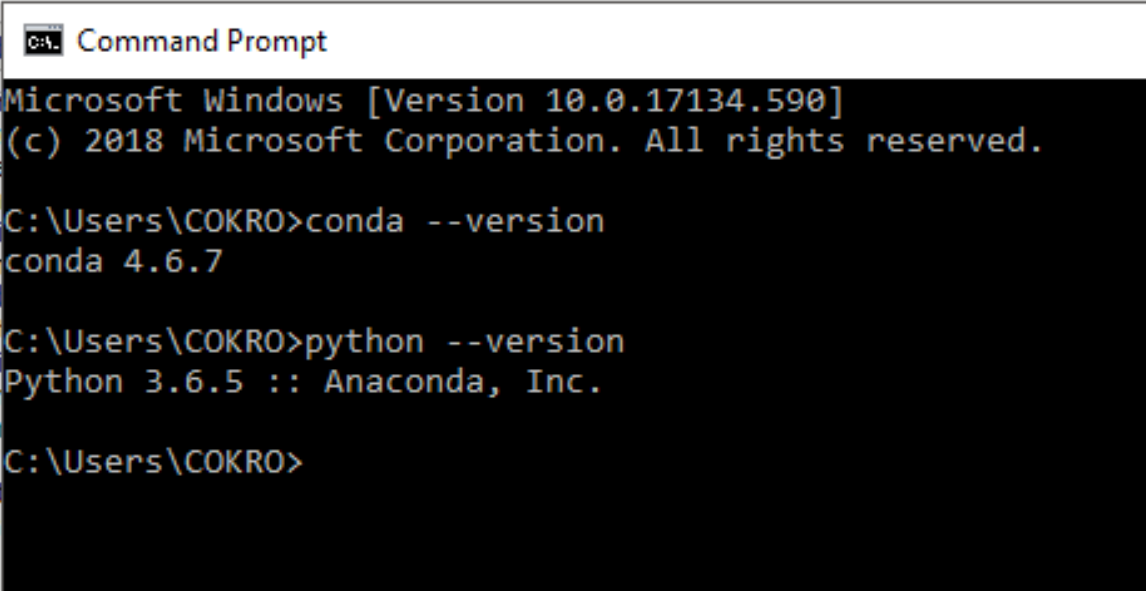
pada codingan tersebut menjadi error dikarenakan variabel atau method digits belum di definisikan. sedangkan untuk kode error pada gambar 1.43 yaitu pada source code

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

pada codingantersebut terjadi error dikarenakan module joblib belum di install atau modul tersebut tidak ada di library python.

(c) Solusi Pemecahan Masalah Error

- untuk memperbaiki error pada gambar 1.42 tinggal mendefinisikan variabel atau method digits, untuk lebih lengkapnya dapat dilihat pada gambar 1.44 dengan cara tersebut maka masalah error dapat diselesaikan.
- sedangkan untuk error joblib bisa dilakukan dengan cara masuk ke cmd administrator kemudian isikan perintah pip install joblib kemudian tekan enter sehingga hasilnya terlihat seperti gambar 1.45 setelah itu coba masuk kembali ke python di cmd dan ketikkan perintah from joblib import dump, load maka hasilnya seperti gambar 1.46.



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

C:\Users\COKRO>conda --version
conda 4.6.7

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

C:\Users\COKRO>
```

Figure 1.37: Tampilan Versi Python dan Anaconda .

=====

6. Error Handling

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

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

C:\WINDOWS\system32>conda --version
conda 4.5.4

C:\WINDOWS\system32>pip install -U scikit-learn
Collecting scikit-learn
  Using cached https://files.pythonhosted.org/packages/ee/c8/c89ebdc0d7dbba6e6fd222daabd257da3c28a967dd7c352d4272b2e1cef6/scikit_learn-0.20.2-cp36-cp36m-win32.whl
Requirement not upgraded as not directly required: numpy>=1.8.2 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (1.14.3)
Requirement not upgraded as not directly required: scipy>=0.13.3 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (1.1.0)
distributed 1.21.8 requires msgpack, which is not installed.
Installing collected packages: scikit-learn
  Found existing installation: scikit-learn 0.19.1
    Uninstalling scikit-learn-0.19.1:
      Successfully uninstalled scikit-learn-0.19.1
Successfully installed scikit-learn-0.20.2
You are using pip version 10.0.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Figure 1.38: Instalasi Library Sikic.

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

## Package Plan ##

  environment location: C:\ProgramData\Anaconda3

  added / updated specs:
    - scikit-learn

The following packages will be UPDATED:

    conda: 4.5.4-py36_0 --> 4.6.7-py36_0

Proceed ([y]/n)? y

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

C:\WINDOWS\system32>
```

Figure 1.39: Instalasi Library Sikic Melalui Conda

(a) Screenshot

(b) Error Code and Error Type

- Module Not Found

module yang dicari tidak ditemukan, karena file yang dicari tidak ada atau belum di instal.

- Type Data Error

type data yang seharusnya diisi oleh data number namun diisi oleh data str/character sedangkan nilai yang bisa dibaca adalah number.

(c) Solution

- For Module Not Found


```
C:\Users\COKRO>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello Anaconda!")
Hello Anaconda!
>>>
```

Figure 1.40: Console Python Include Anaconda

```
Command Prompt - python
C:\Users\COKRO>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from sklearn import datasets
>>> iris = datasets.load_iris()
>>> digits = datasets.load_digits()
>>> print(digits.data)
[[ 0.  0.  5. ...  0.  0.  0.]
 [ 0.  0.  0. ... 10.  0.  0.]
 [ 0.  0.  0. ... 16.  9.  0.]
 ...
 [ 0.  0.  1. ...  6.  0.  0.]
 [ 0.  0.  2. ... 12.  0.  0.]
 [ 0.  0. 10. ... 12.  1.  0.]]
>>> print(iris.data)
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]
 [5.4 3.7 1.5 0.2]
 [4.8 3.4 1.6 0.2]
 [4.8 3.  1.4 0.1]
 [4.3 3.  1.1 0.1]
 [5.8 4.  1.2 0.2]
 [5.7 4.4 1.5 0.4]
 [5.4 3.9 1.3 0.4]
 [5.1 3.5 1.4 0.3]
 [5.7 3.8 1.7 0.3]
 [5.1 3.8 1.5 0.3]
 [5.4 3.4 1.7 0.2]
 [5.1 3.7 1.5 0.4]
 [4.6 3.6 1.  0.2]
 [5.1 3.3 1.7 0.5]
 [4.8 3.4 1.9 0.2]
 [5.  3.  1.6 0.2]
 [5.  3.4 1.6 0.4]
 [5.2 3.5 1.5 0.2]]
```

Figure 1.41: Contoh Codingan Dataset

lakukan instalasi dengan memasukkan code berikut untuk download dan instalasi module JOBLIB

```
conda install -c anaconda joblib
```

- For Data Type Error

ganti isi data menjadi data number, contoh :

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

ganti menjadi

```
clf = SVC(gamma=0.5)
```

~~~~~ c6dbbab89a86daa2bf691a75d9a15c4b56fb15d7 ~~~~~ b1d7802739e393f94f14f567f8e8c88ea

```
Command Prompt - python
C:\Users\COKRO>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 svm
>>> clf = svm.SVC(gamma=0.001, C=100.)
>>> clf.fit(digits.data[:-1], digits.target[:-1])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'digits' is not defined
>>>
```

Figure 1.42: Error Coding 1

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

C:\Users\COKRO>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from joblib import dump, load
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'joblib'
>>>
```

Figure 1.43: Error Coding 2

```
Command Prompt - python
C:\Users\COKRO>python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:23:52) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from sklearn import datasets
>>> iris = datasets.load_iris()
>>> digits = datasets.load_digits()
>>> from sklearn import svm
>>> clf = svm.SVC(gamma=0.0001, C=100.)
>>> 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=0.0001, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
>>> clf.predict(digits.data[-1:])
array([8])
>>>
```

Figure 1.44: Codingan Solusi Untuk Error digits

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

C:\WINDOWS\system32>pip install joblib
Collecting joblib
  Downloading https://files.pythonhosted.org/packages/cd/c1/50a758e8247561e58cb87305b1e90b171b8c767b15b12a1734001f41d356
/joblib-0.13.2-py2.py3-none-any.whl (278kB)
    100% |#####| 286kB 2.3MB/s
distributed 1.21.8 requires msgpack, which is not installed.
Installing collected packages: joblib
Successfully installed joblib-0.13.2
You are using pip version 10.0.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\WINDOWS\system32>
```

Figure 1.45: Codingan Solusi Untuk Error Joblib

```

Command Prompt - python
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

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

```

Figure 1.46: Hasil Solusi Error Joblib

```

>>> 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)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    File "C:\Users\Fathi-PC\Anaconda3\lib\site-packages\sklearn\svm\base.py", line 187,
in fit
    fit(X, y, sample_weight, solver_type, kernel, random_seed=seed)
    File "C:\Users\Fathi-PC\Anaconda3\lib\site-packages\sklearn\svm\base.py", line 254,
in _dense_fit
    max_iter=self.max_iter, random_seed=random_seed)
    File "sklearn\svm\libsvm.pyx", line 58, in sklearn.svm.libsvm.fit
TypeError: must be real number, not str

```

Figure 1.47: Error Type data, yang harus digunakan number sedangkan isinya 'SCALE' pada gamma

```

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

```

Figure 1.48: Error no Module found, modul yang dicari tidak ditemukan atau tidak ada 'JOBLIB'

# Chapter 2

## Related Works

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

### 2.1 Same Topics

Cite every latest journal with same topic

#### 2.1.1 Topic 1

cite for first topic

#### 2.1.2 Topic 2

if you have two topics you can include here to

### 2.2 Same Method

write and cite latest journal with same method

#### 2.2.1 Method 1

cite and paraphrase method 1

#### 2.2.2 Method 2

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

# Chapter 3

## Methods

### 3.1 The data

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

### 3.2 Method 1

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

### 3.3 Method 2

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

# Chapter 4

## Experiment and Result

brief of experiment and result.

### 4.1 Experiment

Please tell how the experiment conducted from method.

### 4.2 Result

Please provide the result of experiment

# Chapter 5

## Conclusion

brief of conclusion

### **5.1 Conclusion of Problems**

Tell about solving the problem

### **5.2 Conclusion of Method**

Tell about solving using method

### **5.3 Conclusion of Experiment**

Tell about solving in the experiment

### **5.4 Conclusion of Result**

tell about result for purpose of this research.

# Chapter 6

## Discussion

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



# Chapter 7

## Discussion

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

# Chapter 8

## Discussion

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

# Chapter 9

## Discussion

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

# Chapter 10

## Discussion

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

# Chapter 11

## Discussion

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

# Chapter 12

## Discussion

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

# Chapter 13

## Discussion

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

# Chapter 14

## Discussion

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



# Appendix A

## Form Penilaian Jurnal

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

| NO | UNSUR                                        | KETERANGAN                                                                                                                                                                        | MAKS | KETERANGAN                                                                                                                                                                                                                                              |
|----|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Keefektifan Judul Artikel                    | Maksimal 12 (dua belas) kata dalam Bahasa Indonesia atau 10 (sepuluh) kata dalam Bahasa Inggris                                                                                   | 2    | a. Tidak lugas dan tidak ringkas (0)<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] Abdillah Baraja. Kecerdasan buatan tinjauan historikal. *Speed-Sentra Penelitian Engineering dan Edukasi*, 1(1), 2008.
- [2] Joshua Eckroth. *Python Artificial Intelligence Projects for Beginners: Get up and running with Artificial Intelligence using 8 smart and exciting AI applications*. Packt Publishing Ltd, 2018.
- [3] Herny Februariyanti and Eri Zuliarso. Klasifikasi dokumen berita teks bahasa indonesia menggunakan ontologi. *Dinamik*, 17(1), 2012.
- [4] Deny Kurniawan. Regresi linier. *R-Foundation for Statistical Computing. Vienna, Austria*, 17, 2008.
- [5] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.
- [6] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.