

# **CERDAS MENGUASAI PYTHON**



---

# CERDAS MENGUASAI PYTHON

## Dalam 24 Jam

---

**Rolly M. Awangga**  
Informatics Research Center



**Kreatif Industri Nusantara**

***Penulis:***

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

***Editor:***

M. Yusril Helmi Setyawan

***Penyunting:***

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

***Desain sampul dan Tata letak:***

Deza Martha Akbar

***Penerbit:***

Kreatif Industri Nusantara

***Redaksi:***

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

***Distributor:***

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara  
apapun tanpa ijin tertulis dari penerbit

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

# CONTRIBUTORS

---

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia



# CONTENTS IN BRIEF

---

<b>1</b>	<b>Judul Bagian Pertama</b>	<b>1</b>
<b>2</b>	<b>Judul Bagian Kedua</b>	<b>7</b>
<b>3</b>	<b>Judul Bagian Ketiga</b>	<b>23</b>
<b>4</b>	<b>Judul Bagian Keempat</b>	<b>39</b>
<b>5</b>	<b>Judul Bagian Kelima</b>	<b>61</b>
<b>6</b>	<b>Judul Bagian Keenam</b>	<b>91</b>





# DAFTAR ISI

---

Daftar Gambar	xv
Daftar Tabel	xix
Foreword	xxxi
Kata Pengantar	xxxiii
Acknowledgments	xxxv
Acronyms	xxxvii
Glossary	xxxix
List of Symbols	xli
Introduction	xliii
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
<b>1 Judul Bagian Pertama</b>	<b>1</b>
1.1 Irvan Rizkiansyah	1
1.2 Python	1
1.2.1 Background	1
1.2.2 Problems	2
	<b>ix</b>

1.2.3	Objective and Contribution	2
1.2.4	Scoop and Environment	2
1.3	Luthfi Muhammad Nabil_1174035	2
1.3.1	Background	2
1.3.2	Problems	3
1.3.3	Objective and Contribution	3
1.3.4	Scoop and Environment	4
1.4	Hagan Rowlenstino/1174040	4
1.4.1	Background	4
1.4.2	Problems	4
1.4.3	Objective and Contribution	4
1.4.4	Scoop and Environment	5
1.5	Rangga Putra Ramdhani_1174056	5
1.5.1	Background	5
1.5.2	Problems	5
1.5.3	Objective and Contribution	5
1.5.4	Scope and Environment	6
<b>2</b>	<b>Judul Bagian Kedua</b>	<b>7</b>
2.1	Kevin Natanael Nainggolan 1174059	7
2.2	Alit Fajar Kurniawan 1174057	15
2.2.1	Teori	15
2.2.2	praktek	16
2.2.3	Keterampilan dan penanganan eror	18
2.3	Muhammad Iqbal Panggabean	18
2.3.1	Teori	18
2.3.2	praktek	20
2.3.3	Keterampilan dan penanganan eror	22
<b>3</b>	<b>Judul Bagian Ketiga</b>	<b>23</b>
3.1	Yusniar Nur Syarif Sidiq/1164089	23
3.1.1	Teori	23
3.1.2	Keterampilan Pemrograman	26
3.1.3	Penanganan Error	29
3.2	Fathi Rabbani/1164074/3C	29
3.2.1	Teori	29
3.2.2	Praktek Pemrograman	32
3.2.3	Handling Error	38

<b>4</b>	<b>Judul Bagian Keempat</b>	<b>39</b>
4.1	Fathi Rabbani / 1164074	39
4.1.1	Teori	39
4.1.2	Praktek	42
4.1.3	Error	44
4.2	Kevin Natanael Nainggolan 1174059	44
4.3	Yusniar Nur Syarif Sidiq/1164089	46
4.3.1	Pemahaman Teori	46
4.3.2	Keterampilan Pemrograman	48
4.3.3	Penanganan Error	50
4.4	Alit Fajar Kurniawan 1174057	51
4.4.1	Pemahaman Teori	51
4.4.2	Keterampilan Pemograman	53
4.4.3	Ketrampilan Penanganan Error	55
4.5	Kevin Natanael Nainggolan 1174059	55
4.6	Muhammad Iqbal Panggabean 1174063	57
4.6.1	Praktek	57
4.6.2	Ketrampilan Penanganan Error	59
<b>5</b>	<b>Judul Bagian Kelima</b>	<b>61</b>
5.1	Alit Fajar Kurniawan	61
5.1.1	No. 1	61
5.1.2	No. 2	62
5.1.3	No. 3	62
5.1.4	No. 4	63
5.1.5	No. 5	63
5.1.6	No. 6	63
5.1.7	No. 7	63
5.1.8	Cek Plagiat	64
5.1.9	No. 1	64
5.1.10	No. 2	64
5.1.11	No. 3	65
5.1.12	No. 4	65
5.1.13	No. 1	66
5.2	Muhammad Iqbal Panggabean	66
5.2.1	Pertanyaan No.. 1	66
5.2.2	Pertanyaan No. 2	67
5.2.3	Pertanyaan No. 3	68

5.2.4	Pertanyaan No. 4	68
5.2.5	Pertanyaan No. 5	68
5.2.6	Pertanyaan No. 6	69
5.2.7	Pertanyaan No. 7	69
5.2.8	Cek Plagiat	69
5.3	Fathi Rabbani / 1164074	69
5.3.1	Teori	69
5.3.2	Praktikum	71
5.3.3	Penanganan Error	72
5.4	Kevin Natanae Nainggolan 1174059	72
5.4.1	TEORI	72
5.4.2	PRAKTEK	74
5.4.3	PENANGANAN ERROR	75
5.5	Yusniar Nur Syarif Sidiq / 1164089	75
5.5.1	Pemahaman Teori	75
5.5.2	Keterampilan Pemrograman	77
5.5.3	Keterampilan Penangan Error	79
5.6	Muhammad Iqbal Panggabean / 1174063	79
5.6.1	Keterampilan Pemrograman	79
5.6.2	Penanganan Error	80

## **6 Judul Bagian Keenam 91**

6.1	Alit Fajar Kurniawan 1174057	91
6.1.1	Pemahaman Teori	91
6.1.2	Keterampilan Pemogramman	97
6.1.3	Keterampilan Penanganan Error	100
6.1.4	Screenshoot Plagiat	101
6.1.5	Screenshoot Kode Program	101
6.2	Fathi Rabbani / 1164074	101
6.2.1	Teori	101
6.2.2	Pemrograman	105
6.2.3	Try Error	106
6.3	Yusniar Nur Syarif Sidiq / 1164089	107
6.3.1	Pemahaman Teori	107
6.3.2	Keterampilan Pemrograman	112
6.3.3	Keterampilan Penanganan Error	114
6.4	Kevin Natanae Nainggolan 1174059	114
6.4.1	TEORI	114

6.4.2	PRAKTEK	118
6.4.3	PENANGANAN EROR	120
Daftar Pustaka		141
Index		143



# DAFTAR GAMBAR

---

4.1	hasil csv pada Excel	42
4.2	Contoh File CSV	47
4.3	Atribute Error	50
4.4	Atribute Error	50
5.1	Hasil cek plagiat	82
5.2	Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dengan looping.	82
5.3	Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.	82
5.4	Hasil dari pembacaan fungsi untuk membaca file csv hasil arduino dan mengembalikan fungsi.	83
5.5	Hasil pembacaan fungsi untuk menanggulangi error menggunakan Try Except.	83
5.6	Hasil cek plagiat.	83
5.7	Check Plagiarisme	83



5.8	Memasang Port USB.	84
5.9	Mencari Device Manager.	84
5.10	Unknown Device.	84
5.11	Update Driver Software.	85
5.12	Browse my computer for driver software.	85
5.13	Open Folder Arduino.	85
5.14	Install.	85
5.15	Fungsi Pyserial.	86
5.16	Cek Plagiat.	86
5.17	Hasil Data Yang Diambil Dari Arduino.	86
5.18	Hasil Data Pengulanganl Dari Arduino.	87
5.19	Membuat File CSV.	88
5.20	Membaca File CSV.	88
5.21	Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino.	88
5.22	Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dengan looping.	89
5.23	Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.	89
5.24	Hasil dari pembacaan fungsi untuk membaca file csv hasil arduino dan mengembalikan fungsi.	89
5.25	Hasil pembacaan fungsi untuk menanggulangi error menggunakan Try Except.	90
6.1	Hasil compile membuat diagram menggunakan Matplotlib.	121
6.2	Hasil compile membuat bar graph menggunakan Matplotlib.	121
6.3	Hasil compile membuat histogram menggunakan Matplotlib.	122
6.4	Hasil compile membuat scatter plot menggunakan Matplotlib.	123
6.5	Hasil compile membuat diagram menggunakan Matplotlib.	124
6.6	Hasil compile membuat Pie Plot menggunakan Matplotlib.	124
6.7	Hasil compile membuat diagram menggunakan Matplotlib.	125

6.8	Hasil compile membuat diagram menggunakan Matplotlib.	125
6.9	Hasil compile membuat subplot menggunakan Matplotlib.	126
6.10	Hasil compile membuat diagram menggunakan Matplotlib.	127
6.11	Hasil compile membuat fungsi Bar Plot menggunakan Matplotlib.	127
6.12	Hasil compile membuat fungsi Scatter Plot menggunakan Matplotlib.	128
6.13	Hasil compile membuat fungsi Pie Plot menggunakan Matplotlib.	128
6.14	Hasil compile membuat fungsi Plot menggunakan Matplotlib.	129
6.15	hasil run penanganan error	130
6.16	hasil run penanganan error plot	131
6.17	hasil run penanganan error scatter	131
6.18	hasil run penanganan error bar	131
6.19	hasil run penanganan error pie	132
6.20	Penggunaan Subplot pada Matplotlib	132
6.21	Hasil dari Penggunaan Hist dari Code tersebut	132
6.22	Hasil Check Data Plagiarisme	132
6.23	Hasil Code Grafik Bar atau Batang	133
6.24	Hasil Code Grafik Titik atau Scatter	133
6.25	Hasil Code Grafik Pie atau Pie	133
6.26	Hasil Code Grafik Plot atau Plot	134
6.27	Hasil Code Try Error	134
6.28	Hasil Melakukan Plot Sumbu X dan Y.	134
6.29	Fungsi Bar.	135
6.30	Fungsi Histogram.	135
6.31	Fungsi Scatter Plot.	135
6.32	Fungsi Area Plot.	136
6.33	Fungsi Pie Plot.	136
6.34	Fungsi Line Graphic.	136

6.35	Fungsi Legend Dan Label.	137
6.36	Fungsi Subplot.	137
6.37	Fungsi Subplot.	137
6.38	Fungsi Bar Praktikum.	138
6.39	Fungsi Scatter Praktikum.	138
6.40	Fungsi Pie Praktikum.	139
6.41	Fungsi Plot Praktikum.	139
6.42	Error Yang Di Dapat.	139

# DAFTAR TABEL

---



# Listings

---

src/chapter3/chap3_1174059.teori.py	7
src/chapter3/chap3_1174059.teori.py	8
src/chapter3/chap3_1174059.teori.py	8
src/chapter3/chap3_1174059.teori.py	9
src/chapter3/chap3_1174059.teori.py	9
src/chapter3/chap3_1174059.teori.py	10
src/chapter3/chap3_1174059.teori.py	10
src/chapter3/chap3_1174059.teori.py	10
src/chapter3/chap3_1174059.teori.py	10
src/chapter3/chap3_1174059_3lib.py	11
src/chapter3/chap3_1174059_3lib.py	12
src/chapter3/chap3_1174059_3lib.py	12
src/chapter3/chap3_1174059_3lib.py	12
src/chapter3/chap3_1174059_3lib.py	13
src/chapter3/chap3_1174059_3lib.py	13
src/chapter3/chap3_1174059_3lib.py	13
src/chapter3/chap3_1174059_3lib.py	13
src/chapter3/chap3_1174059_3lib.py	13

src/chapter3/chap3_1174059_3lib.py	14
src/chapter3/chap3_1174059_main.py	14
src/chapter3/chap3_1174059_main.py	14
src/chapter3/chap3_1174059_teor.py	14
src/chapter2/1174057_teor.py	15
src/chapter2/1174057_teor.py	15
src/chapter2/1174057_teor.py	15
src/chapter2/1174057_teor.py	15
src/chapter2/1174057_teor.py	16
src/chapter2/1174057_teor.py	16
src/chapter2/1174057_teor.py	16
src/chapter2/1174057_teor.py	16
src/chapter2/1174057_teor.py	16
src/chapter2/1174057_praktek.py	17
src/chapter2/1174057_praktek.py	17
src/chapter2/1174057_praktek.py	17
src/chapter2/1174057_praktek.py	17
src/chapter2/1174057_praktek.py	17
src/chapter2/1174057_praktek.py	17
src/chapter2/1174057_praktek.py	18
src/chapter2/1174057_praktek.py	18
src/chapter2/1174057_praktek.py	18
src/chapter2/1174057_praktek.py	18
src/chapter2/1174057_praktek.py	18
src/chapter2/1174057_praktek.py	18
src/errr2.py	18
src/1174063_teor.py	19
src/1174063_teor.py	19
src/1174063_teor.py	19
src/1174063_teor.py	19
src/1174063_teor.py	19
src/1174063_teor.py	20
src/1174063_teor.py	20
src/1174063_teor.py	20
src/1174063_teor.py	20
src/1174063_praktek.py	20
src/1174063_praktek.py	21
src/1174063_praktek.py	21
src/1174063_praktek.py	21

src/1174063_praktek.py	21
src/1174063_praktek.py	21
src/1174063_praktek.py	22
src/1174063_praktek.py	22
src/1174063_praktek.py	22
src/1174063_praktek.py	22
src/1174063_praktek.py	22
src/errr2.py	22
src/chapter2/1164089/1164089_1.py	23
src/chapter2/1164089/Matematika.py	24
src/chapter2/1164089/1164089_2.py	24
src/chapter2/1164089/mtk.py	24
src/chapter2/1164089/1164089_3.py	25
src/chapter2/1164089/1164089_4.py	25
src/chapter2/1164089/1164089_5.py	25
src/chapter2/1164089/1164089_21.py	26
src/chapter2/1164089/1164089_22.py	27
src/chapter2/1164089/1164089_23.py	27
src/chapter2/1164089/1164089_24.py	27
src/chapter2/1164089/1164089_25.py	27
src/chapter2/1164089/1164089_26.py	27
src/chapter2/1164089/1164089_27.py	28
src/chapter2/1164089/1164089_28.py	28
src/chapter2/1164089/1164089_29.py	28
src/chapter2/1164089/1164089_30.py	28
src/chapter2/1164089/1164089_31.py	29
src/chapter2/1164089/1164089_32.py	29
src/chapter2/1164089/1164089_33.py	29
src/chapter3/1164074/praktek.py	29
src/chapter3/1164074/praktek.py	30
src/chapter3/1164074/praktek.py	30
src/chapter3/1164074/praktek.py	30
src/chapter3/1164074/praktek.py	30
src/chapter3/1164074/praktek.py	30
src/chapter3/1164074/praktek.py	31
src/chapter3/1164074/praktek.py	31
src/chapter3/1164074/praktek.py	31
src/chapter3/1164074/praktek.py	31



src/chapter3/1164074/praktek.py	31
src/chapter3/1164074/praktek.py	31
src/chapter3/1164074/praktek.py	32
src/chapter3/1164074/praktek.py	32
src/chapter3/1164074/praktek.py	32
src/chapter3/1164074/praktek.py	33
src/chapter3/1164074/praktek.py	33
src/chapter3/1164074/praktek.py	33
src/chapter3/1164074/praktek.py	34
src/chapter3/1164074/praktek.py	34
src/chapter3/1164074/praktek.py	34
src/chapter3/1164074/praktek.py	34
src/chapter3/1164074/praktek.py	34
src/chapter3/1164074/praktek.py	34
src/chapter3/1164074/3lib.py	35
src/chapter3/1164074/main.py	37
src/chapter3/1164074/praktek.py	38
src/chapter4/coba.csv	40
src/chapter4/coba.csv	41
src/chapter4/coba.csv	41
src/chapter4/coba.csv	41
src/chapter4/coba.csv	42
src/chapter4/coba.csv	42
src/chapter4/coba.csv	42
src/chapter4/1164074/c4_1164074.csv.py	43
src/chapter4/1164074/c4_1164074.csv.py	43
src/chapter4/1164074/c4_1164074_pandas.py	43
src/chapter4/1164074/c4_1164074_pandas.py	43
src/chapter4/1164074/c4_1164074_pandas.py	43
src/chapter4/1164074/c4_1164074_pandas.py	43
src/chapter4/1164074/c4_1164074_pandas.py	43
src/chapter4/1164074/main.py	44
src/chapter4/1164074/main2.py	44
src/chapter4/1164074/1164074_Error.py	44
src/chapter4/1164074/1164074_Error.py	44
src/teoric4.py	44
src/teoric4.py	45
src/teoric4.py	45

src/teoric4.py	45
src/teoric4.py	45
src/teoric4.py	46
src/teoric4.py	46
src/chapter4/1164089.py	47
src/chapter4/1164089.py	48
src/chapter4/1164089/1164089_csv.py	48
src/chapter4/1164089/1164089_csv.py	49
src/chapter4/1164089/1164089_pandas.py	49
src/chapter4/1164089/1164089_pandas.py	49
src/chapter4/1164089/1164089_pandas.py	49
src/chapter4/1164089/1164089_pandas.py	49
src/chapter4/1164089/1164089_pandas.py	49
src/chapter4/1164089/1164089_main.py	50
src/chapter4/1164089/1164089_main2.py	50
src/chapter4/1164089/1164089_pandas.py	50
src/chapter4/1174057csvpandas.py	52
src/chapter4/1174057csvpandas.py	52
src/chapter4/1174057csvpandas.py	52
src/chapter4/1174057csvpandas.py	53
4.1 Fungsi untuk membuka file CSV dengan lib CSV mode list.	53
4.2 Fungsi untuk membuka file CSV dengan lib CSV mode dictionary.	53
4.3 Fungsi untuk membuka file CSV dengan lib Pandas mode list.	53
4.4 Fungsi untuk membuka file CSV dengan lib Pandas mode dictionary.	54
4.5 Fungsi untuk mengubah format tanggal menjadi standar dataframe.	54
4.6 Fungsi untuk mengubah index kolom.	54
4.7 Fungsi untuk mengubah atribut atau nama kolom.	54
4.8 Membuat dan meBaca file CSV menggunakan library 1174006pandas.	54
4.9 Membuat dan mmeBaca file CSV menggunakan library 1174006pandas.	55
src/chapter4/1174057/1174057_error.py	55
src/teoric4.py	55
src/teoric4.py	55
src/teoric4.py	55
src/teoric4.py	56
src/teoric4.py	56
src/teoric4.py	56
src/teoric4.py	57
src/chapter4/1174057/1174057_error.py	57

4.10 Fungsi untuk membuka file CSV dengan lib CSV mode list.	57
4.11 Fungsi untuk membuka file CSV dengan lib CSV mode dictionary.	57
4.12 Fungsi untuk membuka file CSV dengan lib Pandas mode list.	58
4.13 Fungsi untuk membuka file CSV dengan lib Pandas mode dictionary.	58
4.14 Fungsi untuk mengubah format tanggal menjadi standar dataframe.	58
4.15 Fungsi untuk mengubah index kolom.	58
4.16 Fungsi untuk mengubah atribut atau nama kolom.	58
4.17 Membuat dan me baca file CSV menggunakan library 1174006pandas.	59
4.18 Membuat dan mme baca file CSV menggunakan library 1174006pandas.	59
src/chapter4/1174063/1174063_2err.py	59
5.1 Fungsi yang menggunakan pyserial.	64
5.2 Fungsi untuk mendapatkan data dari Arduino.	64
5.3 Fungsi untuk mendapatkan data langsung dari Arduino dengan looping.	64
5.4 Fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.	65
5.5 Fungsi untuk membaca file CSV hasil Arduino dan mengembalikan fungsi.	65
5.6 Fungsi untuk menanggulangi error menggunakan Try Except.	66
5.7 Fungsi yang menggunakan pyserial.	71
src/chapter5/praktek/c4_1164074_realtime.py	71
src/chapter5/praktek/c4_1164074_save.py	71
src/chapter5/praktek/c4_1164074_realtime.py	71
src/chapter5/praktek/c4_1164074_csv.py	72
src/chapter5/penangananerror/c4_1164074_EH.py	72
src/chapter5/teori/1174059.py	72
src/chapter5/teori/1174059.py	72
src/chapter5/teori/1174059.py	73
src/chapter5/teori/1174059.py	73
src/chapter5/teori/1174059.py	73
src/chapter5/teori/1174059.py	73
src/chapter5/teori/1174059.py	74
src/chapter5/praktek/1174059_realtime.py	74
src/chapter5/praktek/1174059_save.py	74
src/chapter5/praktek/1174059_realtimekecsv.py	74
src/chapter5/praktek/1174059_csv.py	75
src/chapter5/penangananerror/1174059_notiferror.py	75
src/chapter5/1164089/1164089_Teori.py	77

src/chapter5/1164089/1164089_realtime.py	77
src/chapter5/1164089/1164089_save.py	78
src/chapter5/1164089/1164089_realtime.py	78
src/chapter5/1164089/1164089csv.py	78
src/chapter5/1164089/1164089_tryExcept.py	79
5.8 Fungsi untuk mendapatkan data dari Arduino.	79
5.9 Fungsi untuk mendapatkan data langsung dari Arduino dengan looping.	79
5.10 Fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.	80
5.11 Fungsi untuk membaca file CSV hasil Arduino dan mengembalikan fungsi.	80
5.12 Fungsi untuk menanggulangi error menggunakan Try Except.	81
src/chapter6/1174057/1174057.py	91
src/chapter6/1174057/1174057.py	92
src/chapter6/1174057/1174057.py	92
src/chapter6/1174057/1174057.py	92
6.1 Kode program membuat diagram menggunakan Matplotlib.	92
6.2 Kode program membuat bar graph menggunakan Matplotlib.	92
6.3 Kode program membuat histogram menggunakan Matplotlib.	93
6.4 Kode program membuat scatter plot menggunakan Matplotlib.	93
6.5 Kode program membuat diagram menggunakan Matplotlib.	94
6.6 Kode program membuat Pie Plot menggunakan Matplotlib.	94
6.7 Kode program membuat diagram menggunakan Matplotlib.	95
6.8 Kode program menggunakan parameter label dengan Matplotlib.	95
6.9 Kode program memanggil fungsi legend dengan Matplotlib.	95
6.10 Kode program membuat diagram menggunakan Matplotlib.	95
6.11 Kode program membuat subplot menggunakan Matplotlib.	96
6.12 Kode program membuat diagram menggunakan Matplotlib.	97
6.13 Kode program membuat fungsi Bar Plot menggunakan Matplotlib.	97
6.14 Kode program membuat fungsi Scatter Plot menggunakan Matplotlib.	98
6.15 Kode program membuat fungsi Pie Plot menggunakan Matplotlib.	99
6.16 Kode program membuat fungsi Plot menggunakan Matplotlib.	99
6.17 Kode program membuat fungsi penanganan error.	100
src/chapter6/1164074/code.py	101
6.18 Jenis Garis	101
6.19 Jenis Titik	102
6.20 Jenis Batang	102

[illegible]

src/chapter6/teori/1174059.py	116
src/chapter6/teori/1174059.py	116
src/chapter6/teori/1174059.py	117
src/chapter6/praktek/1174059/chap6_1174059_bar.py	118
src/chapter6/praktek/1174059/chap6_1174059_pie.py	118
src/chapter6/praktek/1174059/chap6_1174059_scater.py	119
src/chapter6/praktek/1174059/chap6_1174059_plot.py	120
src/chapter6/praktek/1174059/chap6_1174059_penanganan.py	120



# FOREWORD

---

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa





# KATA PENGANTAR

---

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat  
Februari, 2019*



# ACKNOWLEDGMENTS

---

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.



# ACRONYMS

---

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association



# GLOSSARY

---

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald





# SYMBOLS

---

- $A$  Amplitude
- $\&$  Propositional logic symbol
- $a$  Filter Coefficient
  
- $\mathcal{B}$  Number of Beats



# INTRODUCTION

---

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center  
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[1].

$$ABCDEF\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$



# BAB 1

---

## JUDUL BAGIAN PERTAMA

---

### 1.1 Irvan Rizkiansyah

### 1.2 Python

#### 1.2.1 Background

Python adalah sebuah bahasa pemrograman yang bersifat interpreter, interactive, object-oriented, dan dapat beroperasi hampir pada semua platform seperti Windows, Linux, Mac. Python termasuk sebagai bahasa pemrograman yang dapat dengan mudah di pelajari karena sintaks yang jelas dan mudah dipahami, dan dapat dikombinasikan dengan penggunaan modul yang siap pakai, dan struktur data tingkat tinggi yang efisien [2].

Python memiliki kepastakaan atau biasa disebut library yang sangat luas, dan dalam distribusi Python yang telah disediakan, hal tersebut diakibatkan oleh pendistribusian Python yang bebas karena bahasa pemrograman Python merupakan bahasa pemrograman yang freeware atau bebas dalam hal pengembangannya. Python adalah sebuah bahasa pemrograman yang dapat dengan mudah dibaca dan terstruktur, hal tersebut dikarenakan penggunaan sistem identasi, yaitu pemisahan blok-blok

program susunan identasi, jadi untuk menambahkan sub-sub program dalam sebuah blok program, sub program tersebut harus diletakkan pada satu atau lebih spasi dari kolom sebuah blok [3].

Bahasa pemrograman Python dibuat oleh Guido Van Rossum. Dikarenakan para pengembang software atau perangkat lunak lebih cenderung memilih kecepatan dalam menyelesaikan suatu proyek dibandingkan dengan kecepatan proses dari program yang dijalankan, maka dari itu bahasa pemrograman Python dapat dibilang bahasa pemrograman yang kecepatannya dapat melebihi bahasa pemrograman C. Akan tetapi bahasa pemrograman Python lebih lambat dalam memproses suatu program dibandingkan bahasa pemrograman C. dengan berkembangnya kecepatan prosesor dan memori saat ini, mengakibatkan tidak terlihatnya keterlambatan dari sebuah program yang menggunakan bahasa pemrograman Python [4].

### **1.2.2 Problems**

- Kurangnya pemahaman tentang bahasa pemrograman Python
- Kurang mengerti dalam hal fungsi-fungsi yang terdapat pada bahasa pemrograman Python

### **1.2.3 Objective and Contribution**

#### **1.2.3.1 Objective**

- Dapat memahami tentang bahasa pemrograman Python
- Dapat memahami fungsi fungsi yang terdapat pada bahasa pemrograman Python

#### **1.2.3.2 Contribution**

- Dapat membangun sebuah sistem dengan menggunakan bahasa pemrograman Python
- Dapat membangun sebuah alat yang berguna, menggunakan mikrokontroler dan bahasa pemrograman python

### **1.2.4 Scoop and Environment**

- Pengenalan tentang bahasa pemrograman Python
- Pengenalan fungsi-fungsi yang terdapat pada bahasa pemrograman Python

## **1.3 Luthfi Muhammad Nabil\_1174035**

### **1.3.1 Background**

Python adalah sebuah bahasa pemrograman dengan level tinggi yang interaktif, dan mendukung berbagai paradigma pemrograman. Python sudah terkenal pada kалан-

gan programmer sebagai bahasa yang mudah dipahami dan memiliki kompleksitas yang dinamis sehingga dapat dipakai di algoritma maupun platform yang berbagai macam. Python sudah memiliki banyak komunitas pendukung karena penggunaanya yang banyak. Selain pada komunitas biasa, Python sudah diimplementasikan pada banyak perusahaan ternama dan dipasang pada aplikasi yang sudah terkenal seperti pada search engine google yang dimiliki oleh perusahaan Google.

Python mulai dirilis pada tahun 1991 oleh Guido van Rossum sebagai kelanjutan dari bahasa pemrograman ABC dengan memiliki versi yaitu 0.9.0. Nama dari bahasa Python diambil dari program televisi di Inggris bernama Monty Python. Lalu tahun 1995, Guido pindah ke CNRI di Virginia, Amerika sembari melanjutkan pengembangan Python. Versi terakhir yang dikeluarkan telah mencapai 1.6. Pada awalnya, Python adalah bahasa yang dipakai untuk Lalu pada tahun 2000, dirilis Python versi 2.0 yang memiliki peran sebagai bahasa pemrograman tidak berbayar atau open source. Van Rossum sendiri aktif pada development dari Python tetapi sudah bergabung dengan banyak penyumbang. Dibandingkan dengan bahasa lain, Python sudah melewati beberapa versi yang terbatas, mengikuti filosofi dari perubahan berurutan.

Untuk memahami bahasa Python tidak sulit, tetapi instalasi Python cukup memiliki trik tersendiri terlebih untuk pengguna yang baru memasuki lingkup programming. Pada sistem operasi windows, pengguna diharuskan untuk memasuki sistem pada windows untuk mengatur lokasi dari Python yang sudah diinstall. Selain itu, untuk yang terbiasa dengan beberapa pemrograman harus beradaptasi dengan aturan - aturan pada bahasa pemrograman Python seperti penggantian titik koma (;) dengan indentasi. Oleh karena itu, penulis akan membahas mengenai pengenalan singkat mengenai bahasa pemrograman python dan cara instalasi dari python dan library pip.

### 1.3.2 Problems

Sesuai dengan latar belakang yang telah dibahas, penulis merumuskan masalah sebagai berikut :

1. Bagaimana pemaparan singkat mengenai Python?
2. Bagaimana cara melakukan instalasi Python?

### 1.3.3 Objective and Contribution

#### 1.3.3.1 Objective

1. Untuk membahas mengenai Python.
2. Untuk menunjukkan cara instalasi Python.

**1.3.3.2 Contribution** Pada materi ini, penulis menggunakan Python.



### 1.3.4 Scoop and Environment

- Pada Chapter 1 membahas mengenai sejarah, latar belakang, dan keterangan singkat mengenai python tersebut. Chapter ini juga merangkum masalah dan mencari tujuan yang ingin dicapai penulis dalam membuat resume ini.

## 1.4 Hagan Rowlenstino/1174040

### 1.4.1 Background

Python di desain sebagai bahasa pemrograman yang dapat digunakan sehari-hari. Pencipta python, Guido van Rossum, telah menulis seri lengkap tentang sejarah bahasa tersebut. Python diciptakan di awal 1990 di CWI (the Centrum voor Wiskunde and Informatica), tempat kelahiran ALGOL (Algorithmic Language 68). Sebelumnya, Rossum juga telah mengerjakan bahasa pemrograman ABC, yang dikembangkan di CWI sebagai bahasa pengajaran yang menekankan kejelasan. Walaupun project ABC telah di tutup, Rossum banyak belajar dari hal tersebut saat dia mulai membuat Python sebagai alat untuk multimedia dan project penelitian sistem operasi. Dia ingin Python mempunyai tingkatan yang cukup tinggi agar mudah untuk dibaca dan ditulis, juga mirip dengan Java, dan menawarkan portabilitas serta error model yang terdefinisi dengan baik.

Python juga kaya akan vocabulary yang berguna untuk membuat algoritma yang kompleks dengan efisien dikarenakan punya dictionaries yang memiliki string yang kuat dan asosiasi array yang fleksibel. Python menggabungkan antara fleksibilitas tingkat tinggi, kemampuan membaca, dan interface yang terdefinisi dengan baik. Kombinasi tersebut membuat Python cocok untuk menyelesaikan masalah komputasi non-algoritma seperti integrasi dengan web, format data, atau hardware kelas rendah. Python mudah untuk dipelajari karena strukturnya sederhana dan sintaksnya jelas, punya library yang portable dan dapat digunakan di beda perangkat, dan dapat terintegrasi dengan bahasa pemrograman lain seperti C, C++, dan Java.

### 1.4.2 Problems

1. Banyak pemrograman yang penggunaannya kompleks

### 1.4.3 Objective and Contribution

#### 1.4.3.1 Objective

1. Dapat memudahkan pemrograman dengan bahasa pemrograman yang tepat

#### 1.4.3.2 Contribution

1. Menggunakan Python sebagai bahasa pemrograman

### 1.4.4 Scoop and Environment

1. Mengimplementasikan Python dalam pemrograman

## 1.5 Rangga Putra Ramdhani.1174056

### 1.5.1 Background

Python adalah bahasa pemrograman tingkat tinggi untuk keperluan umum yang filosofi desainnya menekankan keterbacaan kode. Sintaksis Python memungkinkan programmer untuk mengekspresikan konsep dalam lebih sedikit baris kode daripada yang mungkin dilakukan dalam bahasa seperti C dan bahasa tersebut menyediakan konstruksi yang dimaksudkan untuk memungkinkan program yang jelas pada skala kecil dan besar.

Python mendukung banyak paradigma pemrograman, termasuk gaya pemrograman berorientasi objek, imperatif dan fungsional. Ini fitur sistem tipe yang sepenuhnya dinamis dan manajemen memori otomatis, mirip dengan Skema, Ruby, Perl dan Tclm dan memiliki perpustakaan standar yang besar dan komprehensif.

Seperti bahasa dinamis lainnya, Python sering digunakan sebagai bahasa scripting, tetapi juga digunakan dalam berbagai konteks non-scripting. Menggunakan alat pihak ketiga, kode Python dapat dikemas ke dalam program yang dapat dieksekusi mandiri. Penerjemah python tersedia untuk banyak sistem operasi.

### 1.5.2 Problems

1. Bagaimana mahasiswa politeknik pos indonesia dapat menggunakan bahasa python.
2. Kenapa mahasiswa politeknik pos indonesia harus belajar bahasa pemrograman python.
3. Bagaimana cara menggunakan bahasa python terhadap web service.

### 1.5.3 Objective and Contribution

#### 1.5.3.1 Objective

1. Mahasiswa politeknik pos indonesia dapat memahami bahasa python secara bertahap.
2. Bahasa pemrograman python dapat dijalankan di Linux, Mac dan Windows.
3. Menggunakan bahasa python dapat mempermudah mahasiswa dalam membuat web service.

### **1.5.3.2 Contribution**

1. Membantu mahasiswa politeknik pos indonesia dalam menyelesaikan masalah pada python.
2. Membantu mahasiswa politeknik pos indonesia memahami bahasa pemrograman python.
3. Mempelajari bahasa python dengan proses pembuatan web service.

### **1.5.4 Scope and Environment**

1. Mahasiswa politeknik pos indonesia dapat memahami bahasa python.
2. Mahasiswa politeknik pos indonesia dapat menjalankan fungsi python.
3. Mahasiswa politeknik pos indonesia dapat membuat web service dengan menggunakan python.

## BAB 2

---

# JUDUL BAGIAN KEDUA

---

### 2.1 Kevin Natanael Nainggolan 1174059

#### 2.1.0.1 Pemahaman Teori

1. Apa itu fungsi, inputan fungsi dan kembalian fungsi dengan contoh kode program

lainnya.

Fungsi adalah bagian dari program yang dapat digunakan ulang.

Berikut merupakan contoh fungsi dan cara pemanggilannya

```
1 Fungsi dalam python diartikan sebagai program yang dapat
   ditugaskan terus menerus tanpa harus mendefinisikannya
   berulang kali, fungsi selalu didefinisikan dengan "def"
   dengan inputan variabel yang telah disediakan untuk
   selanjutnya diproses dengan contoh variabel = input() atau
   value dan untuk kembalian variabel yang sudah di proses itu
   akan menjadi output yang dicontohkan dengan return variabel
   , contoh kode programnya bisa dilihat seperti berikut
2 def Penambahan(var1, var2)
3   var3 = var1+var2
```

```

4     return var3
5 var1 = value1
6 var2 = value2
7 var4 = Penambahan(var1, var2)

```

Fungsi dapat membaca parameter, parameter adalah nilai yang disediakan kepada fungsi, dimana nilai ini akan menentukan output yang akan dihasilkan fungsi.

```

1 Paket atau library adalah koleksi koleksi dari program yang dapat
  digunakan dalam membangun struktur program yang lebih rumit.
  contoh programnya seperti berikut, untuk membuat librarynya
  seperti ini
2 def Penambahan(var1, var2):
3     var3=var1+var2
4     return var3
5 def Pengurangan(var1, var2):
6     var3=var1-var2
7     return var3
8 def Perkalian(var1, var2):
9     var3=var1*var2
10    return var3
11 def Pembagian(var1, var2):
12    var3=var1/var2
13    return var3
14
15 contoh program pemanggilan librarynya seperti ini
16
17 import namafilelibrary
18
19 var1=value
20 var2=value
21
22 var3=namafilelibrary.fungsi1(var1, var2)
23 var4=namafilelibrary.fungsi2(var1, var2)
24 var5=namafilelibrary.fungsi3(var1, var2)

```

Statemen return digunakan untuk keluar dari fungsi. Kita juga dapat menspesifikasikan nilai kembalian.

```

1 Kelas adalah cetakan yang digunakan untuk menciptakan suatu objek
  , objek adalah segala sesuatu yang nyata maupun abstrak,
  atribut adalah nilai data yang terdapat dalam objek, dan
  method adalah apa saja yang dapat dilakukan oleh objek.
  contoh kode programnya sebagai berikut.
2 Kelas
3
4 class namakelas
5     var2 = value1
6
7 Objek
8
9     var1 = namakelas()
10    print(var1.var2)
11
12 atribut
13

```

```

14     var1=value1
15     var2=value2
16
17 method
18
19 class namakelas1:
20     def __init__(self, var6, var7):
21         self.var6 = var6
22         self.var7 = var7
23
24     def fungsi4(self):
25         print("Hello my name is " + self.var6)
26
27     var8 = namakelas1("John", 36)
28     var8.fungsi4()

```

2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya.

Untuk memudahkan dalam pemanggilan fungsi yang di butuhkan, agar dapat dipanggil berulang.

Cara pemanggilannya

```

1 Pemanggilan library dalam python sangat mudah dengan menggunakan
  keyword "\"import\"" library otomatis terpanggil, harus
  diperhatikan, library yang dipanggil harus berada dalam
  folder yang sama dengan fileprogram yang memanggil library
  tersebut. contoh programnya sebagai berikut.
2
3 import namalibrary

```

3. Jelaskan Apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing.

kelas merupakan sebuah blueprint yang mepresentasikan objek.

objek adalah hasil cetakan dari sebuah kelas.

method adalah suatu upaya yang digunakan oleh object.

```

1 Penggunaan from kalkulator import Penambahan dapat diartikan
  sebagai pengambilan fungsi penambahan dalam file library
  bernama kalkulator
2
3 from kalkulator import Penambahan
4
5 var1=value1
6 var2=value2
7 var3=Penambahan(var1, var2)
8 var4=var3.Penambahan()
9 var5=var3.Pengurangan()
10 var6=var3.Perkalian()
11 var7=var3.Pembagian()

```

4. Jelaskan cara pemanggilan library kelas dari instansiasi dan pemakaiannya dengan contoh program lainnya.

#### Cara Pemanggilannya

```

1 Pemanggilan library dalam python sangat mudah dengan menggunakan
  keyword "import" library otomatis terpanggil, harus
  diperhatikan, library yang dipanggil harus berada dalam
  folder yang sama dengan fileprogram yang memanggil library
  tersebut. contoh programnya sebagai berikut.
2
3 import namalibrary

```

5. Jelaskan dengan contoh pemakaian paket dengan perintah from kalkulator import Penambahan disertai dengan contoh kode lainnya.

Penggunaan paket from namafile import, itu berfungsi untuk memanggil file dan fungsinya

```

1 Penggunaan from kalkulator import Penambahan dapat diartikan
  sebagai pengambilan fungsi penambahan dalam file library
  bernama kalkulator
2
3 from kalkulator import Penambahan
4
5 var1=value1
6 var2=value2
7 var3=Penambahan(var1, var2)
8 var4=var3.Penambahan()
9 var5=var3.Pengurangan()
10 var6=var3.Perkalian()
11 var7=var3.Pembagian()

```

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila le library ada di dalam folder.

Pemakaian paket adalah perkumpulan fungsi-fungsi. contoh kodenya adalah sebagai berikut :

```

1 import namalibrary

```

7. Jelaskan dengan contoh kodenya, pemakaian paket kelas apabila le library ada di dalam folder.

```

1 from namalibrary import namafungsi

```

### 2.1.0.2 Ketampilan Pemrograman

1. Buatlah fungsi dengan inputan variabel NPM, dan melakukan print luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk  $NPM \bmod 3 = 0$ , tanda pagar untuk  $NPM \bmod 3 = 1$ , tanda plus untuk  $NPM \bmod 3 = 2$ .

```

1 def penulisan(npm):
2     npm = list(str(npm))
3     angka1 = {"0": "*****", "1": "***", "2": "*****", "3": "*****", "4": "***", "5": "*****", "6": "*****", "7": "*****", "8": "*****", "9": "*****"}
4     angka2 = {"0": "*****", "1": "***", "2": "*****", "3": "*****", "4": "***", "5": "*****", "6": "*****", "7": "*****", "8": "*****", "9": "*****"}
5     angka3 = {"0": "***", "1": "***", "2": "*****", "3": "*****", "4": "***", "5": "*****", "6": "*****", "7": "*****", "8": "*****", "9": "*****"}
6     angka4 = {"0": "***", "1": "*****", "2": "*****", "3": "*****", "4": "*****", "5": "*****", "6": "*****", "7": "*****", "8": "*****", "9": "*****"}
7     angka5 = {"0": "***", "1": "*****", "2": "*****", "3": "*****", "4": "*****", "5": "*****", "6": "*****", "7": "*****", "8": "*****", "9": "*****"}
8     angka6 = {"0": "***", "1": "*****", "2": "*****", "3": "*****", "4": "*****", "5": "*****", "6": "*****", "7": "*****", "8": "*****", "9": "*****"}
9     angka7 = {"0": "***", "1": "*****", "2": "*****", "3": "*****", "4": "*****", "5": "*****", "6": "*****", "7": "*****", "8": "*****", "9": "*****"}
10    angka8 = {"0": "***", "1": "*****", "2": "*****", "3": "*****", "4": "*****", "5": "*****", "6": "*****", "7": "*****", "8": "*****", "9": "*****"}
11    angka9 = {"0": "*****", "1": "*****", "2": "*****", "3": "*****", "4": "*****", "5": "*****", "6": "*****", "7": "*****", "8": "*****", "9": "*****"}
12    angka10 = {"0": "*****", "1": "*****", "2": "*****", "3": "*****", "4": "*****", "5": "*****", "6": "*****", "7": "*****", "8": "*****", "9": "*****"}
13
14    hasil1 = []
15    hasil2 = []
16    hasil3 = []
17    hasil4 = []
18    hasil5 = []
19    hasil6 = []
20    hasil7 = []
21    hasil8 = []
22    hasil9 = []
23    hasil10 = []
24
25    for x in npm:
26        hasil1.append(angka1[x])
27        hasil2.append(angka2[x])
28        hasil3.append(angka3[x])
29        hasil4.append(angka4[x])
30        hasil5.append(angka5[x])
31        hasil6.append(angka6[x])
32        hasil7.append(angka7[x])
33        hasil8.append(angka8[x])
34        hasil9.append(angka9[x])
35        hasil10.append(angka10[x])
36

```



```

37     print(*hasil1 , sep=' ')
38     print(*hasil2 , sep=' ')
39     print(*hasil3 , sep=' ')
40     print(*hasil4 , sep=' ')
41     print(*hasil5 , sep=' ')
42     print(*hasil6 , sep=' ')
43     print(*hasil7 , sep=' ')
44     print(*hasil8 , sep=' ')
45     print(*hasil9 , sep=' ')
46     print(*hasil10 , sep=' ')

```

2. Buatlah fungsi dengan inputan variabel berupa NPM. kemudian dengan menggunakan perulangan mengeluarkan print output sebanyak dua digit belakang NPM.

```

1 def perulangan(npm):
2     NPM = input()
3
4     x = (NPM[5:7])
5     y = int(x)
6     z = 1
7     print("Output : ")
8
9     while z <= y:
10         z+=1
11         print("Halo , "+NPM+" apa kabar?")

```

3. Buatlah fungsi dengan dengan input variabel string bernama NPM dan beri luaran output dengan perulangan berupa tiga karakter belakang dari NPM sebanyak penjumlahan tiga digit tersebut.

```

1 def perulangan_3_digit(npm):
2     NPM = input()
3
4     a = (int(NPM[4]))
5     b = (int(NPM[5]))
6     c = (int(NPM[6]))
7
8     print (a+b+c)

```

4. Buatlah fungsi hello word dengan input variabel string bernama NPM dan beri luaran output berupa digit ketiga dari belakang dari variabel NPM menggunakan akses langsung manipulasi string pada baris ketiga dari variabel NPM.

```

1 def perulangan_3_digit_terakhir(npm):
2     npm = str(npm)
3     bil = npm[-3]
4     print("Halo , "+bil+" apa kabar?")

```

5. buat fungsi program dengan input variabel NPM dan melakukan print nomor npm satu persatu kebawah.

```

1 def down(npm):
2     for i in npm:
3         print (i)

```

6. Buatlah fungsi dengan inputan variabel NPM, didalamnya melakukan penjumlahan dari seluruh digit NPM tersebut, wajib menggunakan perulangan dan atau kondisi.

```

1 def penjumlahan(npm):
2     jumlah = 0
3     for i in npm:
4         jumlah += int(i)
5     print(str(jumlah)+" Adalah hasil penjumlahan")

```

7. Buatlah fungsi dengan inputan variabel NPM, didalamnya melakukan melakukan perkalian dari seluruh digit NPM tersebut, wajib menggunakan perulangan dan atau kondisi.

```

1 def perkalian(npm):
2     jumlah = 0
3     for i in npm:
4         jumlah *= int(i)
5     print(str(jumlah)+" Adalah hasil perkalian")

```

8. Buatlah fungsi dengan inputan variabel NPM, Lakukan print NPM anda tapi hanya digit genap saja. wajib menggunakan perulangan dan atau kondisi.

```

1 def genap():
2     npm = [1,1,7,4,0,5,9]
3     for i in npm:
4         if (i % 2) == 0:
5             print("Bilangan Genapnya : "+str(i))

```

9. Buatlah fungsi dengan inputan variabel NPM, Lakukan print NPM anda tapi hanya digit ganjil saja. wajib menggunakan perulangan dan atau kondisi.

```

1 def ganjil():
2     npm = [1,1,7,4,0,5,9]
3     for i in npm:
4         if (i%2)==1:
5             print("Bilangan Ganjilnya : "+str(i))

```

10. Buatlah fungsi dengan inputan variabel NPM, Lakukan print NPM anda tapi hanya digit yang termasuk bilangan prima saja. wajib menggunakan perulangan dan atau kondisi.

```

1 def prima(npm):
2     npm = str(npm)
3     bil = npm[2]
4     num = int(bil)
5     if num > 1:
6         for i in range(2,num):
7             if (num%i)==0:
8                 print("Bukan Bilangan Prima")
9                 break
10            else:
11                print("Bilangan Primanya :"+str(num))
12    else:
13        print("Tidak Ada Bilangan Prima")

```

11. Buatlah satu library yang berisi fungsi-fungsi dari nomor diatas dengan nama le 3lib.py dan berikan contoh cara pemanggilannya pada le main.py.

```

1 from chap3_1174059_3lib import penulisan

```

12. Buatlah satu library class dengan nama le kelas3lib.py yang merupakan mod-i kasi dari fungsi-fungsi nomor diatas dan berikan contoh cara pemanggilannya pada le main.py.

```

1 from chap3_1174059_3lib import penulisan
2
3 from chap3_1174059_3lib import penulisan
4
5
6
7 print(penulisan(int(input("Masukan NPM :"))))

```

**2.1.0.3 Ketrampilan Penanganan Error** Error yang di dapat dari mengerjakan tugas ini adalah type error, cara menaggulaginya dengan cara mengecek kembali codingannya

kemudian run kembali aplikasinya

berikut contoh Penggunaan fungsi try dan exception

```

1 #1
2 Fungsi dalam python diartikan sebagai program yang dapat ditugaskan
   terus menerus tanpa harus mendefinisikannya berulang kali, fungsi
   selalu didefinisikan dengan "\def\" dengan inputan variabel yang
   telah disediakan untuk selanjutnya diproses dengan contoh
   variabel = input() atau value dan untuk kembalian variabel yang
   sudah di proses itu akan menjadi output yang dicontohkan dengan
   return variabel , contoh kode programnya bisa dilihat seperti
   berikut
3 def Penambahan(var1 , var2)
4     var3 = var1+var2
5     return var3
6 var1 = value1
7 var2 = value2

```

## 2.2 Alit Fajar Kurniawan 1174057

### 2.2.1 Teori

1. Jenis jenis Variable python dan cara pemakaiannya Variabel merupakan tempat menyimpan data, Dalam Python terdapat beberapa variabel dengan berbagai type data diantaranya adalah variabel dengan type data number, string, dan boolean. Dalam python kita dapat membuat variable dengan cara sebagai gambar berikut

```

1 c = 7
2 d = "Alit Fajar Kurniawan"
3
4 print(c)
5 print(d)

```

2. Kode untuk meminta input dari user dan bagaimana melakukan output ke layar

```

1 npm = input()
2 print(npm)

```

3. Operator dasar aritmatika Ada operator penambahan, pengurangan perkalian, perkalian, pembagian, modulus, perpangkatan, dan pembulatan decimal.

```

1 x = 5
2 y = 3
3
4 print(x+y)
5
6 x = 5
7 y = 3
8
9 print(x-y)
10
11 x = 12
12 y = 3
13
14 print(x*y)
15
16 x = 8
17 y = 2
18
19 print(x/y)
20
21 x = 5
22 y = 2
23
24 print(x%y)

```

4. Perulangan Terdapat dua jenis perulangan di dalam python yaitu perulangan while dan perulangan for

```

1 ulang = ["topi", "celana", "baju"]
2 for i in range(ulang):
3     print(i)

```

```

1 coba = 1
2 while i < 6 :
3     print(i)
4     i += 1

```

5. sintak Untuk memilih kondisi, dan kondisi didalam kondisi Pengambilan kondisi If yang digunakan untuk mengantisipasi kondisi yang terjadi saat program dijalankan dan menentukan tindakan apa yang akan diambil sesuai dengan kondisi.

```

1 a = 40
2 b = 300
3 if b > a :
4     print("b lebih besar dari a")

```

```

1 a = 40
2 b = 40
3 if b > a :
4     print("b besar dari a")
5 elif a == b :
6     print("a sama dengan b")

```

```

1 a = 300
2 b = 40
3 if b > a :
4     print("b besar dari a")
5 elif a == b :
6     print("a sama dengan b")
7 else :
8     print("a lebih dari b")

```

6. Jenis-jenis error pada python Syntax Errors adalah keadaan dimana kode python mengalami kesalahan penulisan. ZeroDivisionError adalah error yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol. NameError adalah error yang terjadi saat kode di eksekusi terhadap local name atau global name yang tidak terdefinisi. TypeError adalah error yang terjadi saat dilakukan eksekusi pada suatu operasi atau fungsi dengan type object yang tidak sesuai.
7. Cara memakai try except Cara pemakaian try except adalah sebagai berikut :

```

1 x = 0
2 y = "1"
3 try :
4     z = x + y
5     print(z)
6 except TypeError :
7     print("Perbedaan data")

```

## 2.2.2 praktek

1. Jawaban soal no 1

```

1 print ("+++  +++  +++++++  +++  +++  +++++++")
   +++++++  +++++++")
2 print ("+++  +++  +++++++  +++  +++  +++++++")
   +++++++  +++++++")
3 print ("+++  +++  +++  +++  +++")
   +++")
4 print ("+++  +++  +++  +++  +++  +++  +++")
   +++ ")
5 print ("+++  +++  +++  +++  +++")
   +++++++  +++  +++ ")
6 print ("+++  +++  +++  +++  +++")
   +++++++  +++ ")
7 print ("+++  +++  +++  +++  +++")
   +++ ")
8 print ("+++  +++  +++  +++  +++")
   +++ ")
9 print ("+++  +++  +++  +++  +++")
   +++++++  +++ ")
10 print ("+++  +++  +++  +++  +++")
   +++++++  +++ ")

```

## 2. Jawaban soal no 2

```

1 npm = input("Masukkan NPM :")
2 hitung = 0
3 while(hitung <= 57):
4     print("Hallo , " + str(npm) + "Apa kabar?")
5     hitung = hitung +1

```

## 3. Jawaban soal no 3

```

1 npm = input("Masukkan NPM :")
2 hitung = 0
3 while(hitung <= 6):
4     print("Halo , " + str(npm[4:7]) + " Apa kabar?")
5     hitung = hitung +1

```

## 4. Jawaban soal no 4

```

1 npm = input("Masukkan NPM : ")
2 print("Hello , " + str(npm[-3]) + " Apa kabar?")

```

## 5. Jawaban soal no 5

```

1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 5
7 g = 7
8 h = a+b+c+d+e+f+g
9 i = a*b*c*d*e*f*g
10
11 print("Hasil no 5")
12 print("NPM anda")
13 print(a, b, c, d, e, f, g, h)

```

## 6. Jawaban soal no 6

```
1 print("Hasil no 6")
2 print(h)
```

## 7. Jawaban soal no 7

```
1 print("Hasil no 7")
2 print(i)
```

## 8. Jawaban soal no 8

```
1 print("Hasil no 8")
2 thisset = (a, b, c, d, e, f, g)
3
4 for x in thisset:
5     print(x)
```

## 9. Jawaban soal no 9

```
1 print(d,e,f)
```

## 10. Jawaban soal no 10

```
1 print(a,b,c,g)
```

## 11. Jawaban soal no 11

```
1 print(a,b,c,g)
```

## 2.2.3 Keterampilan dan penanganan error

```
1 c = 1
2 d = "4"
3
4 try:
5     a = c - d
6     print(a)
7 except:
8     print("hampura euy edit errornya")
```

## 2.3 Muhammad Iqbal Panggabean

### 2.3.1 Teori

1. Jenis jenis variable pyton dan cara pemakaiannya Variabel merupakan tempat menyimpan data. Dalam Phyton terdapat beberapa variabel dengan berbagai type data diantaranya adalah variabel dengan type data number, string, dan boolean. Dalam phyton kita dapat membuat variable dengan cara sebagai gambar berikut

```

1 c = 7
2 d = "Gabe Tamvan"
3
4 print(c)
5 print(d)

```

2. Kode untuk meminta input dari user dan bagaimana melakukan output ke layar

```

1 npm = input()
2 print(npm)

```

3. Operator dasar aritmatika Ada operator penambahan, pengurangan perkalian, perkalian, pembagian, modulus, perpangkatan, dan pembulatan decimal.

```

1 x = 5
2 y = 3
3
4 print(x+y)
5
6 x = 5
7 y = 3
8
9 print(x-y)
10
11 x = 12
12 y = 3
13
14 print(x*y)
15
16 x = 8
17 y = 2
18
19 print(x/y)
20
21 x = 5
22 y = 2
23
24 print(x%y)

```

4. Perulangan Terdapat dua jenis perulangan di dalam python yaitu perulangan while dan perulangan for

```

1 ulang = ["topi", "celana", "baju"]
2 for i in range(ulang):
3     print(i)

```

```

1 coba = 1
2 while i < 6 :
3     print(i)
4     i += 1

```

5. sintak Untuk memilih kondisi, dan kondisi didalam kondisi Pengambilan kondisi If yang digunakan untuk mengantisipasi kondisi yang terjadi saat program dijalankan dan menentukan tindakan apa yang akan diambil sesuai dengan kondisi.



```

1 a = 40
2 b = 300
3 if b > a :
4     print("b lebih besar dari a")

```

```

1 a = 40
2 b = 40
3 if b > a :
4     print("b besar dari a")
5 elif a == b :
6     print("a sama dengan b")

```

```

1 a = 300
2 b = 40
3 if b > a :
4     print("b besar dari a")
5 elif a == b :
6     print("a sama dengan b")
7 else :
8     print("a lebih dari b")

```

6. Jenis-jenis error pada python Syntax Errors adalah keadaan dimana kode python mengalami kesalahan penulisan. ZeroDivisionError adalah error yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol. NameError adalah error yang terjadi saat kode di eksekusi terhadap local name atau global name yang tidak terdefinisi. TypeError adalah error yang terjadi saat dilakukan eksekusi pada suatu operasi atau fungsi dengan type object yang tidak sesuai.

7. Cara memakai try except Cara pemakaian try except adalah sebagai berikut :

```

1 x = 0
2 y = "1"
3 try :
4     z = x + y
5     print(z)
6 except TypeError :
7     print("Perbedaan data")

```

## 2.3.2 praktek

1. Jawaban soal no 1

```

1 print ("+++   +++   ++++++++   +++   +++   ++++++++")
2     ++++++++   ++++++++")
3 print ("+++   +++   ++++++++   +++   +++   ++++++++")
4     ++++++++   ++++++++")
5 print ("+++   +++   ++++++++   +++   +++   ++++++++")
6     ++++++++   ++++++++")

```

```

6 print ("+++      +++      +++      +++      +++      +++")
7 print ("+++++++      ++++++++      +++      +++      +++      +++")
8 print ("      +++      +++      +++      +++      +++      +++")
9 print ("      +++      +++      +++      +++      ++++++++      ++++++++")
10 print ("+++++++      ++++++++      +++      +++      ++++++++      ++++++++")

```

## 2. Jawaban soal no 2

```
1 npm = input("Masukkan NPM :")
2 hitung = 0
3 while(hitung <= 63):
4     print("Hallo, " + str(npm) + "Apa kabar?")
5     hitung = hitung + 1
```

### 3. Jawaban soal no 3

```
1 npm = input("Masukkan NPM :")
2 hitung = 0
3 while(hitung <= 6):
4     print("Halo, " + str(npm[4:7]) + " Apa kabar?")
5     hitung = hitung + 1
```

4. Jawaban soal no 4

```
1 npm = input("Masukkan NPM : ")
2 print("Hello , " + str(npm[-3]) + " Apa kabar?")
```

### 5. Jawaban soal no 5

```
1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 6
7 g = 3
8 h = a+b+c+d+e+f+g
9 i = a*b*c*d*e*f*g
10
11 print("Hasil no 5")
12 print("NPM anda")
13 print(a, b, c, d, e, f, g, h)
```

6. Jawaban soal no 6

```
1 print("Hasil no 6")
2 print(h)
```

7. Jawaban soal no 7

```

1 print("Hasil no 7")
2 print (i)

```

#### 8. Jawaban soal no 8

```

1 print("Hasil no 8")
2 thisset = (a, b, c, d, e, f, g)
3
4 for x in thisset:
5     print(x)

```

#### 9. Jawaban soal no 9

```

1 print(d,e,f)

```

#### 10. Jawaban soal no 10

```

1 print(a,b,c,g)

```

#### 11. Jawaban soal no 11

```

1 print(a,b,c,g)

```

### 2.3.3 Keterampilan dan penanganan eror

```

1 c = 1
2 d = "4"
3
4 try:
5     a = c - d
6     print(a)
7 except:
8     print("hampura euy edit errornya")

```

## BAB 3

---

# JUDUL BAGIAN KETIGA

---

### 3.1 Yusniar Nur Syarif Sidiq/1164089

#### 3.1.1 Teori

1. Fungsi merupakan sebuah bagian dari program yang dapat digunakan ulang dan memiliki inputan variabel serta nilai yang akan di kembalikannya. Contohnya adalah source code berikut ini,

```
1 def memanggil(a,b):  
2     x = a + b  
3     return x  
4  
5 a='Hallo'  
6 b=' 1164089'  
7 c=memanggil(a,b)
```

Dalam dalam source code tersebut akan mengeluarkan output Hallo 1164089 ketika kita running di dalam spyder.

2. Library dalam python disini merupakan kumpulan dari fungsi dan cara pemanggilannya adalah dengan melakukan import file librarynya. Sebagai contoh, buat-

lah Matematika.py dan 1164089\_2.py, simpan dalam satu folder. Untuk Matematika.py isikan fungsi sebagai berikut

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 16:44:02 2019
4
5 @author: NS
6 """
7
8 def matematika(a,b,c):
9     r = a + b / c
10    return r

```

Untuk memanggil fungsi tersebut adalah dengan melakukan import Matematika.py pada 1164089\_2.py adalah sebagai berikut,

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 16:47:53 2019
4
5 @author: NS
6 """
7
8 import Matematika
9
10 a=50
11 b=50
12 c=2
13 hasil=Matematika.matematika(a,b,c)

```

3. Class merupakan salah satu cara untuk membuat sebuah kode yang mempunyai objek serta atribut tertentu sehingga akan lebih mudah dalam mengorganisasi berbagai fungsi dan statenya. Objek disini merupakan instansiasi atau perwujudan dari sebuah class. Untuk membuat class yang memiliki objek serta atribut dapat dilihat pada source code berikut ini, dimana kita akan membuat file bernama mtk.py

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 17:10:12 2019
4
5 @author: NS
6 """
7
8 class Mtk:
9     def __init__(self , a , b , c):
10         self.a = a
11         self.b = b
12         self.c = c
13     def Hasil(self):
14         r = self.a + self.b / self.c
15         return r

```

Self tersebut berfungsi untuk menunjukkan variabel lokal dari class tersebut. Untuk memanggil class tersebut kita akan membuat file bernama 1164089\_3.py dan kita akan melakukan import mtk.py pada file tersebut, untuk source codenya dapat dilihat seperti berikut,

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 17:16:05 2019
4
5 @author: NS
6 """
7
8 import mtk
9
10 a=50
11 b=50
12 c=2
13
14 hasilnya = mtk.Mtk(a,b,c)
15
16 jadi=hasilnya.Hasil()
```

#### 4. Cara memanggilnya yaitu

- Pertama import terlebih dahulu filenya
- Buat variabel yang berfungsi menampung data
- Panggil nama classnya dan methodnya
- Gunakan perintah print untuk menampilkannya

Sebagai contoh perhatikan source code ini,

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 17:45:02 2019
4
5 @author: NS
6 """
7
8 import Matematika
9
10 a=50
11 b=50
12 c=2
13 hasil=Matematika.matematika(a,b,c)
14 print(hasil)
```

5. Dimana kita akan melakukan membuka library Matematika.py dan akan melakukan import dari fungsi di dalamnya yaitu matematika, sehingga akan lebih simple dalam penulisan source codenya adalah sebagai berikut,

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 17:51:40 2019
```

```

4
5 @author: NS
6 """
7
8 from Matematika import matematika
9 a=50
10 b=50
11 c=2
12
13 hasil=matematika(a,b,c)

```

6.

7.

### 3.1.2 Keterampilan Pemrograman

#### 1. Soal No 1

```

1 def NPM(npm):
2
3     npm = list(str(npm))
4
5     angka1 = {"0": "+++++ ", "1": " ++", "2": "+++++ ", "3": "
++++++ ", "4": "      ++", "5": "++++++", "6": "+++++ ", "
7": "++++++", "8": "+++++ ", "9": "+++++"}
6     angka2 = {"0": "++ ++", "1": "+++", "2": "++ ++", "3": "
++ ++", "4": "++++", "5": "++ ++", "6": "+++", "
7": "++ ++", "8": "++ ++", "9": "++ ++"}
7     angka3 = {"0": "++ ++", "1": "+++", "2": "++ ++", "3": "
++++ ", "4": "++ ++", "5": "++++++", "6": "++++++ ", "
7": "++ ++", "8": "++ ++", "9": "++ ++"}
8     angka4 = {"0": "++ ++", "1": "+++", "2": "++ ++", "3": "
++++ ", "4": "++++++", "5": "++ ++", "6": "+++", "
7": "++ ++", "8": "++ ++", "9": "++++++"}
9     angka5 = {"0": "++ ++", "1": "+++", "2": "+++", "3": "
++ ++", "4": "+++", "5": "++ ++", "6": "+++", "
7": "+++", "8": "+++", "9": "+++"}
10    angka6 = {"0": "+++++ ", "1": "+++", "2": "++++++", "3": "
++++++ ", "4": "      ++", "5": "+++++", "6": "++++++", "
7": "+++", "8": "++++++", "9": "++++++"}
11
12    hasil1 = []
13    hasil2 = []
14    hasil3 = []
15    hasil4 = []
16    hasil5 = []
17    hasil6 = []
18
19    for x in npm:
20        hasil1.append(angka1[x])
21        hasil2.append(angka2[x])
22        hasil3.append(angka3[x])
23        hasil4.append(angka4[x])
24        hasil5.append(angka5[x])

```

```

25     hasil6.append(angka6[x])
26
27     print(*hasil1, sep=' ')
28     print(*hasil2, sep=' ')
29     print(*hasil3, sep=' ')
30     print(*hasil4, sep=' ')
31     print(*hasil5, sep=' ')
32     print(*hasil6, sep=' ')
33
34 NPM(input("Masukan NPM anda: "))

```

## 2. Soal No 2

```

1 def pengulangan(npm):
2     ulang = 1
3     while(ulang <= 89):
4         print("Halo, "+str(npm)+" apa kabar?")
5         ulang += 1
6
7 pengulangan(int(input("Masukan NPM : ")))

```

## 3. Soal No 3

```

1 def pengulangan_089(npm):
2     hitung = 0
3     npm = str(npm)
4     bilangan = npm[4:7]
5
6     while(hitung < 17):
7         print("Halo, "+bilangan+" apa kabar?")
8         hitung = hitung +1
9
10 pengulangan_089(int(input("Masukan NPM : ")))

```

## 4. Soal No 4

```

1 def perulangan_0(npm):
2     npm = str(npm)
3     bilangan = npm[-3]
4     print("Halo, "+bilangan+" apa kabar?")
5
6 perulangan_0(int(input("Masukan NPM : ")))

```

## 5. Soal No 5

```

1 def down(npm):
2     for i in npm:
3         print(i)
4 down(input("Masukan NPM : "))

```

## 6. Soal No 6

```

1 def pertambahan(npm):
2     nilai = 0
3     for i in npm:

```



```

4         nilai += int(i)
5         print(str(nilai)+" Merupakan Hasil Dari Pertambahan NPM Anda"
6             )
7     pertambahan(input("Masukan NPM : "))

```

## 7. Soal No 7

```

1 def perkalian(npm):
2     nilai = 0
3     for i in npm:
4         nilai *= int(i)
5         print(str(nilai)+" Merupakan Hasil Nilai Perkalian NPM Anda")
6
7     perkalian(input("Masukan NPM : "))

```

## 8. Soal No 8

```

1 def bilgenap():
2     npm = [1,1,6,4,0,8,9]
3     for i in npm:
4         if (i % 2) == 0:
5             print("Bilangan Genapnya Dari NPM : "+str(i))
6     bilgenap()

```

## 9. Soal No 9

```

1 def bilganjil():
2     npm = [1,1,6,4,0,8,9]
3     for i in npm:
4         if (i%2)==1:
5             print("Bilangan Ganjilnya Dari Npm : "+str(i))
6     bilganjil()

```

## 10. Soal No 10

```

1 def bilprima(npm):
2     npm = str(npm)
3     nilai = npm[2]
4     num = int(nilai)
5     if num > 1:
6         for i in range(2,num):
7             if (num%i)==0:
8                 print("Bukan Bilangan Prima")
9                 break
10            else:
11                print("Merupakan Bilangan Primanya :"+str(num))
12    else:
13        print("Tidak Ada Bilangan Prima")
14    bilprima(int(input("Masukan NPM : ")))

```

## 11. Soal No 11

```

1 #contoh salah satu saja
2 lib = __import__('3lib')
3
4 npm = "1164089"
5
6 lib.NPM(npm)
7 lib.Pengulangan(npm)
8
9 print()

```

## 12. Soal No 12

```

1 #contoh salah satu saja
2 from Kelas3lib import kelas3lib
3
4 npm = "1164089"
5
6 klib = kelas3lib(npm)
7
8 klib.NPM()

```

### 3.1.3 Penanganan Errorr

1. Errorr yang saya temui di antaranya adalah Systax Errorr, dimana suatu keadaan script python mengalami kesalahan dalam penulisannya dan solusi dari permasalahan ini adalah dengan memperbaiki script penulisan yang salah. Untuk contoh fungsi trx except dapat dilihat pada source code berikut ini,

```

1 def name(nama):
2     try :
3         print("Hai, "+str(nama))
4     except:
5         print("Errorr")
6
7 name(input("Siapa namamu: "))

```

## 3.2 Fathi Rabbani/1164074/3C

### 3.2.1 Teori

1. Fungsi, Inputan dan Return

- Fungsi adalah sebuah blok code yang digunakan untuk melempar parameter kedalam blok code yang berbeda.

```

1 def coba():
2     print("coba")
3     coba()

```

- Inputan fungsi adalah sebuah fungsi yang memiliki parameter berupa inputan atau data yang bias diinputkan.

```

1 def coba2(nm):
2     print("Namaku :"+str(nm))
3     coba2(input("NAMA:"))

```

- Pengembalian fungsi atau sering juga disebut sebagai return merupakan sebuah pengembalian nilai dari pengekseskuan data pada parameter yang terdapat difungsi.

```

1 def coba3(a,b):
2     x = a+b
3     return x
4
5 a = 7
6 b = 5
7 y = coba3(a,b)
8 print(y)

```

## 2. Paket atau Library Fungsi

paket merupakan sebuah penggunaan library dengan maksud mempermudah dalam eksekusi dan pemanggilan fungsi

```

1 def tambah2(a,b):
2     r = a+b
3     return r

```

berikut ini adalah fungsi yang digunakan untuk memanggil paket atau library

```

1 import praktek
2 a = 1
3 b = 3
4 c = praktek.tambah2(a,b)

```

## 3. Kelas, Objek, Atribut dan Method

- kelas merupakan sebuah blueprint dari objek
- objek merupakan sebuah data hasil eksekusi dari kelas
- atribut merupakan nilai data yang terdapat didalam objek
- method merupakan operasi atau eksekusi yang dilakukan dengan data dari objek

```

1 class fthr:
2     def __init__(self, a, b):
3         self.a = a
4         self.b = b
5     def tambah(self):
6         r = self.a + self.b
7         return r

```

## 4. Library Kelas

- Penggunaan kelas dan datanya

```

1 class fthr:
2     def __init__(self, a, b):
3         self.a = a
4         self.b = b
5     def tambah(self):
6         r = self.a + self.b
7         return r

```

#### ▪ Pemanggilan library kelas

```

1 import fthr
2
3 a = 1
4 b = 3
5 c = lib.fthr(a,b)
6 result=c.tambah2()

```

## 5. Pemanggilan Library Kalkulator

#### ▪ data Kalkulator

```

1 def penambahan(a,b):
2     r = a+b
3     return r

```

#### ▪ data pemanggilan

```

1 from kalkulator import penambahan
2
3 a = 1
4 b = 3
5
6 result = penambahan(a,b)

```

## 6. Penggunaan Paket Fungsi

#### ▪ data Fungsi Kalkulator

```

1 def penambahan(a,b):
2     r = a+b
3     return r

```

#### ▪ data Pemanggil Fungsi

```

1 import kalkulator
2 a = 1
3 b = 3
4 result = kalkulator.penambahan(a,b)

```

## 7. Penggunaan Paket Kelas

#### ▪ data Kelas fthr dari file praktek

```

1 class fthr:
2     def __init__(self, a, b):
3         self.a = a
4         self.b = b
5     def tambah(self):
6         r = self.a + self.b
7         return r

```

#### ▪ data pemanggil

```

1 a = 1
2 b = 3
3 c = lib.fthr(a,b)
4 result=c.tambah2()
5
6 #library kalkulator

```

## 3.2.2 Praktek Pemrograman

```

1 def coba4(n):
2     n = list(str(n))
3
4     angka1 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
5     angka2 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "### ", "5": "##### ", "6": "### ", "7": "##### ", "8": "### ", "9": "### "}
6     angka3 = {"0": "## ", "1": "### ", "2": "### ", "3": "### ", "4": "### ", "5": "### ", "6": "### ", "7": "### ", "8": "### ", "9": "### "}
7     angka4 = {"0": "## ", "1": "### ", "2": "### ", "3": "### ", "4": "### ", "5": "### ", "6": "### ", "7": "### ", "8": "### ", "9": "### "}
8     angka5 = {"0": "## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
9     angka6 = {"0": "## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
10    angka7 = {"0": "## ", "1": "### ", "2": "## ", "3": "### ", "4": "### ", "5": "### ", "6": "### ", "7": "### ", "8": "### ", "9": "### "}
11    angka8 = {"0": "## ", "1": "### ", "2": "## ", "3": "### ", "4": "### ", "5": "### ", "6": "### ", "7": "### ", "8": "### ", "9": "### "}
12    angka9 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "### ", "7": "##### ", "8": "### ", "9": "### "}
13    angka10 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "### ", "5": "##### ", "6": "##### ", "7": "### ", "8": "##### ", "9": "##### "}
14
15    hasil1 = []
16    hasil2 = []

```

```

17     hasil3 = []
18     hasil4 = []
19     hasil5 = []
20     hasil6 = []
21     hasil7 = []
22     hasil8 = []
23     hasil9 = []
24     hasil10 = []
25
26     for x in n:
27         hasil1.append(angka1[x])
28         hasil2.append(angka2[x])
29         hasil3.append(angka3[x])
30         hasil4.append(angka4[x])
31         hasil5.append(angka5[x])
32         hasil6.append(angka6[x])
33         hasil7.append(angka7[x])
34         hasil8.append(angka8[x])
35         hasil9.append(angka9[x])
36         hasil10.append(angka10[x])
37
38     print(*hasil1, sep=' ')
39     print(*hasil2, sep=' ')
40     print(*hasil3, sep=' ')
41     print(*hasil4, sep=' ')
42     print(*hasil5, sep=' ')
43     print(*hasil6, sep=' ')
44     print(*hasil7, sep=' ')
45     print(*hasil8, sep=' ')
46     print(*hasil9, sep=' ')
47     print(*hasil10, sep=' ')
48     coba4(input("NPM: "))

```

```

1 def loop(n):
2     calc = 0
3     while (calc < 74):
4         print("Hello, " + str(n) + " Apa Kabar?")
5         calc = calc + 1
6     loop(int(input("NPM: ")))

```

```

3. def loop3(x):
4     y = 0
5     a = int(x[4])
6     b = int(x[5])
7     c = int(x[6])
8     d = x[4:7]
9     e = a+b+c
10    while y < e:
11        y += 1
12        print("Hello, " + d + " Apa Kabar?")
13    return x
14    loop3(input("NPM: "))

```

```

1 def value4(x):
2     n = x[4]

```

```

3     print("Hello , " + n + " Apa Kabar?")
4     return x
5 value4(input("NPM:"))

```

```

5. def loopdown(x):
2     for i in x:
3         print(i)
4     return i
5 loopdown(input("NPM:"))

```

```

1 def tambah(n):
2     hasil = 0
3     for i in n:
4         hasil += int(i)
5     print(str(hasil))
6
7 tambah(input("NPM:"))

```

```

6. def kali(n):
2     hasil = 0
3     for i in n:
4         hasil *= int(i)
5     print(str(hasil))
6
7 kali(input("NPM:"))

```

```

1 def nilaigenap(x):
2     npm = [a,b,c,d,e,f,g]
3
4     for n in npm:
5         if(n % 2 == 0):
6             if(n != 0):
7                 print(n, end = "")
8 nilaigenap(input("NPM:"))

```

```

8. def nilaiganjil(x):
2     npm = [a,b,c,d,e,f,g]
3
4     for n in npm:
5         if(n % 2 != 0):
6             print(n, end = "")
7 nilaiganjil(input("NPM:"))

```

```

1 def prim(x):
2     npm = [a,b,c,d,e,f,g]
3
4     for n in npm:
5         if(n % 2 == 0):
6             print(end="")
7         else:
8             print(n, end = "")
9 prim(input("NPM:"))

```

## 10. ■ data 3lib.py

```

1
2 #1
3 def coba4(n):
4     n = list(str(n))
5
6     angka1 = {"0": "##### ", "1": "### ", "2": "##### ", "3":
7 : "####", "4": "###   ### ", "5": "##### ", "6": "#####
8 : "7": "##### ", "8": "##### ", "9": "##### "}
9     angka2 = {"0": "##### ", "1": "### ", "2": "##### ", "3":
10 : "####", "4": "###   ### ", "5": "##### ", "6": "##  ##
11 : "7": "##### ", "8": "##  ## ", "9": "##  ## "}
12     angka3 = {"0": "##      ## ", "1": "### ", "2": "      ## ", "3":
13 : "      ## ", "4": "###   ### ", "5": "###           ", "6": "      ##
14 : "7": "      ##      ## ", "8": "##  ## ", "9": "##  ## "}
15     angka4 = {"0": "##      ## ", "1": "### ", "2": "      ## ", "3":
16 : "      ## ", "4": "###   ### ", "5": "###           ", "6": "      ##
17 : "7": "      ##      ## ", "8": "##  ## ", "9": "##  ## "}
18     angka5 = {"0": "##      ## ", "1": "### ", "2": "##### ", "3":
19 : "####", "4": "##### ", "5": "##### ", "6": "#####
20 : "7": "      ##      ## ", "8": "##### ", "9": "##### "}
21     angka6 = {"0": "##      ## ", "1": "### ", "2": "##### ", "3":
22 : "####", "4": "      ##      ## ", "5": "##### ", "6": "#####
23 : "7": "      ##      ## ", "8": "##### ", "9": "##### "}
24     angka7 = {"0": "##      ## ", "1": "### ", "2": "      ## ", "3":
25 : "      ## ", "4": "      ##      ## ", "5": "      ##      ## ", "6": "      ##  ##
26 : "7": "      ##      ## ", "8": "##  ## ", "9": "      ##  ## "}
27     angka8 = {"0": "##      ## ", "1": "### ", "2": "      ## ", "3":
28 : "      ## ", "4": "      ##      ## ", "5": "      ##      ## ", "6": "      ##  ##
29 : "7": "      ##      ## ", "8": "##  ## ", "9": "      ##  ## "}
30     angka9 = {"0": "##### ", "1": "### ", "2": "##### ", "3":
31 : "####", "4": "      ##      ## ", "5": "##### ", "6": "##  ##
32 : "7": "      ##      ## ", "8": "##  ## ", "9": "##  ## "}
33     angka10 = {"0": "##### ", "1": "### ", "2": "##### ", "3":
34 : "####", "4": "      ##      ## ", "5": "##### ", "6": "#####
35 : "7": "      ##      ## ", "8": "##### ", "9": "##### "}
36
37     hasil1 = []
38     hasil2 = []
39     hasil3 = []
40     hasil4 = []
41     hasil5 = []
42     hasil6 = []
43     hasil7 = []
44     hasil8 = []
45     hasil9 = []
46     hasil10 = []
47
48     for x in n:
49         hasil1.append(angka1[x])
50         hasil2.append(angka2[x])
51         hasil3.append(angka3[x])
52         hasil4.append(angka4[x])
53         hasil5.append(angka5[x])
54         hasil6.append(angka6[x])
55         hasil7.append(angka7[x])

```



```
36         hasil8.append(angka8[x])
37         hasil9.append(angka9[x])
38         hasil10.append(angka10[x])
39
40     print(*hasil1, sep=' ')
41     print(*hasil2, sep=' ')
42     print(*hasil3, sep=' ')
43     print(*hasil4, sep=' ')
44     print(*hasil5, sep=' ')
45     print(*hasil6, sep=' ')
46     print(*hasil7, sep=' ')
47     print(*hasil8, sep=' ')
48     print(*hasil9, sep=' ')
49     print(*hasil10, sep=' ')
50
51
52 #2
53 def loop(n):
54     calc = 0
55     while (calc < 74):
56         print("Hello, " + str(n) + " Apa Kabar?")
57         calc = calc + 1
58
59 loop(int(input("NPM:")))
60
61 #3
62 def loop3(x):
63     y = 0
64     a = int(x[4])
65     b = int(x[5])
66     c = int(x[6])
67     d = x[4:7]
68     e = a+b+c
69
70     while y < e:
71         y += 1
72         print("Hello, " + d + " Apa Kabar?")
73     return x
74 loop3(input("NPM:"))
75
76 #4
77 def value4(x):
78     n = x[4]
79     print("Hello, " + n + " Apa Kabar?")
80     return x
81 value4(input("NPM:"))
82
83 #5
84 def loopdown(x):
85     for i in x:
86         print(i)
87     return i
88 loopdown(input("NPM:"))
89
90 #6
91 def tambah(n):
```

```

92     hasil = 0
93     for i in n:
94         hasil += int(i)
95     print(str(hasil))
96
97 tambah(input("NPM:"))
98
99 #7
100 def kali(n):
101     hasil = 0
102     for i in n:
103         hasil *= int(i)
104     print(str(hasil))
105
106 kali(input("NPM:"))
107
108 #8
109 def nilaigenap(x):
110     npm = [a,b,c,d,e,f,g]
111
112     for n in npm:
113         if(n % 2 == 0):
114             if(n != 0):
115                 print(n, end = "")
116 nilaigenap(input("NPM:"))
117
118 #9
119 def nilaiganjil(x):
120     npm = [a,b,c,d,e,f,g]
121
122     for n in npm:
123         if(n % 2 != 0):
124             print(n, end = "")
125 nilaiganjil(input("NPM:"))
126
127 #10
128 def prim(x):
129     npm = [a,b,c,d,e,f,g]
130
131     for n in npm:
132         if(n % 2) == 0:
133             print(end="")
134         else:
135             print(n, end = "")
136 prim(input("NPM:"))

```

#### ▪ data main.py

```

1 import praktik
2
3 a = 1
4 b = 3
5 result = praktik.coba3(a,b)

```

### 3.2.3 Handling Error

Error yang di dapat dari mengerjakan tugas ini adalah type error, cara menaggulaginya dengan cara mengecek kembali codingannya, kemudian run kembali aplikasinya berikut adalah contoh Penggunaan fungsi try dan exception :

```
1 def error(a,b):  
2     try :  
3         c = a+b  
4         print(c)  
5     except TypeError:  
6         print("We Are Different")  
7 a = 10  
8 b = 2
```

## BAB 4

---

# JUDUL BAGIAN KEEMPAT

---

### 4.1 Fathi Rabbani / 1164074

#### 4.1.1 Teori

##### 1. Sejarah dan Penjelasan CSV

Penggunaan dari format file CSV itu sendiri untuk memudahkan pembuatan data dengan menggunakan tanda koma sebagai pembatas dari datanya agar mudah untuk dibaca pada kolom.

CSV sendiri dibuat untuk dapat menangani pembuatan sejumlah data yang berukuran besar, mempermudah program dalam membaca datanya kedalam kolom - kolom. seperti contoh dalam membacanya menggunakan aplikasi Excel sehingga mempermudah dalam proses import dan ekspor datanya. csv sendiri sudah ada pada tahun 1972 dengan pengembangnya adalah IBM namun penggunaannya masuk pada tahun 1983 yang berbarengan dengan adanya SuperCalc spreadsheet.

##### 2. Aplikasi CSV

- Microsoft Excel
- Open Office Calc
- Google Docs
- Libre Office
- Apache Open Office

### 3. Menulis dan Membaca csv di Excel atau Spreadsheet

Menulis, cara menuliskan csv adalah dengan menggunakan tanda baca koma pada bagian data yang ingin dipisah contohnya

```

1 #Membuat data file CSV
2 import csv
3
4 siswa = [
5     ('arslan', 'A', 90),
6     ('bayu', 'B', 85),
7     ('niko', 'A', 80),
8     ('abdul', 'B', 90),
9     ('dahlan', 'C', 70)
10 ]
11
12 # tentukan lokasi file, nama file, dan inisialisasi csv
13 f = open('siswa.csv', 'w')
14 w = csv.writer(f)
15 w.writerow(('Nama', 'Kelas', 'Nilai'))
16
17 # menulis file csv
18 for s in siswa:
19     w.writerow(s)
20
21 # menutup file csv
22 f.close()

```

Membaca, file csv dapat dibaca pada program aplikasi Excel dengan menampilkan hasil data dari setiap data yang dipisah dengan tanda baca koma menjadi kolom  
- kolom hasilnya ada pada 5.7

### 4. Library CSV

CSV atau comma separated value adalah salah satu tipe file yang digunakan secara luas di dunia programming. Tidak hanya itu CSV pun sering digunakan dalam pengolahan informasi yang dihasilkan spreadsheet untuk diproses lebih lanjut melalui mesin analitik. CSV pun dianggap sebagai file yang agnostik karena dapat digunakan oleh berbagai database untuk proses backup data.

### 5. Library Pandas

pandas merupakan library pada pemrograman python yang berguna untuk mengolah dan menyediakan struktur data serta analisa data yang mudah untuk dibaca dan dipahami seperti pada struktur data tabel. pandas dapat melakukan

proses perbandingan data, penggabungan dataset, penanganan dataset yang hilang dll. pandas dapat juga digunakan sebagai pemrosesan data Statistik dengan pembacaan datanya menggunakan struktur Spreadsheet.

## 6. Fungsi pada Library CSV

### ▪ Menulis data CSV

```

1 #Membuat data file CSV
2 import csv
3
4 siswa = [
5     ('arslan', 'A', 90),
6     ('bayu', 'B', 85),
7     ('niko', 'A', 80),
8     ('abdul', 'B', 90),
9     ('dahlan', 'C', 70)
10 ]
11
12 # tentukan lokasi file , nama file , dan inisialisasi csv
13 f = open('siswa.csv', 'w')
14 w = csv.writer(f)
15 w.writerow(('Nama', 'Kelas', 'Nilai'))
16
17 # menulis file csv
18 for s in siswa:
19     w.writerow(s)
20
21 # menutup file csv
22 f.close()

```

### ▪ Hasil dari menullis data CSV

```

1 #Hasil pembuatannya
2 Nama,Kelas,Nilai
3 arslan,A,90
4 bayu,B,85
5 niko,A,80
6 abdul,B,90
7 dahlan,C,70

```

### ▪ Membaca data CSV

```

1 #Membaca data file CSV
2 import csv
3
4 # tentukan lokasi file , nama file , dan inisialisasi csv
5 f = open('siswa.csv', 'r')
6 reader = csv.reader(f)
7
8 # membaca baris per baris
9 for row in reader:
10     print row
11
12 # menutup file csv
13 f.close()

```

### ▪ Hasil pembacaan data CSV

```
1 #Hasil pembacaan datanya
2 [ 'Nama', 'Kelas', 'Nilai' ]
3 [ 'arslan', 'A', '90' ]
4 [ 'bayu', 'B', '85' ]
5 [ 'niko', 'A', '80' ]
6 [ 'abdul', 'B', '90' ]
7 [ 'dahlan', 'C', '70' ]
```

## 7. Fungsi pada Library Pandas

### ▪ Kode

```
1 #Pandas
2 import pandas as pd
3 import numpy as np
4 data = np.array(['a','b','c','d'])
5 karakter = pd.Series(data)
6 print(karakter)
```

### ▪ Hasil

```
1 #hasilnya
2 0 a
3 1 b
4 2 c
5 3 d
```

	A	B	C	D	E	F
1	nama	prodi	kelas	alamat	telpon	ipk
2	fathi	D4 TI	3C	Sariasih 4	82216644485	3.5
3	zulfikar	D4 TI	3C	Sariasih 1	82211644459	3
4	Alwan	D4 TI	3A	Sariasih 4	89775221335	3.75
5	Bayu	D3 TI	3A	Sariasih 4	87523146055	3.3
6						

**Gambar 4.1** hasil csv pada Excel

### 4.1.2 Praktek

#### 1. Fungsi Library CSV List

```

1 # In[1]
2 import csv
3
4 def rdcsv():
5     with open('1164074_csv.csv') as r:
6         read = csv.reader(r)
7         lst = list(read)
8         print(lst)

```

## 2. Fungsi Library CSV Dictionary

```

1 # In[2]
2 def dctcsv():
3
4     with open('1164074_csv.csv') as bk:
5         r = csv.DictReader(bk)
6         l = list(r)
7         print(l)

```

## 3. Fungsi Library pandas List

```

1 # In[1]
2 import pandas as bear
3 coba = bear.read_csv('1164074_pandas.csv')

```

## 4. Fungsi Library pandas Dictionary

```

1 # In[2]
2 coba2 = bear.read_csv('1164074_pandas.csv')
3 form = coba2.to_dict(orient='records')

```

## 5. Fungsi Library pandas Date to DataFrame

```

1 # In[3]
2 coba3 = bear.read_csv('1164074_pandas.csv', parse_dates=['tanggal'])
3 print(coba3)

```

## 6. Fungsi Library pandas Index Column

```

1 # In[4]
2 coba4 = bear.read_csv('1164074_pandas.csv')
3 coba4.set_index('nama')

```

## 7. Fungsi Library pandas Change Att Column

```

1 # In[6]
2 coba5 = bear.read_csv('1164074_pandas.csv', header=0, names=[
    'Nama', 'Tanggal Lahir', 'Program Studi', 'Kelas', 'Alamat
    Mahasiswa', 'IPK Mahasiswa'])
3 print(coba5)

```

## 8. Fungsi Library CSV main.py



```

1 # In[1]: Write
2 import c4.1164074_csv as libcsv
3
4 libcsv.writesv()
5
6 # In[2]: Read
7 libcsv.rdcsv2()

```

## 9. Fungsi Library pandas main2.py

```

1 # In[1]: Write
2 import c4.1164074_pandas as libpandas
3
4 libpandas.writesv()
5
6 # In[2]: Read
7 libpandas.rdcsv()

```

### 4.1.3 Error

#### ▪ Error Code

ValueError: Date column tanggal already in dict

#### ▪ Keterangan Error

data sudah berupa dictionary sehingga tidak memerlukan pemecahan lagi dengan code berikut

```

1 coba3 = bear.read_csv('1164074_pandas.csv', parse_dates={
2     'tanggal':[0, 1]})
3     print(coba3)

```

#### ▪ Solusi

```

1 coba3 = bear.read_csv('1164074_pandas.csv', parse_dates=['tanggal
2     '])
3     print(coba3)

```

## 4.2 Kevin Natanael Nainggolan 1174059

### 1. Apa itu fungsi csv, jelaskan sejarah dan contohnya

- 1 Fungsi : File csv berfungsi untuk mencari data akan menjadi lebih efektif ,
- 2 dan juga mempermudah penginputan data ke dalam database secara sederhana .
- 3 Sejarah : File csv muncul pertama kali sekitar 10 tahun sebelum Personal
- 4 Computer pertama didunia sekitar tahun 1972, akan tetapi
- 5 sebutan file csv digunakan pertama kali pada tahun 1983.

## 2. Aplikasi-aplikasi apa saja yang bisa menciptakan file csv?

- 1 — Notepad
- 2 — Notepad++
- 3 — Microsoft Excel
- 4 — Corel Quatro Pro
- 5 — Apache Open Office.

## 3. Jelaskan bagaimana cara menulis dan membaca file csv di excel atau spreadsheet

- 1 — Buka aplikasi Microsoft Excel kemudian buat dokumen baru
- 2 — Tulis judul kolom untuk setiap informasi yang ingin di masukan atau catat ,
- 3 kemudian tulis semua informasi dalam kolom dengan sesuai.
- 4 — Jika sudah selesai maka save dengan cara pilih menubar File lalu pilih Save
- 5 As
- 6 — Lalu isikan nama file tersebut dan midifikasi dengan memilih tipe file yang
- 7 tersedia tersebut menjadi .csv
- 8 — File csv sudah berhasil terbentuk menggunakan Microsoft Excel
- 9
- 10 #
- 11
- 12 — Buka aplikasi Microsoft Excel kemudian pilih menu Open
- 13 — Cari tempat file csv yang ingin dibuka , kemudian pilih Open
- 14 — File csv sudah berhasil dibaca menggunakan Microsoft Excel

## 4. Jelaskan sejarah library csv

- 1 Pada file csv , tanda koma atau titik koma diartikan sebagai pembatas suatu
- 2 kolom. List-directed memasukan output didefinisikan dalam FORTRAN 77.
- 3 List-directed tipe menggunakan tanda baca koma atau spasi sebagai pembatas ,
- 4 sehingga karakter yang tidak diambil tidak dapat mengandung tanda baca koma
- 5 ataupun spasi. Hal tersebut yang diambil oleh file csv. Tipe csv didukung
- 6 dengan library untuk banyak bahasa pemrograman , kebanyakan yang menspesifikasikan pembatas field , pemisah desimal , pengkodean karakter ,
- 7 dan yang lainnya.

## 5. Jelaskan sejarah library pandas

- 1 Pada tahun 2008, pengembangan pandas dimulai oleh AQR Capital Management.
- 2 Pada akhir tahun 2009 pandas menjadi Open Sourced , dimana disupport oleh
- 3 banyak komunitas atau individu di dunia untuk mengembangkan pandas. Sejak
- 4 tahun 2015, pandas menjadi NumFOCUS proyek sponsor , ini juga membantu

- 5 suksesnya pengembangan dari pandas itu sendiri. pandas merupakan struktur
- 6 data dan data analysis tools untuk bahasa pemrograman Python, dan merupakan
- 7 BSD-licensed library yang menjadikannya memiliki performa yang tinggi.

## 6. Jelaskan fungsi-fungsi yang terdapat di library csv

- 1 – Tanda baca koma atau titik koma : Menjadi pemisah antar kolom
- 2 – Tanda baca kutip dua : Menjadi cara untuk memasukan sebuah kalimat atau untuk
- 3 memasukan karakter spasi sebagai data pada kolom informasi
- 4 – Inputan pada baris pertama akan menjadi Header, dimana akan menjadi nama
- 5 sebuah kolom, dan masih banyak yang lainnya

## 7. Jelaskan fungsi-fungsi yang terdapat di library pandas

- 1 Pada pandas sedikit berbeda, dimana inputan data berbentuk seperti
- 2 peng-inputan pada variabel pada umumnya, hanya saja menggunakan tanda kutip
- 3 satu untuk menandakan sebuah informasi pada kolom kemudian tanda kurung kotak
- 4 yang didalamnya berisi informasi data dari kolom tersebut. dan lain sebagainya.

## 4.3 Yusniar Nur Syarif Sidiq/1164089

### 4.3.1 Pemahaman Teori

1. Apa itu fungsi file csv, jelaskan sejarah dan contoh.

File csv merupakan jenis file khusus yang dapat kita buat dan edit di dalam Excel. File csv akan menyimpan informasi data yang dipisahkan dengan koma atau tanda titik koma, dimana artinya file csv tidak menyimpan data dalam bentuk kolom. Saat pertama kali rilis, excel menggunakan format file dalam bentuk biner yaitu BIFF sebagai format file utama. Namun setelah Microsoft merilis Office System 2007, Excel telah menggantikan format utamanya menjadi XML. Meskipun mendukung format XML baru, Excel masih mendukung format BIFF, tidak hanya itu excel juga telah mendukung format CSV, DBF, SYLK, DIF, dan format-format lainnya. Fungsi dari file csv itu sendiri adalah mempermudah dalam pencarian data dan pengimputan data ke dalam database sederhana. File csv mulai digunakan pada tahun 1983 akan tetapi format file csv sudah ada dari tahun 1972. Contoh file dengan format csv dapat dilihat pada figure 4.2

2. Aplikasi - aplikasi apa saja yang bisa menciptakan file csv.

1	'NOPESE	'TAHUNKEPESERTAAN','NMPENGURUS','NOART','NAMAART'
2	'12160101	'040006','2017','EMA TUMANGGER','12160101604000601','LUKDI
3	'12160101	'040006','2017','EMA TUMANGGER','12160101604000602','EMA TI
4	'12160101	'040005','2017','RAEN MANIK','12160101604000501','RAHMAT TU
5	'12160101	'040005','2017','RAEN MANIK','12160101604000502','RAEN MANI
6	'12160101	'040005','2017','RAEN MANIK','12160101604000503','RAMADHAN
7	'12160101	'040004','2017','ROMA MANIK','12160101604000405','IMOIA SIRN

**Gambar 4.2** Contoh File CSV

Untuk membuat file dengan format CSV, kita dapat menggunakan software bawaan Microfsoft Office yaitu Excel. Bukan hanya Microsoft Excel, kita juga dapat membuat file CSV dengan bantuan text editor. Jika kita ingin membuat file csv secara online dapat menggunakan Google Spreadshare. Apabila OS PC kita menggunakan Linux dapat menggunakan LibreOfficecalc.

3. Jelaskan bagaimana cara menulis dan membaca file csv di excel atau spread-sheet.

Cara membuat file CSV dengan Excel

- Buka software Microsoft Excel
- Pilih new document
- Buatlah judul kolom yang ingin kita rekam
- Isikan informasi - informasi pada setiap kolom
- Simpan dengan menggunakan metode save as
- Cari dan pilih format csv
- Pilih button save untuk melakukan penyimpanan

Cara membaca file CSV dengan Excel

- Buka software Microsoft Excel
- Lakukan perintah open file
- Cari file csv yang sudah kita buat sebelumnya
- Pilih button open untuk membaca file csv pada Microsoft Excel

Cara membaca file csv dari Excel

```

1 # In [01]:
2 import csv
3
4 # In [02]:
5 with open('Datasetyn.csv') as csv_file:
6     csv_reader = csv.reader(csv_file, delimiter=",")
7     line_count = 0
8     for row in csv_reader:
9         print(row)

```

Cara membuat file csv

```

1 with open('AnakKelas.csv', mode='w') as kelas_file:
2     kelas_writer = csv.writer(kelas_file, delimiter=',',
3                               quotechar='"')
4     kelas_writer.writerow(['Yusniar', 'D4 TI', '3C'])
5     kelas_writer.writerow(['Imron', 'D4 TI', '3C'])
6     kelas_writer.writerow(['Cokro', 'D4 TI', '3C'])

```

#### 4. Jelaskan sejarah library csv.

Pada tahun 1972 adalah terbentuknya format file csv namun bukan hanya itu saja, pada saat itupun terbentuk juga yang namanya library pandas. Seiring dengan lahirnya bahasa pemrograman python, library mulai dibuat dan dikembangkan oleh Kevin Altis. Dengan kata lain CSV dibentuk pada tahun 1972 dan sudah satu paket baik dalam librarynya maupun format filenya.

#### 5. Jelaskan sejarah library pandas.

Developer yang bernama Wes McKinney telah mengajarkan pandas pada tahun 2008 ketika ia berada di AQR Capital Management, karena kebutuhan akan alat kinerja tinggi yang fleksibel untuk melakukan analisis kuantitatif pada data keuangan. Sebelum meninggalkan AQR, dia dapat meyakinkan manajemen untuk mengizinkan membuka sumber library. Pegawai AQR lainnya yaitu Chang She, telah bergabung dengan upaya ini pada 2012 sebagai kontributor utama kedua ke library. Pada tahun 2015, pandas telah menandatangani sebagai proyek NumFocus yang disponsori secara fiskal. Pada saat itulah Library Pandas mulai berjalan dan digunakan.

#### 6. Jelaskan fungsi-fungsi yang terdapat di library csv.

Ada dua fungsi pada library csv, yaitu csv.read dan csv.writer. Dimana fungsi tersebut memiliki tugas yang berbeda-beda. Untuk csv.reader bertugas sebagai membaca file csv sedangkan csv.writer bertugas membuat file csv.

#### 7. Jelaskan fungsi-fungsi yang terdapat di library pandas.

Untuk library pandas sama saja dengan library csv namun bedanya hanya cara penulisan source codenya saja. Untuk membaca file csv pada library pandas membutuhkan perintah pandas.read.csv sedangkan untuk membuat file csv membutuhkan perintah pandas.write.csv.

### 4.3.2 Keterampilan Pemrograman

#### 1. Membaca file csv pada lib csv dengan mode list

```

1 # In[01]: Dimana akan melakukan import pada library csv
2 import csv
3
4 # In[02]: Membaca file csv dengan mode list
5 def ReadModeList():
6     with open('1164089.csv') as csv_file:
7         csv_reader = csv.reader(csv_file, delimiter=',')
8         for row in csv_reader:
9             print(row)

```

## 2. Membaca file csv pada lib csv dengan mode dictionary

```

1 # In[03]: Membaca file csv dengan mode dictionary
2 def ReadModeDict():
3     with open('1164089.csv', mode='r') as csv_file:
4         csv_reader = csv.DictReader(csv_file)
5         for row in csv_reader:
6             print(row)

```

## 3. Membaca file csv pada lib pandas dengan mode list

```

1 # In[01]: Melakukan import pada library pandas
2 import pandas as pd
3
4 # In[02]: Membaca file csv dengan mode list
5 def ReadModelListPd():
6     df = pd.read_csv('1164089.csv')
7     print(df)

```

## 4. Membaca file csv pada lib pandas dengan mode dictionary

```

1 # In[02]: membaca file csv dengan mode dictionary
2 def ReadModelDictPd():
3     df = pd.read_csv('1164089.csv')
4     dt = pd.DataFrame.from_dict(df)
5     print(dt)

```

## 5. Mengubah format tanggal menjadi standar DataFrame

```

1 # In[03]: Mengubah format tanggal menjadi standar DataFrame
2 def MengubahFormat():
3     df = pd.read_csv('1164089.csv', parse_dates=['Tanggal Daftar'])
4     print(df)

```

## 6. Mengubah index kolom

```

1 # In[04]: Mengubah Index Kolom
2 def MengubahIndexKolom():
3     df = pd.read_csv('1164089.csv')
4     df.index = ['Row_1', 'Row_2']
5     print(df)

```

## 7. Mengubah atribut atau nama kolom

```

1 # In[05]: Mengubah atribut
2 def MengubahAtribut():
3     df = pd.read_csv('1164089.csv')
4     df.columns = ['Colom_1', 'Colom_2', 'Colom_3', 'Colom_4']
5     print(df)

```

## 8. Membuat program NPM\_main.py dan isikan bagaimana cara membaca file csv dan membuat file csv

```

1 lib = __import__('1164089_csv')
2
3 lib.ReadModeList()
4 lib.ReadModeDict()
5
6 lib.Write()

```

9. Membuat program NPM\_main2.py dan isikan bagaimana cara membaca file csv dan membuat file csv dengan lib pandas

```

1 lib = __import__('1164089_pandas')
2
3 lib.ReadModelListPd()
4 lib.ReadModelDictPd()
5
6 lib.WritePd()

```

### 4.3.3 Penanganan Error

1. Tuisikan peringatan error yang di dapat dari mengerjakan praktek ketiga ini, dan jelaskan cara penanganan error tersebut. Buatlah fungsi try except untuk mengulangi error tersebut.

Dimana error yang saya dapat merupakan atribut error yaitu dimana kondisi penulisan source code atribut salah atau tidak ditemukan, untuk lebih jelasnya dapat dilihat pada figure 4.3

`AttributeError: module '1164089_pandas' has no attribute 'ReadModelListPd'`

**Gambar 4.3** Atribut Error

Cara penganannya yaitu ubah atribut yang terdapat pada file python sama dengan yang kita buat di file sebelumnya. Coontohnya dapat dilihat pada figure 4.4

```

def ReadModelDictPd():
    ...
3 lib.ReadModelListPd()

```

**Gambar 4.4** Atribut Error

Dimana penulisannya harus sama persis dikarenakan Python bersifat case sensitif.

Untuk penulisan Try Ecept dapat dilihat pada source code dibawah ini,

```

1 def TryExcept():
2     try:
3         df = pd.read_csv('1164089.csv')
4         print(df)
5     except SyntaxError:
6         print("Kesalahan Dalam Penulisan Atribut")

```

## 4.4 Alit Fajar Kurniawan 1174057

### 4.4.1 Pemahaman Teori

1.
  - Fungsi : File csv berfungsi melakukan pencarian data agar menjadi lebih mudah dan cepat, dan juga mempermudah penginputan data ke dalam database secara lebih sederhana.
  - Sejarah : Pada 10 tahun yang lalu File csv muncul pertama kali sebelum Personal Computer pertama di dunia sejak sekitar tahun 1972, akan tetapi sebutan file csv digunakan pertama kali pada tahun 1983.
  - Contohnya : Anda dapat mengekspor kontak dari Google ke dalam file CSV, kemudian mengimpornya ke Outlook.
2. Ada banyak aplikasi yang dapat membuat file berformat CSV, diantaranya adalah : Pada Windows
  - Microsoft Excel 2013
  - Microsoft Works
  - CCorel Quattro Pro
  - Apache OpenOffice
  - LibreOffice
  - Microsoft Notepad
  - Intuit Quicken 2015
  - GenScriber

Pada Linux

- Apache OpenOffice
- LibreOffice
- GenScriber

Pada Mac OS

- Microsoft Excel 2011
- Planamesa NeoOffice
- Apache OpenOffice
- LibreOffice
- GenScriber

3. Jelaskan bagaimana cara menulis dan membaca file csv di excel atau spread-sheet?
  - Untuk menulis file csv harus berupa baris dan kolom atau bisa juga di sebut berupa tabel.



- Untuk membacanya file csv dipisahkannya menggunakan koma atau titik koma.
4. sejarah library csv : Library csv menyediakan fungsionalitas untuk membaca dan menulis ke file CSV. Dirancang untuk bekerja di luar kotak dengan file CSV yang dihasilkan Excel, memudahkan untuk bekerja dengan berbagai format CSV. Library csv berisi objek dan kode lain untuk membaca, menulis, dan memproses data ke file CSV.
  5. Sejarah library pandas : Tahun 2008, pengembangan pandas dimulai oleh AQR Capital Management. Pada akhir tahun 2009 pandas menjadi Open Sourced, dimana disupport oleh banyak komunitas atau individu di dunia untuk mengembangkan pandas. Sejak tahun 2015, pandas menjadi NumFOCUS proyek sponsor, ini juga membantu suksesnya pengembangan dari pandas itu sendiri. pandas merupakan struktur data dan data analysis tools untuk bahasa pemrograman Python, dan merupakan BSD-licensed library yang menjadikannya memiliki performa yang tinggi.
  6. Jelaskan fungsi-fungsi yang terdapat di library csv? Terdapat 2 fungsi yang bisa digunakan oleh library csv Pertama, fungsi membaca file csv. fungsi ini bisa menggunakan list dan dictionary Dengan list :

```

1 def readcsv():
2     with open('1174057.csv') as csv_file:
3         csv_reader = csv.reader(csv_file, delimiter=',')
4         line_count = 0
5         for row in csv_reader:
6             if line_count == 0:
7                 print(f'Column names are {", ".join(row)}')
8                 line_count += 1
9             else:
10                print(f'\t NPM : {row[0]} Nama : {row[1]} Kelas :
11                {row[2]}')
12                line_count += 1

```

Dengan dictionary :

```

1 def readdictionary():
2     with open('1174057.csv', mode='r') as csv_file:
3         csv_reader = csv.DictReader(csv_file)
4         line_count = 0
5         for row in csv_reader:
6             if line_count == 0:
7                 print(f'Column names are {", ".join(row)}')
8                 line_count += 1
9             print(f'\t NPM : {row["npm"]} Nama : {row["nama"]}
10             Kelas : {row["kelas"]}')
11             line_count += 1

```

Kedua, fungsi menulis file csv.

```

1 def write():
2     with open('test-tulis.csv', mode='w') as employee_file:

```

```

3     employee_writer = csv.writer(employee_file, delimiter=',',
4     , quotechar='"', quoting=csv.QUOTE_MINIMAL)
5
6     employee_writer.writerow(['Kurniawan', 'Manajer', '
    Agustus'])

```

7. Jelaskan fungsi-fungsi yang terdapat di library pandas Hampir sama dengan library csv, tp library pandas penulisannya lebih sederhana dan terlihat lebih rapih dari pada library csv.

```

1 def readpandas():
2     df = pandas.read_csv('1174057.csv')

```

#### 4.4.2 Keterampilan Pemograman

1. Buatlah fungsi (file terpisah/library dengan nama NPMcsv.py) untuk membuka file csv dengan lib csv mode list.

```

1 #Jawaban No. 1
2 def bukaModeListCsv():
3     with open('teori_alit.csv') as csv_file:
4         csv_reader = csv.reader(csv_file, delimiter=',')
5         for row in csv_reader:
6             print(row[0], row[1], row[2])

```

**Listing 4.1** Fungsi untuk membuka file CSV dengan lib CSV mode list.

2. Buatlah fungsi (file terpisah/library dengan nama NPMcsv.py) untuk membuka file csv dengan lib csv mode dictionary.

```

1 #Jawaban No. 2
2 def bukaModeDictCsv():
3     with open('teori_alit.csv', mode='r') as csv_file:
4         csv_reader = csv.DictReader(csv_file)
5         for row in csv_reader:
6             print(row['npm'], row['nama'], row['kelas'])

```

**Listing 4.2** Fungsi untuk membuka file CSV dengan lib CSV mode dictionary.

3. Buatlah fungsi (file terpisah/library dengan nama NPMpandas.py) untuk membuka file csv dengan lib pandas mode list.

```

1 #Jawaban No. 3
2 def bukaModeListPandas():
3     df = pandas.read_csv('teori_alit.csv')
4     print(df)

```

**Listing 4.3** Fungsi untuk membuka file CSV dengan lib Pandas mode list.

4. Buatlah fungsi (file terpisah/library dengan nama NPMpandas.py) untuk membuka file csv dengan lib pandas mode dictionary.

```

1 #Jawaban No. 4
2 def bukaModeDictPandas():
3     df = pandas.read_csv('teori_alit.csv')
4     dt = pandas.DataFrame.from_dict(df)
5     print(dt)

```

**Listing 4.4** Fungsi untuk membuka file CSV dengan lib Pandas mode dictionary.

5. Buat fungsi baru di NPMpandas.py untuk mengubah format tanggal menjadi standar dataframe.

```

1 #Jawaban No. 5
2 def ubahFormatTanggal():
3     df = pandas.read_csv('teori_alit.csv', parse_dates=['tanggal
4     lahir'])
5     print(df)

```

**Listing 4.5** Fungsi untuk mengubah format tanggal menjadi standar dataframe.

6. Buat fungsi baru di NPMpandas.py untuk mengubah index kolom.

```

1 #Jawaban No. 6
2 def ubahIndexKolom():
3     df = pandas.read_csv('teori_alit.csv')
4     df.index = ['Row_1', 'Row_2']
5     print(df)

```

**Listing 4.6** Fungsi untuk mengubah index kolom.

7. Buat fungsi baru di NPMpandas.py untuk mengubah atribut atau nama kolom.

```

1 #Jawaban No. 7
2 def ubahNamaKolom():
3     df = pandas.read_csv('teori_alit.csv')
4     df.columns = ['Col_1', 'Col_2', 'Col_3', 'Col_4']
5     print(df)

```

**Listing 4.7** Fungsi untuk mengubah atribut atau nama kolom.

8. Buat program main.py yang menggunakan library NPMcsv.py yang membuat dan membaca file csv.

```

1 lib = __import__('1174057csv')
2
3 lib.bukaModeDictCsv()
4
5 lib.tulisCsv()

```

**Listing 4.8** Membuat dan membaca file CSV menggunakan library 1174006pandas.

9. Buat program main2.py yang menggunakan library NPMpandas.py yang membuat dan membaca file csv.

```

1 lib = __import__('1174057pandas')
2
3 lib.bukaModeListPandas()
4 lib.bukaModeDictPandas()
5
6 lib.tulisCsvPandas()

```

**Listing 4.9** Membuat dan membaca file CSV menggunakan library 1174006pandas.

#### 4.4.3 Ketrampilan Penanganan Error

Error yang di dapatkan dari mengerjakan tugas ini adalah type error, mengatasinya dengan cara mengecek kembali codingannya kemudian run kembali aplikasinya berikut contoh Penggunaan fungsi try dan exception

```

1 a = 1
2 b = 10
3 c = "7"
4
5 try :
6     d = a + c
7 except TypeError:
8     print("ini teh beda atuhh type datanya mana bisa bersatu")

```

### 4.5 Kevin Natanael Nainggolan 1174059

#### 1. Apa itu fungsi csv, jelaskan sejarah dan contohnya

- 1 Fungsi : File csv berfungsi untuk mencari data akan menjadi lebih efektif ,
- 2 dan juga mempermudah penginputan data ke dalam database secara sederhana .
- 3 Sejarah : File csv muncul pertama kali sekitar 10 tahun sebelum Personal
- 4 Computer pertama di dunia sekitar tahun 1972, akan tetapi
- 5 sebutan file csv digunakan pertama kali pada tahun 1983.

#### 2. Aplikasi-aplikasi apa saja yang bisa menciptakan file csv?

- 1 – Notepad
- 2 – Notepad++
- 3 – Microsoft Excel
- 4 – Corel Quatro Pro
- 5 – Apache Open Office .

#### 3. Jelaskan bagaimana cara menulis dan membaca file csv di excel atau spread-sheet

- 1 – Buka aplikasi Microsoft Excel kemudian buat dokumen baru
- 2 – Tulis judul kolom untuk setiap informasi yang ingin di masukan atau catat ,

- 3 kemudian tulis semua informasi dalam kolom dengan sesuai.
- 4 — Jika sudah selesai maka save dengan cara pilih menubar File  
lalu pilih Save
- 5 As
- 6 — Lalu isikan nama file tersebut dan midifikasi dengan memilih  
tipe file yang
- 7 tersedia tersebut menjadi .csv
- 8 — File csv sudah berhasil terbentuk menggunakan Microsoft Excel
- 9
- 10 #
- 11
- 12 — Buka aplikasi Microsoft Excel kemudian pilih menu Open
- 13 — Cari tempat file csv yang ingin dibuka, kemudian pilih Open
- 14 — File csv sudah berhasil dibaca menggunakan Microsoft Excel

#### 4. Jelaskan sejarah library csv

- 1 Pada file csv, tanda koma atau titik koma diartikan sebagai  
pembatas suatu
- 2 kolom. List-directed memasukan output didefinisikan dalam FORTRAN  
77.
- 3 List-directed tipe menggunakan tanda baca koma atau spasi sebagi  
pembatas,
- 4 sehinga karakter yang tidak diambil tidak dapat mengandung tanda  
baca koma
- 5 ataupun spasi. Hal tersebut yang diambil oleh file csv. Tipe csv  
didukung
- 6 dengan library untuk banyak bahasa pemrograman, kebanyakan yang  
menspesifikasikan pembatas field, pemisah desimal, pengkodean  
karakter,
- 8 dan yang lainnya.

#### 5. Jelaskan sejarah library pandas

- 1 Pada tahun 2008, pengembangan pandas dimulai oleh AQR Capital  
Management.
- 2 Pada akhir tahun 2009 pandas menjadi Open Sourced, dimana  
disupport oleh
- 3 banyak komunitas atau individu di dunia untuk mengembangkan  
pandas. Sejak
- 4 tahun 2015, pandas menjadi NumFOCUS proyek sponsor, ini juga  
membantu
- 5 suksesnya pengembangan dari pandas itu sendiri. pandas merupakan  
struktur
- 6 data dan data analysis tools untuk bahasa pemrograman Python, dan  
merupakan
- 7 BSD-licensed library yang menjadikannya memiliki performa yang  
tinggi.

#### 6. Jelaskan fungsi-fungsi yang terdapat di library csv

- 1 — Tanda baca koma atau titik koma : Menjadi pemisah antar kolom
- 2 — Tanda baca kutip dua : Menjadi cara untuk memasukan sebuah  
kalimat atau untuk
- 3 memasukan karakter spasi sebagai data pada kolom informasi

- 4 – Inputan pada baris pertama akan menjadi Header, dimana akan menjadi nama
- 5 sebuah kolom, dan masih banyak yang lainnya

## 7. Jelaskan fungsi-fungsi yang terdapat di library pandas

- 1 Pada pandas sedikit berbeda, dimana inputan data berbentuk seperti
- 2 peng-inputan pada variabel pada umumnya, hanya saja menggunakan tanda kutip
- 3 satu untuk menandakan sebuah informasi pada kolom kemudian tanda kurung kotak
- 4 yang didalamnya berisi informasi data dari kolom tersebut. dan lain sebagainya.

```

1 a = 1
2 b = 10
3 c = "7"
4
5 try :
6     d = a + c
7 except TypeError:
8     print("ini teh beda atuhh type datanya mana bisa bersatu")

```

## 4.6 Muhammad Iqbal Panggabean 1174063

### 4.6.1 Praktek

1. Buatlah fungsi (file terpisah/library dengan nama NPMcsv.py) untuk membuka file csv dengan lib csv mode list.

```

1 #Jawaban No. 1
2 with open('1174063 teori.csv') as csv_file:
3     csv_reader = csv.reader(csv_file, delimiter=',')
4     line_count = 0
5     for row in csv_reader:
6         if line_count == 0:

```

**Listing 4.10** Fungsi untuk membuka file CSV dengan lib CSV mode list.

2. Buatlah fungsi (file terpisah/library dengan nama NPMcsv.py) untuk membuka file csv dengan lib csv mode dictionary.

```

1         line_count += 1
2     else:
3         print(f'\t{row[0]} dengan nama {row[1]} kelas {
4         row[2]}.')
5         line_count += 1
6         print(f'Processed {line_count} lines.')

```

**Listing 4.11** Fungsi untuk membuka file CSV dengan lib CSV mode dictionary.

3. Buatlah fungsi (file terpisah/library dengan nama NPMpandas.py) untuk membuka file csv dengan lib pandas mode list.

```
1 df = pandas.read_csv('1174063 teori.csv')
2 print(df)
3
4 #Jawaban No. 4
```

**Listing 4.12** Fungsi untuk membuka file CSV dengan lib Pandas mode list.

4. Buatlah fungsi (file terpisah/library dengan nama NPMpandas.py) untuk membuka file csv dengan lib pandas mode dictionary.

```
1 df = pandas.read_csv('1174063 teori.csv')
2 print(df)
3
4 #Jawaban No. 4
```

**Listing 4.13** Fungsi untuk membuka file CSV dengan lib Pandas mode dictionary.

5. Buat fungsi baru di NPMpandas.py untuk mengubah format tanggal menjadi standar dataframe.

```
1 df = pandas.read_csv('1174063 teori.csv')
2 dt = pandas.DataFrame.from_dict(df)
3 print(dt)
4
5 #Jawaban No. 5
```

**Listing 4.14** Fungsi untuk mengubah format tanggal menjadi standar dataframe.

6. Buat fungsi baru di NPMpandas.py untuk mengubah index kolom.

```
1 df = pandas.read_csv('1174063 teori.csv', parse_dates=['
2 tanggal lahir'])
3 print(df)
4
5 #Jawaban No. 6
```

**Listing 4.15** Fungsi untuk mengubah index kolom.

7. Buat fungsi baru di NPMpandas.py untuk mengubah atribut atau nama kolom.

```
1 df = pandas.read_csv('1174063 teori.csv')
2 df.index = ['Row_1', 'Row_2']
3 print(df)
4
5 #Jawaban No. 7
```

**Listing 4.16** Fungsi untuk mengubah atribut atau nama kolom.

8. Buat program main.py yang menggunakan library NPMcsv.py yang membuat dan membaca file csv.

```

1 lib = __import__('1174063csv')
2
3 lib.bukaModeDictCsv()
4
5 lib.tulisCsv()

```

**Listing 4.17** Membuat dan membaca file CSV menggunakan library 1174063pandas.

9. Buat program main2.py yang menggunakan library NPMpandas.py yang membuat dan membaca file csv.

```

1 lib = __import__('1174063pandas')
2
3 lib.bukaModeListPandas()
4 lib.bukaModeDictPandas()
5
6 lib.tulisCsvPandas()

```

**Listing 4.18** Membuat dan membaca file CSV menggunakan library 1174063pandas.

## 4.6.2 Ketrampilan Penanganan Error

Error yang di dapat dari mengerjakan tugas ini adalah type error, cara menaggulaginya dengan cara mengecek kembali codingannya kemudian run kembali aplikasinya berikut contoh Penggunaan fungsi try dan exception

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Mar 27 02:50:06 2019
4
5 @author: Muhammad iqbal
6 """
7
8 def ErrorGais():
9     import csv
10    try:
11        with open('1174063csv.py', newline='') as csvfile:
12            reader = csv.DictReader(csvfile)
13            for row in reader:
14                print(row['npm'], row['nama'], row['key'])
15    except KeyError:
16        print("ERRROOOOORRRRRR")
17
18 ErrorGais()

```





## BAB 5

---

# JUDUL BAGIAN KELIMA

---

### 5.1 Alit Fajar Kurniawan

## Pemahaman Teori

#### 5.1.1 No. 1

Apa itu fungsi device manager di windows dan folder /dev di linux?

Device manager adalah perangkat lunak untuk menampilkan seluruh perangkat keras yang diinisialisasi atau sudah terhubung dengan sistem operasi Windows. Perangkat keras tersebut bisa berupa harddisk, kartu VGA, sound, keyboard, perangkat USB dan lain-lainnya. fungsi lain dari device manager yaitu, Menunjukkan status mengenai suatu perangkat keras, Mengelola driver perangkat keras, Menonaktifkan dan mengaktifkan perangkat keras, Menunjukkan informasi detail mengenai suatu perangkat keras, Mengidentifikasi konflik antar perangkat keras, dan Memberitahukan terjadinya masalah pada perangkat keras.

Folder /dev merupakan representasi dari drive yang terhubung ke sistem operasi

Linux dan oleh sistem dianggap sebagai file-file direktori. Biasanya sering ditampilkan direktori seperti `/dev/sda1` yang mewakili Drive SATA pertama dalam sistem.

### 5.1.2 No. 2

Jelaskan langkah-langkah instalasi driver dari arduino!

Berikut ini adalah langkah-langkah instalasi driver dari Arduino UNO di Windows:

1. Pertama pastikan Arduino IDE telah terinstall.
2. Lalu hubungkan port USB Arduino Uno ke port USB PC.
3. Kemudian PC anda akan mendeteksi perangkat baru yang terpasang dan akan muncul pop seperti ini.
4. Karena Arduino Uno baru pertama kali terpasang, maka akan muncul pop up error seperti ini.
5. Buka "Start" lalu cari Device Manager, kemudian klik "Device Manager".
6. Setelah Device Manager terbuka, silahkan cari "Unknown Device" yang berada di Other Device.
7. Kemudian klik kanan pada "Unknown Device", lalu pilih "Update Driver Software".
8. Setelah itu muncul window baru, lalu pilih "Browse my computer for driver software".
9. Lalu cari folder yang terinstall Arduino IDE dengan mengklik browse. Kemudian klik "Next".
10. Windows akan mencari dan menginstall driver yang berada pada folder tersebut.
11. Setelah itu akan muncul window, lalu klik "Install".
12. Jika berhasil terinstal maka akan muncul window seperti ini.

### 5.1.3 No. 3

#### ▪ Membaca Baudrate dari Komputer

1. Pertama buka "Start". Cari "Device Manager", lalu klik.
2. Kemudian pilih "Ports (COM & LPT)".
3. Klik dua kali pada COM yang terhubung.
4. Pilih tab "Port Settings", lalu lihat di "Bit per second".

- Membaca Port dari Komputer

1. Pertama buka "Start". Cari "Device Manager", lalu klik.
2. Kemudian pilih "Ports (COM & LPT)".
3. Port dari Arduino telah terbaca oleh PC.

#### 5.1.4 No. 4

Jelaskan sejarah library pyserial!

PySerial adalah paket Python yang memfasilitasi komunikasi serial antara PC dengan perangkat keras eksternal. PySerial menyediakan antarmuka untuk berkomunikasi melalui protokol komunikasi serial. Komunikasi serial adalah salah satu protokol komunikasi komputer tertua. Protokol komunikasi serial mendahului spesifikasi USB yang digunakan oleh komputer dan perangkat keras lain seperti mouse, keyboard, dan webcam. USB adalah singkatan dari Universal Serial Bus. USB dan dibangun di atas dan memperluas antarmuka komunikasi serial asli.

#### 5.1.5 No. 5

Jelaskan fungsi-fungsi apa saja yang dipakai dari library pyserial!

Fungsi-fungsi yang dipakai dari library PySerial, yaitu:

1. Serial - fungsi ini untuk membuka port serial.
2. write(data) - fungsi ini menulis data lewat port serial.
3. readline - fungsi ini membaca sebuah string dari port serial.
4. read(size) - fungsi ini untuk membaca jumlah byte dari port serial.
5. close - fungsi ini untuk menutup port serial.

#### 5.1.6 No. 6

Jelaskan kenapa butuh perulangan dan tidak butuh perulangan dalam membaca serial!

Pada saat membaca serial di Arduino diperlukan perulangan agar bisa membaca data secara berulang kali sehingga data yang muncul banyak. Sedangkan apabila tidak membutuhkan perulangan maka Arduino hanya akan membaca data sekali saja.

#### 5.1.7 No. 7

Jelaskan bagaimana cara membuat fungsi yang menggunakan pyserial!

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Apr  2 20:59:08 2019
4
5 @author: alitfajarkurniawan
6 """
7 import serial
8
9 def cobaarduino():
10     ser = serial.Serial("COM10",9600)
11     print(ser.name)
12
13 cobaarduino()

```

**Listing 5.1** Fungsi yang menggunakan pyserial.

### 5.1.8 Cek Plagiat

## Keterampilan Pemrograman

### 5.1.9 No. 1

Buatlah fungsi file terpisah dengan nama NPM realtime.py untuk mendapatkan data langsung dari arduino

```

1 import serial
2
3 def getData():
4     ser = serial.Serial('COM10',9600)
5     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
6
7 getData()

```

**Listing 5.2** Fungsi untuk mendapatkan data dari Arduino.

### 5.1.10 No. 2

Buatlah fungsi file terpisah dengan nama NPM save.py untuk mendapatkan data langsung dari arduino dengan looping

```

1 import serial
2
3 def getDataLoop():
4     ser = serial.Serial('COM10',9600)
5     while (1):
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7

```

```
8 getDataLoop()
```

**Listing 5.3** Fungsi untuk mendapatkan data langsung dari Arduino dengan looping.

### 5.1.11 No. 3

Buatlah fungsi file terpisah dengan nama NPM realtime.py untuk mendapatkan data dari arduino dan langsung ditulis kedalam file csv

```
1 import serial
2 import csv
3
4 def writeCsv():
5     ser = serial.Serial('COM10', 9600)
6     with open('gps.csv', mode='w') as csv_file:
7         fieldnames = ['location']
8         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
9
10        writer.writeheader()
11        while (1):
12            data = ser.readline().decode("utf-8").strip('\n').strip('
\r')
13            writer.writerow({'location': data})
14
15 writeCsv()
```

**Listing 5.4** Fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.

### 5.1.12 No. 4

Buatlah fungsi file terpisah dengan nama NPM csv.py untuk membaca file csv hasil arduino dan mengembalikan ke fungsi

```
1 import csv
2
3 def readCsv():
4     with open('gps.csv', mode='r') as csv_file:
5         reader = csv.DictReader(csv_file)
6         for row in reader:
7             print(row['location'])
8
9 readCsv()
```

**Listing 5.5** Fungsi untuk membaca file CSV hasil Arduino dan mengembalikan fungsi.

## Penanganan Error

### 5.1.13 No. 1

Tuliskan peringatan error yang didapat dari mengerjakan praktek kelima ini, dan jelaskan cara penanganan error tersebut. dan Buatlah satu fungsi yang menggunakan try except untuk menanggulangi error tersebut.

Peringatan error di praktek kelima ini, yaitu:

- **Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- **Name Error** NameError adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Solusinya adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.
- **Type Error** TypeError adalah exception yang akan terjadi apabila pada saat dilakukannya eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusi dari error ini adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

Fungsi yang menggunakan try except untuk menanggulangi error.

```

1 import serial
2
3 def tryExceptError():
4     try:
5         ser = serial.Serial('COM10', 9600)
6         print(sre.readline().decode("utf-8").strip('\n').strip('\r'))
7     except SyntaxError:
8         print("Kesalahan penulisan syntax")
9     except NameError:
10        print("Variable tersebut tidak ada")
11    except TypeError:
12        print("Tipe data salah")
13    except:
14        print("Terjadi sebuah kesalahan")
15
16 tryExceptError()
```

**Listing 5.6** Fungsi untuk menanggulangi error menggunakan Try Except.

## 5.2 Muhammad Iqbal Panggabean

# Pemahaman Teori

### 5.2.1 Pertanyaan No.. 1

Apa itu fungsi device manager di windows dan folder /dev di linux?

Fungsi device manager antara lain :

1. Mengidentifikasi konflik antar perangkat keras.
2. Menunjukkan informasi detil suatu hardware.
3. Disable dan Enable hardware.
4. Mengelola driver hardware.
5. Menunjukkan status suatu hardware.

Folder /dev berisi file device, baik device blok maupun device karakter. Di dalamnya setodaknya ada file biner yang bernama MAKEDEV untuk membuat device secara manual.

### 5.2.2 Pertanyaan No. 2

Jelaskan langkah-langkah instalasi driver dari arduino!

Berikut ini adalah langkah-langkah instalasi driver dari Arduino UNO di Windows:

1. Hubungkan sistem minimum Arduino Uno ke komputer dengan kabel USB type B (kabel Printer).
2. Lalu pada bagian kanan desktop PC anda, akan muncul popup Installing device driver software.
3. Sistem operasi Windows tidak menyediakan driver untuk Arduino Uno.
4. Buka Device Manager, caranya pada bagian Search Program and Files lalu ketikkan device manager (tanpa tanda petik). Kemudian bagian Control Panel akan muncul halaman Device Manager, selanjutnya klik untuk menjalankan.
5. Cari yang bernama Unknown device yang berada pada bagian Other device, biasanya ada tanda seru berwarna kuning, itu disebabkan karena penginstallan tidak berjalan dengan sempurna.
6. Klik kanan pada Unknown device kemudian pilih Update Driver Software.
7. Pilih Browse my computer for driver software.
8. Arahkan lokasi folder ke folder ..arduino-1.0.5 drivers. Pastikan check-box lalu centang include subfolders. Klik Next untuk melanjutkan instalasi driver.
9. Kemudian lanjutkan dengan mengklik Install pada tampilan Windows Security.
10. Jika instalasi driver berhasil maka akan muncul Windows has successfully updated your driver software.
11. Perhatikan dan ingat nama COM Arduino Uno, karena nama COM ini yang akan digunakan untuk meng-upload program nantinya.



### 5.2.3 Pertanyaan No. 3

Jelaskan bagaimana cara membaca baudrate dan port dari komputer yang sudah terinstall driver!

#### Membaca Port dari Komputer

1. Hubungkan modul TX-RX serial dengan komputer melalui serial port menggunakan DB9 cable extension.
2. Buka Hyper Terminal dengan menekan start kemudian All programs lalu Accessories kemudian Communications lalu Hyper Terminal.
3. Ketik nama untuk Connection Description, misal coba, kemudian tekan OK.
4. Pada Connect to, pilihlah COM port yang dipakai di Connect using, kemudian tekan OK.
5. Masukkan nilai-nilai port settingnya, sesuai dengan DCE-nya. Kemudian tekan OK.

### 5.2.4 Pertanyaan No. 4

Jelaskan sejarah library pyserial!

PySerial adalah library/modul Python siap-pakai dan gratis yang dibuat untuk memudahkan kita dalam membuat program komunikasi data serial RS232 dalam bahasa Python. Jika modul USB-2REL dapat kita kontrol dengan mudah menggunakan Python dan PyUSB (lihat pembahasannya di sini dan di sini), maka modul SER-2REL juga dapat kita kontrol dengan mudah menggunakan Python dengan bantuan modul PySerial.

### 5.2.5 Pertanyaan No. 5

Jelaskan fungsi-fungsi apa saja yang dipakai dari library pyserial!

Fungsi-fungsi yang dipakai dari library PySerial, yaitu:

1. Serial - fungsi ini untuk membuka port serial.
2. write(data) - fungsi ini menulis data lewat port
3. serial.close() - fungsi ini untuk menutup port serial.
4. readline() - fungsi ini membaca sebuah string dari port serial.
5. read(size) - fungsi ini untuk membaca jumlah byte dari port serial.

### 5.2.6 Pertanyaan No. 6

Jelaskan kenapa butuh perulangan dan tidak butuh perulangan dalam membaca serial!

Pada saat membaca serial di Arduino diperlukan perulangan agar bisa membaca data secara berulang kali sehingga data yang muncul banyak. Sedangkan apabila tidak membutuhkan perulangan maka Arduino hanya akan membaca data sekali saja.

### 5.2.7 Pertanyaan No. 7

Jelaskan bagaimana cara membuat fungsi yang menggunakan pyserial!

Fungsi yang berada pada Python, dibuat dengan nama kata kunci def kemudian diikuti dengan nama fungsinya pada python. Seperti halnya dengan blok kode yang lain, kita juga harus memberikan identasi untuk menuliskan isi fungsi.

### 5.2.8 Cek Plagiat

## 5.3 Fathi Rabbani / 1164074

### 5.3.1 Teori

#### 1. Fungsi Device manager pada Windows dan folder /dev pada Linux

Device Manager dalam komputer Windows merupakan perluasan dari Microsoft Management Console. Device Manager menampilkan seluruh hardware yang bisa di-inisialisasi (dikenali) oleh Windows. Tampilannya sudah terorganisir (dikelompokkan) sedemikian rupa sehingga akan memudahkan pengelolaan setiap hardware yang ada. berikut ini adalah kegunaan dari Device Manager :

- Menunjukkan status suatu hardware
- Menunjukkan informasi detail suatu hardware
- Mengelola driver hardware
- Disable dan Enable hardware
- Meng-identifikasi konflik antar hardware, dll.

folder /dev pada linux merupakan direktori yang berfungsi untuk menyimpan konfigurasi device atau hardware dari sistem, seperti harddisk (hda, sda), terminal (tty) etc.

#### 2. Instalasi Driver Arduino

Berikut ini adalah tahapan dalam melakukan proses instalasi Driver Arduino pada OS Windows :

- (a) Download terlebih dahulu Arduino IDE yang akan digunakan (yang terbaru 1.8.9) lalu install.
- (b) Hubungkan perangkat Arduino Uno ke komputer dengan kabel USB type B
- (c) Lalu pada bagian kanan didesktop PC anda, akan muncul popup Installing device driver software
- (d) Jika pada desktop tidak muncul popup maka buka Windows Device Manager (Start<Control Panel<Hardware) dan cari Unknown Device lalu klik kanan dan Pilih Update Driver.
- (e) Pada screen selanjutnya pilih Browse my computer for driver software lalu klik Next.
- (f) Lalu klik tombol Browse dan pilih lokasi pengistalan Arduino IDE yang telah diinstall, dan klik OK.
- (g) Anda akan menerima pemberitahuan bahwa drivernya akan diinstall, lalu klik tombol Install dan Driver Arduino anda akan terupdate dengan sendirinya.

### 3. Membaca baudrate dari Komputer

setelah melakukan Instalasi IDE dan Driver Arduino pada Komputer maka anda dapat melakukan check baud ratenya pada IDE yang akan menampilkan datanya melalui Toolbar Menu<Serial Monitor.

### 4. Sejarah PySerial

PySerial merupakan sebuah library data yang digunakan untuk melakukan komunikasi ke port serial yang diutamakan adalah sistem mikrokontroller. Py-Serial pertama kali diluncurkan pada tahun 2002 hingga sekarang.

### 5. Fungsi dari library PySerial

- (a) serial - membuka port serial
- (b) write() - untuk menulis data
- (c) readline - membaca sebuah string
- (d) read() - membaca data
- (e) close - menutup port serial

### 6. Kenapa Butuh Loop dan Tidak dalam membaca Serial

Karena perulangan digunakan untuk membaca seluruh data pada serial yang ada setiap baris. Perulangan digunakan agar data dapat muncul secara terus menerus atau realtime. jika tidak dibuatkan perulangan maka data yang terbaca akan sekali saja dan tidak menghasilkan nilai yang bagus untuk akhirnya.

### 7. Membuat fungsi menggunakan PySerial

```

1 import serial
2
3 def tryARD():
4     ser = serial.Serial("COM4", 9600)
5     print(ser.name)
6
7 tryARD()

```

**Listing 5.7** Fungsi yang menggunakan pyserial.

### 5.3.2 Praktikum

#### 1. Membuat fungsi untuk mendapatkan data langsung dari arduino

```

1 import serial
2 import csv
3
4 def rSerial():
5     ser = serial.Serial('COM3', 9600, timeout=2)
6     x = ser.readline()
7     return x

```

#### 2. Membuat fungsi untuk mendapatkan data langsung dari arduino dengan looping

```

1 import serial
2
3 def rSerialLoop():
4     ser = serial.Serial('COM3', 9600, timeout=2)
5     lst = []
6     while True:
7         x = ser.readline()
8         if x:
9             print(x)
10            lst.append(x)
11        else:
12            break
13    return lst

```

#### 3. Membuat fungsi untuk mendapatkan data langsung dari arduino lalu me write data tersebut ke dalam file csv

```

1 def wCSV():
2     ser = serial.Serial('COM3', 9600)
3     with open('1164074_try.csv', mode='w') as csv_file:
4         fieldnames = ['coba']
5         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
6
7         writer.writeheader()
8         while (1):
9             data = ser.readline().decode("utf-8").strip('\n').
10            strip('\r')
11            writer.writerow({'coba': data})
12 wCSV()

```

4. Membuat fungsi untuk membaca csv yang berisikan data dari arduino dan mengembalikan fungsi

```

1 import csv
2
3 def rCSV():
4     with open('1164074_try.csv', 'r') as f:
5         read = csv.reader(f)
6         lst = list(read)
7         return lst

```

### 5.3.3 Penanganan Error

```

1 import serial
2
3 def tryExceptError():
4     try:
5         ser = serial.Serial('COM3', 9600)
6         print(sre.readline().decode("utf-8").strip('\n').strip('\r'))
7     except SyntaxError:
8         print("Syntax Error")
9     except NameError:
10        print("Variable tidak dimiliki")
11    except TypeError:
12        print("Tipe data tidak benar")
13    except:
14        print("Terjadi sebuah kesalahan pada code")
15 tryExceptError()

```

## 5.4 Kevin Natanae Nainggolan 1174059

### 5.4.1 TEORI

1. Apa itu fungsi device manager di windows dan folder /dev di linux

```

1 Device Manager : device manager berfungsi untuk menampilkan dan
   mengelola
2 semua hardware yang sudah dinstall ataupun dapat di instalasi ke
   dalam
3 perangkat windows.
4
5 folder /dev pada linux: Di dalam sistem operasinya, perangkat
   yang terhubung
6 akan dianggap sebagai file. Di dalam folder /dev file-file
   tersebut ada

```

2. Jelaskan langkah-langkah instalasi driver dari arduino

```

1 Langkah – langkah instalasi driver arduino :
2     –download file driver arduino terlebih dahulu dan masukkan ke
   dalam
3     directory yang diinginkan
4     –hubungkan arduino uno anda ke pc anda dengan kabel USB yang
   tersedia

```

- 5     –lalu windows akan memunculkan pop up yang memberitahu bahwa ingin
- 6     menginstall driver, tapi nanti tidak akan menemukan drivernya
- 7     –buka Device Manager
- 8     –cari unknown device di dalam Device Manager di dalam tab other device
- 9     –klik kanan pada unknown device tersebut lalu pilih update driver software
- 10    –pilih browse my computer for driver software lalu masukkan directory
- 11    dimana anda menyimpan driver arduino yang telah anda download tadi
- 12    –setelah itu klik install dan tunggu hingga proses selesai

### 3. Jelaskan bagaimana cara membaca baudrate dan port dari komputer yang sudah terinstall driver

- 1     Untuk melihat atau membaca BAUDRATE dan PORT kita hanya perlu menginstal
- 2     Arduino IDE, setelah itu buka menu serial monitor yang berada di tab TOOLS.
- 3     Dari sana akan terlihat baik BAUDRATE dan PORT yang sedang digunakan oleh
- 4     arduino anda.

### 4. Jelaskan sejarah library pyserial

- 1     PySerial merupakan sebuah library yang digunakan untuk komunikasi ke port
- 2     serial terutama untuk MICROCONTROLLER. PySerial melakukan debut pertamanya
- 3     pada tahun 2002 yang semakin berkembang dalam setiap versinya hingga tahun
- 4     2017 lalu.

### 5. Jelaskan fungsi-fungsi apa saja yang dipakai dari library pyserial

- 1     –stop() untuk menghentikan pembacaan program
- 2     –serial.to\_bytes(sequence): berfungsi untuk mengubah sequence ke dalam bytes
- 3     agar dapat diupload ke arduino.
- 4     –close(): untuk menutup port dan menghentikan pembacaan program

### 6. Jelaskan kenapa butuh perulangan dalam tidak butuh perulangan dalam membaca serial

- 1     Dengan menggunakan pengulangan kita dapat mengambil data berkali-kali tanpa
- 2     harus mengeksekusi file python tersebut berulang-ulang. Tanpa perulangan
- 3     juga penting karena dapat digunakan di saat tertentu seperti jika ingin
- 4     mengukur suhu ruangan yang hanya dilakukan pada saat tertentu tidak terus
- 5     menerus.

## 7. Jelaskan bagaimana cara membuat fungsi yang menggunakan pyserial

```

1 Untuk membuat fungsi yang menggunakan Pyserial kita hanya perlu
  untuk
2 menginisialisasi pembubatan fungsi dengan menggunakan
3 def namafungsi() : lalu masukkan pyserial tersebut dengan
  indentasi.
4 Atau cukup dengan menggunakan fungsi while loop dengan menggunakan
  while true :

```

### 5.4.2 PRAKTEK

#### 1. Buatlah fungsi untuk mendapatkan data langsung dari arduino

```

1 import serial
2
3 arduino = serial.Serial('COM6',9600)
4 def sensorjarak():
5     data = int(arduino.readline())
6     print(data)
7
8 try:
9     while (True):
10         sensorjarak()
11 except KeyboardInterrupt:
12     arduino.close()
13     print('Sebentar aja cukup ^_^')

```

#### 2. Buatlah fungsi untuk mendapatkan data langsung dari arduino dengan looping

```

1 import serial
2
3
4
5 def LOOP():
6     ser = serial.Serial('COM6',9600)
7     while (1):
8         print(ser.readline().decode("utf-8").strip('\n').strip('\r\
          r'))
9 LOOP()

```

#### 3. Buatlah fungsi untuk mendapatkan data dari arduino dan langsung ditulis kedalam file csv

```

1 import serial
2 import csv
3
4 def getData():
5     ser = serial.Serial('COM6',9600)
6     print(ser.readline().decode("utf-8").strip('\n').strip('\r\
          r'))
7     getData()
8
9 def writeCsv():
10     ser = serial.Serial('COM6',9600)
11     with open('sibin.csv', mode='w') as csv_file:

```

```

12     fieldnames = ['jaraknya']
13     writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
14     writer.writeheader()
15     while (1):
16         data = ser.readline().decode("utf-8").strip('\n').
strip('\r')
17         writer.writerow({'jaraknya': data})
18 writeCsv()

```

4. Buatlah fungsi untuk membaca le csv hasil arduino dan mengembalikan ke fungsi

```

1 import csv
2
3
4
5 def macacsv():
6     with open('sibin.csv', mode='r') as csv_file:
7         reader = csv.DictReader(csv_file)
8         for row in reader:
9             print(row['jaraknya'])
10 macacsv()

```

### 5.4.3 PENANGANAN ERROR

1. Tuliskan peringatan error yang didapat dari mengerjakan praktek ketiga ini, dan jelaskan cara penanganan error tersebut. dan Buatlah satu fungsi yang menggunakan gunakan try except untuk menanggulangi error tersebut

```

1 import serial
2
3
4
5 def panggil():
6     try:
7         ser = serial.Serial('COM6',9600)
8         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
9     except SyntaxError:
10         print("Kesalahan penulisan syntax")
11     except NameError:
12         print("Variable tersebut tidak ada")
13     except TypeError:
14         print("Tipe data salah")
15 panggil()

```

## 5.5 Yusniar Nur Syarif Sidiq / 1164089

### 5.5.1 Pemahaman Teori

1. Apa itu fungsi device manager di windows dan folder /dev di linux?



Device manager merupakan perangkat lunak untuk menampilkan seluruh perangkat keras yang di-inisialisasi atau dikenali oleh sistem operasi Windows. Device Manager membantu dalam mengelola atau memange semua perangkat keras yang terpasang dan terdeteksi dalam sistem Windows. Fungsi device manager yaitu menunjukkan informasi detail mengenai suatu hardware, mengelola driver hardware, menaktifkan dan nonaktifkan hardware, dan memberikan terjadinya masalah pada hardware. Folder /dev merupakan representasi dari drive yang terhubung ke sistem operasi Libux dan sistem akan menganggap sebagai file-file direktori.

## 2. Jelaskan Langkah-langkah instalasi driver arduino

- Install terlebih dahulu Arduino IDE pada PC anda
- Pasangkan port USB Arduino Uno pada ports USB PC anda, figure 5.8
- Bula Start dan cari Device Manager, figure 5.9
- Carilah Unknown Device yang berbeda di Other Device, figue 5.10
- Klik kanan lalu pilih Update Driver Software, figure 5.11
- Akan muncul windows baru lalu pilih menu Browse my computer for driver software, figure 5.12
- Cari folder Arduino lalu klik next, dan windows akan mencari, figure 5.13
- Lalu klik install, figure 5.14

## 3. Jelaskan bagaimana cara membaca baudrate dan port dari komputer yang sudah terinstall driver !

### Cara Membaca Baudrate

- Pertama kita buka Device Manager
- Lalu pilih Port (COM & LPT)
- Klik dua kali pada COM yang telah tersambung
- Pilih menu Port Setting dan disitulah kita dapat melihat baudrate

### Cara Membaca Port

- Pertama kita buka Device Manager
- Lalu pilih Port (COM & LPT)
- Dan lihat apabila Arduino sudah tersambung maka akan terbaca

## 4. Jelaskan sejarah Library pyserial !.

PySerial adalah paket Python yang memfasilitasi komunikasi serial antara PC dengan perangkat keras eksternal. PySerial menyediakan antarmuka untuk berkomunikasi melalui protokol komunikasi serial. Komunikasi serial adalah salah satu protokol komunikasi komputer tertua. Protokol komunikasi serial mendahului spesifikasi USB yang digunakan oleh komputer dan perangkat keras

lain seperti mouse, keyboard, dan webcam. USB adalah singkatan dari Universal Serial Bus. USB dibangun di atas dan memperluas antarmuka komunikasi serial asli.

## 5. Jelaskan fungsi-fungsi apa saja yang dipakai pada lib pyserial !

### Fungsi-fungsi Pada Lib Pyserial

- write(data) berfungsi untuk menulis data melalui port serial
- readline() berfungsi untuk membaca string dari port serial
- read(size) berfungsi untuk membaca jumlah byte dalam port serial
- Serial berfungsi untuk membuka port serial
- close() berfungsi untuk menutup port serial

## 6. Jelaskan kenapa butuh perulangan dan tidak butuh perulangan dalam membaca serial !

Karena agar kita mendapatkan data yang cukup banyak supaya semakin akurat maka dari itu kita membutuhkan perulangan dan bila tidak dilakukan perulangan maka data hanya muncul sekali saja.

## 7. Jelaskan bagaimana cara membuat fungsi yang menggunakan pyserial !

### Membuat Fungsi Dengan Pyserial

```

1 # In[01] Melakukan import pada module serial
2 import serial
3
4 # In[02]: Membuat Fungsi menggunakan Pyserial
5 def Arduino():
6     ser = serial.Serial(110200)
7     print(ser.name)
8
9 Arduino()
```

Apabila di running dalam spyder maka hasilnya akan seperti pada figure 5.15  
Cek Plagiat dapat dilihat pada figure 5.16

## 5.5.2 Keterampilan Pemrograman

### 1. Buatlah fungsi (file terpisah atau library dengan nama NPM\_realtime.py) untuk mendapatkan data langsung dari arduino !

```

1 # In[01]: Melakukan import pada module serial
2 import serial
3
4 # In[02]: Membuat fungsing mengambil data
5 def mengambilData():
6     ser = serial.Serial('COM1', 9600)
7     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
8
9 mengambilData()
```

Hasil output dari source code diatas dapat dilihat pada figure 5.17

2. Buatlah fungsi (file terpisah atau library dengan nama NPM\_save.py) untuk mendapatkan data langsung dari arduino dengan looping !

```

1 # In[01]: Melakukan import module serial
2 import serial
3
4 # In[02]: Membuat fungsi looping atau pengulangan
5 def Looping():
6     ser = serial.Serial('COM1',9600)
7     while (1):
8         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
9
10 Looping()

```

Dimana source code diatas akan melakukan pengulangan data, seperti pada figure 5.18

3. Buatlah fungsi (file terpisah atau library dengan nama NPM\_realtime.py) untuk mendapatkan data dari arduino dan langsung ditulisa kedalam file csv !

```

1 # In[01]: Melakukan import pada module serial
2 import serial
3
4 # In[02]: Membuat fungsing mengambil data
5 def mengambilData():
6     ser = serial.Serial('COM1',9600)
7     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
8
9     mengambilData()

```

Dimana kita akan membuat file csv dari data arduino, maka outputnya dapat dilihat pada figure 5.19

4. Buatlah fungsi (file terpisah atai library dengan nama NPM\_csv.py) untuk membaca file csv hasil arduino dan mengembalikan ke fungsi !

```

1 # In[01]: Import module csv
2 import csv
3
4 # In[02]: Membuat fungsi untuk membaca file csv
5 def read():
6     with open('1164089.csv', mode='r') as csv_file:
7         reader = csv.DictReader(csv_file)
8         for row in reader:
9             print(row['space'])
10
11 read()

```

Dimana source code diatas akan membaca file csv, seperti pada figure 5.20

### 5.5.3 Keterampilan Penangan Error

1. Dimana error yang saya dapat yaitu syntaxerror yaitu dimana terjadi salah pengetikan pada source code dan cara penanganannya yaitu cari source code yang salah dan rubah sesuai dengan code yang dapat dibaca oleh python. Dimana fungsi try except adalah sebagai berikut,

```

1 # In[01]: Melakukan import modul serial
2 import serial
3
4 # In[02]: Membuat fungsi trxExcept
5 def teError():
6     try:
7         ser = serial.Serial('COM1',9600)
8         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
9     except SyntaxError:
10        print("Kesalahan Dalam Penulisan Syntax")
11
12 teError()

```

## 5.6 Muhammad Iqbal Panggabean / 1174063

### 5.6.1 Keterampilan Pemrograman

1. Buatlah fungsi file terpisah dengan nama NPM realtime.py untuk mendapatkan data langsung dari arduino

```

1 import serial
2
3 def getData():
4     ser = serial.Serial('COM3',9600)
5     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
6
7     getData()

```

**Listing 5.8** Fungsi untuk mendapatkan data dari Arduino.

2. Buatlah fungsi file terpisah dengan nama NPM save.py untuk mendapatkan data langsung dari arduino dengan looping

```

1 import serial
2
3 def getDataLoop():
4     ser = serial.Serial('COM3',9600)
5     while (1):
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7
8     getDataLoop()

```

**Listing 5.9** Fungsi untuk mendapatkan data langsung dari Arduino dengan looping.

3. Buatlah fungsi file terpisah dengan nama NPM realtime.py untuk mendapatkan data dari arduino dan langsung ditulis kedalam file csv

```

1 import serial
2 import csv
3
4 def writeCsv():
5     ser = serial.Serial('COM3', 9600)
6     with open('speedometer.csv', mode='w') as csv_file:
7         fieldnames = ['speed']
8         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
9
10        writer.writeheader()
11        while (1):
12            data = ser.readline().decode("utf-8").strip('\n').
13            strip('\r')
14            writer.writerow({'speed': data})
15 writeCsv()

```

**Listing 5.10** Fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.

4. Buatlah fungsi file terpisah dengan nama NPM csv.py untuk membaca file csv hasil arduino dan mengembalikan ke fungsi

```

1 import csv
2
3 def readCsv():
4     with open('speedometer.csv', mode='r') as csv_file:
5         reader = csv.DictReader(csv_file)
6         for row in reader:
7             print(row['speed'])
8
9 readCsv()

```

**Listing 5.11** Fungsi untuk membaca file CSV hasil Arduino dan mengembalikan fungsi.

## 5.6.2 Penanganan Error

Tuliskan peringatan error yang didapat dari mengerjakan praktek kelima ini, dan jelaskan cara penanganan error tersebut. dan Buatlah satu fungsi yang menggunakan try except untuk menanggulangi error tersebut.

Peringatan error di praktek kelima ini, yaitu:

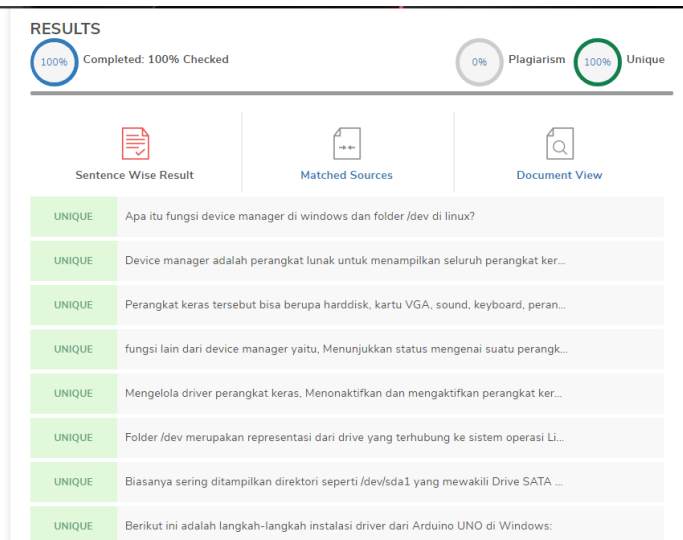
- Syntax Errors Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- Name Error NameError adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Solusinya adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.

- Type Error `TypeError` adalah exception yang akan terjadi apabila pada saat dilakukannya eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusi dari error ini adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

Fungsi yang menggunakan try except untuk menanggulangi error.

```
1 import serial
2
3 def tryExceptError():
4     try:
5         ser = serial.Serial('COM3', 9600)
6         print(sre.readline().decode("utf-8").strip('\n').strip('\r'))
7     except SyntaxError:
8         print("Kesalahan penulisan syntax")
9     except NameError:
10        print("Variable tersebut tidak ada")
11    except TypeError:
12        print("Tipe data salah")
13    except:
14        print("Terjadi sebuah kesalahan")
15
16 tryExceptError()
```

**Listing 5.12** Fungsi untuk menanggulangi error menggunakan Try Except.



**Gambar 5.1** Hasil cek plagiat

[illegible]

**Gambar 5.2** Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dengan looping.

```
location;
DeviceExample.ino

A simple demonstration of TinyGPS++ with an attached GPS module

Testing TinyGPS++ library v. 1.0.2

by Mikal Hart

Location: -6.876163,107.575996

Location: -6.876163,107.575996

Location: -6.876163,107.575996

Location: -6.876162,107.575996

Location: -6.876162,107.575996

Location: -6.876162,107.575996
```

**Gambar 5.3** Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.

```

C:\Users\ASPIRE E15\Desktop\New folder\src\5\1174042\Praktek>python 1174057_csv.py
DeviceExample.ino
A simple demonstration of TinyGPS++ with an attached GPS module
Testing TinyGPS++ library v. 1.0.2
By Mikal Hart

Location: -6.876163,107.575996
Location: -6.876163,107.575996
Location: -6.876163,107.575996
Location: -6.876162,107.575996
Location: -6.876162,107.575996
Location: -6.876162,107.575996
Location: -6.876160,107.576004
Location: -6.876160,107.576004
Location: -6.876160,107.576004

```

**Gambar 5.4** Hasil dari pembacaan fungsi untuk membaca file csv hasil arduino dan mengembalikan fungsi.

```




C:\Users\ASPIRE E15\Desktop\New folder\src\5\1174042\Praktek>python 1174057.py
Variable tersebut tidak ada

```

**Gambar 5.5** Hasil pembacaan fungsi untuk menanggulangi error menggunakan Try Except.

**RESULTS**

100% Completed: 100% Checked 0% Plagiarism 100% Unique




 Sentence Wise Result  Matched Sources  Document View

UNIQUE	Apa itu fungsi device manager di windows dan folder /dev di linux?
UNIQUE	Folder /dev berisi file device, baik device blok maupun device karakter.
UNIQUE	Di dalamnya setidaknya ada file biner yang bernama MAKEDEV untuk membuat ...
UNIQUE	Berikut ini adalah langkah-langkah instalasi driver dari Arduino UNO di Windows:
UNIQUE	1. Hubungkan sistem minimum Arduino Uno ke komputer dengan kabel USB typ...
UNIQUE	2. Lalu pada bagian kanan desktop PC anda, akan muncul pop up "Installing d...
UNIQUE	3. Sistem operasi Windows tidak menyediakan driver untuk Arduino Uno.
UNIQUE	4. Buka Device Manager, caranya pada bagian Search Program and Files lalu ke...
UNIQUE	5. Kemudian bagian Control Panel akan muncul halaman Device Manager, selanjutnya...

**Gambar 5.6** Hasil cek plagiat.

**RESULTS**

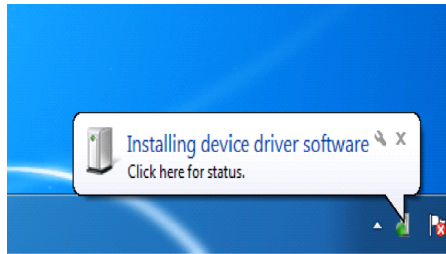
100% Completed: 100% Checked 0% Plagiarism 100% Unique

 Sentence Wise Result  Matched Sources  Document View

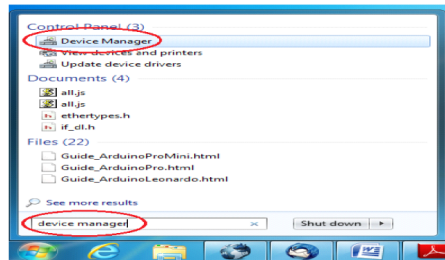
UNIQUE	Fungsi Device manager pada Windows dan folder /dev pada Linux Device Manager...
UNIQUE	Device Manager menampilkan seluruh hardware yang bisa dikenal oleh OS Windo...
UNIQUE	Tampilannya sudah dikelompokkan sehingga akan memudahkan pengelolaan seti...

**Gambar 5.7** Check Plagiarisme

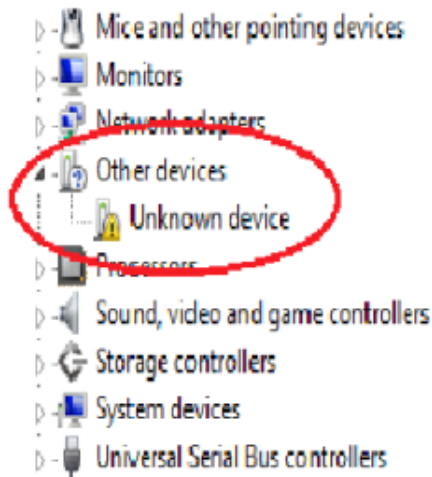




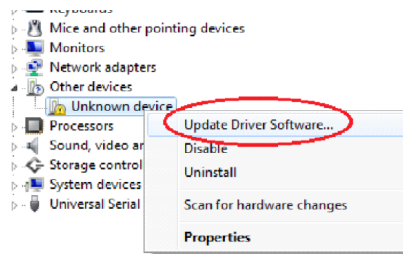
**Gambar 5.8** Memasang Port USB.



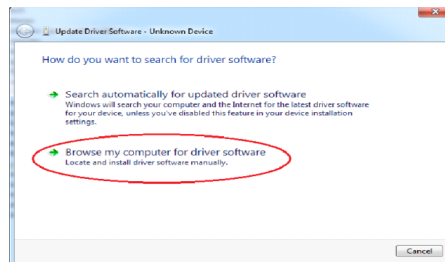
**Gambar 5.9** Mencari Device Manager.



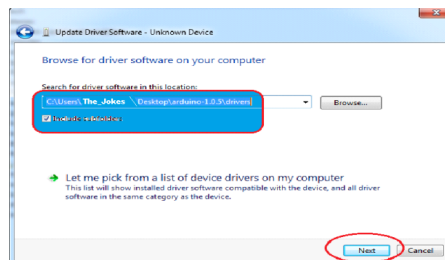
**Gambar 5.10** Unknown Device.



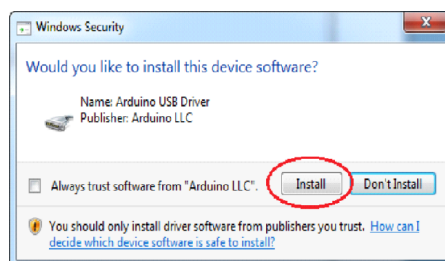
**Gambar 5.11** Update Driver Software.



**Gambar 5.12** Browse my computer for driver software.



**Gambar 5.13** Open Folder Arduino.



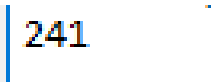
**Gambar 5.14** Install.

```
In [4]: runfile('D:/Perkuliahan/Semester 4/Mengulang Web Service/TH 5/2b_grup2/
src/chapter5/1164089/1164089_Teori.py', wdir='D:/Perkuliahan/Semester 4/Mengulang
Web Service/TH 5/2b_grup2/src/chapter5/1164089')
```

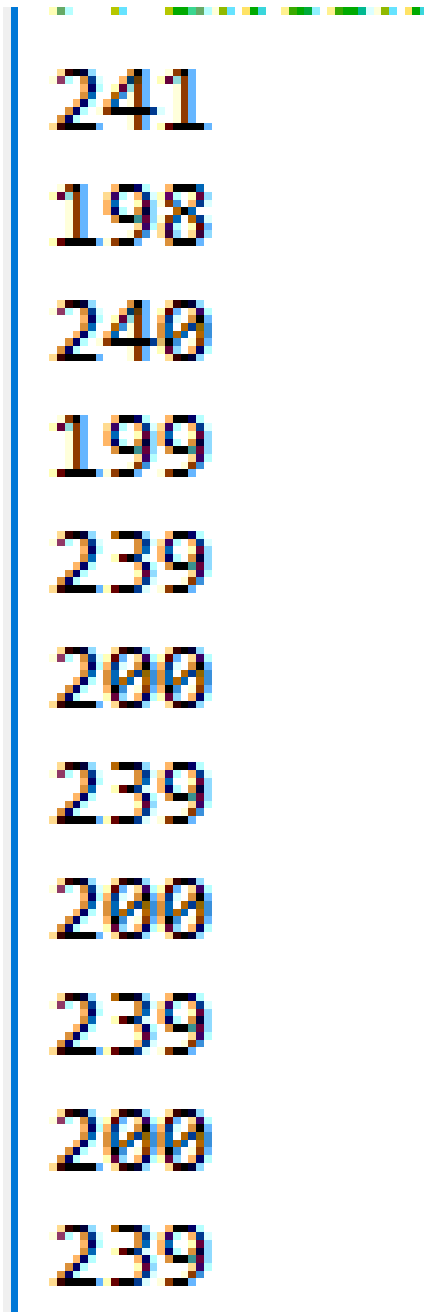
Gambar 5.15 Fungsi Pyserial.



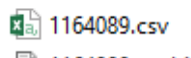
Gambar 5.16 Cek Plagiat.



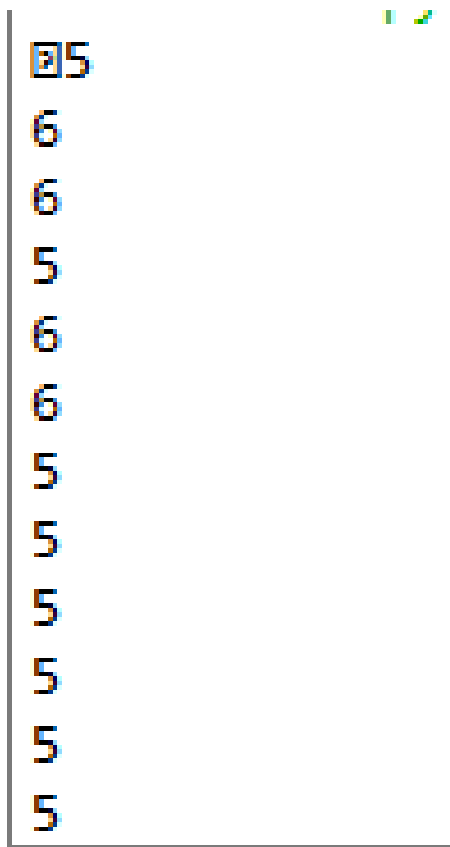
Gambar 5.17 Hasil Data Yang Diambil Dari Arduino.



**Gambar 5.18** Hasil Data Pengulanganl Dari Arduino.



**Gambar 5.19** Membuat File CSV.



**Gambar 5.20** Membaca File CSV.

```
D:\MATA_KULIAH_POLITEKNIK_POS_INDONESIA\PAKROLLYPEHR03\chapter5praktek\maudicommit>python 1174063_realtime.py  
DeviceExample.Ino
```

**Gambar 5.21** Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino.

```

D:\WATA KULIAH POLITEKNIK_POS_INDONESIA\PAKROLLYPEMRO3\chapter5praktek\maudicommit>python 1174063_save.py
DeviceExample.ino
A simple demonstration of TinyGPS++ with an attached GPS module
Testing TinyGPS++ library v. 1.0.2
by Mikal Hart

Location: INVALID
Location: -6.876183,107.575920
Location: -6.876187,107.575927
Location: -6.876187,107.575927
Location: -6.876187,107.575927
Location: -6.876190,107.575927
Location: -6.876190,107.575927
Location: -6.876190,107.575927
Location: -6.876190,107.575927
Location: -6.876190,107.575927
Location: -6.876190,107.575935
Location: -6.876190,107.575935
Location: -6.876190,107.575935
Location: -6.876190,107.575935
Location: -6.876190,107.575935
Location: -6.876190,107.575935

```

**Gambar 5.22** Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dengan looping.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	speed																				
2																					
3	DeviceExample.ino																				
4																					
5	A simple demonstration of TinyGPS++ with an attached GPS module																				
6																					
7	Testing TinyGPS++ library v. 1.0.2																				
8																					
9	by Mikal Hart																				
10																					
11																					
12																					
13	Location: -6.876188,107.576034																				
14	Location: -6.876190,107.576034																				
15	Location: -6.876190,107.576034																				
16																					
17	Location: -6.876190,107.576034																				
18																					
19	Location: -6.876190,107.576034																				
20																					
21	Location: -6.876194,107.576034																				
22																					

**Gambar 5.23** Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.

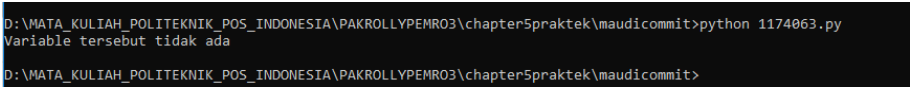
```

D:\WATA KULIAH POLITEKNIK_POS_INDONESIA\PAKROLLYPEMRO3\chapter5praktek\maudicommit>python 1174063_csv.py
DeviceExample.ino
A simple demonstration of TinyGPS++ with an attached GPS module
Testing TinyGPS++ library v. 1.0.2
by Mikal Hart

Location: -6.876188,107.576034
Location: -6.876190,107.576034
Location: -6.876190,107.576034
Location: -6.876190,107.576034
Location: -6.876194,107.576034
Location: -6.876194,107.576034
Location: -6.876194,107.576034

```

**Gambar 5.24** Hasil dari pembacaan fungsi untuk membaca file csv hasil arduino dan mengembalikan fungsi.



```
D:\MATA_KULIAH_POLITEKNIK_POS_INDONESIA\PAKROLLYPEMR03\chapter5praktek\maudicommit>python 1174063.py
Variable tersebut tidak ada
D:\MATA_KULIAH_POLITEKNIK_POS_INDONESIA\PAKROLLYPEMR03\chapter5praktek\maudicommit>
```

**Gambar 5.25** Hasil pembacaan fungsi untuk menanggulangi error menggunakan Try Except.

## BAB 6

---

# JUDUL BAGIAN KEENAM

---

### 6.1 Alit Fajar Kurniawan 1174057

#### 6.1.1 Pemahaman Teori

1. Apa itu fungsi library matplotlib ?

Matplotlib adalah suatu library Python 2D yang dapat menghasilkan plot dengan kualitas yang tinggi dalam berbagai format dan dapat digunakan di berbagai platform. Matplotlib berfungsi sebagai pembuat grafik di berbagai platform, seperti Python dan Jupyter. Grafik yang dibuat menggunakan Matplotlib bisa dibuat dalam berbagai bentuk, seperti grafik garis, batang, lingkaran, histogram, dan sebagainya.

2. Jelaskan langkah-langkah membuat sumbu X dan Y di matplotlib

- Pertama import library Matplotlib.

```
from matplotlib import pyplot as plt
```

- Buat variabel x yang menampung list untuk sumbu x dan variabel y yang menampung list untuk sumbu y.



```
1 x=[1,2,3]
2 y=[5,2,4]
```

- Panggil fungsi plot dan isi parameter pertama dengan variabel x dan parameter kedua dengan variabel y.

```
1 plt.plot(x,y)
```

- Lalu panggil plot tadi dengan memanggil fungsi show.

```
1 plt.show()
```

## Kode Program

```
1 from matplotlib import pyplot as plt
2
3 x=[1,2,3]
4 y=[5,2,4]
5
6 plt.plot(x,y)
7
8 plt.show()
```

**Listing 6.1** Kode program membuat diagram menggunakan Matplotlib.

## Hasil Compile

3. Jelaskan bagaimana perbedaan fungsi dan cara pakai untuk berbagai jenis (bar, histogram, scatter, dll) jenis plot di matplotlib !

### ▪ Bar Graph

Perbedaan bar graph dengan jenis plot yang lain adalah bar graph menggunakan bar atau batang-batang untuk membandingkan data di antara berbagai kategori.

### Kode Program

```
1 from matplotlib import pyplot as plt
2
3 plt.bar
4     ([2013,2014,2015,2016,2017,2018],[20000,25000,30000,35000,40000,45000])
5 label="KFC",color='b',width=.3)
6 plt.bar
7     ([2012.7,2013.7,2014.7,2015.7,2016.7,2017.7],[9000,9500,10000,15000,20000,25000])
8 label="MCD",color='r',width=.3)
9 plt.bar
10     ([2013.3,2014.3,2015.3,2016.3,2017.3,2018.3],[2000,2500,3000,3500,4000,4500])
11 label="PHD",color='y',width=.3)
12 plt.legend()
13 plt.xlabel('Tahun')
```

```

11 plt.ylabel('Jumlah Pengunjung')
12 plt.title('Pengunjung restoran junkfood Dari Tahun ke Tahun')
13 plt.show()

```

**Listing 6.2** Kode program membuat bar graph menggunakan Matplotlib.

## Hasil Compile

### ■ Histogram

Perbedaan histogram dengan jenis plot yang lain adalah histogram akan membuat plot dimana plot yang dimunculkan merupakan gabungan dari beberapa data yang telah dikelompokkan.

### Kode Program

```

1 import matplotlib.pyplot as plt
2 orang =
    [11,22,16,9,10,15,22,55,62,45,21,22,34,42,42,4,2,102,95,85,55,110,1
3 umur = [0,10,20,30,40,50,60,70,80,90,100]
4 plt.hist(orang, umur, histtype='bar', rwidth=0.8, color='r')
5 plt.xlabel('Umur')
6 plt.ylabel('Jumlah Orang')
7 plt.title('Histogram')
8 plt.show()

```

**Listing 6.3** Kode program membuat histogram menggunakan Matplotlib.

## Hasil Compile

### ■ Scatter Plot

Perbedaan scatter plot dengan jenis plot lain adalah scatter plot menampilkan data sebagai kumpulan titik, masing-masing memiliki nilai satu variabel yang menentukan posisi pada sumbu horizontal dan nilai variabel lain menentukan posisi pada sumbu vertikal.

### Kode Program

```

1 import matplotlib.pyplot as plt
2 x = [1,1.5,2,2.5,3,3.5,3.6]
3 y = [7.5,8,8.5,9,9.5,10,10.5]
4
5 x1=[8,8.5,9,9.5,10,10.5,11]
6 y1=[3,3.5,3.7,4,4.5,5,5.2]
7
8 plt.scatter(x,y, label='keuntungan tinggi simpanan rendah',
    color='g')
9 plt.scatter(x1,y1,label='keuntungan rendah simpanan tinggi',
    color='b')
10 plt.xlabel('simpanan dalam ratusan')
11 plt.ylabel('pendapatan dalam ribuan')
12 plt.title('Scatter Plot')
13 plt.legend()
14 plt.show()

```

**Listing 6.4** Kode program membuat scatter plot menggunakan Matplotlib.

## Hasil Compile

### ▪ Area Plot

Perbedaan area plot dengan jenis plot lain adalah area plot digunakan untuk melacak perubahan dari waktu ke waktu untuk dua atau lebih kelompok terkait yang membentuk satu kategori secara keseluruhan.

#### Kode Program

```

1 import matplotlib.pyplot as plt
2 hari = [1,2,3,4,5]
3
4 tidur =[7,8,6,11,7]
5 makan = [2,3,4,3,4]
6 kuliah =[7,8,7,2,5]
7 jalan_jalan = [8,5,7,8,13]
8
9 plt.plot([],[],color='c', label='Tidur', linewidth=5)
10 plt.plot([],[],color='b', label='Futsal', linewidth=5)
11 plt.plot([],[],color='y', label='Kuliah', linewidth=5)
12 plt.plot([],[],color='k', label='Youtube-an', linewidth=5)
13
14 plt.stackplot(hari,tidur,makan,kuliah,jalan_jalan, colors=['c',
15                 'm','y','k'])
16 plt.xlabel('x')
17 plt.ylabel('y')
18 plt.title('Kegiatan Sehari-hari')
19 plt.legend()
20 plt.show()

```

**Listing 6.5** Kode program membuat diagram menggunakan Matplotlib.

## Hasil Compile

### ▪ Pie Plot

Perbedaan pie plot dengan jenis plot lain adalah pie plot digunakan untuk menunjukkan persentase atau data proporsional di mana setiap potongan pie mewakili kategori.

#### Kode Program

```

1 import matplotlib.pyplot as plt
2
3 potong = [7,2,2,12]
4 kegiatan = ['Tidur','Futsal','Kuliah','Youtube-an']
5 kolom = ['c','m','y','g']
6
7 plt.pie(potong,
8         labels=kegiatan,
9         colors=kolom,
10        startangle=90,
11        shadow=True,
12        explode=(0,0,0.2,0),
13        autopct='%1.1f%%')
14
15 plt.title('Kegiatan Sehari-hari')

```

```
16 plt.show()
```

**Listing 6.6** Kode program membuat Pie Plot menggunakan Matplotlib.

### Hasil Compile

#### ▪ Line Graph

Perbedaan line graph dengan jenis plot lain adalah line graph menampilkan diagram dalam bentuk garis.

#### Kode Program

```
1 from matplotlib import pyplot as plt
2
3 y = [4000,6000,10000,13000,14000,17000]
4 x = [2014,2015,2016,2017,2018,2019]
5 plt.plot(x,y)
6 plt.title('Peminat bahasa pemrograman Python')
7 plt.ylabel('Tahun')
8 plt.xlabel('Jumlah Pengguna sistem operasi Python')
9 plt.show()
```

**Listing 6.7** Kode program membuat diagram menggunakan Matplotlib.

### Hasil Compile

4. Jelaskan bagaimana cara menggunakan legend dan label serta kaitannya fungsi tersebut !

- Untuk menggunakan legend definisikan parameter label di tiap fungsi plot. Parameter label digunakan untuk memberikan label pada line sebagai pembeda antar line.

```
1 plt.plot(x,y,'b',label='Team Bapao',linewidth=1)
2 plt.plot(x2,y2,'r',label='Ateam',linewidth=1)
```

**Listing 6.8** Kode program menggunakan parameter label dengan Matplotlib.

- Kemudian panggil fungsi legend.

```
1 plt.legend()
```

**Listing 6.9** Kode program memanggil fungsi legend dengan Matplotlib.

### Kode Program

```
1 from matplotlib import pyplot as plt
2
3 x = [2014,2015,2016,2017,2018,2019]
4 y = [76,87,105,122,148,170]
5 x2 = [2014,2015,2016,2017,2018,2019]
6 y2 = [78,97,114,134,146,167]
7 plt.plot(x,y,'b',label='Team Bapao',linewidth=1)
8 plt.plot(x2,y2,'r',label='Ateam',linewidth=1)
9 plt.title('Dunia Youtube')
10 plt.ylabel('Jumlah Pendukung')
```

```

11 plt.xlabel('Tahun')
12 plt.legend()
13 plt.grid(True, color='k')
14 plt.show()

```

**Listing 6.10** Kode program membuat diagram menggunakan Matplotlib.

## Hasil Compile

5. Jelaskan apa fungsi dari subplot di matplotlib, dan bagaimana cara kerja dari fungsi subplot, sertakan ilustrasi dan gambar sendiri dan apa parameternya jika ingin menggambar plot dengan 9 subplot di dalamnya!

Fungsi subplot adalah untuk membuat beberapa plot di dalam satu gambar.

Cara kerja subplot, yaitu fungsi subplot memiliki parameter pertama adalah jumlah kolom, parameter kedua adalah jumlah baris, dan parameter ketiga adalah index plot keberapanya.

## Kode Program

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 t = np.arange(0.0, 9.0, 1)
5 s = [1,2,3,4,5,6,7,8,9]
6
7 for i in range(1, 10):
8     plt.subplot(3,3,i)
9     plt.xticks([], plt.yticks([]))
10    plt.title('subplot(1,2,'+str(i)+'')
11    plt.plot(t,s,'-y')
12
13 plt.show()

```

**Listing 6.11** Kode program membuat subplot menggunakan Matplotlib.

## Hasil Compile

6. Sebutkan semua parameter color yang bisa digunakan (contoh: m,c,r,k,... dkk)!

- 'b' (blue)
- 'g' (green)
- 'r' (red)
- 'c' (cyan)
- 'm' (magenta)
- 'y' (yellow)
- 'k' (black)
- 'w' (white)

7. Jelaskan bagaimana cara kerja dari fungsi hist, sertakan ilustrasi dan gambar sendiri!

Cara kerja dari fungsi hist yaitu fungsi hist akan menerima parameter yang diberikan, kemudian fungsi hist akan dieksekusi sesuai dengan parameter yang diberikan.

### Kode Program

```
1 import matplotlib.pyplot as plt
2 orang =
   [11,22,16,9,10,15,22,55,62,45,21,22,34,42,42,4,2,102,95,85,55,110,120,
3 umur = [0,10,20,30,40,50,60,70,80,90,100]
4 plt.hist(orang, umur, histtype='bar', rwidth=0.8, color='r')
5 plt.xlabel('Umur')
6 plt.ylabel('Jumlah Orang')
7 plt.title('Histogram')
8 plt.show()
```

**Listing 6.12** Kode program membuat diagram menggunakan Matplotlib.

### Hasil Compile

8. Jelaskan lebih mendalam tentang parameter dari fungsi pie diantaranya labels, colors, startangle, shadow, explode, autopct!

- labels : untuk memberikan label di tiap persentase.
- colors : untuk memberikan warna di tiap persentase.
- startangle : untuk memutar plot sesuai dengan derajat yang ditentukan.
- shadow : untuk memberikan bayangan pada plot.
- explode : untuk memisahkan antar tiap potongan pie pada plot.
- autopct : untuk menentukan jumlah angka dibelakang koma.

## 6.1.2 Keterampilan Pemogramman

1. Buatlah librari fungsi (file terpisah/library dengan nama NPMbar.py) untuk plot dengan jumlah subplot adalah  $NPM \bmod 3 + 2$ !

### Kode Program

```
1 from matplotlib import pyplot as plt
2
3 def bar():
4
5     hasil = 1174057 % 3 + 2
6
```

```

7     for i in range(1, hasil+1):
8         plt.subplot(2,2,i)
9         plt.bar
10        ([2012.7,2013.7,2014.7,2015.7,2016.7,2017.7],[9000,9500,10000,15000,20000,25000])
11
12        label="KFC",color='b',width=.3)
13        plt.bar
14        ([2013,2014,2015,2016,2017,2018],[20000,25000,30000,35000,40000,45000])
15
16        label="MCD",color='r',width=.3)
17        plt.bar
18        ([2013.3,2014.3,2015.3,2016.3,2017.3,2018.3],[2000,2500,3000,3500,4000,4500])
19
20        label="PHD",color='y',width=.3)
21        plt.legend()
22        plt.xlabel('Tahun')
23        plt.ylabel('Jumlah Pengunjung')
24        plt.title('Pengunjung restoran junkfood tahun ke Tahun')
25        plt.subplots_adjust(wspace=1, hspace=.7)
26
27    plt.show()

```

**Listing 6.13** Kode program membuat fungsi Bar Plot menggunakan Matplotlib.

## Hasil Compile

2. Buatlah librari fungsi (file terpisah/library dengan nama NPMscatter.py) untuk plot dengan jumlah subplot NPM mod 3 + 2!

## Kode Program

```

1 from matplotlib import pyplot as plt
2
3 def scatter():
4
5     hasil = 1174006 % 3 + 2
6
7     x = [1,1.5,2,2.5,3,3.5,3.6]
8     y = [7.5,8,8.5,9,9.5,10,10.5]
9
10    x1=[8,8.5,9,9.5,10,10.5,11]
11    y1=[3,3.5,3.7,4,4.5,5,5.2]
12
13    for i in range(1, hasil+1):
14        plt.subplot(2,2,i)
15        plt.scatter(x,y, label='keuntungan tinggi simpanan rendah',color='g')
16        plt.scatter(x1,y1, label='keuntungan rendah simpanan tinggi',color='b')
17        plt.xlabel('simpanan dalam ratusan')
18        plt.ylabel('pendapatan dalam ribuan')
19        plt.title('Scatter Plot')
20        plt.legend()
21        plt.subplots_adjust(wspace=1.1, hspace=.7)

```

```

22 plt.show()
23

```

**Listing 6.14** Kode program membuat fungsi Scatter Plot menggunakan Matplotlib.

### Hasil Compile

3. Buatlah librari fungsi (file terpisah/library dengan nama NPMpie.py) untuk plot dengan jumlah subplot NPM mod 3 + 2!

### Kode Program

```

1 from matplotlib import pyplot as plt
2
3 def pie():
4
5     hasil = 1174006 % 3 + 2
6
7     potong = [7,4,2,12]
8     kegiatan = ['Tidur', 'Makan', 'Kuliah', 'Youtube-an']
9     kolom = ['c', 'm', 'y', 'g']
10
11     for i in range(1, hasil+1):
12         plt.subplot(2,2,i)
13         plt.pie(potong,
14               labels=kegiatan,
15               colors=kolom,
16               startangle=90,
17               shadow=True,
18               explode=(0,0,0.2,0),
19               autopct='%1.1f%%')
20         plt.title('Kegiatan Sehari-hari')
21         plt.subplots_adjust(hspace=.4)
22
23     plt.show()

```

**Listing 6.15** Kode program membuat fungsi Pie Plot menggunakan Matplotlib.

### Hasil Compile

4. Buatlah librari fungsi (file terpisah/library dengan nama NPMplot.py) untuk plot dengan jumlah subplot NPM mod 3 + 2

### Kode Program

```

1 from matplotlib import pyplot as plt
2
3 def plot():
4
5     hasil = 1174006 % 3 + 2
6
7     x = [2014,2015,2016,2017,2018,2019]

```



```

8   y = [76,87,105,122,148,170]
9   x2 = [2014,2015,2016,2017,2018,2019]
10  y2 = [78,97,114,134,146,167]
11
12  for i in range(1, hasil+1):
13      plt.subplot(2,2,i)
14      plt.plot(x,y,'b',label='Team Bapao', linewidth=1)
15      plt.plot(x2,y2,'r',label='Team Iron Man',linewidth=1)
16      plt.title('Dunia Youtube')
17      plt.ylabel('Jumlah Penonton')
18      plt.xlabel('Tahun')
19      plt.legend()
20      plt.grid(True,color='k')
21      plt.subplots_adjust(wspace=.4, hspace=.7)
22
23  plt.show()

```

**Listing 6.16** Kode program membuat fungsi Plot menggunakan Matplotlib.

## Hasil Compile

### 6.1.3 Keterampilan Penanganan Error

Tuliskan peringatan error yang didapat dari mengerjakan praktek keenam ini, dan jelaskan cara penanganan error tersebut. dan Buatlah satu fungsi yang menggunakan try except untuk menanggulangi error tersebut.

Peringatan error di praktek kelima ini, yaitu:

- Syntax Errors Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- Name Error NameError adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Solusinya adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.
- Type Error TypeError adalah exception yang akan terjadi apabila pada saat dilakukannya eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusi dari error ini adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

Fungsi yang menggunakan try except untuk menanggulangi error.

## Kode Program

```

1  from matplotlib import pyplot as plt
2
3  def tryExceptError():

```

```

4     try :
5         a=[1,2,3]
6         y=[5,2,4]
7         plt.plot(x,y)
8         plt.show()
9     except SyntaxError:
10        print("Kesalahan penulisan syntax")
11    except NameError:
12        print("Variable tersebut tidak ada")
13    except TypeError:
14        print("Tipe data salah")
15    except:
16        print("Terjadi sebuah kesalahan")
17
18 tryExceptError()

```

**Listing 6.17** Kode program membuat fungsi penanganan error.

## Hasil Compile

### 6.1.4 Screenshoot Plagiat

### 6.1.5 Screenshoot Kode Program

## 6.2 Fathi Rabbani / 1164074

### 6.2.1 Teori

#### 1. Matplotlib

Library yang ada pada pemrograman Python yang berguna untuk membuat data grafik yang berdimensi 2.

#### 2. Membuat Sumbu X dan Y

membuat sumbu X dan Y pada penggunaan Library Matplotlib adalah dengan menjelaskan setiap detail data array yang dimiliki sebagai contohnya ada pada Code Berikut :

```

1 import matplotlib.pyplot as plt
2
3 x = (2, 4, 8, 10, 12)
4 y = (4, 8, 12, 16, 20)
5 plt.plot(x, y)
6 plt.show()

```

#### 3. Penggunaan Jenis Plot di Matplotlib

ada berbagai macam Jenis PLOT yang ada pada Matplotlib, diantaranya adalah :

```

1 # Garis
2 import matplotlib.pyplot as plt
3

```

```

4 x = (2, 4, 8, 10, 12)
5 y = (4, 8, 12, 16, 20)
6 plt.plot(x, y)
7 plt.show()

```

**Listing 6.18** Jenis Garis

```

1 # Titik
2 plt.scatter(x,y)
3 plt.show()

```

**Listing 6.19** Jenis Titik

```

1 # Batang
2 b=[2,4,6,3,50,100,5,7,81,42,99,80]
3 num_bins = 3
4 n, bins, patches = plt.hist(b, num_bins, facecolor = 'blue')
5 plt.show()

```

**Listing 6.20** Jenis Batang

```

1 # Pie
2 labels = 'Travel', 'Main Game', 'Makan', 'Membaca'
3 sizes = [5, 35, 30, 30]
4 explode = (0, 0.1, 0, 0)
5
6 fig1, ax1 = plt.subplots()
7 ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.0f%%',
8         shadow=True, startangle=90)
9 ax1.axis('equal')
10
11 plt.show()

```

**Listing 6.21** Jenis Pie

#### 4. Menggunakan Legend dan Label

Legend pada matplotlib digunakan untuk menunjukkan penggunaan grafik yang ditampilkan. contohnya dapat dilihat pada code berikut :

```

1 # Legend
2 plt.plot([1, 2, 3])
3 plt.legend(['garis simple'])

```

**Listing 6.22** Code penggunaan Legend pada Matplotlib

Label pada matplotlib digunakan untuk menambahkan data Text pada grafik agar mudah untuk dibaca.

```

1 # Label
2 plt.plot([1, 2, 3])
3 plt.ylabel('nilai Y')
4 plt.xlabel('nilai X')

```

**Listing 6.23** Code penggunaan Label pada Matplotlib

## 5. Fungsi subplot di Matplotlib

Fungsi yang digunakan untuk menambahkan beberapa diagram sekaligus dalam satu sintaks. contoh code dan penggunaannya dapat dilihat pada Code dan Gambar 6.20

```
1 # Subplot
2 plt.subplot(3, 3, 1)
3 plt.plot(x, y)
4
5 plt.subplot(3, 3, 2)
6 plt.plot(x, y)
7
8 plt.subplot(3, 3, 3)
9 plt.plot(x, y)
10
11 plt.subplot(3, 3, 4)
12 plt.plot(x, y)
13
14 plt.subplot(3, 3, 5)
15 plt.plot(x, y)
16
17 plt.subplot(3, 3, 6)
18 plt.plot(x, y)
19
20 plt.subplot(3, 3, 7)
21 plt.plot(x, y)
22
23 plt.subplot(3, 3, 8)
24 plt.plot(x, y)
25
26 plt.subplot(3, 3, 9)
27 plt.plot(x, y)
```

**Listing 6.24** Code penggunaan Subplot pada Matplotlib

## 6. Parameter Color

Penggunaan warna memiliki peran penting dalam membuat tampilan Grafik lebih menarik dan tidak membosankan, berikut ini adalah daftar penggunaan warna pada Matplotlib :

- b : Untuk memberikan warna biru
- g : Untuk memberikan warna hijau
- r : Untuk memberikan warna merah
- c : Untuk memberikan warna biru muda
- m : Untuk memberikan warna pink
- y : Untuk memberikan warna kuning
- k : Untuk memberikan warna hitam
- w : Untuk memberikan warna putih

## 7. Cara kerja Fungsi hist

Fungsi Hist merupakan fungsi yang berguna untuk membuat data grafik Batang yang memiliki nilai banyak (Array). contoh Code dan penjelasannya dapat dilihat pada Gambar 6.21

```

1 # Hist
2 x = [27, 28, 29, 30, 90, 91, 92, 93, 150, 151, 152, 153]
3 nK = 5
4 plt.hist(x, nK, facecolor='blue', alpha=0.5)
5 plt.show()

```

**Listing 6.25** Code penggunaan Subplot pada Matplotlib

## 8. Keterangan Lebih tentang Parameter pada Fungsi Pie

### ▪ Labels

Isi dengan tipe data list dan tidak wajib untuk digunakan. Fungsi parameter labels untuk memberi label pada setiap pecahan data yang ada pada grafik pie yang ditampilkan.

### ▪ Colors

Tipe data array atau sejenis dan tidak wajib untuk digunakan. Fungsi parameter colors untuk mengganti warna pada setiap pecahan yang ada. Jika tidak digunakan atau ditentukan, maka warna yang akan dipakai adalah warna yang aktif atau standar.

### ▪ Startangle

Tipe data pecahan atau float, tidak wajib untuk digunakan. Fungsi parameter startangle adalah fungsi untuk memutar grafik agar berubah posisi dengan acuan yaitu angle awalan dari grafik pie.

### ▪ Shadow

Bertipe data boolean dan tidak wajib digunakan. Fungsi parameter shadow digunakan untuk membuat bayangan pada bawah grafik pie yang ditampilkan.

### ▪ Explode

Bertipe data array atau sejenis dan tidak wajib digunakan. Fungsi parameter explode adalah menentukan radius untuk mengimbangi setiap pecahan pada grafik pie. Jika radius lebih dari 0 maka pecahan akan mulai menjauh dari pusat dan terlihat seperti keluar dari grafik lingkaran tersebut.

### ▪ Autopct

Bertipe data string atau fungsi dan tidak wajib digunakan. Fungsi parameter autopct adalah memberi label pada irisan dengan labelnya berupa fungsi atau string.

## 9. Check Plagiarisme

## 6.2.2 Pemrograman

### 1. Bar atau Grafik Batang

```

1 import matplotlib.pyplot as plt
2
3 def databar():
4
5     for i in range(1, 5):
6         plt.subplot(2, 2, i)
7         plt.bar(['DOTA 2'], ['0'], label = "MOBA",
8               color = 'r')
9         plt.bar(['PUBG'], ['199000'], label = "Battleroyal",
10              color = 'y')
11         plt.bar(['CSGO'], ['199000'], label = "FPP",
12              color = 'b')
13         plt.bar(['RDR2'], ['780000'], label = "Adventure",
14              color = 'g')
15         plt.legend()
16         plt.xlabel('Game')
17         plt.ylabel('Harga')
18         plt.title('Harga Game tahun 2018')
19     plt.show()
20 databar()

```

hasilnya dapat dilihat pada gambar 6.23

### 2. Scatter atau Grafik Titik

```

1 import matplotlib.pyplot as plt
2
3 def datascatter():
4
5     x = [1, 3, 5, 7, 9]
6     y = [11, 13, 15, 17, 19]
7     X = [1, 3, 5, 7, 9]
8     Y = [12, 14, 16, 18, 20]
9
10    for b in range(1, 5):
11        plt.subplot(2, 2, b)
12        plt.scatter(x, y, label = 'Data 1')
13        plt.scatter(X, Y, label = 'Data 2')
14        plt.xlabel('sumbu X')
15        plt.ylabel('sumbu Y')
16        plt.legend()
17    plt.show()
18 datascatter()

```

hasilnya dapat dilihat pada gambar 6.24

### 3. Pie atau Grafik Pie

```

1 import matplotlib.pyplot as plt
2
3 def datapie():
4
5     x = [25, 15, 25, 35]

```

```

6     y = ['PUBG', 'CSGO', 'DOTA 2', 'APEX LEGEND']
7     c = ['y', 'b', 'r', 'g']
8     for i in range(1, 5):
9         plt.subplot(2,2,i)
10        plt.pie(x,
11              labels = y,
12              colors = c,
13              startangle=90,
14              shadow= True,
15              explode=(0, 0, 0, 0.2),
16              autopct='%1.0f%%')
17        plt.subplots_adjust(hspace=.4)
18    plt.show()
19    datapie()

```

hasilnya dapat dilihat pada gambar 6.25

#### 4. Plot atau Grafik Plot

```

1 import matplotlib.pyplot as plt
2
3 def dataplot():
4
5     x = [1, 3, 5, 7, 9]
6     y = [11, 13, 15, 17, 19]
7     X = [1, 3, 5, 7, 9]
8     Y = [12, 14, 16, 18, 20]
9
10    for a in range(1, 5):
11        plt.subplot(2, 2, a)
12        plt.plot(x, y, label = 'Data 1')
13        plt.plot(X, Y, label = 'Data 2')
14        plt.xlabel('sumbu X')
15        plt.ylabel('sumbu Y')
16        plt.legend()
17    plt.show()
18    dataplot()

```

hasilnya dapat dilihat pada gambar 6.26

### 6.2.3 Try Error

```

1 lastlinelastline
2 def tryExceptError():
3     a = 1, 2, 3
4     b = "1", "2", "3"
5     try:
6         plt.plot(x, b)
7         plt.show()
8     except IndexError:
9         print('check datanya')
10    print('variable error')

```

hasil dapat dilihat pada gambar 6.27

## 6.3 Yusniar Nur Syarif Sidiq / 1164089

### 6.3.1 Pemahaman Teori

#### 1. Apa itu fungsi library matplotlib ?

Library Matplotlib adalah library python dengan 2D yang dapat menghasilkan plot dengan kualitas yang cukup tinggi dalam berbagai format dan dapat digunakan di berbagai macam platform. Fungsi dari Library Matplotlib ini yaitu membuat grafik dalam berbagai bentuk seperti grafik diagram batang, berbentuk garis, lingkaran, histogram, dan lain sebagainya.

#### 2. Jelaskan langkah - langkah membuat sumbu X dan Y di matplotlib !

- Lakukan Import Library Matplotlib terlebih dahulu dengan source code seperti berikut,

```
1 # In[1]: Import matplotlib
2 from matplotlib import pyplot as plt
```

- Buatlah dua variabel yang dapat menampung nilai dengan source code seperti berikut,

```
1 x=[2,2,3]
2 y=[3,2,1]
```

- Panggil variabel tersebut dengan fungsi plot seperti source code berikut ini,

```
1 plt.plot(x,y)
```

- Untuk menampilkan hasilnya gunakan show seperti source code berikut ini,

```
1 plt.show()
```

- Sehingga hasilnya dapat kita lihat di figure 6.28

#### 3. Jelaskan bagaimana perbedaan fungsi dan cara pakai untuk berbagai jenis (bar, histogram, scatter, dll) jenis plot di matplotlib !

- Diagram Batang atau Bar Graphic dimana fungsi ini akan melakukan plot dan menampilkannya dengan bentuk diagram batang, untuk membuat source codenya perhatikan dibawah dan hasilnya akan ditampilkan pada figure 6.28.

```
1 from matplotlib import pyplot as plt
2
3 plt.bar
   ([2013.7,2014.7,2015.7,2016.7,2017.7,2018.7],[36000,40000,35000,37000,42000,45000])
4 label="Cengek",color='y',width=.3)
5 plt.bar
   ([2013,2014,2015,2016,2017,2018],[42000,38000,39500,36000,38500,40000])
```



```

6 label="Cabe Merah",color='r',width=.3)
7 plt.bar
  ([2013.3,2014.3,2015.3,2016.3,2017.3,2018.3],[42000,40000,41000,41500,41000,41500])
8 label="Cabe Hijau",color='g',width=.3)
9 plt.legend()
10 plt.xlabel('Tahun')
11 plt.ylabel('Harga /kg')
12 plt.title('Harga Cabe Dari Tahun Ke Tahun')
13 plt.show()

```

- Histogram, dimana pada fungsi ini akan melakukan plot penggabungan data yang telah dikelompokkan, untuk membuatnya perhatikan source code dibawah dan hasilnya akan ditampilkan pada figure 6.30.

```

1 import matplotlib.pyplot as plt
2
3 jumlah = [10,10,20,30,22,100,25,40,60,70,80,90,99,43,12,12,12]
4 angka = [0,10,20,30,40,50,60,70,80,90,100]
5 plt.hist(jumlah, angka, histtype='bar', rwidth=0.8)
6 plt.xlabel('Angka')
7 plt.ylabel('Jumlah Angka Yang Keluar')
8 plt.title('Grafik Histogram')
9 plt.show()

```

- Scatter Plot, dimana fungsi ini akan melakukan plot berbentuk titik-titik yang masing-masing memiliki nilai variabel, untuk membuatnya perhatikan source code dibawah dan hasilnya akan ditampilkan pada figure ??.

```

1 import matplotlib.pyplot as plt
2 x = [1,1.5,2,2.5,3,3.5,3.6]
3 y = [7.5,8,8.5,9,9.5,10,10.5]
4
5 x1=[8,8.5,9,9.5,10,10.5,11]
6 y1=[3,3.5,3.7,4,4.5,5,5.2]
7
8 plt.scatter(x,y, label='Tabungan 1',color='y')
9 plt.scatter(x1,y1,label='Tabungan 2',color='c')
10 plt.xlabel('Tabungan 2')
11 plt.ylabel('Tabungan 1')
12 plt.title('Grafik Scatter Plot')
13 plt.legend()
14 plt.show()

```

- Area Plot, dimana fungsi ini akan melakukan pelacakan terhadap perubahan antar dua kelompok atau lebih yang terkait satu kategori secara keseluruhan, untuk membuatnya perhatikan source code dibawah dan hasilnya akan ditampilkan pada figure 6.32.

```

1 import matplotlib.pyplot as plt
2 Download_Game = [1,2,3,4,5]
3
4 Dota = [1,2,3,4,5]
5 ML = [5,4,3,2,1]
6 FF = [2,1,4,5,3]

```

```

7  PUBG = [3,5,4,1,2]
8
9  plt.plot([],[],color='r', label='Dota', linewidth=5)
10 plt.plot([],[],color='b', label='Mobile Legend', linewidth=5)
11 plt.plot([],[],color='g', label='Free Fire', linewidth=5)
12 plt.plot([],[],color='y', label='PUBG', linewidth=5)
13
14 plt.stackplot(Download_Game,Dota,ML,FF,PUBG, colors=['r','b','g','y'])
15
16 plt.xlabel('Sumbu x')
17 plt.ylabel('Sumbu y')
18 plt.title('Area Plot')
19 plt.legend()
20 plt.show()

```

- Pie Plot, dimana pada fungsi ini akan digunakan dalam menunjukkan presentase yang mewakili setiap kategori, untuk membuatnya perhatikan source code dibawah dan hasilnya akan ditampilkan pada figure 6.33.

```

1  import matplotlib.pyplot as plt
2
3  Download_Game = [1,2,3,4,5]
4
5  Dota =[1,2,3,4,5]
6  ML = [5,4,3,2,1]
7  FF =[2,1,4,5,3]
8  PUBG = [3,5,4,1,2]
9  potong = [3,1,1,12]
10 Game = ['Dota','Mobile Legend','Free Fire','PUBG']
11 kolom = ['r','b','g','y']
12
13 plt.pie(potong,
14       labels=Game,
15       colors=kolom,
16       startangle=90,
17       shadow=True,
18       explode=(0,0,0.2,0),
19       autopct='%1.1f%%')
20
21 plt.title('Pie Plot')
22 plt.show()

```

- Line Graphic, dimana fungsi ini akan melakukan plot dengan bentuk line, untuk membuatnya perhatikan source code dibawah dan hasilnya akan ditampilkan pada figure 6.34.

```

1  from matplotlib import pyplot as plt
2
3  y = [1,2,3,4,5,6]
4  x = [1,2,3,4,5,6]
5  plt.plot(x,y)
6  plt.title('Line')
7  plt.ylabel('Sumbu Y')
8  plt.xlabel('Sumbu X')
9  plt.show()

```

4. Jelaskan bagaimana cara menggunakan legend dan label serta kaitannya fungsi tersebut !

- Pertama kita akan membuat 4 variabel terlebih dahulu, perhatikan source code berikut,

```
1 from matplotlib import pyplot as plt
2
3 x = [1,2,3,4,5,6]
4 y = [11,12,13,14,15,16]
5 x2 = [1,2,3,4,5,6]
6 y2 = [17,18,19,20,21,22]
```

- Dalam membuat fungsi legend kita akan mendefinisikan parameter label pada fungsi plot, dimana parameter label ini akan digunakan untuk memberikan keterangan pada tiap line, perhatikan source code dibawah ini,

```
1 plt.plot(x,y, 'b', label='Label 1', linewidth=1)
2 plt.plot(x2,y2, 'r', label='Label 2', linewidth=1)
```

- Selanjutnya kita akan memberikan title dan nama pada setiap sumbu, gunakan source code seperti berikut,

```
1 plt.title('Penggunaan Legend Dan Label')
2 plt.ylabel('Sumbu Y')
3 plt.xlabel('Sumbu X')
```

- Panggil fungsi legend dan tampilkan di console gunakan source code berikut,

```
1 plt.legend()
2 plt.grid(True, color='g')
3 plt.show()
```

- Hasilnya dapat kita lihat pada figure 6.35

5. Jelaskan apa fungsi dari subplot di matplotlib, dan bagaimana cara kerja dari fungsi subplot, sertakan ilustrasi dan gambar sendiri dan apa parameternya jika ingin menggambarkan plot dengan 9 subplot di dalamnya !

Fungsi Subplot digunakan untuk membuat plot di dalam satu gambar, dimana fungsi subplot memiliki parameter pertama akan menentukan kolomnya, dan pada parameter kedua akan menentukan jumlah barisnya, dan parameter ketiga akan menentukan index plotnya, untuk membuatnta perhatikan source code dibawah dan hasilnya akan ditampilkan pada figure 6.36.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 t = np.arange(0.0, 9.0, 1)
5 s = [2,4,6,8,12,14,16,18,20]
6
7 for i in range(1, 5):
8     plt.subplot(3,3,i)
9     plt.xticks([]), plt.yticks([])
```

```

10 plt.title('subplot(1, '+str(i)+' )')
11 plt.plot(t,s,'-y')
12
13 plt.show()

```

6. Sebutkan semua parameter color yang bisa digunakan (contoh: m,c,r,k,...dkk) !

Ada beberapa parameter warna yang dapat kita gunakan pada library matplotlib ini yaitu :

- r menunjukkan arti red
- b menunjukkan arti blue
- g menunjukkan arti green
- c menunjukkan arti cyan
- y menunjukkan arti yellow
- m menunjukkan arti magenta
- k menunjukkan arti black
- w menunjukkan arti white

7. Jelaskan bagaimana cara kerja dari fungsi hist, sertakan ilustrasi dan gambar sendiri !

Cara kerja dari fungsi hist atau biasa kita sebut dengan histogram, dimana akan melakukan eksekusi terhadap data yang telah dikelompokkan dan akan ditampilkan perhitungan jumlah data yang keluar, misalnya saya akan memprediksi jumlah angka yang akan keluar ada berapa, maka dibutuhkan lah source code seperti berikut,

```

1 import matplotlib.pyplot as plt
2
3 jumlah = [10,10,20,30,22,100,25,40,60,70,80,90,99,43,12,12,12]
4 angka = [0,10,20,30,40,50,60,70,80,90,100]
5 plt.hist(jumlah, angka, histtype='bar', rwidth=0.8)
6 plt.xlabel('Angka')
7 plt.ylabel('Jumlah Angka Yang Keluar')
8 plt.title('Grafik Histogram')
9 plt.show()

```

Hasil dari source code tersebut akan diperlihatkan pada figure 6.37.

8. Jelaskan lebih mendalam tentang parameter dari fungsi pie diantaranya labels, colors, startangle, shadow, explode, autopct !

- Labels digunakan untuk memberikan keterangan pada tiap presentase
- Colors digunakan untuk memberikan warna pada tiap presentase agar terlihat berbeda-beda
- Startangle digunakan untuk memutar plot dengan derajat yang telah ditentukan

- Shadow digunakan untuk memberikan efek bayangan pada plot
- Explode digunakan untuk memisahkan tiap potongan pie pada plot
- Autopct digunakan untuk menentukan jumlah angka dibelakang koma

### 6.3.2 Keterampilan Pemrograman

1. Buatlah Library Fungsi dengan nama NPM\_bar.py untuk plot dengan jumlah subplot adalah NPM mod 3 + 2 !

```

1 from matplotlib import pyplot as plt
2
3 def bar():
4
5     for i in range(1, 5):
6         plt.subplot(2,2,i)
7         plt.bar
8         ([2013.7,2014.7,2015.7,2016.7,2017.7,2018.7],[36000,40000,35000,37000,4
9
10        label="Cengek", color='y', width=.3)
11        plt.bar
12        ([2013,2014,2015,2016,2017,2018],[42000,38000,39500,36000,38500,40000]
13
14        label="Cabe Merah", color='r', width=.3)
15        plt.bar
16        ([2013.3,2014.3,2015.3,2016.3,2017.3,2018.3],[42000,40000,41000,41500,4
17
18        label="Cabe Hijau", color='g', width=.3)
19        plt.legend()
20        plt.xlabel('Tahun')
21        plt.ylabel('Harga /kg')
22        plt.title('Harga Cabe Dari Tahun Ke Tahun')
23        plt.subplots_adjust(wspace=1.1, hspace=.7)
24        plt.show()
25
26 bar()

```

Hasil dari source code tersebut akan menampilkan 4 biji bar, dikarenakan figure yang terlalu besar saya sceenshoot seperti pada figure 6.38.

2. Buatlah Library Fungsi dengan nama NPM\_scatter.py untuk plot dengan jumlah subplot NPM mod 3 + 2 !

```

1 from matplotlib import pyplot as plt
2
3 def scatter():
4
5     x = [1,1.1,1.2,1.3,1.4,1.5,1.6]
6     y = [5,5.1,5.2,5.3,5.4,5.5,5.6]
7
8     x1=[8,8.5,9,9.5,10,10.5,11]
9     y1=[3,3.5,3.7,4,4.5,5,5.2]
10
11     for i in range(1, 5):
12         plt.subplot(2,2,i)

```

```

13     plt.scatter(x,y, label='Contoh 1',color='y')
14     plt.scatter(x1,y1,label='Contoh 2',color='c')
15     plt.xlabel('Contoh 2')
16     plt.ylabel('Contoh 1')
17     plt.title('Grafik Scatter Plot')
18     plt.legend()
19     plt.subplots_adjust(wspace=1.1, hspace=.7)
20
21     plt.show()
22 scatter()

```

Pada source code tersebut dapat dilihat pada figure 6.39.

3. Buatlah Library Fungsi dengan nama NPM\_pie.py untuk plot dengan jumlah subplot NPM mod 3 + 2 !

```

1 from matplotlib import pyplot as plt
2
3 def pie():
4
5     potong = [4,4,4,4]
6     Game = ['Dota', 'Mobile Legend', 'Free Fire', 'PUBG']
7     kolom = ['r', 'b', 'g', 'y']
8
9     for i in range(1, 5):
10        plt.subplot(2,2,i)
11        plt.pie(potong,
12              labels=Game,
13              colors=kolom,
14              startangle=90,
15              shadow= True,
16              explode=(0,0,0,0.2),
17              autopct='%1.1f%%')
18        plt.title('Pie Plot')
19        plt.subplots_adjust(hspace=.4)
20
21    plt.show()
22 pie()

```

Pada source code tersebut dapat dilihat pada figure 6.40.

4. Buatlah Library Fungsi dengan nama NPM\_plot.py untuk plot dengan jumlah subplot NPM mod 3 + 2 !

```

1 from matplotlib import pyplot as plt
2
3 def plot():
4
5     x = [1,2,3,4,5,6]
6     y = [11,12,13,14,15,16]
7     x2 = [1,2,3,4,5,6]
8     y2 = [17,18,19,20,21,22]
9
10    for i in range(1, 5):
11        plt.subplot(2,2,i)
12        plt.plot(x,y, 'b', label='Label 1', linewidth=1)

```

```

13     plt.plot(x2,y2,'r',label='Label 2',linewidth=1)
14     plt.title('Fungsi Plot')
15     plt.ylabel('Sumbu Y')
16     plt.xlabel('Sumbu X')
17     plt.legend()
18     plt.grid(True,color='g')
19     plt.subplots_adjust(wspace=.4, hspace=.7)
20
21     plt.show()
22 plot()

```

Pada source code tersebut dapat dilihat pada figure 6.41.

### 6.3.3 Keterampilan Penanganan Error

1. Error yang saya dapatkan terlihat pada figure 6.42 yang merupakan NameError dikarenakan variabel yang saya deklarasikan tidak sesuai, cara membenarkannya kita ganti variabel tersebut menjadi x dan y, dan untuk membuat fungsi try exceptnya dapat perhatikan source code berikut ini,

```

1 # In[07] Fungsi TryExcept
2 def tryExceptError():
3     try:
4         b=[2,2,3]
5         a=[3,2,1]
6         plt.plot(x,y)
7         plt.show()
8     except NameError:
9         print("Tidak Ada Variabel Tersebut")
10
11 tryExceptError()

```

## 6.4 Kevin Natanae Nainggolan 1174059

### 6.4.1 TEORI

1. Apa itu fungsi library matplotlib  
Library plotting 2 dimensi Python yang menciptakan gambar publikasi bermutu di dalam berbagai macam format hardcopy
2. Jelaskan langkah-langkah membuat sumbu X dan Y di matplotlib  
Membuat sumbu x dan y pada matplotlib, kita bisa membuatnya menggunakan list untuk mempermudah penyimpanan nilai setiap sumbunya. Seperti kode di bawah

```

1 from matplotlib import pyplot as plt
2
3 plt.plot([1,2,7],[4,3,9])
4 plt.show()

```

3. Jelaskan bagaimana perbedaan fungsi dan cara pakai untuk berbagai jenis(bar,histogram,scatter) jenis plot di matplotlib

Untuk membedakan fungsi plot yang digunakan adalah dengan melihat bentuk grafik yang akan di tampilkan sesuai dengan perintah yang digunakan pada pemogramannya, dan untuk cara pengguna plot tersebut bisa dilihat sebagai berikut

```
1 from matplotlib import pyplot as plt
```

(a) bar

Perhatikan kode dalam membentuk diagram bar seperti berikut

```
1 plt.bar([1,3,5,7,9],[20,21,31,33,18],
2 label="Pisang (Sisir)",color='Y',width=.5)
3 plt.bar([2,4,6,8,10],[10,12,19,10,15],
4 label="Rambutan(Kg)", color='K',width=.5)
5 plt.legend()
6 plt.xlabel('Minggu')
7 plt.ylabel('Terjual (buah)')
8 plt.title('BAR')
9 plt.show()
```

(b) histogram

Dalam penggunaanya plot bar x dan y dapat diatur dengan angka koma

```
1 population_age =
   [40,12,11,22,16,9,10,15,22,55,62,45,5,22,4,32,35,55,33]
2 bins = [0,10,20,30,40,50,60,70]
3 plt.hist(population_age, bins, histtype='bar', rwidth=0.8)
4 plt.xlabel('Rentan Usia')
5 plt.ylabel('Jumlah Orang')
6 plt.title('Histogram')
7 plt.show()
```

(c) scatter

Diagram yang penampilannya dengan titik titik sebagai penandanya

```
1 x = [1,1.5,2,2.5,3,3.5,3.9]
2 y = [7,8.2,8.4,9.6,9.8,10,10.2]
3 x1=[8,8.5,9,9.7,10,10.5,10.7]
4 y1=[3,3.5,3.7,4,4.5,5,5.2]
5 plt.scatter(x,y, label='Lokasi A',color='B')
6 plt.scatter(x1,y1,label='Lokasi B',color='R')
7 plt.xlabel('Penyimpanan(ratus)')
8 plt.ylabel('Pendapatan(ribu)')
9 plt.title('SCATTER')
10 plt.legend()
11 plt.show()
```

(d) line

Perhatikan kode dalam membentuk diagram line seperti berikut

```
1 x=[1,1,7]
2 y=[4,5,9]
3 plt.plot(x,y)
4 plt.show
```



## (e) stack plot

Penggunaan stack plot ini seperti diagram line, dengan warna yang mengisi-nya, serta antar line itu bisa berdekatan. Berikut Contoh penggunaannya

```

1 ari = [1,2,3,4,5,6,7]
2 modom = [8,8,6,9,7,8,9]
3 mangan = [2,3,4,3,2,7,5]
4 karejo = [7,8,7,6,5,13,2]
5 marmeam = [7,6,7,8,4,7,15]
6 potong = [7,2,2,13]
7 kegiatan = ['modom', 'mangan', 'karejo', 'marmeam']
8 cols = ['c', 'm', 'r', 'b']
9
10 plt.plot([], [], color='m', label='Modom', linewidth=5)
11 plt.plot([], [], color='c', label='Mangan', linewidth=5)
12 plt.plot([], [], color='r', label='Karejo', linewidth=5)
13 plt.plot([], [], color='k', label='Marmeam', linewidth=5)
14 plt.stackplot(ari, modom, mangan, karejo, marmeam, colors=['m', 'c',
    'r', 'b'])
15
16 plt.xlabel('Hari')
17 plt.ylabel('Berapa Lama')
18 plt.title('Stack Plot')
19 plt.legend()
20 plt.show()
21 plt.pie(potong, labels=kegiatan, colors=cols, startangle=0, shadow
    = True, explode=(0,0,0.1,0), autopct='%1.1f%%')
22 plt.title('Pie Plot')
23 plt.show()

```

#### 4. Jelaskan bagaimana cara menggunakan legend dan label serta kaitannya dengan fungsi tersebut

Penggunaan legend untuk memudahkan dalam membaca grafik. Dalam penggunaan legend dan label perhatikan code berikut

```
1 plt.legend()
```

#### 5. Jelaskan apa fungsi dari subplot di matplotlib, dan bagaimana cara kerja dari fungsi subplot, sertakan ilustrasi dan gambar sendiri dan apa parameternya jika ingin menggambar plot dengan 9 subplot di dalamnya

fungsi dari subplot dari matplotlib untuk bisa membuat lebih dari 1 grafik dalam sebuah program. Untuk parameternya sendiri kami menggunakan t1 dan t2. Cara kerjanya sendiri bisa d cek sebagai berikut

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def f(t):
5     return np.exp(-t) * np.cos(2*np.pi*t)
6     t1 = np.arange(0.0, 5.0, 0.1)
7     t2 = np.arange(0.0, 5.0, 0.02)
8     plt.subplot(331)
9     plt.plot(t1, f(t1), 'bo', t2, f(t2))
10    plt.subplot(332)

```

```

11 plt.plot(t2, np.cos(2*np.pi*t2))
12 plt.subplot(333)
13 plt.plot(t1, f(t1), 'bo', t2, f(t2))
14 plt.subplot(334)
15 plt.plot(t2, np.cos(2*np.pi*t2))
16 plt.subplot(335)
17 plt.plot(t1, f(t1), 'bo', t2, f(t2))
18 plt.subplot(336)
19 plt.plot(t2, np.cos(2*np.pi*t2))
20 plt.subplot(337)
21 plt.plot(t1, f(t1), 'bo', t2, f(t2))
22 plt.subplot(338)
23 plt.plot(t1, f(t1), 'bo', t2, f(t2))
24 plt.subplot(339)
25 plt.plot(t2, np.cos(2*np.pi*t2))
26 plt.show()

```

6. Sebutkan semua parameter color yang bisa digunakan (contoh: m,c,r,k,... dkk)

- R untuk warna Red atau Merah
- G untuk warna Green atau Hijau
- B untuk warna Blue atau Biru
- C untuk warna Cyan atau Biru Muda
- M untuk warna Magenta atau Merah Tua
- Y untuk warna Yellow Atau Kuning
- K untuk warna blacK atau Hitam

7. Jelaskan bagaimana cara kerja dari fungsi hist, sertakan ilustrasi dan gambar sendiri

Fungsi hist digunakan untuk menjumlahkan beberapa data yang memenuhi kriteria pramater yang kita tentukan, seprti contoh kode dibawah

```

1 import matplotlib.pyplot as plt
2
3 x = [9, 13, 11, 12, 41, 21, 25, 35, 40, 45, 10, 50, 53, 29, 22]
4 num_bins = 5
5 n, bins, patches = plt.hist(x, num_bins, facecolor='red', alpha
6                               =0.5)
7 plt.show()

```

8. Jelaskan lebih mendalam tentang parameter dari fungsi pie diantaranya labels, colors, startangle, shadow, explode, autopct

- labels : Isi dengan tipe data list dan tidak wajib untuk digunakan. Fungsi parameter labels untuk memberi label pada setiap pecahan data yang ada pada grafik pie yang ditampilkan.
- colors : Tipe data array atau sejenis dan tidak wajib untuk digunakan. Fungsi parameter colors untuk mengganti warna pada setiap pecahan yang ada.

Jika tidak digunakan atau ditentukan, maka warna yang akan dipakai adalah warna yang aktif atau standar.

- `startangle` : Tipe data pecahan atau float, tidak wajib untuk digunakan. Fungsi parameter `startangle` adalah fungsi untuk memutar grafik agar berubah posisi dengan acuan yaitu angle awalan dari grafik pie.
- `shadow` : Bertipe data boolean dan tidak wajib digunakan. Fungsi parameter `shadow` digunakan untuk membuat bayangan pada bawah grafik pie yang ditampilkan.
- `explode` : Bertipe data array atau sejenis dan tidak wajib digunakan. Fungsi parameter `explode` adalah menentukan radius untuk mengimbangi setiap pecahan pada grafik pie. Jika radius lebih dari 0 maka pecahan akan mulai menjauh dari pusat dan terlihat seperti keluar dari grafik lingkaran tersebut.
- `autopct` : Bertipe data string atau fungsi dan tidak wajib digunakan. Fungsi parameter `autopct` adalah memberi label pada irisan dengan labelnya berupa fungsi atau string.

## 6.4.2 PRAKTEK

1. Buatlah librari fungsi (le terpisah/library dengan nama NPM `bar.py`) untuk plot dengan jumlah subplot adalah  $\text{NPM} \bmod 3 + 2$

```

1 def bar(npm):
2     import matplotlib.pyplot as plt
3     import random as r
4     i = (int(npm)%3)+2
5     rowcol = ""
6     if i%2==0:
7         row=i/2
8         rowcol=str(int(row))+ "2"
9     if i%3==0:
10        row=i/3
11        rowcol=str(int(row))+ "3"
12    a=1
13    while a<=i:
14        y = [r.randint(1,10),r.randint(1,10),r.randint(1,10),r.
15            randint(1,10)]
16        plt.subplot(int(rowcol+str(a)))
17        plt.bar([1,2,3,4], y)
18        a+=1
19    plt.show()
20    return i
21 bar("1174059")

```

2. Buatlah librari fungsi (le terpisah/library dengan nama NPM `scatter.py`) untuk plot dengan jumlah subplot  $\text{NPM} \bmod 3 + 2$

```

1 def pie(npm):
2     import matplotlib.pyplot as plt
3     import random as r
4     i = (int(npm)%3)+2

```

```

5 row = ""
6 col = ""
7 if i%2==0:
8     row=i/2
9     col="2"
10 if i%3==0:
11     row=i/3
12     col="3"
13 a=1
14 labels = 'T', 'B', 'N', 'K'
15 fig, axs = plt.subplots(int(row), int(col))
16 while a<=i:
17     c=0
18     fracs = []
19     wow=0
20     while c!=3:
21         num = r.randint(1,33)
22         wow = wow+num
23         fracs.append(num)
24         c+=1
25     fracs.append(100-wow)
26     plt.subplot(str(int(row)),col,a)
27     plt.pie(fracs,labels=labels,startangle=90,shadow= True,
28         explode=[0.1,0.1,0.1,0],autopct='%1.1f%%')
29     a+=1
30     plt.show()
31 return i
a = pie("1174059")

```

3. Buatlah librari fungsi (le terpisah/library dengan nama NPM pie.py) untuk plot dengan jumlah subplot NPM mod 3 + 2

```

1 def scatter(npm):
2     import matplotlib.pyplot as plt
3     import random as r
4     i = (int(npm)%3)+2
5     rowcol = ""
6     if i%2==0:
7         row=i/2
8         rowcol=str(int(row))+ "2"
9     if i%3==0:
10         row=i/3
11         rowcol=str(int(row))+ "3"
12     a=1
13     while a<=i:
14         plt.subplot(int(rowcol+str(a)))
15         y = [r.randint(1,10),r.randint(1,10),r.randint(1,10),r.
16             randint(1,10),r.randint(1,10),r.randint(1,10),r.randint(1,10)]
17         plt.plot([1,2,3,4,5,6,7], y, 'ro')
18         a+=1
19     plt.show()
20     return i
scatter("1174059")

```

4. Buatlah librari fungsi (le terpisah/library dengan nama NPM plot.py) untuk plot dengan jumlah subplot NPM mod 3 + 2

```

1 def plot(npm):
2     import matplotlib.pyplot as plt
3     import random as r
4     i = (int(npm)%3)+2
5     rowcol = ""
6     if i%2==0:
7         row=i/2
8         rowcol=str(int(row))+ "2"
9     if i%3==0:
10        row=i/3
11        rowcol=str(int(row))+ "3"
12    a=1
13    while a<=i:
14        plt.subplot(int(rowcol+str(a)))
15        y = [r.randint(1,10),r.randint(1,10),r.randint(1,10),r.
16            randint(1,10),r.randint(1,10)]
17        plt.plot([1,2,3,4,5], y)
18        a+=1
19    plt.show()
20    return i
21 plot("1174059")

```

### 6.4.3 PENANGANAN EROR

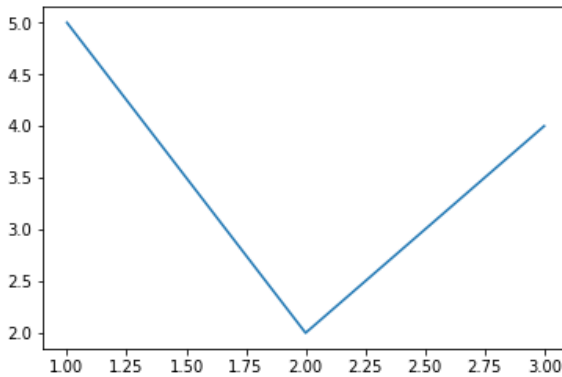
Tuliskan peringatan error yang didapat dari mengerjakan praktek ketiga ini, dan jelaskan cara penanganan error tersebut. dan Buatlah satu fungsi yang menggunakan gunakan try except untuk menanggulangi error tersebut

```

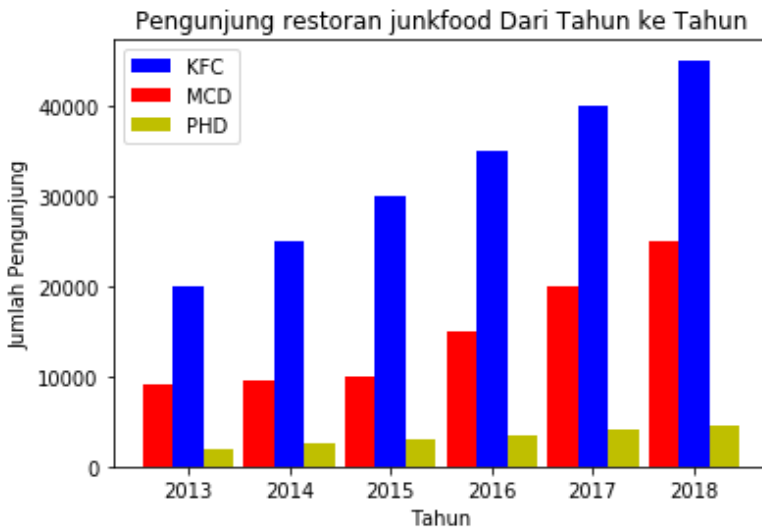
1 def fungsi(npm):
2     try:
3         import matplotlib.pyplot as plt
4         import random as r
5         i = (int(npm)%3)+2
6         rowcol = ""
7         if i%2==0:
8             row=i/2
9             rowcol=str(int(row))+ "2"
10        if i%3==0:
11            row=i/3
12            rowcol=str(int(row))+ "3"
13        a=1
14        while a<=i:
15            y = [r.randint(1,10),r.randint(1,10),r.randint(1,10),r.randint
16                (1,10)]
17            plt.subplot(int(rowcol+str(4)))
18            plt.bar([1,2,3,4], y)
19            a+=1
20        plt.show()
21    except ValueError:
22        print("Salah pada index subplot")
23    return i
24 fungsi("1174059")

```

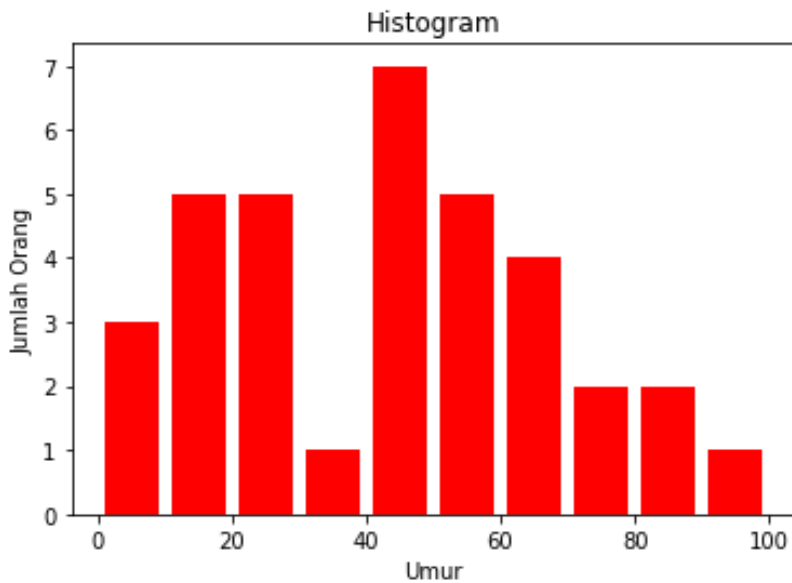
```
In [3]: runfile('D:/Data Alit/SEMESTER IV/Pemrogramman III/1174057/1174057.py',
wdir='D:/Data Alit/SEMESTER IV/Pemrogramman III/1174057')
Reloaded modules: 1174006_bar, 1174006_scatter, 1174006_pie, 1174006_plot
```



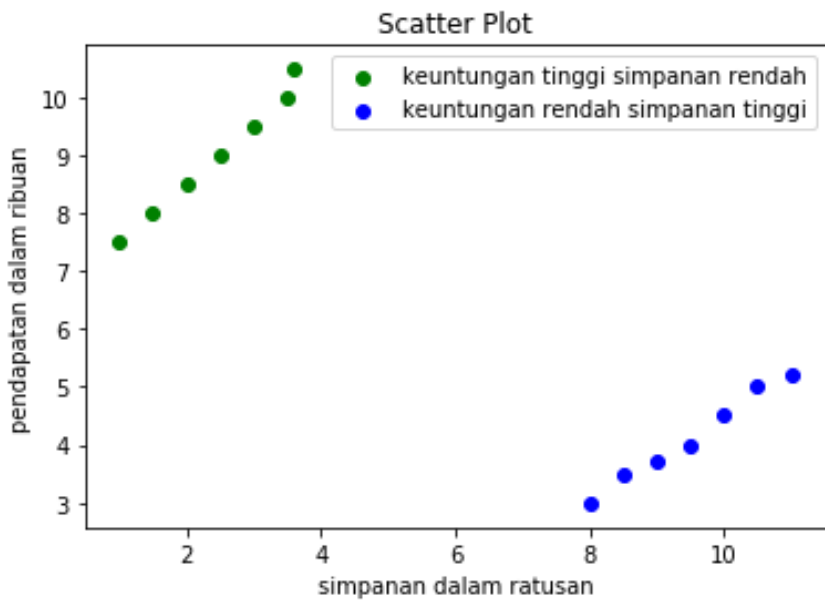
**Gambar 6.1** Hasil compile membuat diagram menggunakan Matplotlib.



**Gambar 6.2** Hasil compile membuat bar graph menggunakan Matplotlib.

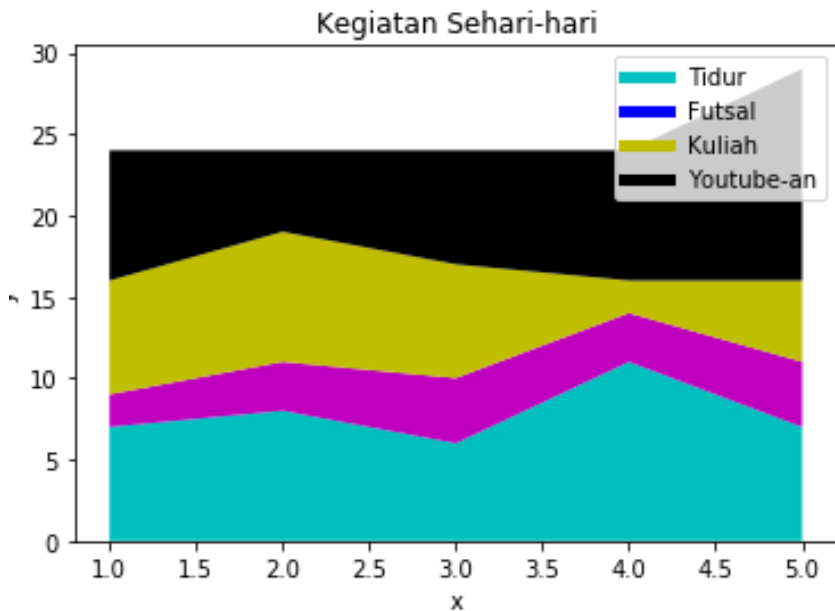


**Gambar 6.3** Hasil compile membuat histogram menggunakan Matplotlib.

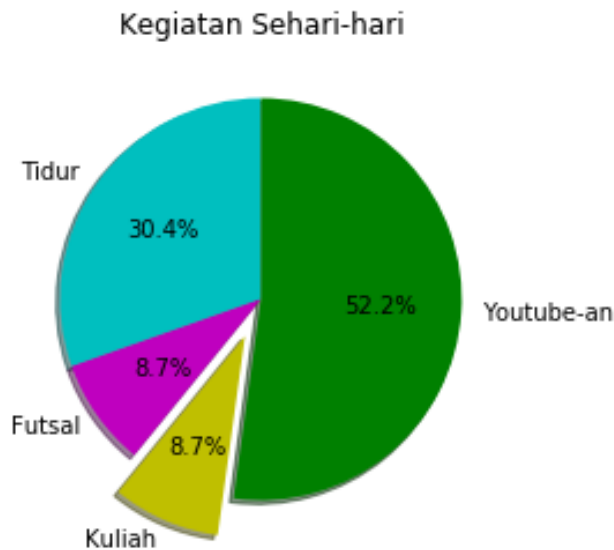


**Gambar 6.4** Hasil compile membuat scatter plot menggunakan Matplotlib.

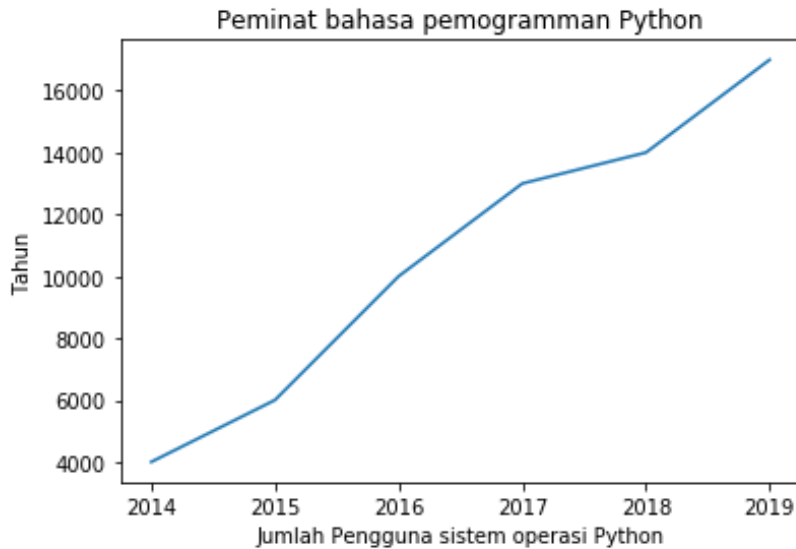




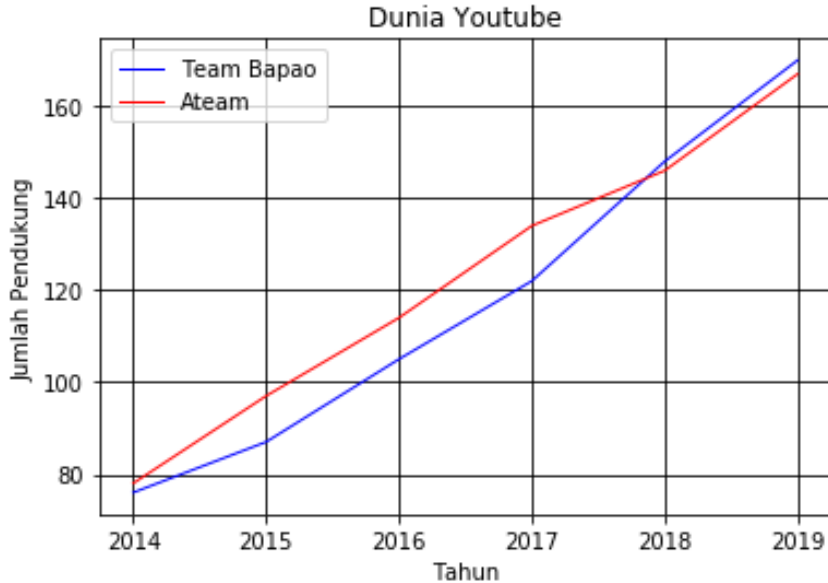
**Gambar 6.5** Hasil compile membuat diagram menggunakan Matplotlib.



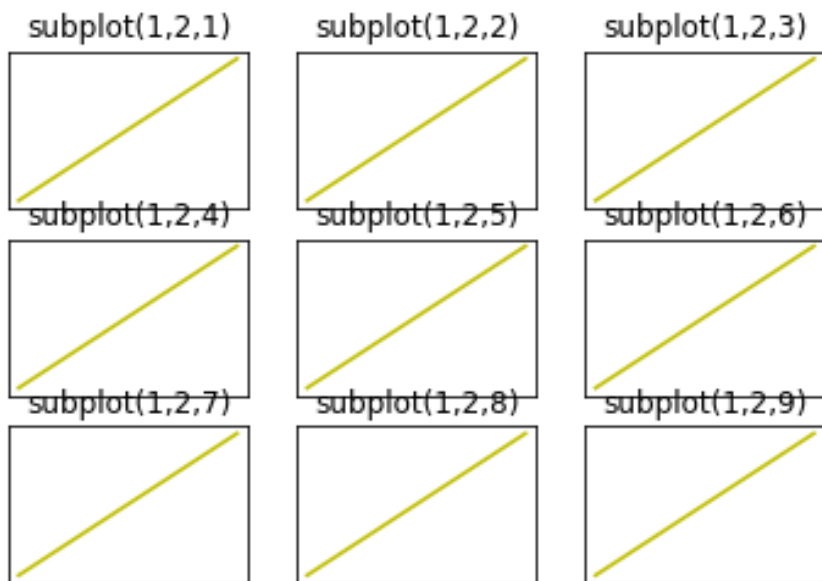
**Gambar 6.6** Hasil compile membuat Pie Plot menggunakan Matplotlib.



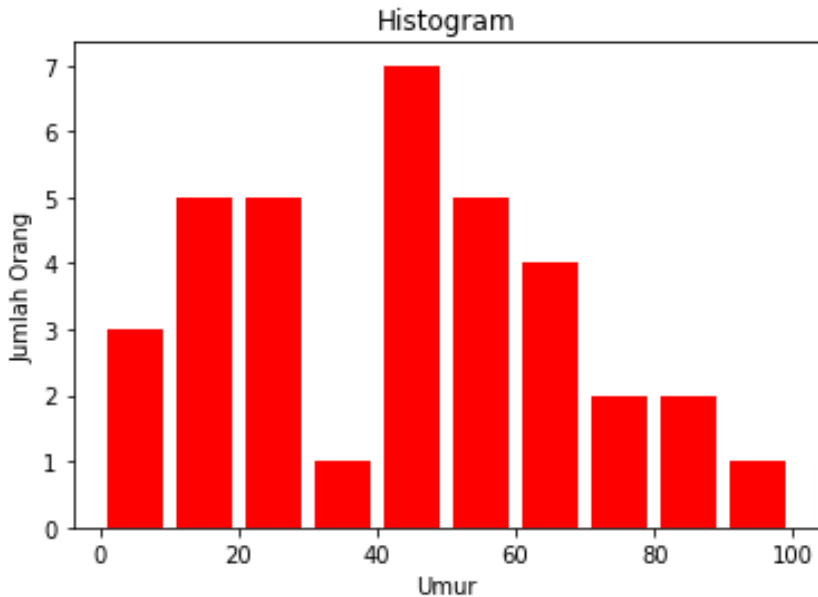
**Gambar 6.7** Hasil compile membuat diagram menggunakan Matplotlib.



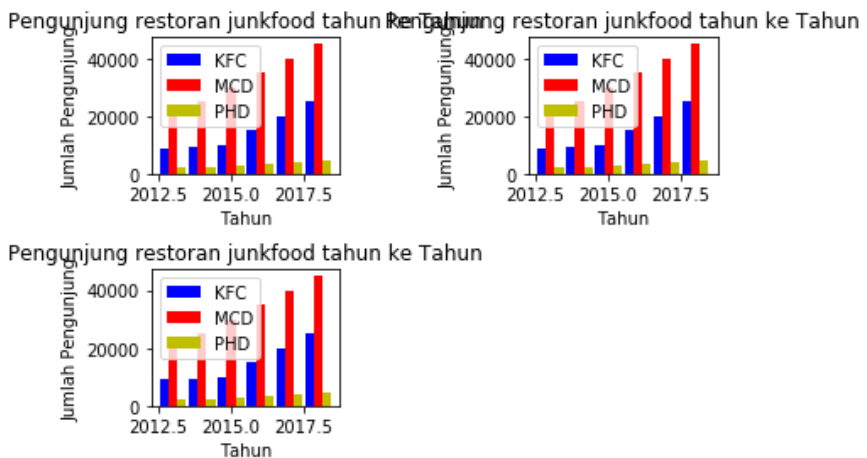
**Gambar 6.8** Hasil compile membuat diagram menggunakan Matplotlib.



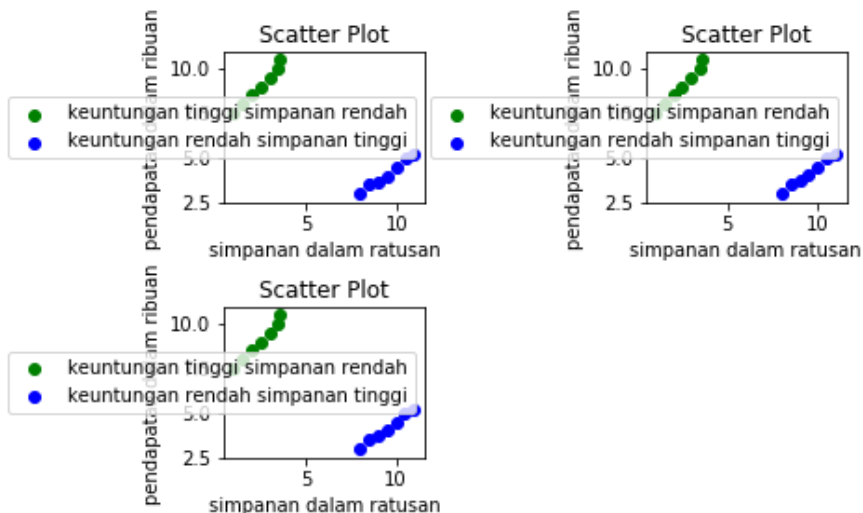
**Gambar 6.9** Hasil compile membuat subplot menggunakan Matplotlib.



**Gambar 6.10** Hasil compile membuat diagram menggunakan Matplotlib.

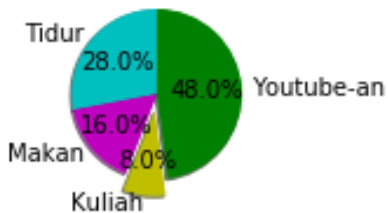


**Gambar 6.11** Hasil compile membuat fungsi Bar Plot menggunakan Matplotlib.

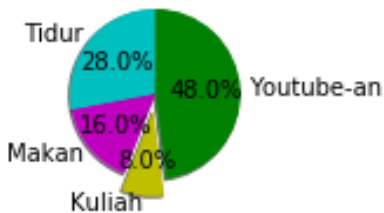


**Gambar 6.12** Hasil compile membuat fungsi Scatter Plot menggunakan Matplotlib.

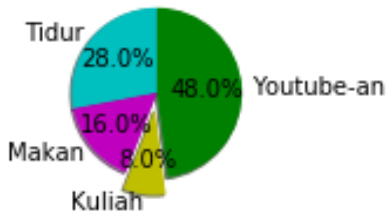
**Kegiatan Sehari-hari**



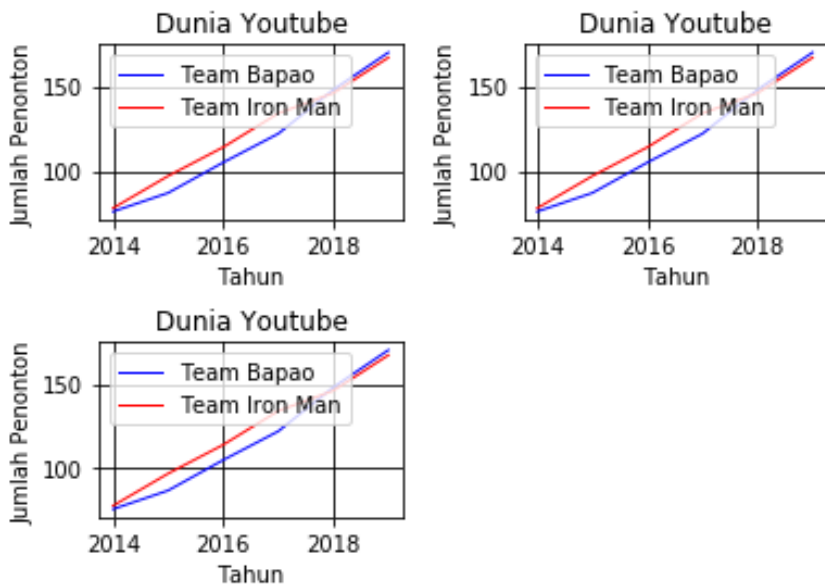
**Kegiatan Sehari-hari**



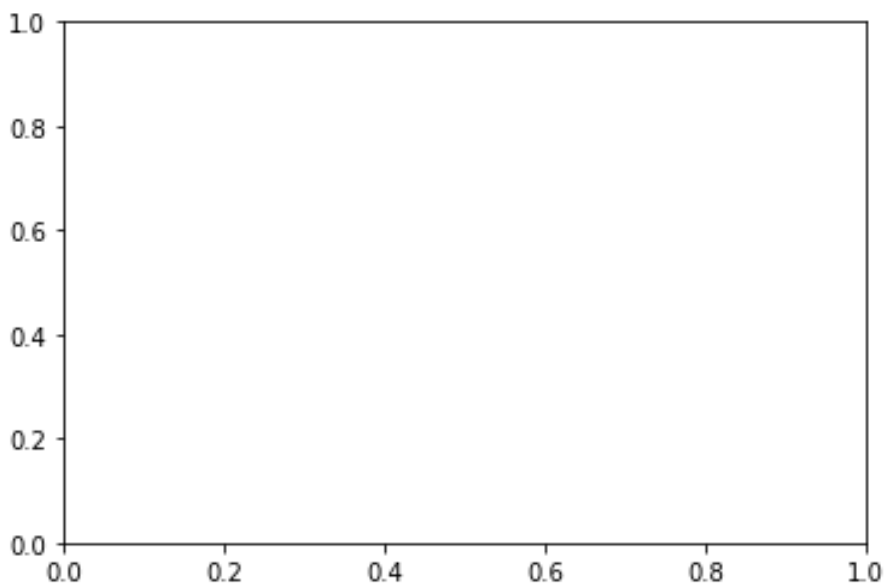
**Kegiatan Sehari-hari**



**Gambar 6.13** Hasil compile membuat fungsi Pie Plot menggunakan Matplotlib.



**Gambar 6.14** Hasil compile membuat fungsi Plot menggunakan Matplotlib.



**Gambar 6.15** hasil run penanganan error

### RESULTS

100%

Completed: 100% Checked

0%

Plagiarism

100%

Unique

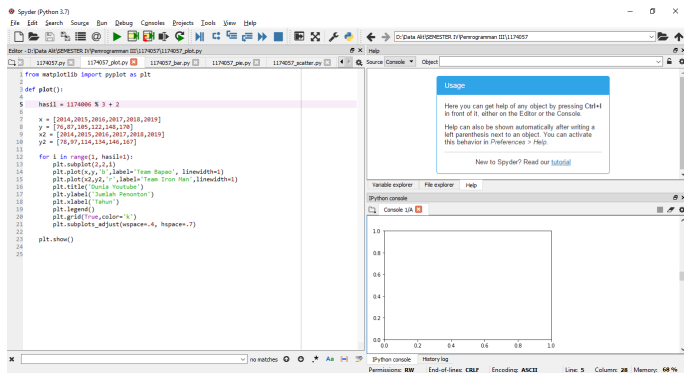
---

Sentence Wise Result

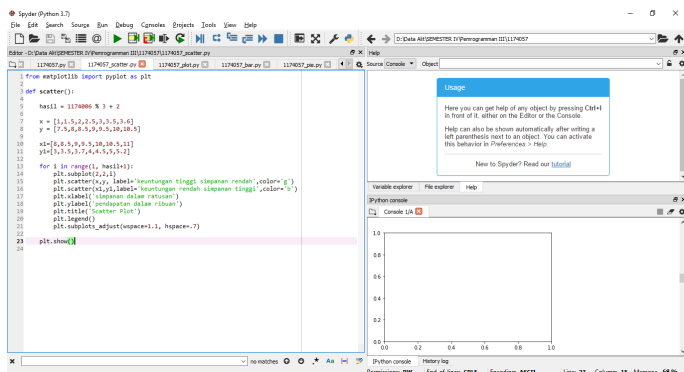
Matched Sources

Document View

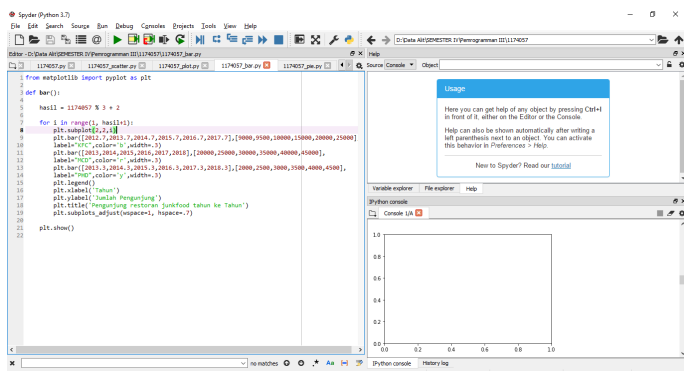
UNIQUE	\subitem Matplotlib adalah suatu library Python 2D yang dapat menghasilkan plot ...
UNIQUE	Matplotlib berfungsi sebagai pembuat grafik di berbagai platform, seperti Python d...
UNIQUE	Grafik yang dibuat menggunakan Matplotlib bisa dibuat dalam berbagai bentuk, se...
UNIQUE	\item Jelaskan langkah-langkah membuat sumbu X dan Y di matplotlib
UNIQUE	\item Buat variabel x yang menampung list untuk sumbu x dan variabel y yang men...
UNIQUE	\item Panggil fungsi plot dan isi parameter pertama dengan variabel x dan paramet...
UNIQUE	\item Lalu panggil plot tadi dengan memanggil fungsi show.



Gambar 6.16 hasil run penanganan error plot

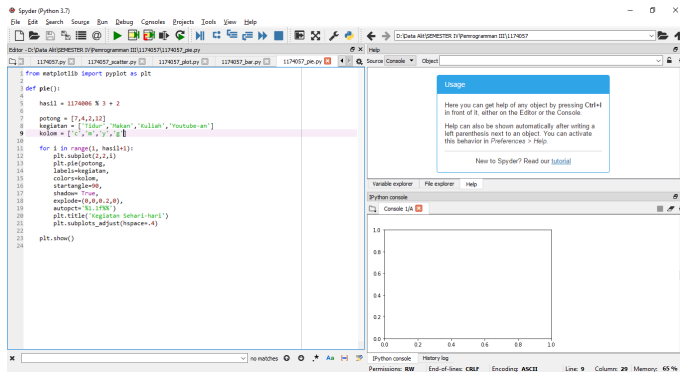


Gambar 6.17 hasil run penanganan error scatter

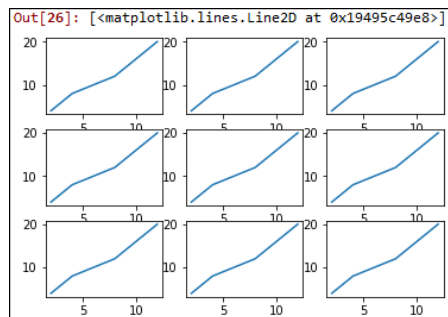


Gambar 6.18 hasil run penanganan error bar

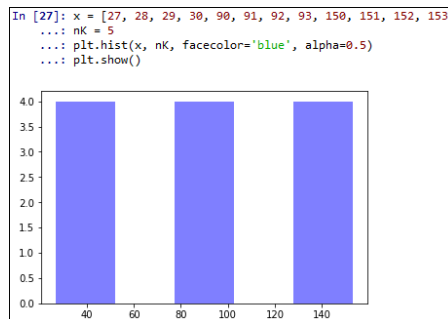




**Gambar 6.19** hasil run penanganan error pie



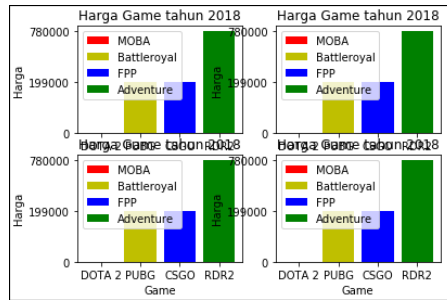
**Gambar 6.20** Penggunaan Subplot pada Matplotlib



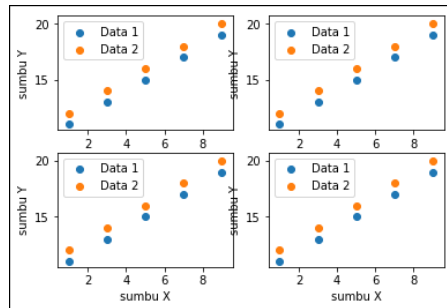
**Gambar 6.21** Hasil dari Penggunaan Hist dari Code tersebut



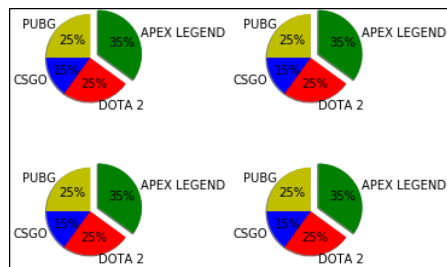
**Gambar 6.22** Hasil Check Data Plagiarisme



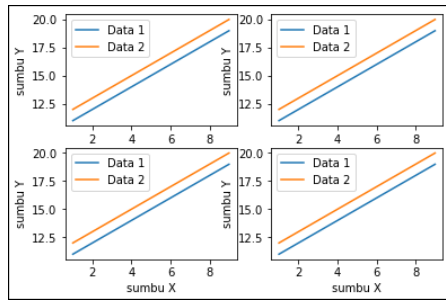
**Gambar 6.23** Hasil Code Grafik Bar atau Batang



**Gambar 6.24** Hasil Code Grafik Titik atau Scatter



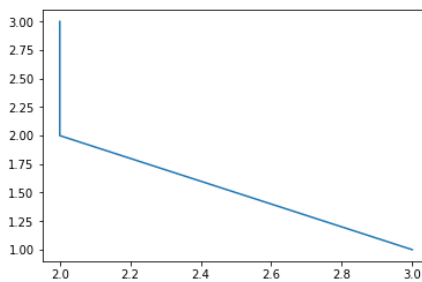
**Gambar 6.25** Hasil Code Grafik Pie atau Pie



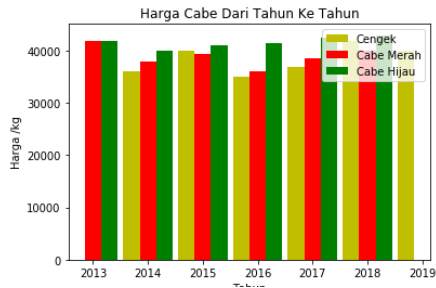
**Gambar 6.26** Hasil Code Grafik Plot atau Plot

```
In [55]: def tryExceptError():
...:     a = 1, 2, 3
...:     b = "1", "2", "3"
...:     try:
...:         plt.plot(x, b)
...:         plt.show()
...:     except IndexError:
...:         print('check datanya')
...:     print('variable error')
variable error
```

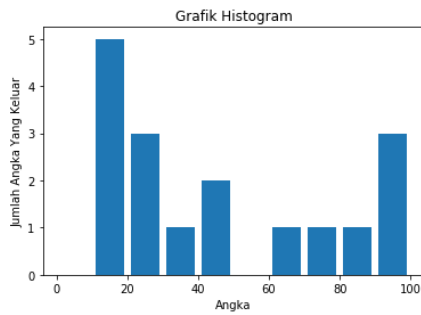
**Gambar 6.27** Hasil Code Try Error



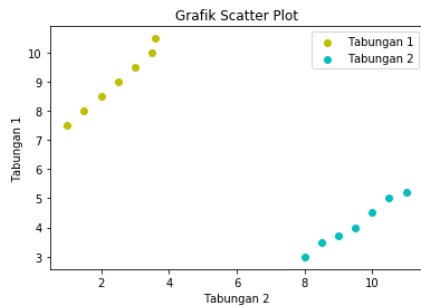
**Gambar 6.28** Hasil Melakukan Plot Sumbu X dan Y.



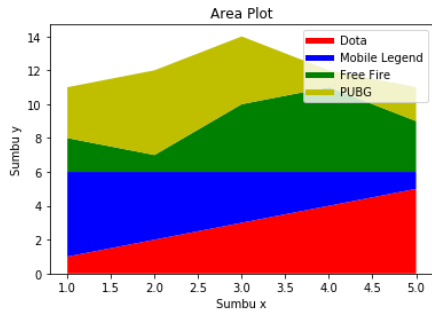
**Gambar 6.29** Fungsi Bar.



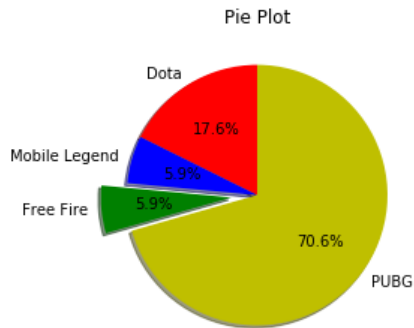
**Gambar 6.30** Fungsi Histogram.



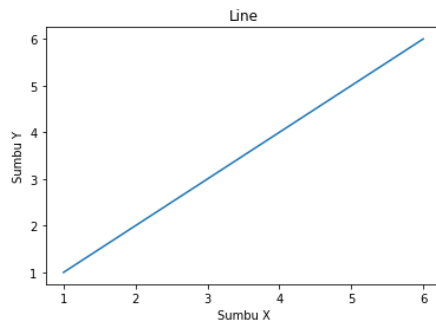
**Gambar 6.31** Fungsi Scatter Plot.



**Gambar 6.32** Fungsi Area Plot.



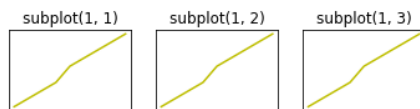
**Gambar 6.33** Fungsi Pie Plot.



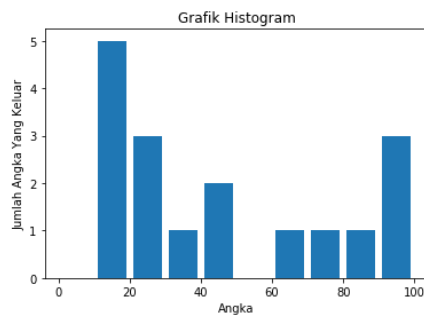
**Gambar 6.34** Fungsi Line Graphic.



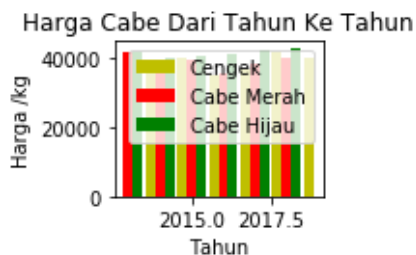
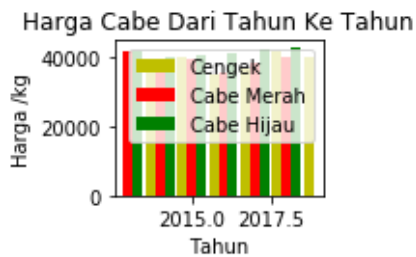
**Gambar 6.35** Fungsi Legend Dan Label.



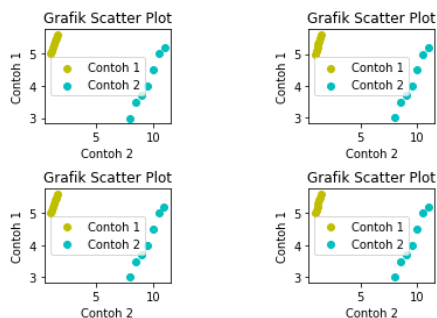
**Gambar 6.36** Fungsi Subplot.



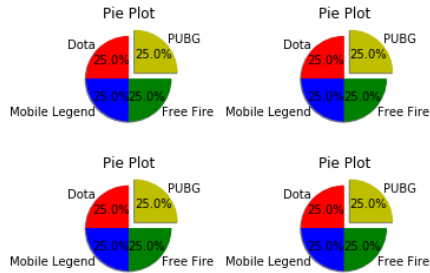
**Gambar 6.37** Fungsi Subplot.



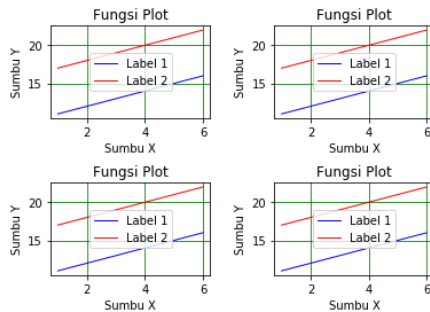
**Gambar 6.38** Fungsi Bar Praktikum.



**Gambar 6.39** Fungsi Scatter Praktikum.



**Gambar 6.40** Fungsi Pie Praktikum.



**Gambar 6.41** Fungsi Plot Praktikum.

```
In [15]: a=[2,2,3]
...: b=[3,2,1]
...:
...: plt.plot(x,y)
...:
...: plt.show()
Traceback (most recent call last):
  File "<ipython-input-15-9388a9e39054>", line 4, in <module>
    plt.plot(x,y)
NameError: name 'x' is not defined
```

**Gambar 6.42** Error Yang Di Dapat.





# DAFTAR PUSTAKA

---

1. R. Awangga, "Sampeu: Servicing web map tile service over web map service to increase computation performance," in *IOP Conference Series: Earth and Environmental Science*, vol. 145, no. 1. IOP Publishing, 2018, p. 012057.
2. D. A. Prasetya and I. Nurviyanto, "Deteksi wajah metode viola jones pada opencv menggunakan pemrograman python," 2012.
3. T. R. Perkasa, H. Widyantara, and P. Susanto, "Rancang bangun pendeteksi gerak menggunakan metode image subtraction pada single board computer (sbc)," *Journal JCONES*, vol. 3, no. 2, 2014.
4. M. Miftakhuddin, W. Suadi, and B. A. Pratomo, "Implementasi key-value store dengan struktur data list dan tree menggunakan python," *Institut Teknologi Sepuluh Noverber*.



# Index

---

disruptif, xliii  
modern, xliii