

CERDAS MENGUASAI PYTHON

Penulis:

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:

M. Yusril Helmi Setyawan

Penyunting:

Syafrial Fachrie Pane

Khaera Tunnisia

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. Sariasisih 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

1 Mengenal Python dan Anaconda	1
2 Pemograman Dasar	11
3 Praktek Pemograman Dasar	19
4 Fungsi dan Kelas	23
5 Praktek Fungsi dan Kelas	27
6 Library CSV dan Pandas	33
7 Praktek Library CSV dan Pandas	51
8 Komunikasi Perangkat Keras	65
9 Praktek Komunikasi Perangkat Keras	101
10 Matplotlib	121
11 Praktek Matplotlib	151

DAFTAR ISI

Daftar Gambar	xv
Daftar Tabel	xix
Foreword	xxix
Kata Pengantar	xxxii
Acknowledgments	xxxiii
Acronyms	xxxv
Glossary	xxxvii
List of Symbols	xxxix
Introduction	xli
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
1 Mengenal Python dan Anaconda	1
1.1 D. Irga B. Naufal Fakhri D4 TI 2C	1
1.1.1 Sejarah Python	1
1.1.2 Tanggal Rilis Python	2
1.1.3 Perbedaan Python 2 dengan Python 3	3

1.1.4	Penggunaan Python di perusahaan dunia	4
1.1.5	Cara menginstall Anaconda	5
1.1.6	Cara menggunakan Spyder pada Anaconda	7
1.1.7	Membuat Hello World di Spyder	9
2	Pemrograman Dasar	11
2.1	D. Irga B. Naufal Fakhri	11
2.1.1	Teori	11
3	Praktek Pemrograman Dasar	19
3.1	D. Irga B. Naufal Fakhri	19
3.1.1	Ketrampilan Pemrograman	19
3.1.2	Ketrampilan Penanganan Error	22
4	Fungsi dan Kelas	23
4.1	D. Irga B. Naufal Fakhri	23
4.1.1	Pemahaman Teori	23
5	Praktek Fungsi dan Kelas	27
5.1	D. Irga B. Naufal Fakhri	27
5.1.1	Keterampilan Pemrograman	27
5.1.2	Ketrampilan Penanganan Error	30
6	Library CSV dan Pandas	33
6.1	D. Irga B. Naufal Fakhri	33
6.1.1	Pemahaman Teori	33
6.2	Fanny Shafira Damayanti 1174069	36
6.2.1	Pemahaman Teori	36
6.3	Aulyardha Anindita 1174054	39
6.3.1	Pemahaman Teori	39
6.4	Nurul Izza Hamka 1174062	42
6.4.1	Pemahaman Teori	42
6.5	Chandra Kirana Poetral 1174079	45
6.5.1	Teori	45
7	Praktek Library CSV dan Pandas	51
7.1	D. Irga B. Naufal Fakhri	51
7.1.1	Soal 1	51

7.1.2	Soal 2	52
7.1.3	Soal 3	52
7.1.4	Soal 4	52
7.1.5	Soal 5	52
7.1.6	Soal 6	52
7.1.7	Soal 7	53
7.1.8	Soal 8	53
7.1.9	Soal 9	53
7.1.10	Keterampilan Penanganan Error	54
7.2	Fanny Shafira Damayanti 1174069	54
7.2.1	Keterampilan Pemrograman	54
7.2.2	Penanganan Error	55
7.3	Aulyardha Anindita 1174054	56
7.3.1	Keterampilan Pemrograman	56
7.3.2	Keterampilan Penanganan Error	57
7.4	Nurul Izza Hamka 1174062 Teori	58
7.5	Keterampilan Pemrograman	58
7.6	Ketrampilan Penanganan Error	60
7.7	Chandra Kirana Poetra	60
7.7.1	Soal 1	60
7.7.2	Soal 2	61
7.7.3	Soal 3	61
7.7.4	Soal 4	61
7.7.5	Soal 5	61
7.7.6	Soal 6	61
7.7.7	Soal 7	62
7.7.8	Soal 8	62
7.7.9	Soal 9	62
7.7.10	Keterampilan Penanganan Error	63
8	Komunikasi Perangkat Keras	65
8.1	D. Irga B. Naufal Fakhri	65
8.1.1	Pemahaman Teori	65
8.2	Difa Al Fansha	68
8.2.1	Soal Nomor 1	68
8.2.2	Soal Nomor 2	69
8.2.3	Soal Nomor 3	72
8.2.4	Soal Nomor 4	72

8.2.5	Soal Nomor 5	72
8.2.6	Soal Nomor 6	72
8.2.7	Soal Nomor 7	72
8.3	Fanny Shafira Damayanti 1174069	73
8.3.1	Pemahaman Teori	73
8.4	Aulyardha Anindita 1174054	76
8.4.1	Pemahaman Teori	76
8.5	Nurul Izza Hamka 1174062 Teori	81
8.6	Tia Nur Candida 1174086	83
8.6.1	Soal No. 1	84
8.6.2	Soal No. 2	84
8.6.3	Soal No. 3	85
8.6.4	Soal No. 4	86
8.6.5	Soal No. 5	86
8.6.6	Soal No. 6	86
8.6.7	Soal No. 7	86
8.6.8	Cek Plagiat	87
8.6.9	Kode Program	87
8.7	Chandra Kirana Poetra	87
8.7.1	Pemahaman Teori	87
9	Praktek Komunikasi Perangkat Keras	101
9.1	D. Irga B. Naufal Fakhri	101
9.1.1	Soal 1	102
9.1.2	Soal 2	102
9.1.3	Soal 3	102
9.1.4	Soal 4	103
9.1.5	Ketrampilan Penanganan Error	103
9.2	Tia Nur Candida	104
9.2.1	No. 1	104
9.2.2	No. 2	105
9.2.3	No. 3	105
9.2.4	No. 4	105
9.2.5	Kode Program Praktek	106
9.2.6	Cek Plagiat Praktek	106
9.2.7	No. 1	106
9.2.8	Kode Program Penanganan Error	107
9.2.9	Cek Plagiat Penanganan Error	107

9.3	Fanny Shafira Damayanti 1174069	107
9.3.1	Keterampilan Pemrograman	107
9.3.2	Keterampilan Penanganan Error	108
9.4	Aulyardha Anindita 1174054	109
9.4.1	Keterampilan Pemrograman	109
9.4.2	Keterampilan Penanganan Error	110
9.5	Nurul Izza Hamka 1174062	111
9.5.1	Pemrograman	111
9.5.2	Penanganan Error	112
9.6	Chandra Kirana Poetra	112
9.6.1	Soal 1	112
9.6.2	Soal 2	113
9.6.3	Soal 3	113
9.6.4	Soal 4	113
9.6.5	Ketrampilan Penanganan Error	114
9.7	Difa Al Fansha	115
9.7.1	Nomor 1	115
9.7.2	Nomor 2	115
9.7.3	Nomor 3	115
9.7.4	Nomor 4	116
10	Matplotlib	121
10.1	D. Irga B. Naufal Fakhri	121
10.1.1	Pemahaman Teori	121
10.2	Fanny Shafira Damayanti 1174069	125
10.2.1	Pemahaman Teori	125
10.3	Tia Nur Candida 1174086	128
10.3.1	Teori	128
10.4	Aulyardha Anindita 1174054	132
10.4.1	Pemahaman Teori	132
10.5	Nurul Izza Hamka 1174062 Teori	137
10.6	Difa Al Fansha	140
10.6.1	Pemahaman Teori	140
10.7	Chandra Kirana Poetra	142
10.7.1	Pemahaman Teori	142
11	Praktek Matplotlib	151
11.1	D. Irga B. Naufal Fakhri	151

11.1.1	Buatlah library fungsi (file terpisah/library dengan nama NPM/textunderscore bar.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2	151
11.1.2	Buatlah library fungsi (file terpisah/library dengan nama NPM/textunderscore scatter.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2	152
11.1.3	Buatlah library fungsi (file terpisah/library dengan nama NPM/textunderscore pie.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2	153
11.1.4	Buatlah library fungsi (file terpisah/library dengan nama NPM/textunderscore pie.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2	155
11.1.5	Keterampilan Penanganan Error	156
11.2	Chandra Kirana Poetra	156
11.2.1	Buatlah librari fungsi (file terpisah/library dengan nama NPM bar.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2	156
11.2.2	Buatlah librari fungsi (file terpisah/library dengan nama NPM scatter.py) untuk plot dengan jumlah subplot NPM mod 3 + 2	157
11.2.3	Buatlah librari fungsi (file terpisah/library dengan nama NPM pie.py) untuk plot dengan jumlah subplot NPM mod 3 + 2	158
11.2.4	Buatlah librari fungsi (file terpisah/library dengan nama NPM plot.py) untuk plot dengan jumlah subplot NPM mod 3 + 2	158
11.2.5	Keterampilan Penanganan Error	159
	Daftar Pustaka	161
	Index	163

DAFTAR GAMBAR

1.1	Tampilan Instalasi 1	5
1.2	Tampilan Instalasi 2	6
1.3	Tampilan Instalasi 3	6
1.4	Tampilan Instalasi 4	6
1.5	Tampilan Instalasi 5	6
1.6	Tampilan Instalasi 6	7
1.7	Tampilan Instalasi 7	7
1.8	Tampilan Instalasi 8	7
1.9	Tampilan Instalasi 9	8
1.10	Tampilan awal Anaconda	8
1.11	Tampilan spider	8
1.12	Tampilan new file pada spider	9
1.13	print("Hello World")	9

1.14	Dialog simpan file	9
1.15	Hasil Program	10
6.1	Buat data CSV	46
6.2	Menyimpan CSV	47
6.3	Menyimpan CSV	48
8.1	Setup	66
8.2	Instalasi	66
8.3	Instalasi Berhasil	66
8.4	Device Manager	67
8.5	Arduino Terdeteksi	67
8.6	Plagiarisme	68
8.7	Licence Agreement	69
8.8	Installation Options	70
8.9	Installation Folder	70
8.10	Installation	71
8.11	Installing	71
8.12	Driver baru	73
8.13	Buka device manager	73
8.14	klik "unknown device"	74
8.15	Klik update driver software	74
8.16	Pilih browse	74
8.17	Cari folder instalasinya	74
8.18	install	75
8.19	selesai	75
8.20	plagiarisme	76
8.21	Setup	77
8.22	Instalasi	77
8.23	Search Device Manager	78

8.24	Unknown Device	78
8.25	Update Driver Software	78
8.26	Mencari folder untuk driver software	79
8.27	Lokasi folder	79
8.28	Install PAda tampilan WIndows Security	79
8.29	Pesan Peringatan Berhasil	80
8.30	Peringatan COM Arduino Uno	80
8.31	Plagiarisme	81
8.32	Hasil pembuatan fungsi pyserial.	96
8.33	Hasil cek plagiat.	97
8.34	Kode program file 1174086.py.	97
8.35	Driver Manager	97
8.36	Port	98
8.37	Instalasi	98
8.38	browse my computer	98
8.39	Driver Arduino	99
8.40	Install driver	99
8.41	Selesai	99
8.42	Result	99
8.43	Plagiarisme	100
9.1	Membaca Serial tanpa loop	102
9.2	Membaca Serial dengan loop	102
9.3	Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino.	117
9.4	Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dengan looping.	117
9.5	Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.	117
9.6	Hasil dari pembacaan fungsi untuk membaca file csv hasil arduino dan mengembalikan fungsi.	118

9.7	Hasil pembacaan fungsi untuk menanggulangi error menggunakan Try Except.	119
9.8	Cek Plagiarisme punya chandra	120
10.1	SubPlot	147
10.2	Diagram Histogram	147
10.3	Hasil compile membuat diagram menggunakan Matplotlib.	147
10.4	Hasil compile subplot menggunakan Matplotlib.	148
10.5	Hasil compile membuat diagram menggunakan Matplotlib.	149
10.6	Subplot Dita	149
10.7	Histogram Dita	150

DAFTAR TABEL

Listings

2.1	Contoh kode variable Boolean.	11
2.2	Contoh kode variable String.	11
2.3	Contoh kode variable Integer.	12
2.4	Contoh kode variable Float.	12
2.5	Contoh kode variable Hexadecimal.	12
2.6	Contoh kode variable Complex.	12
2.7	Contoh kode variable List.	12
2.8	Contoh kode variable Tuple.	12
2.9	Contoh kode variable Set.	12
2.10	Contoh kode variable Dictionary.	13
2.11	Contoh kode input dan outputnya.	13
2.12	Contoh kode operasi pertambahan.	13
2.13	Contoh kode operasi pengurangan.	13
2.14	Contoh kode operasi perkalian.	13
2.15	Contoh kode operasi pembagian.	14
2.16	Contoh kode operasi modulus.	14
2.17	Contoh kode operasi perpangkatan.	14
2.18	Contoh kode operasi pembulatan hasil pembagian kebawah.	14

src/3/1174066/1174066.py	29
src/3/1174066/1174066.py	29
src/3/1174066/1174066.py	29
src/3/1174066/main_1174066.py	30
src/3/1174066/main_1174066.py	30
src/3/1174066/main_1174066.py	31
src/4/1174066/Teori/1174066.csv	33
src/4/1174066/Teori/1174066.csv	34
src/4/1174066/Teori/1174066_csv.py	34
src/4/1174066/Teori/1174066_csv.py	35
src/4/1174066/Teori/1174066_csv.py	35
src/4/1174066/Teori/1174066_pandas.py	36
src/4/1174066/Teori/1174066_pandas.py	36
src/4/1174066/Teori/1174066_pandas.py	36
src/4/1174069/Teori/1174069_csvteori.py	37
src/4/1174054/Teori/contohcsv.py	40
src/4/1174062/Teori/CSV.py	42
src/4/1174079/Teori/1174079.csv	45
src/4/1174079/Teori/contohcsvchandra.py	49
src/4/1174079/Teori/contohcsvwrite.py	49
src/4/1174079/Teori/contohcsvwriteparameter.py	49
src/4/1174079/Teori/pandasreadchandra.py	50
src/4/1174079/Teori/pandaswritechandra.py	50
src/4/1174066/Praktek/lib_1174066_csv.py	51
src/4/1174066/Praktek/lib_1174066_csv.py	52
src/4/1174066/Praktek/lib_1174066_pandas.py	52
7.1 lib_1174066_csv.py	53
7.2 main.py	53
7.3 main2.py	53
7.4 main2.py	54
src/4/1174069/Praktek/1174069_csv.py	54
src/4/1174069/Praktek/1174069_csv.py	54
src/4/1174069/Praktek/1174069_pandas.py	54
src/4/1174069/Praktek/1174069_pandas.py	54

src/4/1174069/Praktek/1174069_pandas.py	55
src/4/1174069/Praktek/1174069_pandas.py	55
src/4/1174069/Praktek/1174069_pandas.py	55
src/4/1174069/Praktek/main.py	55
src/4/1174069/Praktek/main2.py	55
src/4/1174069/Praktek/1174069.py	56
src/4/1174054/Praktek/1174054csv.py	56
src/4/1174054/Praktek/1174054csv.py	56
src/4/1174054/Praktek/1174054pandas.py	56
src/4/1174054/Praktek/1174054pandas.py	57
src/4/1174054/Praktek/main.py	57
src/4/1174054/Praktek/main2.py	57
src/4/1174054/Praktek/1174054.py	58
src/4/1174062/Praktek/1174062csv.py	58
src/4/1174062/Praktek/1174062csv.py	58
src/4/1174062/Praktek/1174062Pandas.py	59
src/4/1174062/Praktek/main.py	59
src/4/1174062/Praktek/main2.py	59
src/4/1174062/Praktek/1174062.py	60
src/4/1174079/Praktek/lib_1174079_csv.py	60
src/4/1174079/Praktek/lib_1174079_csv.py	61
src/4/1174079/Praktek/lib_1174079_pandas.py	62
src/4/1174079/Praktek/lib_1174079_pandas.py	62
src/4/1174079/Praktek/lib_1174079_csv.py	62
7.5 main.py	62
7.6 main2.py	62
7.7 main2.py	63
src/4/1174079/Praktek/contoherrorpandas.py	63

src/5/1174066/Teori/1174066.py	68
src/5/1174069/Teori/1174069_teori.py	76
src/5/1174054/Teori/1174054.py	81
8.1 Fungsi yang menggunakan pyserial.	86
src/5/1174079/Teori/1174079.py	88
src/5/1174066/Praktek/arduinotemprature.ino	101
src/5/1174066/Praktek/1174066_realtime.py	102
src/5/1174066/Praktek/1174066_save.py	102
9.1 "Kode python"	102
9.2 "Data yang telah ditulis ke file csv"	103
src/5/1174066/Praktek/1174066_csv.py	103
src/5/1174066/Praktek/1174066_error.py	104
9.3 Fungsi untuk mendapatkan data dari Arduino.	104
9.4 Fungsi untuk mendapatkan data langsung dari Arduino dengan looping.	105
9.5 Fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.	105
9.6 Fungsi untuk membaca file CSV hasil Arduino dan mengembalikan fungsi.	106
9.7 Fungsi untuk menanggulangi error menggunakan Try Except.	106
src/5/1174069/Praktek/1174069_realtime.py	107
src/5/1174069/Praktek/1174069_save.py	107
src/5/1174069/Praktek/1174069_realtime.py	108
src/5/1174069/Praktek/1174069_csv.py	108
src/5/1174069/Praktek/1174069_praktek.py	109
src/5/1174054/Praktek/1174054realtime.py	109
src/5/1174054/Praktek/1174054save.py	109
src/5/1174054/Praktek/1174054realtime.py	110
src/5/1174054/Praktek/1174054csv.py	110
src/5/1174054/Praktek/1174054.py	110
src/5/1174062/Praktek/1174062realtime.py	111
src/5/1174062/Praktek/1174062save.py	111
src/5/1174062/Praktek/1174062realtime.py	111
src/5/1174062/Praktek/1174062csv.py	112
src/5/1174062/Praktek/1174062.py	112
src/5/1174079/Praktek/1174079realtime.py	112
src/5/1174079/Praktek/1174079save.py	113
src/5/1174079/Praktek/1174079realtime.py	113

src/5/1174079/Praktek/1174079csv.py	113
src/5/1174079/Praktek/1174079.py	114
src/5/1174076/Praktek/1174076_realtime.py	115
src/5/1174076/Praktek/1174076_save.py	115
src/5/1174076/Praktek/1174076_realtime.py	115
src/5/1174076/Praktek/1174076_csv.py	116
src/6/1174066/Teori/1174066.py	121
src/6/1174066/Teori/1174066.py	122
src/6/1174066/Teori/1174066.py	123
src/6/1174066/Teori/1174066.py	123
src/6/1174066/Teori/1174066.py	123
src/6/1174066/Teori/1174066.py	124
src/6/1174066/Teori/1174066.py	125
src/6/1174069/Teori/1174069_teori.py	125
src/6/1174069/Teori/1174069_teori.py	126
src/6/1174069/Teori/1174069_teori.py	127
src/6/1174069/Teori/1174069_teori.py	127
src/6/1174069/Teori/1174069_teori.py	127
src/6/1174069/Teori/1174069_teori.py	128
src/6/1174086/1174086.py	129
10.1 Kode program menggunakan parameter label dengan Matplotlib.	130
10.2 Kode program memanggil fungsi legend dengan Matplotlib.	130
10.3 Kode program membuat diagram menggunakan Matplotlib.	130
10.4 Kode program membuat subplot menggunakan Matplotlib.	131
10.5 Kode program membuat diagram menggunakan Matplotlib.	132
src/6/1174054/Teori/1174054.py	133

src/6/1174054/Teori/1174054.py	133
src/6/1174054/Teori/1174054.py	133
src/6/1174054/Teori/1174054.py	134
src/6/1174054/Teori/1174054.py	134
src/6/1174054/Teori/1174054.py	134
src/6/1174054/Teori/1174054.py	135
src/6/1174054/Teori/1174054.py	135
src/6/1174054/Teori/1174054.py	135
src/6/1174062/Teori/1174062.py	137
src/6/1174062/Teori/1174062.py	138
src/6/1174062/Teori/1174062.py	138
src/6/1174062/Teori/1174062.py	139
src/6/1174076/Teori/1174076.py	140
src/6/1174076/Teori/1174076.py	140
src/6/1174076/Teori/1174076.py	140
src/6/1174076/Teori/1174076.py	141
src/6/1174076/Teori/1174076.py	141
src/6/1174076/Teori/1174076.py	142
src/6/1174079/Teori/1174079.py	143
src/6/1174079/Teori/1174079.py	144
src/6/1174079/Teori/1174079.py	144
src/6/1174079/Teori/1174079.py	144
src/6/1174079/Teori/1174079.py	145
src/6/1174079/Teori/1174079.py	145
src/6/1174079/Teori/1174079.py	146
src/6/1174066/Praktek/11174066_bar.py	151
src/6/1174066/Praktek/main.py	152
src/6/1174066/Praktek/11174066_scatter.py	152
src/6/1174066/Praktek/main.py	153
src/6/1174066/Praktek/11174066_pie.py	154
src/6/1174066/Praktek/main.py	154

src/6/1174066/Praktek/l1174066_plot.py	155
src/6/1174066/Praktek/main.py	155
src/6/1174066/Praktek/l1174066_error.py	156
src/6/1174079/Praktek/l1174079_Bar.py	156
src/6/1174079/Praktek/main.py	157
src/6/1174079/Praktek/l1174079_scatter.py	157
src/6/1174079/Praktek/main.py	157
src/6/1174079/Praktek/l1174079_pie.py	158
src/6/1174079/Praktek/main.py	158
src/6/1174079/Praktek/l1174079_plot.py	158
src/6/1174079/Praktek/main.py	159
src/6/1174079/Praktek/error.py	159

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan flask 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

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].

$$ABC\mathcal{DEF}\alpha\beta\Gamma\Delta \sum_{def}^{abc} \quad (I.1)$$

BAB 1

MENGENAL PYTHON DAN ANACONDA

1.1 D. Irga B. Naufal Fakhri D4 TI 2C

1.1.1 Sejarah Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.

Python diciptakan oleh Guido van Rossum di Scitchting Mathematisch Centrum (CWI) di Belanda pada tahun 1990-an. Bahasa python terinspirasi dari bahasa pemrograman ABC dan merupakan kelanjutan dari bahasa tersebut. Nama python sendiri bukan berasal dari nama ular python namun karena Guido adalah penggemar grup komedi Inggris bernama Monty Python. Guido masih menjadi penulis utama untuk python, walaupun python bersifat open source sehingga ribuan orang juga berkontribusi dalam mengembangkan python.

Di tahun 1995, Guido melanjutkan pembuatan python di Corporation for National Research Initiative (CNRI) di Virginia Amerika, dimana dia merilis beberapa versi

dari python. Pada Mei 2000, Guido dan tim Python pindah ke BeOpen.com dan membentuk tim BeOpen PythonLabs. Di bulan Oktober pada tahun yang sama, tim python pindah ke Digital Creation (sekarang menjadi Perusahaan Zope). Pada tahun 2001, dibentuklah Organisasi Python yaitu Python Software Foundation (PSF). PSF merupakan organisasi nirlaba yang dibuat khusus untuk semua hal yang berkaitan dengan hak intelektual Python. Perusahaan Zope menjadi anggota sponsor dari PSF.

1.1.2 Tanggal Rilis Python

Semua versi python yang dirilis bersifat open source. Dalam sejarahnya, hampir semua rilis python menggunakan lisensi GFL-compatible. Berikut adalah versi major dan minor python berikut tanggal rilisnya.

- Python 1.0 – Januari 1994
- Python 1.2 – 10 April 1995
- Python 1.3 – 12 Oktober 1995
- Python 1.4 – 25 Oktober 1996
- Python 1.5 – 31 Desember 1997
- Python 1.6 – 5 September 2000
- Python 2.0 – 16 Oktober 2000
- Python 2.1 – 17 April 2001
- Python 2.2 – 21 Desember 2001
- Python 2.3 – 29 Juli 2003
- Python 2.4 – 30 Nopember 2004
- Python 2.5 – 19 September 2006
- Python 2.6 – 1 Oktober 2008
- Python 2.7 – 3 Juli 2010
- Python 3.0 – 3 Desember 2008
- Python 3.1 – 27 Juni 2009
- Python 3.2 – 20 Februari 2011
- Python 3.3 – 29 September 2012
- Python 3.4 – 16 Maret 2014
- Python 3.5 – 13 September 2015
- Python 3.6 – 23 Desember 2016

1.1.3 Perbedaan Python 2 dengan Python 3

Pada Python 2 dan Python 3 memiliki kesamaan kapabilitas namun cara penggunaannya berbeda

- Print

Pada python2, print lebih seperti statement daripada fungsi

```
1 print "Saya Belajar Python"
```

sedangkan pada python3, print digunakan sebagai fungsi

```
1 print("Saya Belajar Python")
```

- Pembagian pada Interger

Pada Python 2, semua tipe data angka yang tidak mengandung desimal akan diperlakukan sebagai integer. Terlihat mudah pada awalnya, ketika mencoba untuk membagi kedua integer akan didapatkan tipe data float.

```
1 3 / 2 = 1.5
```

Python 2 menggunakan floor division atau dibulatkan ke nilai paling rendah misalnya 1.5 jadi 1, 2.6 jadi 2 dan seterusnya. Pada Python 2.7 akan menjadi seperti ini:

```
1 3
2 4
3 x = 3 / 2
4 print a
5 #Output
6 1
```

Untuk desimal maka tambahkan .0 setelah bilangan dan menjadi seperti ini 3.0 / 2.0 untuk mendapatkan hasil 1.5 Pada Python 3, pembagian pada bilangan integer lebih intuitif:

```
1 a = 3 / 2
2 print(a)
3 #Output
4 1.5
```

Kita juga masih bisa melakukan 3.0 / 2.0 untuk mendapatkan 1.5 namun untuk mendapatkan floor division maka pada Python 3 gunakan //:

```
1 b = 3 // 2
2 print(b)
3 #Output
4 1
```

Fitur pada Python 3 ini tidak bisa digunakan pada Python 2.7

- Dukungan Unicode

Ketika bahasa pemrograman menangani tipe data string (yang mana merupakan sekumpulan karakter), mereka bisa melakukan beberapa cara berbeda sehingga komputer dapat mengubah angka ke huruf dan simbol lain. Python 2 menggunakan alfabet ASCII secara default, sehingga ketika kita mengetik "Halo!" maka Python 2 menangani string sebagai ASCII. Terbatas pada beberapa ratus karakter, ASCII mungkin bukan pilihan yang fleksibel untuk menangani proses encoding terutama yang non English.

Untuk menggunakan unicode yang lebih luwes, mendukung lebih dari 128,000 karakter maka kita harus mengetik u"Halo!", dengan tambahan u di depannya yang mana berarti Unicode.

Python 3 menggunakan Unicode secara default, yang mana menyelamatkan programmer dari tambahan kode lagi, lebih hemat waktu dan mudah untuk diisikan dan ditampilkan. Karena Unicode mendukung berbagai karakter linguistik yang beragam termasuk menampilkan emoji, penggunaan karakter secara default dengan encoding memastikan perangkat mobile didukung oleh program yang kita buat.

Jika kita ingin kode Python 3 kita mendukung Python 2, tambahkan u di depan string.

1.1.4 Penggunaan Python di perusahaan dunia

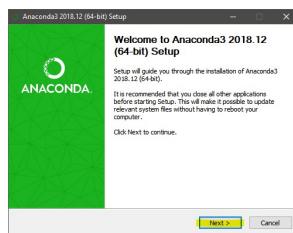
1. Google adalah perusahaan besar yang menggunakan banyak kode Python di dalam mesin pencarinya. Dan mesin pencari google adalah yang paling terkenal di dunia.
2. Youtube, situs video terbesar dan terpopuler di dunia, sebagian besar kodennya ditulis dalam bahasa Python.
3. Facebook, media sosial terbesar di dunia, menggunakan Tornado, sebuah framework Python untuk menampilkan timeline.
4. Instagram, siapa yang tidak kenal. Instagram menggunakan Django, framework python sebagai mesin pengolah sisi server dari aplikasinya.
5. Pinterest, banyak menggunakan python untuk membangun aplikasinya.
6. Dropbox, barangkali Anda adalah salah seorang pengguna layanan ini. Dropbox menggunakan python baik di sisi server maupun di sisi pengguna layanannya.
7. Quora, salah satu situs tanya jawab terbesar di dunia, dibangun menggunakan Python.
8. NASA, badan antariksa Amerika ini menggunakan Python untuk bidang sainsnya.

9. NSA, badan mata – mata Amerika banyak menggunakan Python untuk analisa kriptografi dan intelijen
10. Blender, Maya, software pembuat animasi 3D terkenal, menggunakan Python sebagai salah satu bahasa skrip pemrogramannya.
11. Raspberry Pi, komputer mini yang banyak digunakan sebagai mikrokontroller, menggunakan Python sebagai bahasa utamanya.
12. ESRI, produsen terkenal pembuat software pemetaan GIS banyak menggunakan Python di produknya.

Untuk lebih lengkapnya bisa mengunjungi www.python.org/about/success/

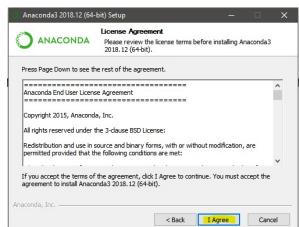
1.1.5 Cara menginstall Anaconda

1. Pastikan anda telah menginstall python dan anda mengetahui versi dari python yang telah anda install
2. Download Anaconda dari website www.anaconda.com/distribution
3. pilih sesuai dengan versi python anda, jika versi anda python3 maka pilih python3
4. Setelah itu buka file yang telah anda download
5. Setelah muncul gambar dibawah ini, tekan next

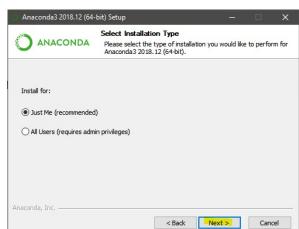


Gambar 1.1 Tampilan Instalasi 1

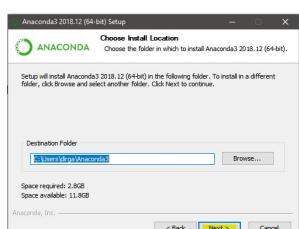
6. Baca license agreement lalu tekan 'I Agree'
7. Setelah itu pilih mau diinstall pada user yang sedang anda pakai atau kesemua user, direkomendasikan untuk memilih just me yaitu hanya user yang sedang dipakai saja
8. Catat tempat dimana anda akan menginstall anaconda, lalu tekan 'Next'
9. Setelah itu anda diberi pilihan, direkomendasikan untuk tidak mengubah pilihan tersebut, lalu tekan 'Install'



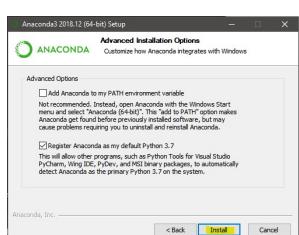
Gambar 1.2 Tampilan Instalasi 2



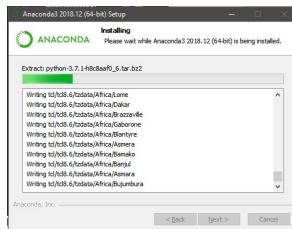
Gambar 1.3 Tampilan Instalasi 3



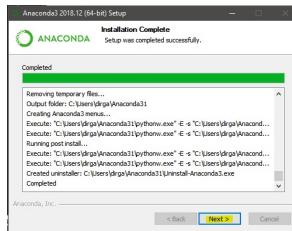
Gambar 1.4 Tampilan Instalasi 4



Gambar 1.5 Tampilan Instalasi 5

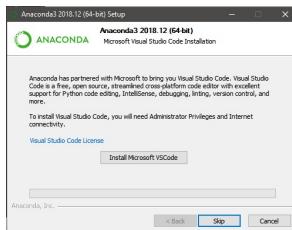


Gambar 1.6 Tampilan Instalasi 6



Gambar 1.7 Tampilan Instalasi 7

10. Tunggu sampai instalasi selesai
11. Setelah selesai tekan 'Next'
12. Setelah itu ada opsi untuk memilih untuk meinstall visual studio code, jika anda berminat klik 'Install VSCode' jika tidak tekan 'Skip'



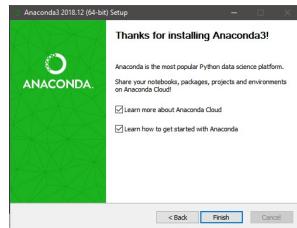
Gambar 1.8 Tampilan Instalasi 8

13. Tekan 'Finish' untuk menyelesaikan instalasi

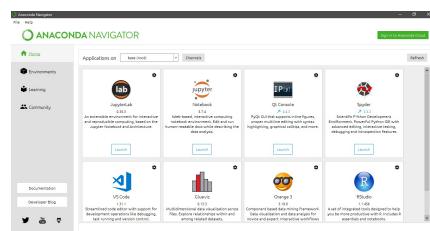
1.1.6 Cara menggunakan Spyder pada Anaconda

Pertama buka aplikasi Anaconda sampai muncul seperti ini

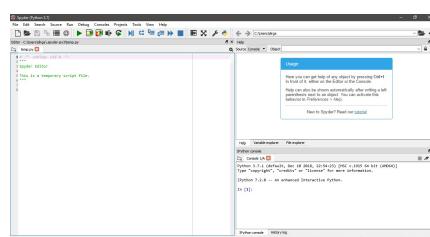
Setelah itu tekan Launch dibawah logo Spyder Tunggu sampai muncul seperti ini



Gambar 1.9 Tampilan Instalasi 9



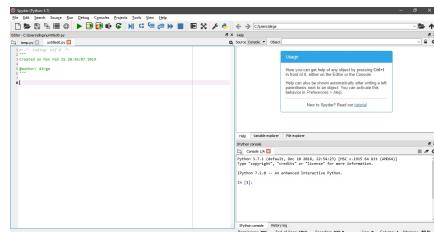
Gambar 1.10 Tampilan awal Anaconda



Gambar 1.11 Tampilan spider

1.1.7 Membuat Hello World di Spyder

Setelah membuka spyder seperti gambar di section sebelumnya tekan menu File lalu klik New File atau bisa menggunakan kombinasi tombol Ctrl + N sampai muncul seperti ini



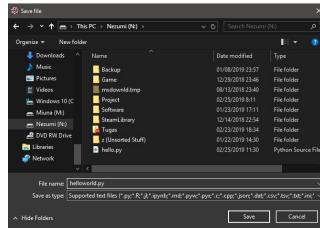
Gambar 1.12 Tampilan new file pada spider

Karena kita menggunakan Python3.7 maka kita menggunakan fungsi print() untuk memunculkan teks Hello World yang akan kita buat, tuliskan print("Hello World") pada teks editor di Spyder

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Feb 25 20:36:07 2019
4
5 @author: dirga
6 """
7
8 print("Hello World")
```

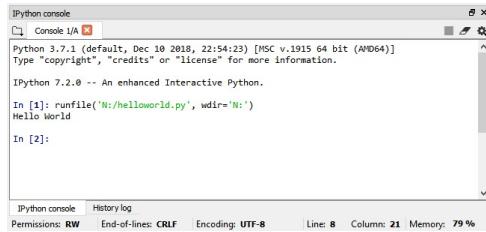
Gambar 1.13 print("Hello World")

setelah itu tekan tombol play berwarna hijau diatas, karena kita belum save file yang kita buat maka akan muncul dialog simpan file, pilih tempat dan nama file yang akan disimpan contohnya helloworld.py



Gambar 1.14 Dialog simpan file

setelah itu tekan run maka hasil dari program yang kita buat tadi ada dibagian console yang berada di pinggir kanan bawah



The screenshot shows an IPython console window. The title bar says "IPython console". The main area displays the following text:

```
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.2.0 -- An enhanced Interactive Python.

In [1]: runfile('N:/helloworld.py', wdir='N:')
Hello World

In [2]:
```

At the bottom of the window, there are tabs for "IPython console" and "History log". Below the tabs, status information is displayed: Permissions: RW, End-of-lines: CRLF, Encoding: UTF-8, Line: 8, Column: 21, Memory: 79 %.

Gambar 1.15 Hasil Program

BAB 2

PEMOGRAMAN DASAR

2.1 D. Irga B. Naufal Fakhri

2.1.1 Teori

1. Jenis-jenis variabel pada python dan cara penggunaannya:

(a) Boolean

```
1 #Boolean  
2 varbool = True  
3 print(varbool)
```

Listing 2.1 Contoh kode variable Boolean.

(b) String

```
1 #String  
2 varstr = "D. Irga B. Naufal Fakhri"  
3 print(varstr)
```

Listing 2.2 Contoh kode variable String.

(c) Integer

```

1 #Integer
2 varint = 1174066
3 print(varint)

```

Listing 2.3 Contoh kode variable Integer.

(d) Float

```

1 #Float
2 varflt = 1.1
3 print(varflt)

```

Listing 2.4 Contoh kode variable Float.

(e) Hexadecimal

```

1 #Hexadecimal
2 varhex = 0x1
3 print(varhex)

```

Listing 2.5 Contoh kode variable Hexadecimal.

(f) Complex

```

1 #Complex
2 varcmp = 7j
3 print(varcmp)

```

Listing 2.6 Contoh kode variable Complex.

(g) List

```

1 #List
2 varli = [1,2,3]
3 print(varli)
4 print(varli[2])

```

Listing 2.7 Contoh kode variable List.

(h) Tuple

```

1 #Tuple
2 vartu = (1,2,3)
3 print(vartu)
4 print(vartu[2])

```

Listing 2.8 Contoh kode variable Tuple.

(i) Set

```

1 #Set
2 varset = {1,2,3}
3 print(varset)

```

Listing 2.9 Contoh kode variable Set.

(j) Dictionary

```

1 #Dictionary
2 vardic = {1: 'satu', 2: 'dua', 'tiga':3}
3 print(vardic)
4 print(vardic[1])

```

Listing 2.10 Contoh kode variable Dictionary.

2. Permintaan Input dari user dan Outputnya

```

1 #Input dan Output
2 nama = input('Tolong masukan nama anda: ')
3 print('Hallo, '+nama)

```

Listing 2.11 Contoh kode input dan outputnya.

3. Operator dasar aritmatika dan perubahan tipe data variable

Operator dasar aritmatika

- (a) Perjumlahan (+) Operator ini berfungsi untuk melakukan operasi perjumlahan.

```

1 #Input dan Output
2 nama = input('Tolong masukan nama anda: ')
3 print('Hallo, '+nama)
4
5 #Operator Dasar Aritmatika
6 #Perjumlahan
7 var1 = 1
8 var2 = 2
9 hasil = var1 + var2
10 print(hasil)

```

Listing 2.12 Contoh kode operasi pertambahan.

- (b) Pengurangan (-) Operator ini berfungsi untuk melakukan operasi pengurangan.

```

1 #Pengurangan
2 var1 = 4
3 var2 = 3
4 hasil = var1 - var2
5 print(hasil)

```

Listing 2.13 Contoh kode operasi pengurangan.

- (c) Perkalian (*) Operator ini dipergunakan untuk melakukan operasi perkalian.

```

1 #Perkalian
2 var1 = 5
3 var2 = 6
4 hasil = var1 * var2
5 print(hasil)

```

Listing 2.14 Contoh kode operasi perkalian.

- (d) Pembagian (/) Operator ini dipergunakan untuk melakukan operasi pembagian.

```

1 #Pembagian
2 var1 = 8
3 var2 = 2
4 hasil = var1 / var2
5 print(hasil)

```

Listing 2.15 Contoh kode operasi pembagian.

- (e) Modulus (Operator ini dipergunakan untuk melakukan operasi modulus.

```

1 #Modulus
2 var1 = 7
3 var2 = 3
4 hasil = var1 % var2
5 print(hasil)

```

Listing 2.16 Contoh kode operasi modulus.

- (f) Perpangkatan (***) Operator ini dipergunakan untuk melakukan operasi perpangkatan.

```

1 #Perpangkatan
2 var1 = 10
3 var2 = 2
4 hasil = var1 ** var2
5 print(hasil)

```

Listing 2.17 Contoh kode operasi perpangkatan.

- (g) Pembulatan Kebawah Pada Hasil Pembagian (//) Operator ini dipergunakan untuk melakukan operasi pembulatan hasil bagi.

```

1 #Pembulatan_Kebawah_Pada_Hasil_Pembagian
2 var1 = 11
3 var2 = 3
4 hasil = var1 // var2
5 print(hasil)

```

Listing 2.18 Contoh kode operasi pembulatan hasil pembagian kebawah.

Perubahan tipe data variable

- (a) String menjadi Integer

```

1 #String menjadi Integer
2 varstr = '1'
3 varint = int(varstr)
4 print(varint)

```

Listing 2.19 Contoh kode variable string menjadi integer.

- (b) Integer menjadi String

```

1 #Integer menjadi String
2 varint = 2
3 varstr = str(varint)
4 print(varstr)

```

Listing 2.20 Contoh kode variable integer menjadi string.

4. Sintak perulangan (looping), jenis-jenisnya, dan penggunaannya.

- (a) While Loop While Loop adalah perulangan yang mengeksekusi statement terus menerus selama kondisi bernilai True.

```

1 #While Loop
2 hitung = 1
3 while (hitung < 6):
4     print(hitung)
5     hitung += 1

```

Listing 2.21 Contoh kode penggunaan while loop.

- (b) For Loop For Loop adalah pengulangan berdasarkan kondisi yang telah ditentukan biasanya kondisi pertambahan seperti 1 sampai 5

```

1 #For Loop
2 angka = [1, 2, 3, 4, 5]
3 for a in angka:
4     print(a)

```

Listing 2.22 Contoh kode penggunaan for loop.

- (c) Nested Loop Nested Loop merupakan pengulangan yang ada di dalam pengulangan

```

1 #Nested Loop
2 i = 1
3 while(i < 5):
4     j = 1
5     while(j <= 5):
6         print('TEST')
7         j += 1
8     i += 1

```

Listing 2.23 Contoh kode penggunaan nested loop.

5. Sintak kondisi dan penggunaannya.

- (a) If Kondisi ini digunakan untuk mengecek apabila kondisi tersebut dipenuhi akan mengeksekusi kode didalamnya.

```

1 #Kondisi
2 #If
3 var = 1
4 if var > 0:
5     print(var, "adalah angka lebih dari 0")

```

Listing 2.24 Contoh kode penggunaan if.

- (b) If Else Kondisi ini digunakan untuk mengecek apabila kondisi tersebut dipenuhi akan mengeksekusi kode didalamnya dan didalamnya memiliki dua kondisi.

```

1 #If Else
2 var = -1
3 if var > 0:
4     print(var, "adalah angka kurang dari 0")
5 else:
6     print(var, "adalah angka lebih dari 0")

```

Listing 2.25 Contoh kode penggunaan if else.

- (c) Elif Kondisi ini digunakan untuk mengecek apabila kondisi tersebut dipenuhi akan mengeksekusi kode didalamnya dan didalamnya memiliki dua kondisi atau lebih.

```

1 #Elif
2 var = 0
3 if var > 0:
4     print(var, "adalah angka lebih dari 0")
5 elif var < 0:
6     print(var, "adalah angka kurang dari 0")
7 else:
8     print(var, "adalah angka yang sama dengan 0")

```

Listing 2.26 Contoh kode penggunaan elif.

- (d) Kondisi di dalam kondisi Kondisi ini digunakan saat kondisi memerlukan kondisi lagi didalamnya

```

1 #Kondisi di dalam kondisi
2 if var > 0:
3     print(var, "adalah angka lebih dari 0")
4     if var > 10:
5         print(var, "adalah angka lebih dari 10.")
6     else:
7         print(var, "adalah angka lebih dari 10.")
8 elif var < 0:
9     print(var, "adalah angka kurang dari 0")
10 else:
11     print(var, "adalah angka yang sama dengan 0")

```

Listing 2.27 Contoh kode penggunaan kondisi di dalam kondisi.

6. Jenis-jenis error pada python dan cara mengatasinya.

- Syntax Errors Syntax Errors adalah kesalahan pada penulisan syntax atau kode. Solusinya adalah memperbaiki penulisan syntax atau kode
- Zero Division Error ZeroDivisionError adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0). Solusinya adalah tidak membagi suatu yang hasilnya nol.

- Name Error NameError adalah exception saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi atau tidak ada. Solusinya adalah memastikan variabel atau function yang akan dipanggil ada didalam program atau tidak salah mengetikannya.
- Type Error TypeError adalah exception saat melakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkoversi varibelnya sesuai dengan tipe data sesuai dengan yang akan digunakan.

7. Cara pemakaian Try Except.

```
1 #Try Except
2 try :
3     print(tidakada)
4 except:
5     print("Terjadi kesalahan penulisan kode")
6 finally:
7     print("Try except telah selesai")
```

Listing 2.28 Contoh kode penggunaan try except.

BAB 3

PRAKTEK PEMOGRAMAN DASAR

3.1 D. Irga B. Naufal Fakhri

3.1.1 Ketrampilan Pemrograman

1. Jawaban Soal 1

```
1 #Jawaban No. 1
2
3 print(1174066%3)
4 #mod3 = 1
5
6 print(" ##  ## #######    ###  #####  #######  #######")
7 print("###  ###  #######  #######  ##  ##  ##")
8 print(" ##  ##  ##  ##  ##  ##  ##  ##  ##  ##")
9 print(" ##  ##  ##  #######  ##  ##  #######  #######")
10 print(" ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##")
11 print(" ##  ##  ##  ##  ##  ##  #######  #######  ##")
12 #Cara kedua Penulisan NPM tanda pagar menurut https://
   stackoverflow.com/questions/9632995/how-to-easily-print-ascii
   -art-text
13 text = "1174066"
14 from PIL import Image, ImageDraw, ImageFont
```

```

15 import numpy as np
16 myfont = ImageFont.truetype("verdanab.ttf", 12)
17 size = myfont.getsize(text)
18 img = Image.new("1", size, "black")
19 draw = ImageDraw.Draw(img)
20 draw.text((0, 0), text, "white", font=myfont)
21 pixels = np.array(img, dtype=np.uint8)
22 chars = np.array([' ', '#'], dtype="U1")[pixels]
23 strings = chars.view('U' + str(chars.shape[1])).flatten()
24 print("\n".join(strings))

```

2. Jawaban Soal 2

```

1 #Jawaban No. 2
2
3 npm = input("Input : ")
4 i = 1
5 while(i <= int(npm[5:])):
6     print("Hallo, "+str(npm)+" apa kabar?")
7     i += 1

```

3. Jawaban Soal 3

```

1 #Jawaban No. 3
2
3 npm = input("Input : ")
4 i = 1
5 akhir = int(npm[4:5]) + int(npm[5:6]) + int(npm[6:7])
6 while(i <= akhir):
7     print("Hallo, "+str(npm[4:7])+" apa kabar?")
8     i += 1

```

4. Jawaban Soal 4

```

1 #Jawaban No. 4
2
3 npm = input("Input : ")
4 print("Hallo, "+str(npm[-3])+" apa kabar?")

```

5. Jawaban Soal 5

```

1 #Jawaban No. 5
2
3 a = 1
4 b = 1
5 c = 7
6 d = 4
7 e = 0
8 f = 6
9 g = 6
10
11 npm = [a,b,c,d,e,f,g]
12
13 for n in npm:
14     print(n, end="")
15 print()

```

6. Jawaban Soal 6

```
1 #Jawaban No. 6
2
3 print(a+b+c+d+e+f+g)
```

7. Jawaban Soal 7

```
1 #Jawaban No. 7
2
3 print(a*b*c*d*e*f*g)
```

8. Jawaban Soal 8

```
1 #Jawaban No. 8
2
3 for n in npm:
4     print(n)
```

9. Jawaban Soal 9

```
1 #Jawaban No. 9
2
3 for n in npm:
4     if(n % 2 == 0):
5         if(n != 0):
6             print(n, end =",")
7
8 print()
```

10. Jawaban Soal 10

```
1 #Jawaban No. 10
2
3 for n in npm:
4     if(n % 2 != 0):
5         print(n, end =",")
6
7 print()
```

11. Jawaban Soal 11

```
1 #Jawaban No. 11
2
3 for n in npm:
4     for x in range(2,n):
5         if n%x != 0:
6
7             break
8         print(n, "Bilangan prima")
9         break
```

3.1.2 Ketrampilan Penanganan Error

1. Jawaban Soal No. 1

- Syntax Errors Syntax Errors adalah kesalahan pada penulisan syntax atau kode. Solusinya adalah memperbaiki penulisan syntax atau kode
- Zero Division Error ZeroDivisonError adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0). Solusinya adalah tidak membagi suatu yang hasilnya nol.
- Name Error NameError adalah exception saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi atau tidak ada. Solusinya adalah memastikan variabel atau function yang akan dipanggil ada didalam program atau tidak salah mengetikannya.
- Type Error TypeError adalah exception saat melakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkoversi varibelnya sesuai dengan tipe data sesuai dengan yang akan digunakan.

2. Jawaban Soal No. 2

```
1 a = "1"
2 b = 1
3
4 try :
5     print(a+b)
6 except TypeError:
7     print("Berbeda type datanya sehingga tidak bisa dijumlahkan")
```

BAB 4

FUNGSI DAN KELAS

4.1 D. Irga B. Naufal Fakhri

4.1.1 Pemahaman Teori

1. Fungsi

Fungsi adalah blok blok kode yang terorganisir yang dapat digunakan kembali didalam program yang digunakan untuk melakukan suatu perintah yang telah diberikan. Untuk membuat fungsi kita harus menggunakan def kemudian nama fungsinya dan (variable)nya diakhiri oleh tanda :

```
1 #Contoh fungsi inputan
2 def printteks( str ):
3     print (str)
4     return
5
6 printteks("Ini adalah Contoh fungsi")
```

Listing 4.1 Contoh kode fungsi inputan ke fungsi.

Fungsi juga berguna untuk melemparkan variable contohnya

```

1 #Contoh fungsi outputan
2 def output( str ):
3     full = "Ini adalah contoh dari " + str
4     return full
5
6 print(output("fungsi outputan"))

```

Listing 4.2 Contoh kode fungsi outputan ke fungsi.

2. Paket(Package) atau Library

Paket atau yang biasa disebut dengan library adalah kumpulan kode-kode fungsi atau method pada python yang dapat dipanggil kedalam program python yang kita buat. Package berada di file terpisah dari main program cara memanggil package: Pastikan file package ada didalam folder yang sama lalu ditambah import dengan nama filenya tanpa extensi (.py)

```

1 #Contoh import library atau package
2 import fungsi_11174066
3
4 input(fungsi_11174066.hello("Dirga"))

```

Listing 4.3 Contoh import package atau library.

3. Kelas (Class), Objek (Object), Atribut (Attribute), dan Method

Kelas(Class) adalah sebuah blueprint(cetakan) dari sebuah objek. Objek(Object) adalah hasil cetakan dari sebuah kelas(class). Atribut(Attribute) adalah nilai data yang ada didalam sebuah object. Method adalah sesuatu yang bisa dilakukan oleh object.

```

1 #Contoh Class
2 class Mahasiswa:
3     def __init__(self ,nama ,npm ,kelas):
4         self.nama = nama
5         self.npm = npm
6         self.kelas = kelas
7
8     def datadiri(self):
9         print ("Nama: ", self.nama, " , NPM: ", self.npm, " ,
10        Kelas: ", self.kelas)
11
12 #Pembuatan Object
13 mhs1 = Mahasiswa("D. Irga",11174066,"D4TI2C")
14 mhs1.datadiri()

```

Listing 4.4 Contoh import package atau library.

4. Cara memanggil library dari instansiasi

Cara memanggilnya:

- Pertama kita import filenya
- kemudian buat variablenya jika menggunakan variable untuk menampung data

- Kemudian panggil nama classnya(file) dan panggil fungsinya
- Kemudian menggunakan perintah print untuk menampilkan data

```

1 #Contoh package nomor 4
2 def Tambah(x,y):
3     z = x + y
4     return z

```

Listing 4.5 Contoh package atau library.

```

1 #Contoh cara memanggil library atau package
2 import fungsi_1174066
3
4 x = 3
5 y = 7
6 print(fungsi_1174066.Tambah(x,y))

```

Listing 4.6 Contoh import package atau library.

5. Contoh pemakaian paket dengan perintah from kalkulator import Penambahan
Pemakaian package(paket) dengan perintah from namafilenya import berfungsi untuk memanggil fungsi dari nama filenya

```

1 #Contoh cara memanggil library atau package dari from import
2 from fungsi_1174066 import Tambah
3
4 x = 3
5 y = 7
6 print(Tambah(x,y))

```

Listing 4.7 Contoh import package atau library.

6. Jelaskan dengan contoh kode, pemakaian paket fungsi didalam folder
Jika file paket ada didalam folder maka kita harus menambahkan lokasi filenya ada didalam folder apa dengan cara menggunakan namafolder.namafile

```

1 #Contoh cara memanggil library atau package dari folder
2 from folder_1174066.fungsi_1174066 import Tambah
3
4 x = 3
5 y = 7
6 print(Tambah(x,y))

```

Listing 4.8 Contoh import package atau library didalam folder.

7. Jelaskan dengan contoh kode, pemakaian paket fungsi didalam folder
Jika file paket ada didalam folder maka kita harus menambahkan lokasi filenya ada didalam folder apa dengan cara menggunakan namafolder.namafile

```

1 #Contoh cara memanggil library atau package dari folder
2 from folder_1174066.fungsi_1174066 import Tambah
3
4 x = 3

```

```
5 y = 7  
6 print(Tambah(x,y))
```

Listing 4.9 Contoh import package atau library didalam folder.

BAB 5

PRAKTEK FUNGSI DAN KELAS

5.1 D. Irga B. Naufal Fakhri

5.1.1 Keterampilan Pemrograman

1. Jawaban nomor 1

```
1 #Jawaban nomor 1
2 def npm(npm):
3     text = str(npm)
4     from PIL import Image, ImageDraw, ImageFont
5     import numpy as np
6     myfont = ImageFont.truetype("verdanab.ttf", 12)
7     size = myfont.getsize(text)
8     img = Image.new("1",size,"black")
9     draw = ImageDraw.Draw(img)
10    draw.text((0, 0), text, "white", font=myfont)
11    pixels = np.array(img, dtype=np.uint8)
12    if int(text) % 3 == 0:
13        chars = np.array([' ', '*', '*'], dtype="U1")[pixels]
14    elif int(text) % 3 == 1:
15        chars = np.array([' ', '#', '#'], dtype="U1")[pixels]
16    elif int(text) % 3 == 2:
```

```

17     chars = np.array([' ', '+'], dtype="UI")[pixels]
18     strings = chars.view('U' + str(chars.shape[1])).flatten()
19     print("\n".join(strings))

```

2. Jawaban nomor 2

```

1 #Jawaban nomor 2
2 def loop(npm):
3     i = 1
4     while(i <= int(npm[5:])):
5         print("Hallo, "+str(npm)+" apa kabar?")
6         i += 1
7
8 loop(input("Masukan NPM : "))

```

3. Jawaban nomor 3

```

1 #Jawaban nomor 3
2 def looptambah(npm):
3     i = 1
4     akhir = int(npm[4:5]) + int(npm[5:6]) + int(npm[6:7])
5     while(i <= akhir):
6         print("Hallo, "+str(npm[4:7])+" apa kabar?")
7         i += 1
8
9 looptambah(input("Input: "))

```

4. Jawaban nomor 4

```

1 #Jawaban nomor 4
2 def hallo(npm):
3     print("Hallo, "+str(npm[-3:])+" apa kabar?")
4
5 hallo(input("Input: "))

```

5. Jawaban nomor 5

```

1 #Jawaban nomor 5
2 def array(npm):
3     for x in npm:
4         print(x)

```

6. Jawaban nomor 6

```

1 #Jawaban nomor 6
2 def jumlah(npm):
3     npm = list(map(int, npm))
4     hasil = 0
5     for x in npm:
6         hasil += x
7     print("Hasil perjumlahan dari npm anda adalah: " + str(hasil))
8
9 jumlah(input("NPM: "))

```

7. Jawaban nomor 7

```
1 #Jawaban nomor 7
2 def kali(npm):
3     npm = list(map(int, npm))
4     hasil = 0
5     for x in npm:
6         hasil *= x
7     print("Hasil perkalian dari npm anda adalah: " + str(hasil))
8
9 kali(input("NPM: "))
```

8. Jawaban nomor 8

```
1 #Jawaban nomor 8
2 def printgenap(npm):
3     npm = list(map(int, npm))
4     for x in npm:
5         if(x % 2 == 0):
6             if(x != 0):
7                 print(x, end = "")
```

```
8
9 printgenap(input("NPM: "))
```

9. Jawaban nomor 9

```
1 #Jawaban nomor 9
2 def printganjil(npm):
3     npm = list(map(int, npm))
4     for x in npm:
5         if(x % 2 != 0):
6             print(x, end = "")
```

```
7
8 printganjil(input("NPM: "))
```

10. Jawaban nomor 10

```
1 #Jawaban nomor 10
2 def printprima(npm):
3     npm = list(map(int, npm))
4     prima = []
5     for n in npm:
6         bilPrima = True
7         if n == 0 or n == 1:
8             bilPrima = False
9             for x in range(2, n):
10                 if n % x == 0:
11                     bilPrima = False
12             if bilPrima:
13                 prima.append(n)
14
15             for p in prima:
16                 print(p, end = "")
```

```
17
18 printprima(input("NPM: "))
```

11. Jawaban nomor 11

```

1 #Pemanggilan library 3lib
2 from lib_1174066 import *
3
4 npm(1174066)
5 loop(input("Masukan NPM : "))
6 looptambah(input("Input: "))
7 hallo(input("Input: "))
8 array(input("NPM: "))
9 jumlah(input("NPM: "))
10 kali(input("NPM: "))
11 printgenap(input("NPM: "))
12 printganjil(input("NPM: "))
13 printprima(input("NPM: "))

```

12. Jawaban nomor 12

```

1 #Pemanggilan kelas 3lib
2 from kelas3lib_1174066 import kelas3lib
3 npm = "1174066"
4 kelas = kelas3lib(npm)
5
6 kelas.npmprint()
7 kelas.loop()
8 kelas.looptambah()
9 kelas.hallo()
10 kelas.array()
11 kelas.jumlah()
12 kelas.kali()
13 kelas.printgenap()
14 kelas.printganjil()
15 kelas.printprima()

```

5.1.2 Ketrampilan Penanganan Error

- Syntax Errors

Syntax Errors adalah kesalahan pada penulisan syntax atau kode. Solusinya adalah memperbaiki penulisan syntax atau kode

- Zero Division Error

ZeroDivisionError adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0). Solusinya adalah tidak membagi suatu yang hasilnya nol.

- Name Error

NameError adalah exception saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi atau tidak ada. Solusinya adalah memastikan variabel atau function yang akan dipanggil ada didalam program atau tidak salah mengetikannya.

- Type Error

TypeError adalah exception saat melakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkoversi varibelnya sesuai dengan tipe data sesuai dengan yang akan digunakan.

```
1 #Pemanggilan kelas 3lib
2 from kelas3lib_1174066 import kelas3lib
3 npm = "1174066"
4 kelas = kelas3lib(npm)
5
6 kelas.npmprint()
7 kelas.loop()
8 kelas.looptambah()
9 kelas.hallo()
10 kelas.array()
11 kelas.jumlah()
12 kelas.kali()
13 kelas.printgenap()
14 kelas.printganjil()
15 kelas.printprima()
```


BAB 6

LIBRARY CSV DAN PANDAS

6.1 D. Irga B. Naufal Fakhri

6.1.1 Pemahaman Teori

1. CSV

CSV (Comma Separated Values file) adalah sebuah tipe file text biasa yang memiliki penataan khusus yang biasanya berfungsi untuk mengelola data. sesuai dengan namanya file csv memisahkan setiap data menggunakan koma (,).

Format data CSV pertama kali digunakan pada tahun 1978 pada complier FORTRAN 77, kemudian nama CSV baru muncul dan mulai digunakan pada tahun 1983

Contoh data pada csv:

```
1 npm, nama , kelas , tanggal lahir  
2 11714066,D. Irga B. Naufal Fakhri ,D4 Teknik Informatika 2C  
   ,11/07/98
```

2. Aplikasi yang bisa menciptakan CSV

Semua aplikasi teks editor seperti notepad++, vscode, sublime ataupun notepad dapat menciptakan CSV termasuk aplikasi spreadsheet seperti Microsoft Excel, Libre Office

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

- Buka Microsoft Excel 2019-nya lalu buat dokumen baru
- Isikan data sesuai dengan kebutuhan, yang paling atas akan menjadi header dari file csv
- Setelah memasukkan data, klik file lalu klik Save As
- Pilih Browse dan pilih tempat menyimpannya akan dimana
- Masukkan nama file pada File Name
- Lalu pada Save As Type pilih CSV (comma delimited) (*.csv)
- Maka hasil file akan seperti ini

```

1 npm , nama , kelas , tanggal lahir
2 11714066,D. Irga B. Naufal Fakhri ,D4 Teknik Informatika 2C
   ,11/07/98

```

4. Jelaskan sejarah library csv

Module csv mengimplementasikan kelas untuk membaca dan menulis data kedalam format CSV. Hal ini memungkinkan programmer untuk "tulis data ini dalam format yang disukai oleh Excel," atau "baca data dari file yang dihasilkan oleh Excel," tanpa mengetahui detail yang tepat dari format CSV yang digunakan oleh Excel. Pemrogram juga dapat menggambarkan format CSV yang dipahami oleh aplikasi lain atau menentukan format CSV tujuan khusus untuk mereka sendiri.

5. Jelaskan sejarah library pandas

pandas adalah sebuah library open source dan berlisensi BSD yang menyediakan performa yang tinggi, mudah digunakan struktur data dan data analisis untuk python.

6. Jelaskan fungsi-fungsi yang terdapat di library csv

- csv.reader

Berfungsi untuk membaca dan mengembalikan data kedalam variable dari file csv. Fungsi reader dirancang untuk mengambil data pada setiap baris didalam file dan membuat daftar semua kolom. Kemudian, tinggal dipilih kolom mana yang diinginkan untuk data variabel.

```

1 def read(namofile):
2     with open(namofile) 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'\tNomor NPM: {row[0]} Nama: {row[1]}
Kelas: {row[2]}.')
11        line_count += 1
12        print(f'Processed {line_count} lines.')
13
14 read('1174066.csv')

```

- **csv.writer**

Berfungsi untuk menuliskan data dari variable kedalam file csv. Fungsi writer akan membuat objek yang cocok untuk menulis. Untuk mengulang data yang ada di atas baris, gunakan fungsi writerow.

```

1 def write(namofile):
2     with open(namofile, mode='w') as employee_file:
3         employee_writer = csv.writer(employee_file, delimiter=
4             ', ', quotechar='"', quoting=csv.QUOTE_MINIMAL)
5
6         employee_writer.writerow(['npm', 'nama', 'kelas'])
7         employee_writer.writerow(['1174066', 'D. Irga 3', 'D4
Teknik Informatika 2C'])
7
8 write('1174066_baru.csv')

```

- **csv.register_dialect**

Mendaftarkan dialect pada csv

- **csv.unregister_dialect**

Menghapus dialect yang diasosiasi dengan nama dari registry dialect

- **csv.list_dialects**

Mengembalikan dialect yang diasosiasi dengan nama

- **csv.field_size_limit**

Mengembalikan ukuran field maksimum yang diizinkan oleh parser.

- **csv.DictReader**

Berfungsi untuk membaca dan mengembalikan data kedalam variable dictionary dari file csv.

```

1 def readdict(namofile):
2     with open(namofile, 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'\tNomor NPM: {row["npm"]} Nama: {row["nama"]}
']) Kelas: {row["kelas"]}.')
10                line_count += 1
11                print(f'Processed {line_count} lines.')

```

```

12
13 readdict('1174066.csv')

```

7. Jelaskan fungsi-fungsi yang terdapat di library pandas

- `pandas.read_csv`

Berfungsi untuk membaca dan mengembalikan data kedalam format DataFrame dari file csv.

```

1 def readpanda(namafile):
2     df = pandas.read_csv(namafile)
3     print(df)
4
5 readpanda('1174066.csv')

```

- `to_dict`

Berfungsi untuk membaca dan mengembalikan data kedalam format dictionary dari file csv.

```

1 def readpandadict(namafile):
2     df = pandas.read_csv(namafile).to_dict()
3     print(df)
4
5 readpandadict('1174066.csv')

```

- `to_csv`

Berfungsi untuk mengedit data didalam csv dan menulisnya kedalam file csv

```

1 def writepanda(namafile):
2     df = pandas.read_csv(namafile,
3                           index_col='npm',
4                           parse_dates=['tanggallahir'],
5                           header=0,
6                           names=['npm', 'nama', 'kelas', 'tanggallahir'])
7     df.to_csv('1174066_edit.csv')
8 writepanda('1174066.csv')

```

6.2 Fanny Shafira Damayanti | 1174069

6.2.1 Pemahaman Teori

1. Fungsi CSV, sejarah dan contoh

CSV (comma separated value) digunakan pada tahun 1983 merupakan suatu tipe file yang digunakan untuk pengolahan informasi yang dihasilkan spreadsheet yang di proses melalui mesin analitik. CSV juga digunakan sebagai file yang agnostik karena bisa digunakan oleh berbagai database untuk backup data.

Contoh :

```
1 import csv
2
3 siswa = [
4     ('fanny', 'A', 90),
5     ('shafira', 'B', 85),
6     ('dama', 'A', 80),
7     ('yanti', 'B', 90),
8
9 ]
10
11 # tentukan lokasi file , nama file , dan inisialisasi csv
12 f = open('siswa.csv', 'w')
13 w = csv.writer(f)
14 w.writerow(['Nama', 'Kelas', 'Nilai'])
15
16 # menulis file csv
17 for s in siswa:
18     w.writerow(s)
19
20 # menutup file csv
21 f.close()
```

2. Aplikasi untuk membuat file CSV

- Microsoft Excel
- Spyder
- Apple Number
- LibreOffice
- Apple Office Calc
- Apache Open Office Calc

3. Cara menulis dan membaca file .csv di Excel atau Spreadsheet

Cara menulis :

Buka Ms. Excel, lalu buat file nya, ketika akan di save ganti jenis filenya menjadi .csv

- Download Template CSV.
- Buka file spreadsheet Anda di Excel.
- Buat dokumen baru di Excel.
- Tambahkan judul kolom, lalu ketikkan informasi dalam kolom tersebut
- Klik File , dan pilih Save As
- Masukkan nama file, lalu pilih CSV (Comma delimited) (* csv) dari dropdown Save as type .

Cara membaca :

- Buka Microsoft Excel.
- Mulai / buka spreadsheet
- Pilih tab Data
- Pilih opsi Dari Teks. (Jika opsi berwarna abu-abu, Anda mungkin perlu membuka spreadsheet / workbook baru).
- Temukan dan pilih file .csv yang telah Anda unduh dari Kotive. Klik pada file dan kemudian klik Impor.
- Panduan impor Teks akan terbuka. Pastikan opsi Dibatasi dipilih. Klik tombol Berikutnya.
- Pilih Koma di bawah Pembatas. Kualifikasi Teks harus menunjukkan “(tanda kutip ganda). Klik tombol Selesai. Anda mungkin ditanya Di mana Anda ingin meletakkan data? Klik pada sel kiri atas. Klik tombol OK.
- Excel menampilkan data di buku kerja Anda

4. Sejarah Libary CSV Inisiatif standardisasi utama - mentransformasikan "definisi fuzzy de facto" menjadi definisi yang lebih tepat dan de jure - adalah pada tahun 2005, dengan RFC4180, mendefinisikan CSV sebagai Tipe Konten MIME. Kemudian, pada 2013, beberapa kekurangan RFC4180 ditangani oleh rekomendasi W3C.

Pada 2014 IETF menerbitkan RFC7111 yang menjelaskan aplikasi fragmen URI pada dokumen CSV. RFC7111 menentukan bagaimana rentang baris, kolom, dan sel dapat dipilih dari dokumen CSV menggunakan indeks posisi.

Pada 2015 W3C, dalam upaya untuk meningkatkan CSV dengan semantik formal, mempublikasikan draft rekomendasi pertama untuk standar metadata CSV, yang dimulai sebagai rekomendasi pada bulan Desember tahun yang sama.

5. Sejarah Libary Pandas

Pandas muncul ketika ada bahasa pemrograman R dan Matlab. Pengembang Wes McKinney mulai mengerjakan pandas pada 2008 ketika di AQR Capital Management karena kebutuhan akan alat kinerja tinggi yang fleksibel untuk melakukan analisis kuantitatif pada data keuangan. Sebelum meninggalkan AQR, dia bisa meyakinkan manajemen untuk mengizinkannya membuka sumber perpustakaan.

Pegawai AQR lainnya, Chang She, bergabung dengan upaya ini pada 2012 sebagai kontributor utama kedua ke perpustakaan.

Pada 2015, panda menandatangani sebagai proyek NumFOCUS yang disponsori secara fiskal, sebuah badan amal nirlaba 501 (c) (3) di Amerika Serikat.

6. Fungsi yang terdapat pada Library CSV

- csv.reader berfungsi untuk membaca modul csv.

- csv.writer berfungsi untuk menulis modul csv.
- csv.writerows berfungsi untuk menambahkan baris baru.

7. Fungsi yang terdapat pada Library Pandas

Dengan panda kita dapat dengan mudah merubah data (CSV, excel, JSON atau SQL) menjadi sebuah object data yang terdiri dari baris dan kolom yang disebut dengan DataFrame. Fitur :

- DataFrame Object untuk manipulasi data dengan pengindeksan terintegrasi.
- Alat untuk membaca dan menulis data antara struktur data dalam memori dan berbagai format file.
- Penyelarasan data dan penanganan terpadu pada kehilangan data.
- Membentuk kembali dan memutar set data.
- Seleksi berbasis label, pengindeksan fantastis, dan melakukan subset kumpulan data besar.
- Penyisipan dan penghapusan kolom struktur data.
- Memungkinkan operasi split-apply-combine pada Data set.
- Menghubungkan dan menggabungkan Data set.
- Pengindeksan hierarki untuk bekerja dengan data dimensi tinggi dalam struktur data dimensi rendah.
- Fungsionalitas seri waktu: Pembuatan rentang tanggal dan konversi frekuensi.
- Menyediakan penyaringan data (sorting and filtering).

6.3 Aulyardha Anindita | 1174054

6.3.1 Pemahaman Teori

1. Fungsi File CSV, Sejarah, dan Contoh

CSV (Comma Separated Value) merupakan salah satu tipe file yang sering digunakan secara luas didalam dunia programming. CSV berfungsi dalam pengolahan informasi yang dihasilkan spreadsheet untuk diproses lebih lanjut menggunakan mesin analitik. CSV juga digunakan oleh berbagai database untuk proses backup data.

Pada tahun 1972 CSV memberikan format data berupa tanggal lebih awal pada komputer pribadi lebih dari satu dekade : kompiler IBM Fortran dibawah OS /360 yang kemudian disetujui pada tahun 1978. CSV pertama kali digunakan pada tahun 1983. CSV lebih mudah untuk diketik daripada data yang selaras dengan kolom tetap dan cenderung menghasilkan hasil yang salah jika suatu nilai ditinjau satu kolom dari lokasi yang dituju. pada tahun 2014 IETF menerbitkan RFC7111 yang menjelaskan fragmen URI pada dokumen CSV. RFC7111

menentukan bagaimana rentang baris, kolom, dan sel dapat dipilih dari dokumen CSV menggunakan indeks posisi. dan pada tahun 2015 W3C dalam upaya meningkatkan CSV dengan semantik formal mempublikasikan draft rekomendasi pertama untuk standar metadata CSV yang dimulai sebagai rekomendasi pada bulan desember di tahun yang sama.

Contoh CSV :

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

2. Aplikasi Yang Menciptakan File CSV

- a. Microsoft Excel
- b. Python (Spyder)
- c. Apple Numbers
- d. LibrareOffice
- e. Apple Office Calc
- f. Apache Open Office Calc.

3. Cara Menulis dan Membaca File CSV di Excel

a. Cara Menulis File CSV

- Downloadlah terlebih dahulu Template CSV.
- Kemudian Buka file spreadsheet Anda di Excel.
- Selanjutnya, Buat dokumen baru di Excel.
- Tambahkan judul kolom, lalu ketikkan informasi dalam kolom tersebut
- Klik File lalu pilih Save As
- Masukkan nama file, lalu pilih CSV (Comma delimited) (* csv) dari dropdown Save as type .

b. Cara Membaca File CSV

- Pertama, buka microsoft excelnya
- Kemudian buka spreadsheet
- Selanjutnya, pilihlah tab Data
- Kemudian pilih opsi dari teks. (Jika opsi berwarna abu-abu, Anda mungkin perlu membuka spreadsheet / workbook baru).
- Cari dan pilih file .csv yang telah Anda unduh dari Kotive. Klik pada file dan kemudian klik Impor. Panduan impor Teks akan terbuka. Pastikan opsi Dibatasi dipilih.
- Klik tombol Berikutnya. Pilih Koma di bawah Pembatas. Kualifikasi Teks harus menunjukkan “(tanda kutip ganda).
- Klik tombol Selesai. Anda mungkin ditanya Di mana Anda ingin meletakkan data? Klik pada sel kiri atas. Klik tombol OK.
- Excel menampilkan data di buku kerja Anda

4. Sejarah Library CSV

Inisiatif standardisasi utama - mentransformasikan ”definisi fuzzy de facto” menjadi definisi yang lebih tepat dan de jure - adalah pada tahun 2005, dengan RFC4180, mendefinisikan CSV sebagai Tipe Konten MIME. Kemudian, pada 2013, beberapa kekurangan RFC4180 ditangani oleh rekomendasi W3C.

Pada 2014 IETF menerbitkan RFC7111 yang menjelaskan aplikasi fragmen URI pada dokumen CSV. RFC7111 menentukan bagaimana rentang baris, kolom, dan sel dapat dipilih dari dokumen CSV menggunakan indeks posisi.

Pada 2015 W3C, dalam upaya untuk meningkatkan CSV dengan semantik formal, mempublikasikan draft rekomendasi pertama untuk standar metadata CSV, yang dimulai sebagai rekomendasi pada bulan Desember tahun yang sama.

5. Sejarah Library Pandas

Library pandas pertama kali muncul ketika ada R dan Matlab. R dan Matlab merupakan bahasa pemrograman yang berfokus pada data yang besar.

Pengembang Library Pandas adalah Wes McKinney. Dia mulai mengerjakan pandas pada tahun 2008 ketika di AQR Capital Management karena kebutuhan akan alat kinerja tinggi yang fleksibel untuk melakukan suatu analisis kuantitatif pada data keuangan. Pegawai AQR lainnya adalah Chang She bergabung pada tahun 2012 sebagai kontributor utama kedua ke perpustakaan. Dan pada tahun 2015 pandas menandatangani sebagai proyek NumFOCUS yang disponsori secara fiskal, yang merupakan badan amal nirlaba 501 (c) (3) di Amerika Serikat.

6. Fungsi-fungsi Yang Terdapat di Library CSV

- reader() berfungsi untuk membaca data oleh module CSV

- write() berfungsi untuk menulis isi data
- writerows() berfungsi untuk menambahkan baris baru pada file

7. Fungsi-fungsi Yang Terdapat di Library Pandas

- DataFrame, Object untuk manipulasi data dengan pengindeksan terintegrasi.
- Alat untuk membaca dan menulis data antara struktur data dalam memori dan berbagai format file.
- Penyelarasan data dan penanganan terpadu pada kehilangan data.
- Membentuk kembali dan memutar set data.
- Seleksi berbasis label, pengindeksan fantastis, dan melakukan subset kumpulan data besar.
- Penyisipan dan penghapusan kolom struktur data.
- Memungkinkan operasi split-apply-combine pada Data set.
- Menghubungkan dan menggabungkan Data set.
- Pengindeksan hierarki untuk bekerja dengan data dimensi tinggi dalam struktur data dimensi rendah.
- Fungsionalitas seri waktu: Pembuatan rentang tanggal dan konversi frekuensi.
- Menyediakan penyaringan data (sorting dan filtering).

6.4 Nurul Izza Hamka | 1174062

6.4.1 Pemahaman Teori

1. Sejarah CSV, Fungsi File CSV , dan Contoh :

Sejarah : CSV (Comma Separated Values) adalah nilai yang dipisahkan oleh tanda koma, CSV ini mulai digunakan pada tahun 1983. Pada tahun 2005 dengan RCF4180, mendefinisikan CSV sebagai konten MIME. Kemudian pada tahun 2013 ada beberapa kekurangan pada RCF4180 dan ini berhasil ditrangkan oleh W3C. Kemudian pada tahun 2004 IETF menerbitkan RCF7111, kemudian pada tahun berikutnya yaitu 2015 W3C untuk meningkatkan CSV mempublikasikan drafts of recommendations iuntuk CSV.

Fungsi : File CSV dapat dibuat atau diedit di Excel. File CSV ini menyimpan Informasi yang dipisahkan oleh tanda koma. Jika data yang kita simpan dalam bentuk CSV akan sangat mudah untuk memindahkan dari satu program ke program lainnya.

```
1 #Contoh No 1
2
3 import csv
4
```

```

5 Mahasiswa = [
6     ('Izzah', 'C', 'TI', 2017),
7     ('Fanny', 'C', 'TI', 2017),
8     ('Dita', 'C', 'TI', 2017),
9     ('Nadia', 'A', 'MB', 2017),
10    ('Sri', 'B', 'LB', 2017)
11 ]
12
13 # tentukan lokasi file , nama file , dan inisialisasi csv
14 f = open('Mahasiswa.csv', 'w')
15 w = csv.writer(f)
16 w.writerow(['Nama', 'Kelas', 'Jurusan', 'Angkatan'])
17
18 # menulis file csv
19 for s in Mahasiswa:
20     w.writerow(s)
21
22 # menutup file csv
23 f.close()

```

2. Aplikasi yang bisa menciptakan file CSV

Format file CSV bisa dibuat di Microsoft Excel, Apple Numbers, LibreOffice, dan Apple Office Calc, dan Apache Open Office Calc.

3. Cara menulis dan membaca file CSV di Excel atau Spreadsheet Cara Menulis CSV di Excel :

- Download Template CSV,
- Buka file spreadsheet Anda di Excel,
- Buat dokumen baru di Excel,
- Tambahkan judul kolom, lalu ketikkan informasi dalam kolom tersebut,
- Klik File , dan pilih Save As ,
- Masukkan nama file, lalu pilih CSV (Comma delimited) (* csv) dari drop-down Save as type.

Cara Menimport CSV di Excel :

- Mulai / buka spreadsheet ,

- Pilih tab Data,
 - Pilih opsi Dari Teks. (Jika opsi berwarna abu-abu, Anda mungkin perlu membuka spreadsheet / workbook baru),
 - Temukan dan pilih file .csv yang telah Anda unduh dari Kotive.
 - Klik pada file dan kemudian klik Impor,
 - Panduan impor Teks akan terbuka. Pastikan opsi Dibatasi dipilih.
 - Klik tombol Berikutnya,
 - Pilih Koma di bawah Pembatas. Kualifikasi Teks harus menunjukkan “(tanda kutip ganda). Klik tombol Selesai,
 - Anda mungkin ditanya Di mana Anda ingin meletakkan data? Klik pada sel kiri atas. Klik tombol OK,
 - Excel menampilkan data di buku kerja Anda.
4. Sejarah Library CSV Tahun 2005, dengan RFC4180, mendefinisikan CSV sebagai Tipe Konten MIME. Kemudian, pada 2013, beberapa kekurangan RFC4180 ditangani oleh rekomendasi W3C. Kemudian Pada 2014 IETF menerbitkan RFC7111 yang menjelaskan aplikasi fragmen URI pada dokumen CSV. Dan Pada 2015 W3C, dalam upaya untuk meningkatkan CSV dengan semantik formal, mempublikasikan draft rekomendasi pertama untuk standar metadata CSV, yang dimulai sebagai rekomendasi pada bulan Desember tahun yang sama.
5. Sejarah Library Pandas
- Pandas di Tulis oleh Wes McKinney, dan Pandas ini pertama kali di luncurkan pada tanggal 11 Januari 2008. Ditulis dengan Python, Cython, C. Sistem operasi yang digunakan adalah Lintas-platform.
6. Fungsi Dalam Library CSV
- Ada beberapa Fungsi dari library CSV yaitu Perpustakaan csv berisi objek dan kode yang menyediakan fungsinalitas untuk membaca dan menulis ke file CSV. File CSV ini dirancang dan dihasilkan Excel.
7. Fungsi Dalam Library Pandas Fungsi dari File pandas ini kita dapat mengelolah suatu data dan bentuk pengelolaannya seperti join, distinct, group by, agregasi. Pandas ini juga dapat membaca file dari berbagai format seperti CSV. Pandas adalah open source dari python yang menyediakan analisis data dan struktur yang sangat mudah digunakan, dan pandas ini tersedia untuk semua instalasi Python. Dengan panda kita dapat dengan mudah merubah data (CSV, excel, JSON atau SQL) menjadi sebuah object data yang terdiri dari baris dan kolom yang disebut dengan DataFrame.

6.5 Chandra Kirana Poetra| 1174079

6.5.1 Teori

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

CSV (Comma separated Values) adalah format data yang telah ada sebelum personal komputer hadir 10 tahun sebelumnya, CSV file pada tahun 1972 telah didukung oleh IBM Fortran, nama CSV sendiri mulai digunakan pada tahun 1983, menggunakan koma dan spasi sebagai pembatas, jadi karakter string yang tidak dikutip tidak bisa mengandung koma atau spasi, list CSV lebih mudah untuk diketik daripada data yang diatur berdasarkan kolumn, dan lebih terhindar dari hasil yang salah jika di cetak pada media seperti punched card.

1 NO;Nama;Kelas
2 1;Chandra Kirana P;D4TI2C
3 2;Bakti Qilan Mufid;D4TI2C
4 3;Gani ;D4TI2C

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

- Notepad
- OpenOffice Calc
- Excel
- Google docs

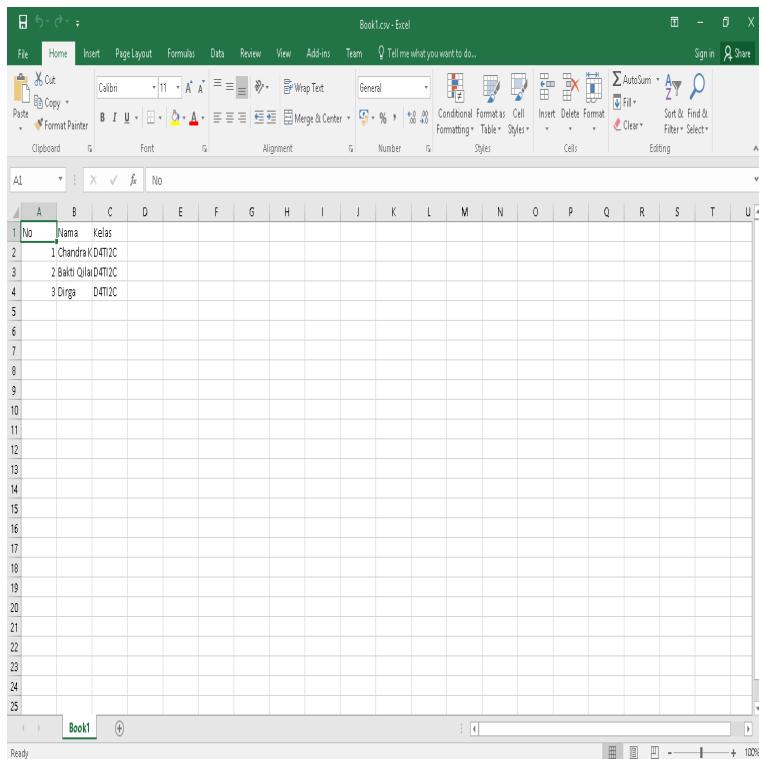
CSV file adalah sebuah file berisi text jadi CSV file bisa dibuat oleh text editor apapun.

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

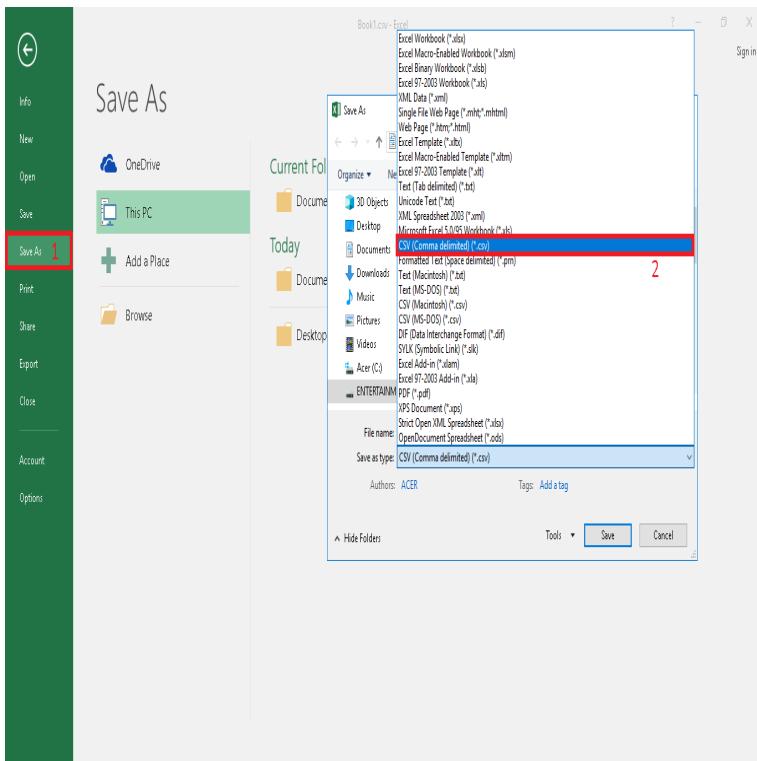
- Pertama buka terdahulu Program Excel anda
- Lalu sertakan data contoh seperti screenshot di bawah
- Simpan dengan ekstensi csv
- untuk membuka file csv, cari file yang anda simpan tadi

4. Jelaskan sejarah library csv

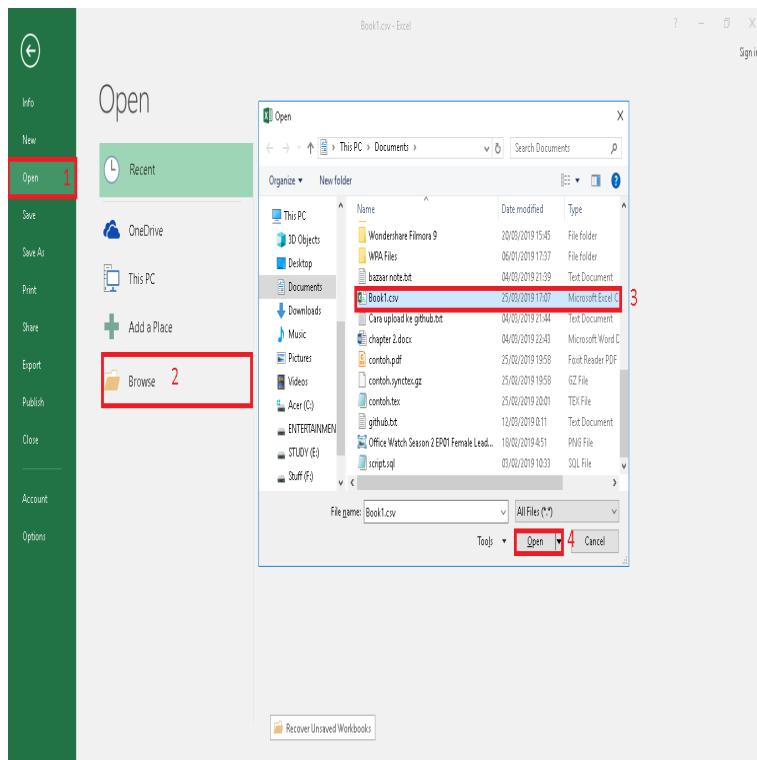
CSV (Comma separated Values) adalah format data yang telah ada sebelum personal komputer hadir 10 tahun sebelumnya, CSV file pada tahun 1972 telah didukung oleh IBM Fortran, nama CSV sendiri mulai digunakan pada tahun 1983, menggunakan koma dan spasi sebagai pembatas, jadi karakter string yang tidak dikutip tidak bisa mengandung koma atau spasi, list CSV lebih mudah untuk diketik daripada data yang diatur berdasarkan kolumn, dan lebih terhindar dari hasil yang salah jika di cetak pada media seperti punched card



Gambar 6.1 Buat data CSV



Gambar 6.2 Menyimpan CSV



Gambar 6.3 Menyimpan CSV

5. Jelaskan sejarah library pandas

Pandas (Python Data Analysis Library) nama ini didapat dari istilah panel data dan juga istilah econometrik untuk data terstruktur multidimensi, Developer Wes McKinney mulai mengerjakan pandas pada tahun 2008, ketika saat AQR capital management sedang membutuhkan alat yang memiliki performa tinggi untuk melakukan analisis kuantitatif tentang data finansial, sebelum keluar dari AQR, dia bisa membuat management percaya untuk supaya dia bisa membuat library open source. Chang She, bergabung pada tahun 2012 sebagai kontributor ke 2 untuk library pandas.

6. Jelaskan fungsi-fungsi yang terdapat di library csv

- fungsi open di CSV

Digunakan untuk membuka dan membaca file csv

```

1 with open('contohchandra.txt') as csv_file:
2     csv_reader = csv.reader(csv_file, delimiter=',')
3     line_count = 0
4     for row in csv_reader:
5         if line_count == 0:
6             print(f'Column names are {" ".join(row)}')
7             line_count += 1
8         else:
9             print(f'\t{row[0]} works in the {row[1]}'
10                department, and was born in {row[2]}.')
11             line_count += 1
12     print(f'Processed {line_count} lines.')

```

- fungsi write di csv

Digunakan untuk menulis data ke file csv menggunakan writerow

```

1 import csv
2 with open('contohchandra.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(['aaaaaaaaa', 'Accounting', 'November'])
6     employee_writer.writerow(['bbbbbb Meyers', 'IT', 'March'])

```

- fungsi write dengan parameter di csv

Digunakan untuk menulis data ke file csv menggunakan writerow

```

1 import csv
2 with open('employee_file2.csv', mode='w') as csv_file:
3     fieldnames = ['emp_name', 'dept', 'birth_month']
4     writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
5
6     writer.writeheader()
7     writer.writerow({'emp_name': 'John Smith', 'dept': 'Accounting',
8                      'birth_month': 'November'})
8     writer.writerow({'emp_name': 'Erica Meyers', 'dept': 'IT',
9                      'birth_month': 'March'})

```

7. Jelaskan fungsi-fungsi yang terdapat di library pandas

- fungsi pandas read csv

Digunakan untuk membuka, menganalisis, dan membaca file csv

```
1 import pandas  
2 df = pandas.read_csv('contohchandrapandas.csv')  
3 print(df)
```

- fungsi pandas to csv Digunakan untuk menulis data ke file csv

BAB 7

PRAKTEK LIBRARY CSV DAN PANDAS

7.1 D. Irga B. Naufal Fakhri

7.1.1 Soal 1

Buatlah fungsi untuk membuka file csv dengan lib csv mode list

```
1 def read(namafile):
2     with open(namafile) 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'\tNomor NPM: {row[0]} Nama: {row[1]} Kelas: {row[2]}.')
11                 line_count += 1
12             print(f'Processed {line_count} lines.')
```

7.1.2 Soal 2

Buatlah fungsi untuk membuka file csv dengan lib csv mode dictionary

```

1 def readdict(namafile):
2     with open(namafile, 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'\tNomor NPM: {row["npm"]} Nama: {row["nama"]}')
10            Kelas: {row["kelas"]})
11            line_count += 1
12            print(f'Processed {line_count} lines.')

```

7.1.3 Soal 3

Buatlah fungsi untuk membuka file csv dengan lib pandas mode list

```

1 def readpanda(namafile):
2     df = pandas.read_csv(namafile)
3     print(df)

```

7.1.4 Soal 4

Buatlah fungsi untuk membuka file csv dengan lib pandas mode dictionary

```

1 def readpandadict(namafile):
2     df = pandas.read_csv(namafile).to_dict()
3     print(df)
4
5 #readpandadict('1174066.csv')

```

7.1.5 Soal 5

Buat fungsi baru untuk mengubah format tanggal menjadi standar dataframe

```

1 def readrubahTanggal(namafile):
2     df = pandas.read_csv(namafile, index_col='npm', parse_dates=[0,
3     'tanggallahir'])
4     print(df)

```

7.1.6 Soal 6

Buat fungsi baru untuk mengubah index kolom

```

1 def changeIndex(namafile):
2     df = pandas.read_csv(namafile, index_col='npm')
3     print(df)

```

7.1.7 Soal 7

Buat fungsi baru untuk mengubah atribut atau nama kolom

```

1 def renamekolom(namafile):
2     df = pandas.read_csv(namafile, header=0, names=['Nomor Induk
3         Mahasiswa', 'Nama Lengkap', 'Kelas dan Jurusan', 'Tanggal Lahir'])
4     print(df)

```

7.1.8 Soal 8

Buat program main yang menggunakan library NPM csv yang membuat dan membaca file csv

```

1 def write(namafile):
2     with open(namafile, mode='w') as employee_file:
3         tulis_file = csv.writer(employee_file, delimiter=',',
4             quotechar='"', quoting=csv.QUOTE_MINIMAL)
5
6         tulis_file.writerow(['npm', 'nama', 'kelas', 'tanggallahir'])
7         tulis_file.writerow(['1174066', 'D. Irga 3', 'D4 Teknik
8             Informatika 2C', '11/07/98'])

```

Listing 7.1 lib_1174066.csv.py

```

1 import lib_1174066_csv
2
3 # Membuat file csv baru yang isinya npm, nama, kelas dan tanggallahir
4 lib_1174066_csv.write('1174066_csv_baru.csv')
5
6 # Membaca file csv
7 # List
8 lib_1174066_csv.read('1174066.csv')
9 # Dictionary
10 lib_1174066_csv.readdict('1174066.csv')

```

Listing 7.2 main.py

7.1.9 Soal 9

Buat program main2.py yang menggunakan library NPM pandas.py yang membuat dan membaca file csv

```

1 def writepanda(namafile, filebaru):
2     df = pandas.read_csv(namafile,
3             index_col='npm',
4             parse_dates=['tanggallahir'],
5             header=0,
6             names=['npm', 'nama', 'kelas', 'tanggallahir'])
7     df.to_csv(filebaru)

```

Listing 7.3 main2.py

```

1 import lib_1174066_pandas
2
3 # Membuat file baru
4 lib_1174066_pandas.writepanda('1174066.csv','1174066_baru_pandas.csv')
5
6 # Membaca file csv panda
7 # Dataframe
8 lib_1174066_pandas.readpanda('1174066.csv')
9 # Dictionary
10 lib_1174066_pandas.readpandadict('1174066.csv')

```

Listing 7.4 main2.py

7.1.10 Keterampilan Penanganan Error

Pada praktikum saat ini saya tidak mendapatkan error

7.2 Fanny Shafira Damayanti | 1174069

7.2.1 Keterampilan Pemrograman

1. NO 1

```

1 #No. 1
2 def bukaModeListCsv():
3     with open('teori.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])

```

2. NO 2

```

1 #No. 2
2 def bukaModeDictCsv():
3     with open('teori.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'])

```

3. NO 3

```

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

```

4. NO 4

```

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

```

5. NO 5

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

6. NO 6

```
1 #No. 5
2 def ubahFormatTanggal():
3     df = pandas.read_csv('teori.csv', parse_dates=['tanggal_lahir',
4     ''])
5     print(df)
```

7. NO 7

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

8. NO 8

```
1 lib = __import__('1174069_csv')
2
3 lib.bukaModeListCsv()
4 lib.bukaModeDictCsv()
5
6 lib.tulisCsv()
```

9. NO 9

```
1 lib = __import__('1174069_pandas')
2
3 lib.bukaModeListPandas()
4 lib.bukaModeDictPandas()
5
6 lib.tulisCsvPandas()
```

7.2.2 Penanganan Error

1. Peringatan error yang terdapat pada praktikum chapter 4 ini yaitu :

- Syntax Errors Syntax Errors terjadi ketika ada kesalahan dalam menuliskan kode. Solusinya adalah memperbaiki penulisan kode yang salah.
- Name Error NameError terjadi ketika salah mengetikan kode local name yang tidak terdefinisi. Solusinya adalah menuliskan kode dengan benar agar function nya dapat terpanggil.

- Type Error TypeError terjadi pada saat eksekusi terhadap fungsi dengan tipe objek tidak sesuai. Solusinya mengkonversi variablenya harus sesuai dengan tipe datanya.

Contoh Penggunaan TryExcept

```

1 #Fungsi Try Except
2 def bacaCsvPandas():
3     try:
4         df = pandas.read_csv('teori.csv')
5         print(dt)
6     except SyntaxError:
7         print("Kesalahan penulisan syntax")
8     except NameError:
9         print("Variable tersebut tidak ada")
10    except TypeError:
11        print("Tipe data salah")
12    except:
13        print("Terjadi sebuah kesalahan")

```

7.3 Aulyardha Anindita | 1174054

7.3.1 Keterampilan Pemrograman

1. Jawaban Soal No. 1

```

1 #Jawaban No. 1
2 def bukaModeListCsv():
3     with open('teori.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])

```

2. Jawaban Soal No. 2

```

1 #Jawaban No. 2
2 def bukaModeDictCsv():
3     with open('teori.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'])

```

3. Jawaban Soal No. 3

```

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

```

4. Jawaban Soal No. 4

```

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

```

5. Jawaban Soal No. 5

```

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

```

6. Jawaban Soal No. 6

```

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

```

7. Jawaban Soal No. 7

```

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

```

8. Jawaban Soal No. 8

```

1 lib = __import__('1174054csv')
2
3 lib.bukaModeListCsv()
4 lib.bukaModeDictCsv()
5
6 lib.tulisCsv()

```

9. Jawaban Soal No. 9

```

1 lib = __import__('1174054pandas')
2
3 lib.bukaModeListPandas()
4 lib.bukaModeDictPandas()
5
6 lib.tulisCsvPandas()

```

7.3.2 Keterampilan Penanganan Error

Peringatan error di praktik keempat ini, yaitu:

- Syntax Errors Syntax Errors adalah keadaan dimana pada kode python mngalami kesalahan dalam penulisan. Untuk mengatasinya yaitu dengan memperbaiki penulisan kode yang salah
- Name Error NameError adalah suatu keadaan atau exception yang terjadi ketika kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Untuk mengatasinya yaitu dengan memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.
- Type Error TypeError adalah suatu keadaan atau exception yang akan terjadi apabila pada saat dilakukannya eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Untuk mengatasinya yaitu dengan mengkoversi variabelnya sesuai dengan tipe data yang akan digunakan.

Fungsi yang menggunakan try except

```

1 #Fungsi Try Except
2 def bacaCsvPandas () :
3     try :
4         df = pandas.read_csv ('teori.csv')
5         print (dt)
6     except SyntaxError :
7         print ("Kesalahan penulisan syntax")
8     except NameError :
9         print ("Variable tersebut tidak ada")
10    except TypeError :
11        print ("Tipe data salah")
12    except :
13        print ("Terjadi sebuah kesalahan")
```

7.4 Nurul Izza Hamka | 1174062 | Teori

7.5 Keterampilan Pemrograman

1. NO1

```

1 #Jawaban No. 1
2 def bukaModeListCsv () :
3     with open ('teori.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])
```

2. NO2

```

1 #Jawaban No. 2
2 def bukaModeDictCsv () :
3     with open ('teori.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' ])
```

3. NO3

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

4. NO4

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

5. NO5

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

6. NO6

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

7. NO7

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

8. NO8

```
1 lib = __import__('1174006csv')
2
3 lib.bukaModeListCsv()
4 lib.bukaModeDictCsv()
5
6 lib.tulisCsv()
```

9. NO9

```
1 lib = __import__('1174006pandas')
2
3 lib.bukaModeListPandas()
4 lib.bukaModeDictPandas()
5
6 lib.tulisCsvPandas()
```

7.6 Ketrampilan Penanganan Error

- NameError Error yang terjadi adalah NameError yang mana terjadi kesalahan pada saat melakukan eksekusi dan tidak dapat terdefinisi. Penanganan yang dapat dilakukan adalah memastikan bahwa Variable dan Function yang akan kita panggil pastikan semua benar.
- SyntaxError Tipe Error ini adalah ada kesalahan pada penulisan, untuk itu pastikan penulisannya benar.
- TypeError Error ini terjadi pada saat eksekusi terhadap dan operasi dan fungsi masuk kedalam objek sedangkan typenya salah.
- Menggunakan Try Except

```

1 #Fungsi Try Except
2 def bacaCsvPandas () :
3     try :
4         df = pandas.read_csv ('teori.csv')
5         print(dt)
6     except SyntaxError:
7         print("Kesalahan penulisan syntax")
8     except NameError:
9         print("Variable tersebut tidak ada")
10    except TypeError:
11        print("Tipe data salah")
12    except:
13        print("Terjadi sebuah kesalahan")

```

7.7 Chandra Kirana Poetra

7.7.1 Soal 1

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

```

1 def readcsvfile (namafайл):
2     with open(namafайл) 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'\tNomor No: {row[0]} Nama: {row[1]} NPM: {row
11 [2]}.')
12                 line_count += 1
13                 print(f'Processed {line_count} lines.')

```

7.7.2 Soal 2

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

```

1 def readdengandict(a):
2     with open(a, 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'\tNomor No: {row["No"]} Nama: {row["Nama"]} NPM:
{row["NPM"]}')
10            line_count += 1
11            print(f'Processed {line_count} lines.')

```

7.7.3 Soal 3

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

```

1 def exampleofhowtoreadpanda(a):
2     df = pandas.read_csv(a)
3     print(df)

```

7.7.4 Soal 4

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

```

1 def exampleofreadpandausingdict(a):
2     df = pandas.read_csv(a).to_dict()
3     print(df)

```

7.7.5 Soal 5

Buat fungsi baru di NPM pandas.py untuk mengubah format tanggal menjadi standar dataframe

```

1 def exampleofreadchangedateformatpandas(a):
2     df = pandas.read_csv(a, index_col='npm', parse_dates=['
tanggallahir'])
3     print(df)

```

7.7.6 Soal 6

Buat fungsi baru di NPM pandas.py untuk mengubah index kolom

```

1 def exampleofhowtochangeindexinpandas(a):
2     df = pandas.read_csv(a, index_col='npm')
3     print(df)

```

7.7.7 Soal 7

Buat fungsi baru di NPM pandas.py untuk mengubah atribut atau nama kolom

```

1 def exampleofhowtorenameacolumn(a):
2     df = pandas.read_csv(a, header=0, names=['Student ID', 'Full Name',
3     'Class', 'Date of birth'])
4     print(df)

```

7.7.8 Soal 8

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

```

1 def write(a):
2     with open(a, mode='w') as exampleofhowtowrite:
3         tulis_file = csv.writer(exampleofhowtowrite, delimiter=',',
4         quotechar='"', quoting=csv.QUOTE_MINIMAL)
5
6         tulis_file.writerow(['No', 'Nama', 'NPM', 'tanggallahir'])
7         tulis_file.writerow(['1', 'Chandra Kirana Poetra', '1174079',
8         '05/09/99'])
9
10    import lib_1174079_csv
11    # buat file baru
12    lib_1174079_csv.write('1174079_csv_baru.csv')
13    # read dengan
14    # Dictionary
15    lib_1174079_csv.readdengandict('1174079.csv')
16    # List
17    lib_1174079_csv.readcsvfile('1174079.csv')

```

Listing 7.5 main.py

7.7.9 Soal 9

Buat program main2.py yang menggunakan library NPM pandas.py yang membuat dan membaca file csv

```

1 def examplehowtowritepanda(a,b):
2     df = pandas.read_csv(a,
3             index_col='npm',
4             parse_dates=['tanggallahir'],
5             header=0,
6             names=['No', 'Nama', 'NPM', 'tanggallahir'])
7     df.to_csv(b)

```

Listing 7.6 main2.py

```
1 import lib_1174079_pandas
2 # filebaru
3 lib_1174079_pandas . writepanda('117079.csv','1174066_baru_pandas.csv')
4 # baca file csv dengan library panda
5 # Dataframe
6 lib_1174079_pandas . exampleofhowtoreadpanda('117079.csv')
7 # Dictionary
8 lib_1174079_pandas . exampleofreadpandausingdict('117079.csv')
```

Listing 7.7 main2.py

7.7.10 Keterampilan Penanganan Error

Unicode Decode Error Terjadi ketika kesalahan decode untuk unicode terdeteksi, typo semacamnya

```
1 def nyobaintrycatch():
2     try :
3         da = pandas.read_csv("1174079.csv")
4         print(da)
5     except SyntaxError:
6         print("Cek apabila ada typo")
7     except NameError:
8         print("Variable tidak ditemukan")
9     except UnicodeDecodeError:
10        print("ada yang salah di penulisan")
11    except :
12        ("terjadi kesalahan")
```


BAB 8

KOMUNIKASI PERANGKAT KERAS

8.1 D. Irga B. Naufal Fakhri

8.1.1 Pemahaman Teori

1. Device Manager pada Windows dan folder /dev pada Linux

Device manager adalah system tools yang fungsinya untuk mengidentifikasi dan mengelola hardware serta driver yang diperlukan oleh hardware yang dihubungkan. Device Manager juga bisa mengenali hardware dan menginstall drivernya secara otomatis, tapi kalo driver dari hardware tersebut tidak ada pada koleksi driver Windows, maka Device Manager akan memberikan notifikasi dan tanda khusus bahwa hardware tersebut membutuhkan driver terpisah agar bisa terinstall dengan benar.

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

2. Jelaskan langkah-langkah instalasi driver dari arduino

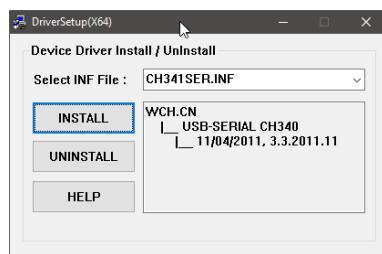
Apabila anda menggunakan Arduino versi original maka drivernya akan diinstall saat anda menginstall Arduino IDE saat anda menhubungkan Arduino anda ke PC menggunakan kabel USB. Jika anda menggunakan Arduino Uno SMD Clone seperti saya maka anda harus menginstall drivernya terlebih dahulu setelah menginstall Arduino Uno, Caranya:

- Buka program setup.exe pada folder DRIVER1



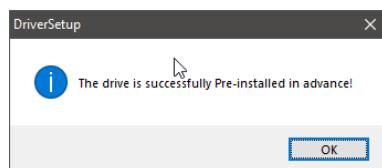
Gambar 8.1 Setup

- Setelah itu klik Install



Gambar 8.2 Instalasi

- Setelah muncul seperti dibawah ini tekan OK

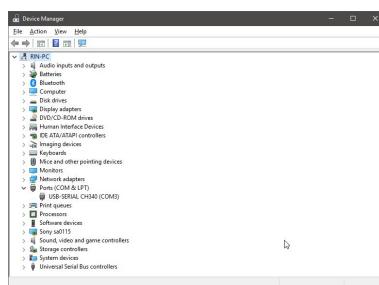


Gambar 8.3 Instalasi Berhasil

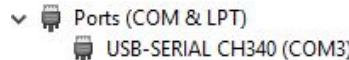
- Hubungkan Arduino ke PC, lalu buka device manager
- Lalu pilih pada bagian port apabila seperti ini maka driver arduino telah terinstall

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

Cara membaca baudrate adalah dengan cara membuka arduino ide lalu mengclik serial monitor yang iconnya seperti kaca pembesar(Cari). Untuk mengecek



Gambar 8.4 Device Manager



Gambar 8.5 Arduino Terdeteksi

port kita bisa melihatnya melalui device manager pada bagian ports, yang ada tulisan COMXX (XX adalah angka dari COM) itu adalah portnya

4. Jelaskan sejarah library pyserial

pySerial adalah modul API Python untuk mengakses port serial. pySerial menyediakan API yang seragam di berbagai sistem operasi, termasuk Windows, Linux, dan BSD.

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

- **Serial()**

Berfungsi untuk membuka port serial.

- **Write()**

Berfungsi untuk mengirimkan data string ke port serial dan mengembalikan nomor bytes yang terkirim.

- **Read(size)**

Berfungsi untuk membaca data dari port serial.

- **Readline(size)**

Berfungsi untuk membaca line sampai line terakhir (EOL).

- **Close()**

Berfungsi untuk menutup pembacaan port serial.

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

Perulangan dalam python berfungsi untuk mengulangi kode/perintah yang ada didalam perulangan tersebut. ada dua macam perulangan pada python, yang

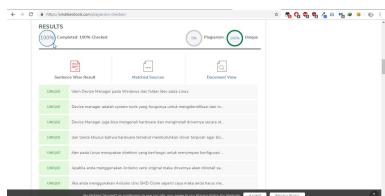
pertama for dan yang kedua while. Perbedaan for dan while adalah for yaitu perulangan yang menghitung (Counted Loop) sedangkan while adalah perulangan yang tidak terhitung (Uncounted Loop). Penggunaan for itu biasanya jika untuk mengulangi kode yang akan ditentukan berapa kali diulangnya sedangkan while digunakan jika ada syarat tertentu untuk mengulangi kode itu dan tidak menentu berapa banyak perulangannya. Mengapa diperlukan perulangan, karena agar data yang dibaca tidak hanya satu kali saja namun berkali kali, dengan adanya perulangan kita bisa membaca datanya berulang kali. Sehingga data yang kita baca dapat muncul lebih dari satu. Sedangkan kalau kita tidak memakai perulangan maka data yang muncul hanya satu.

7. Jelaskan bagaimana cara membuat fungsi yang menggunakan pyserial

Pembuatan fungsi sama seperti pembuatan fungsi seperti biasanya namun method dari pyserial dimasukkan kedalam fungsi dan dipanggil fungsi yang kita buat tadi

```
1 import serial
2
3 def ReadSerial():
4     ser = serial.Serial("COM3", 9600)
5     read = ser.readline()
6     print(read)
7
8 ReadSerial()
```

8. Scan Plagiarisme



Gambar 8.6 Plagiarisme

8.2 Difa Al Fansha

8.2.1 Soal Nomor 1

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

Jawab :

- Fungsi Device Manager di Sistem Operasi Windows

Device manager merupakan program yang mengatur device atau perangkat yang terhubung dengan komputer / laptop.

- Fungsi Folder /dev di Sistem Operasi Linux

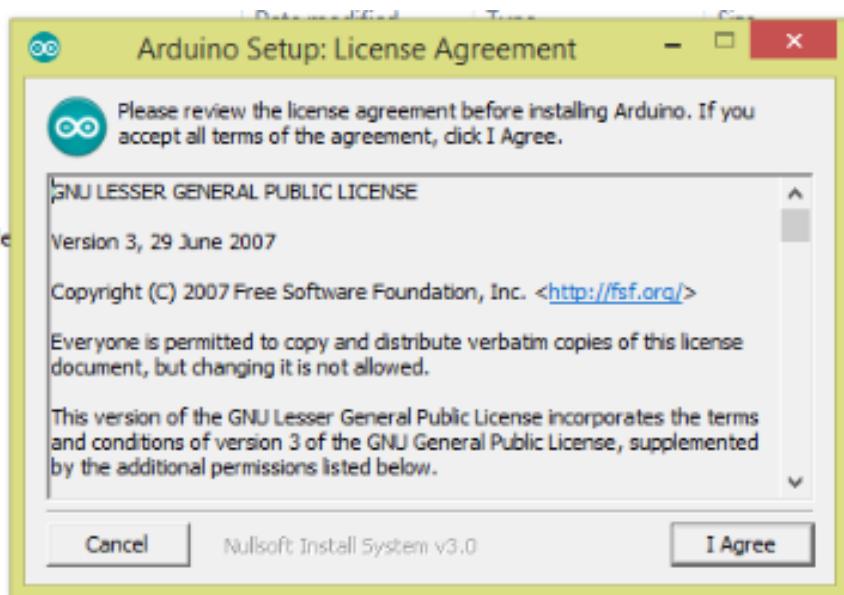
Device manager pada linux berada pada folder /dev yang mempunyai arti device. folder ini berisi konfigurasi device pada sistem.

8.2.2 Soal Nomor 2

Jelaskan langkah-langkah instalasi driver dari arduino!

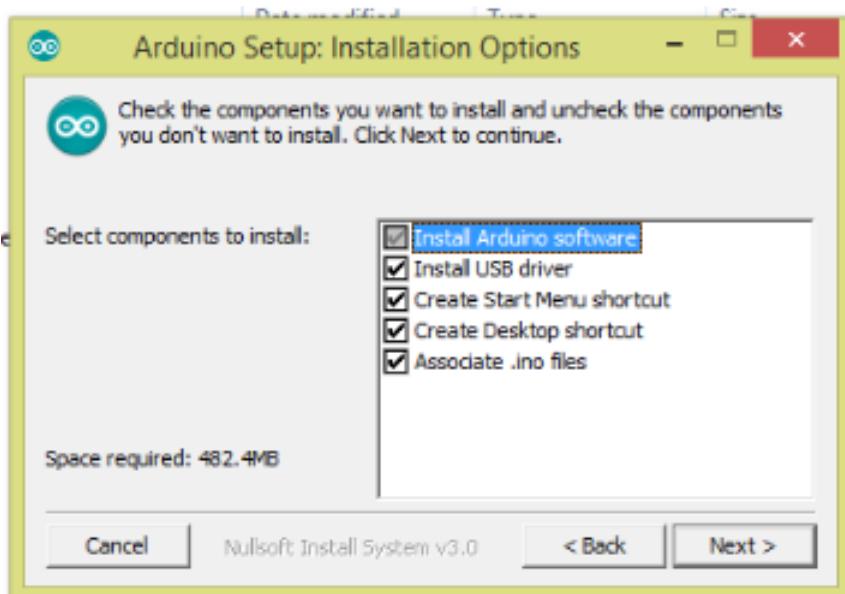
Jawab :

1. Download Software IDE Arduino di <https://www.arduino.cc/en/Main/Software>
2. Setelah di download jalankan software tersebut.
3. Pilih I Agree.

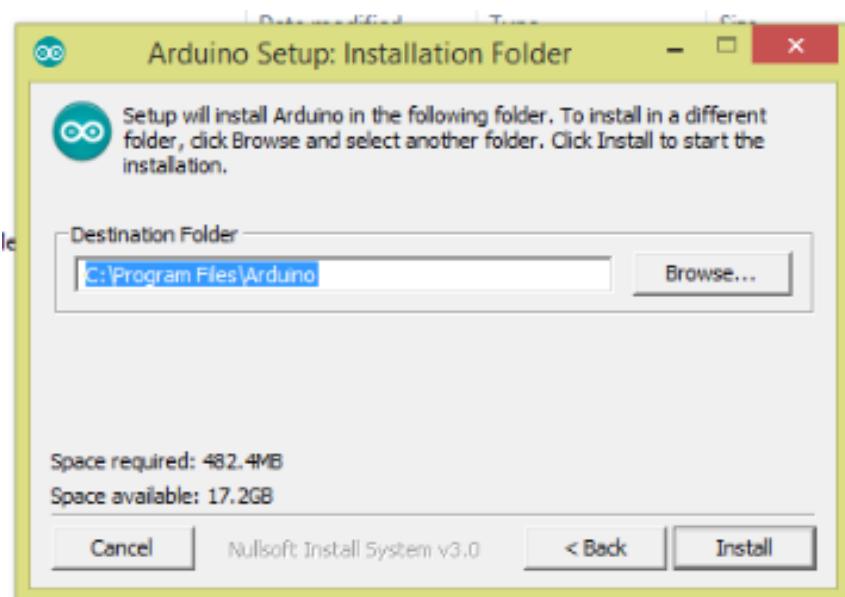


Gambar 8.7 Licence Agreement

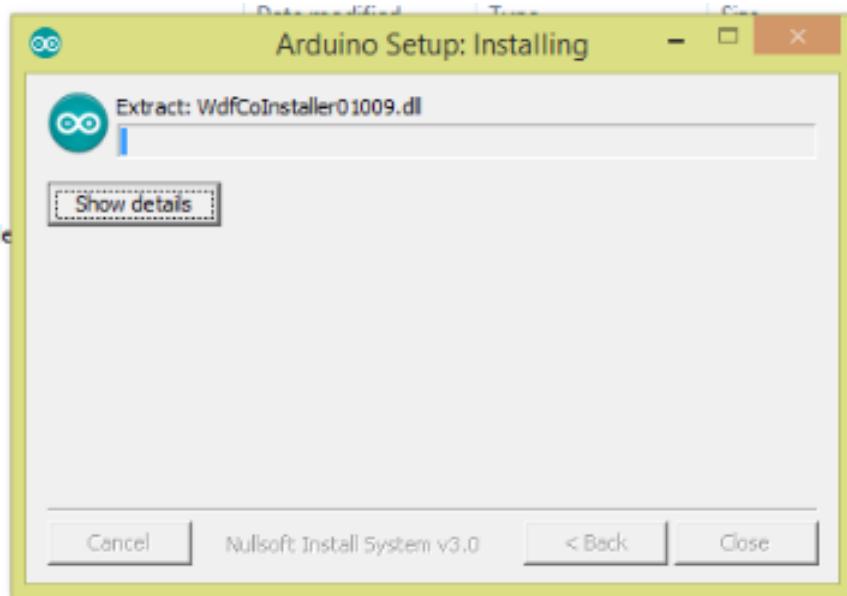
4. Lalu Next.
5. Pilih directory, lalu tekan Install.
6. Tunggu hingga proses selesai
7. Jika sudah selesai, tekan close



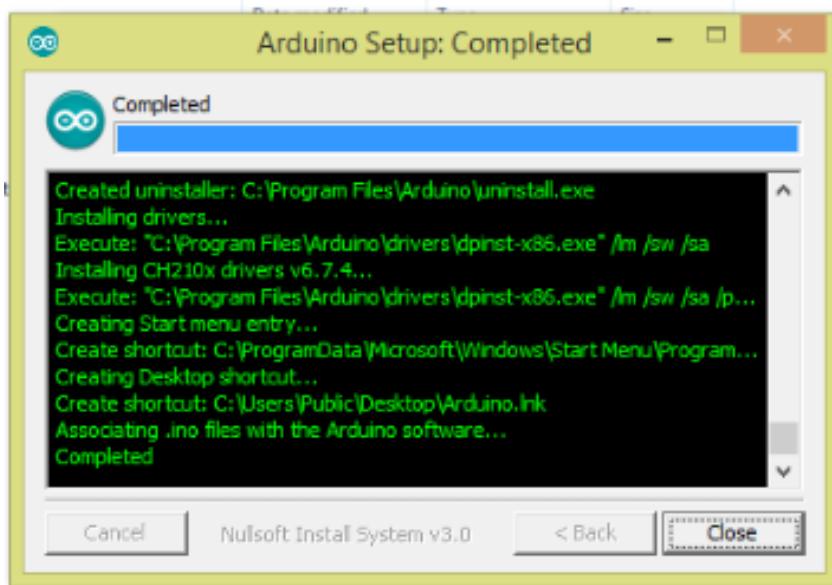
Gambar 8.8 Installation Options



Gambar 8.9 Installation Folder



Gambar 8.10 Installation



Gambar 8.11 Installing

8.2.3 Soal Nomor 3

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

Jawab :

baud rate dan port akan langsung terbaca saat arduino dicolokkan ke komputer, sistem akan membaca informasi device yang terhubung.

8.2.4 Soal Nomor 4

Jelaskan sejarah library pyserial! Jawab :

Salah satu cara untuk melakukan komunikasi melalui serial menggunakan python adalah dengan menggunakan modul pyserial. PySerial diluncurkan pada tahun 2002.

8.2.5 Soal Nomor 5

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

Jawab :

- stop() berguna untuk menghentikan program yang berjalan.
- readline() berguna untuk membaca sebuah string dari port serial.
- read(size)berguna untuk membaca jumlah byte dari port serial.
- close()berguna untuk menutup port serial.

8.2.6 Soal Nomor 6

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

Jawab :

- Perulangan

Perulangan diperlukan untuk untuk mengulangi perintah agar lebih mudah dan tidak terjadi penumpukan kodingan. Perulangan dijalankan jika kondisi benar dan akan berhenti jika kondisi salah.

- Tidak membutuhkan perulangan

Apabila perintah dijalankan sekali, kita tidak memerlukan perulangan.

8.2.7 Soal Nomor 7

Jelaskan bagaimana cara membuat fungsi yang menggunakan pyserial!

Jawab :

Definisikan nama fungsi dengan cara def namaFungsi(): lalu masukkan isi fungsi

8.3 Fanny Shafira Damayanti | 1174069

8.3.1 Pemahaman Teori

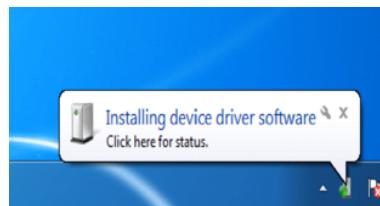
1. Fungsi Device manager di Windows dan folder /dev di Linux

Device Manager di Windows berfungsi untuk mengelola semua Hardware yang berada di Windows.

Folder /dev berisikan file device pada Linux.

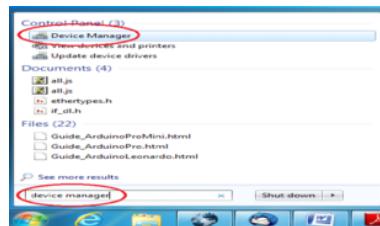
2. Langkah-langkah Instalasi driver dari arduino

- Download Software Arduino IDE
- Hubungkan port USB Arduino ke port pc.
- setelah itu pc akan mendeteksi driver baru



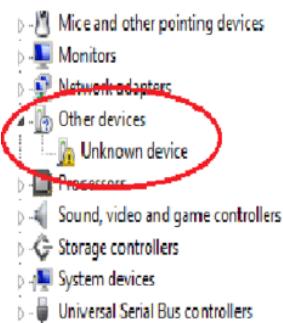
Gambar 8.12 Driver baru

- Buka device manager

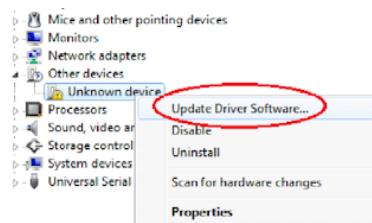


Gambar 8.13 Buka device manager

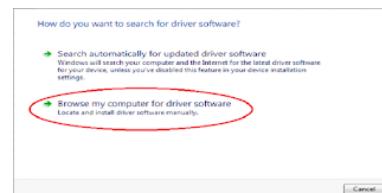
- setelah itu, cari "Unknown device"
- Klik kanan pada "Unknown device" lalu klik Update driver software
- Pilih Browse my computer for driver software
- Cari folder installan Arduino nya
- Lalu klik next
- setelah itu klik install



Gambar 8.14 klik "unknown device"



Gambar 8.15 Klik update driver software



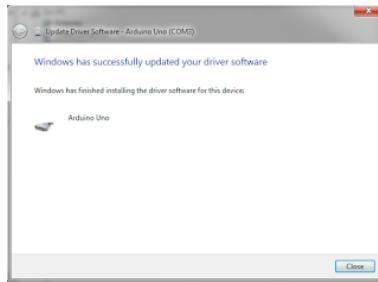
Gambar 8.16 Pilih browse



Gambar 8.17 Cari folder instalasinya



Gambar 8.18 install



Gambar 8.19 selesai

- Arduino telah berhasil di install
3. Cara membaca baudrate dan port dari komputer yang sudah terinstall driver
Untuk membaca baudrate dan port yaitu dengan cara :
- menginstall Arduino IDE
 - setelah itu buka menu serial monitor yang berada di tab tools
 - lalu akan terlihat baudrate dan port yang sedang digunakan.
4. Sejarah library pyserial PySerial merupakan sebuah library yang digunakan untuk komunikasi ke port serial terutama untuk mikrokontroller. PySerial pertama kali diluncurkan pada tahun 2002 yang makin berkembang dalam setiap versinya hingga tahun 2017 lalu.
5. Fungsi-fungsi yang di pakai dari libarary pyserial
- `stop()`
: untuk menghentikan pembacaan program
 - `serial.to_bytes(sequence)`
: berfungsi untuk mengubah sequence ke dalam bytes agar dapat dikirim ke dalam arduino.
 - `close()`
: untuk menutup port dan menghentikan pembacaan program

6. kenapa butuh perulangan dan tidak butuh perulangan 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. Sedangkan kalau tidak memakai perulangan, maka data hanya muncul satu kali, tidak berulang.

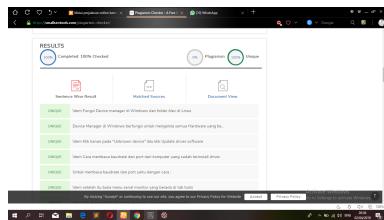
7. Cara membuat fungsi menggunakan pyserial

```

1 import serial
2
3 def testArduino():
4     ser = serial.Serial("COM5", 115200)
5     print(ser.name)
6
7 testArduino()

```

8. Scan Plagiarisme



Gambar 8.20 plagiarisme

8.4 Aulyardha Anindita | 1174054

8.4.1 Pemahaman Teori

1. Fungsi Device Manager dan folder /dev

- Fungsi Device Manager

Device manager dalam windows merupakan perluasan dari microsoft management console. Device manager mampu menampilkan seluruh hardware yang bisa diinisialisasi (dikenali) oleh windows. Device manager disini memiliki beberapa fungsi diantaranya :

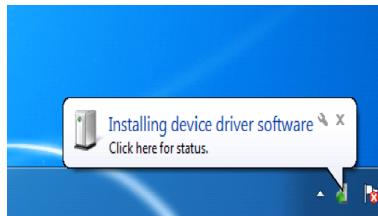
- a. Menunjukkan status suatu hardware
- b. Menunjukkan informasi detail suatu hardware
- c. Mengelola driver hardware
- d. Disable dan enable hardware
- e. Memberikan pesan error terjadinya problem status device

- Fungsi Folder /dev

Folder/dev adalah suatu representasi dari drive yang terhubung ke dalam sistem operasi linux dan sistem menganggapnya sebagai file-file direktori. Biasanya sering ditampilkan direktori seperti /dev/sdal yang mewakili Drive SATA pertama dalam sistem.

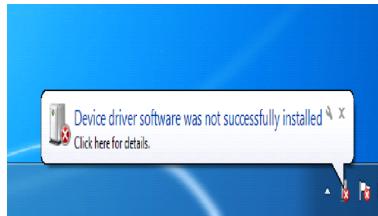
2. Langkah-langkah instalasi driver dari arduino

- Pertama, download dan instal terlebih dahulu Arduino IDE
- Hubungkan port USB Arduino UNO ke Port USB PC, maka PC akan mendekati keberadaan perangkat baru dan akan muncul pop up berupa pesan install device driver software. Seperti pada gambar berikut :



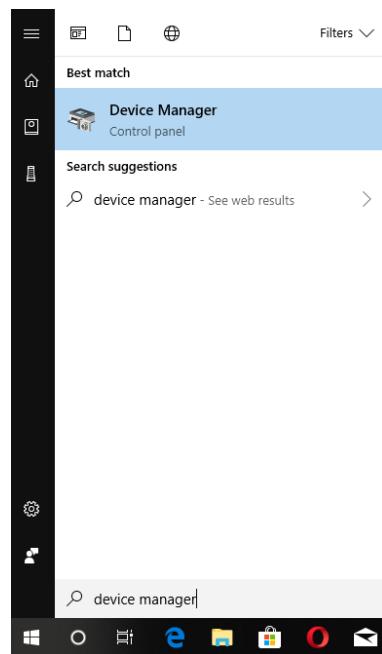
Gambar 8.21 Setup

- Dalam Windows tidak menyediakan menyediakan driver untuk Arduino Uno seperti yang terlihat pada gambar dibawah ini, sehingga proses instalasinya harus dilakukan secara manual.

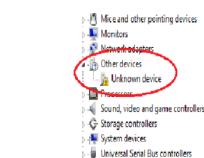


Gambar 8.22 Instalasi

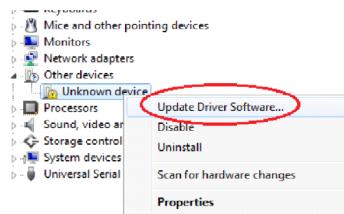
- Selanjutnya, buka start windows kemudian ketikkan device manager lalu tekan enter, maka akan muncul device manager. Klik untuk menjalankan. Seperti pada gambar berikut :
- Kemudian Carilah Unknown device pada bagian Other device, biasanya terdapat tanda seru berwarna kuning hal itu disebabkan karena penginstallan tidak berjalan dengan sempurna.
- Klik kanan pada “Unknown device” kemudian pilih Update Driver Software



Gambar 8.23 Search Device Manager

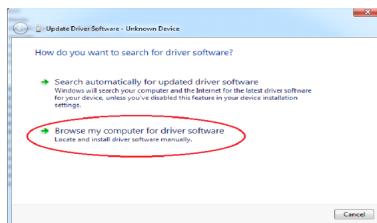


Gambar 8.24 Unknown Device



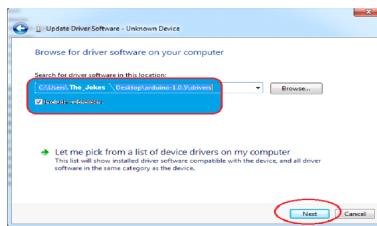
Gambar 8.25 Update Driver Software

- Pilih Browse my computer for driver software



Gambar 8.26 Mencari folder untuk driver software

- Arahkan lokasi folder ke folder arduino kemudian checkbox lalu centang include subfolders. setelah itu, Klik Next untuk melanjutkan instalasi driver.



Gambar 8.27 Lokasi folder

- Kemudian lanjutkan dengan mengklik Install pada tampilan Windows Security



Gambar 8.28 Install PAda tampilan WIndows Security

- Jika instalasi driver berhasil maka akan muncul Windows has successfully updated your driver software.
- Perhatikan dan ingat nama COM Arduino Uno, karena nama COM ini yang akan digunakan untuk meng-upload program nantinya

3. Cara Membaca Baudrate dan Port

Cara membaca baudrate yaitu dengan membuka arduino ide yang telah diinstal lalu klik serial monitor dimana iconnya seperti kaca pembesar (cari).



Gambar 8.29 Pesan Peringatan Berhasil



Gambar 8.30 Peringatan COM Arduino Uno

Cara membaca port yaitu dengan melalui device manager pada bagian port yang terdapat tulisan COMXX dimana itu adalah portnya.

4. Sejarah Library Pyserial

PySerial merupakan sebuah library yang digunakan untuk komunikasi ke port serial terutama untuk mikrokontroller. PySerial pertama kali diluncurkan pada tahun 2002 yang makin berkembang dalam setiap versinya hingga tahun 2017 lalu.

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. Fungsi-fungsi pada Library Pyserial

- `Serial()`, berfungsi untuk membuka port serial
- `Write()`, berfungsi untuk menulis data lewat port serial
- `readline(size)`, berfungsi untuk membaca sebuah string dari port serial
- `read(size)`, berfungsi untuk membaca jumlah byte dari port serial
- `close()`, berfungsi untuk menutup port serial

6. Mengapa butuh perulangan dan tidak butuh perulangan dalam membaca serial

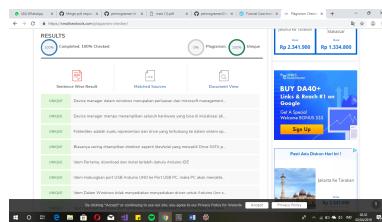
Ketika kita akan membaca serial di Arduino diperlukan perulangan yang berfungsi

untuk membaca data secara berulang kali sehingga data yang muncul banyak. Sedangkan apabila tidak membutuhkan perulangan maka Arduino hanya akan membaca data satu kali saja.

7. Cara Membuat fungsi menggunakan Pyserial

```
1 #Teori Membuat Fungsi
2 import serial
3
4 def testArduino():
5     ser = serial.Serial("COM5", 115200)
6     print(ser.name)
7
8 testArduino()
```

8. Cek Plagiarisme



Gambar 8.31 Plagiarisme

8.5 Nurul Izza Hamka | 1174062 | Teori

1. Fungsi Device Manager di Windows dan Folder / Dev di Linux Device manager ini adalah untuk menampilkan semua hardware yang bisa dikenali oleh windows, tampilan di device manager ini sudah dikelompokkan dengan rapi serta teratur,dengan adanya device manager ini akan sangat membantu dalam memanajemen setiap hardware yang ada dalam windows.

Fungsi dari device manager ini adalah seperti :

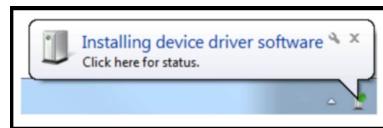
1. Mengelola setiap driver didalam hardware,
2. Mengidentifikasi konfig yang terjadi dalam hardware,
3. Menunjukka setiap informasi secara detail suatu hardware,
4. Menampilkan status dalam hardware.

Sedangka pada Linux ia melihat hardware apa saja yang yang telah terpasang melalui command dan terminal ataupun secara Visual.

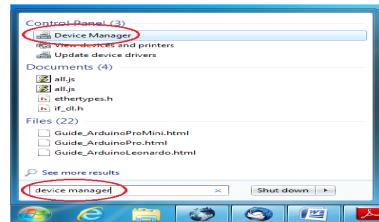
2. Langkah-Langkah Instalasi Driver dari Arduino :

Adapun langkah-langkah dalam menginstal arduino di Windows :

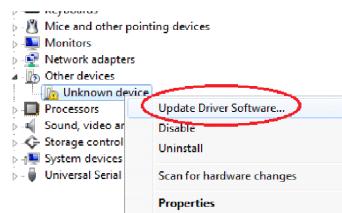
- (a) Pertama Hubungkan Port USB arduino UNO ke komputer atau Laptop,



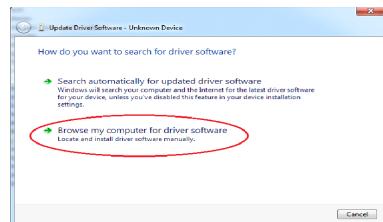
- (b) Setelah terhubung, laptop akan mengidentifikasi terlebih dahulu,
- (c) Selanjutnya windows akan menginstall di driver,pastikan device manager kita telah terinstall,



- (d) Di device manager pada bagian Ports, akan muncul nama baru yaitu Arduino UNO,
- (e) Klik kanan Arduino Uno tersebut kemudian pilih Update Driver Software,



- (f) Setelah itu Browser my Computer for driver software,lalu klik Next untuk install



- (g) Jika berhasil, berarti Arduino telah berhasil di install.

3. Cara membaca baudrate dan Port dari Komputer yang telah Terinstall Driver:
 1. Buka device manager,



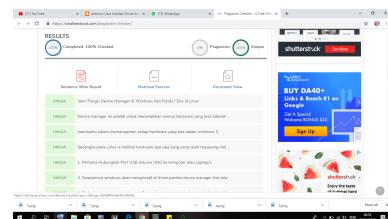
2. Pilih port (COM & LPT),
3. Kemudian double klik yang COM,
4. Pilih tab yang Port Setting, dan lihat pada Bit per Second,

Kemudian Cara Membaca Port Dari Komputer : 1. Buka device manager,
2. Pilih port (COM & LPT),
3. Selanjutnya akan muncul jika port arduino telah terbaca oleh PC.

4. Sejarah Library Pyserial: Pyserial adalah perpustakaan yang menyediakan dukungan untuk sebuah koneksi serial "RS-232" melalui berbagai perangkat yang berbeda. Pyserial ini adalah paket python yang memfasilitasi komunikasi serial antara PC dengan perangkat keras eksternal. Pyserial ini menyediakan interface yang dapat digunakan untuk berkomunikasi melalui protokol komunikasi serial.
5. Fungsi-Fungsi Yang Dipakai Dari Library Pyserial: 1. Serial,digunakan untuk membuka port serial,
2. Write(data), digunakan untuk menulis data lewat port serial,
3. Readline(),digunakan untuk membaca sebuah string dari port serial,
4. Read(size),digunakan untuk membaca jumlah byte dari port serial,
5. Close(),digunakan untuk menutup port serial.
6. Kenapa Butuh Perulangan dan Tidak Butuh Perulangan Dalam Membaca Serial:
ketika membaca serial pada arduino maka akan dibutuhkan perulangan untuk bisa membaca setiap data secara berulang, sehingga data-data yang muncul bisa banyak.
Sedangkan jika kita tidak membutuhkan perulangan maka Arduino hanya akan membaca data sekali saja.
7. Bagaimana Cara Membuat Fungsi Yang Menggunakan pyserial.
8. Plagiarisme

8.6 Tia Nur Candida | 1174086

Pemahaman Teori



8.6.1 Soal No. 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 me-manage semua perangkat keras yang terspasang dan terdeteksi dalam sistem Windows. Perangkat keras tersebut bisa berupa harddisk, kartu VGA, sound, keyboard, perangkat USB dan lain-lainnya.

Fungsi device manager antara lain :

1. Menunjukkan status mengenai suatu perangkat keras.
2. Menunjukkan informasi detail mengenai suatu perangkat keras.
3. Mengelola driver perangkat keras.
4. Menonaktifkan dan mengaktifkan perangkat keras.
5. Mengidentifikasi konflik antar perangkat keras.
6. 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.

8.6.2 Soal 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.

8.6.3 Soal No. 3

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

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.

8.6.4 Soal No. 4

Jelaskan sejarah library pyserial!

PySerial adalah paket Python yang menfasilitasi 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.

8.6.5 Soal 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.

8.6.6 Soal 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.

8.6.7 Soal No. 7

Jelaskan bagaimana cara membuat fungsi yang menggunakan pyserial!

```

1 import serial
2
3 def testArduino():
4     ser = serial.Serial("COM5", 115200)
5     print(ser.name)
6
7 testArduino()

```

Listing 8.1 Fungsi yang menggunakan pyserial.

8.6.8 Cek Plagiat

8.6.9 Kode Program

8.7 Chandra Kirana Poetra

8.7.1 Pemahaman Teori

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

Device manager adalah suatu aplikasi kecil bagian dari control panel yang ada di sistem operasi windows, aplikasi ini membuat user bisa melihat dan mengontrol hardware yang terpasang ke komputer. ketika suatu hardware tidak bekerja, hardware yang terdeteksi akan terdapat notifikasi bagi user untuk ditangani. di deivce manager ini user bisa enable atau disable hardware, melihat informasi technical, dan lain lain. device manager sudah ada di windows sejak versi 95

/dev di linux merupakan lokasi spesial atau tempat penyimpanan file file perangkat. didalamnya anda bisa menemukan hda1 dan hda2 yang merupakan partisi system, cdrom dan fd0 adalah drive CD-ROM dan juga floppy drive, dan lain lain.

2. Jelaskan langkah-langkah instalasi driver dari arduino

Download dari halaman sini dulu <https://www.arduino.cc/en/main/software> lalu ikuti saja langkah nya, langkah dibawah ini adalah langkah alternatif jika terjadi error, langkah ini akan menginstall driver arduino dari device manager

- Buka driver manager
- Buka bagian port
- Klik Kanan lalu update driver
- pilih Browse my computer
- Cari tempat driver berada, jika anda sebelumnya sudah menginstall arduino ide maka driver ada di folder instalasi arduino
- Klik next saja lalu pilih install
- Driver selesai terinstall
- Cek hasil instalasi

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

Baudrate (bits per second) adalah suatu takaran yang digunakan untuk mengukur informasi yang akan di sampaikan atau dikirim dan cara membacanya sendiri adalah dengan cara membuka aplikasi arduino IDEnya lalu mengklik icon serial monitor. Port sendiri bisa kita lihat di bagian device manager pada opsi port, tinggal dibuka saja dropdownnya maka akan tampil port number berapanya

4. Jelaskan sejarah library pyserial

pySerial adalah modul yang digunakan untuk mengakses port, menyediakan backend untuk python pada platform windwos, OSX, linux, BSD dan lain lain,

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

- Serial()

Berfungsi untuk mendefinisikan port, diikuti dengan parameter lainnya seperti baudrate, bytesize dan lain lain

- open()

Berfungsi untuk membuka port

- close()

Berfungsi untuk Menutup port

- write(data)

Berfungsi untuk menulis data berdasarkan parameter

- flush()

Operasi Flush memaksa data file untuk dikirim ke file cache di ram dan disana akan dialihkan ke sistem operasi untuk dikirimkan ke disk datayang akan di simpan

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

Perulangan sendiri merupakan kode/perintah yang digunakan untuk melakukan suatu kegiatan/tujuan yang sama secara berulang ulang. for dan while adalah perulangan dalam pyhton. For dan while merupakan perulangan namun memiliki fungsi yang berbeda, for menggunakan perhitungan dalam penggunaannya jadi kode akan dieksekusi sesuai dengan jumlah yang telah ditentukan, sedangkan while akan terus mengulang perintah-perintahnya ketika kondisi masih sesuai dengan apa yang ditulis di kondisi Mengapa diperlukan perulangan? Karena kode program akan menjadi lebih sederhana dibandingkan dengan cara manual satu persatu dan juga lebih efficient dengan meminimalisir baris kode

7. Jelaskan bagaimana cara membuat fungsi yang menggunakan pyserial

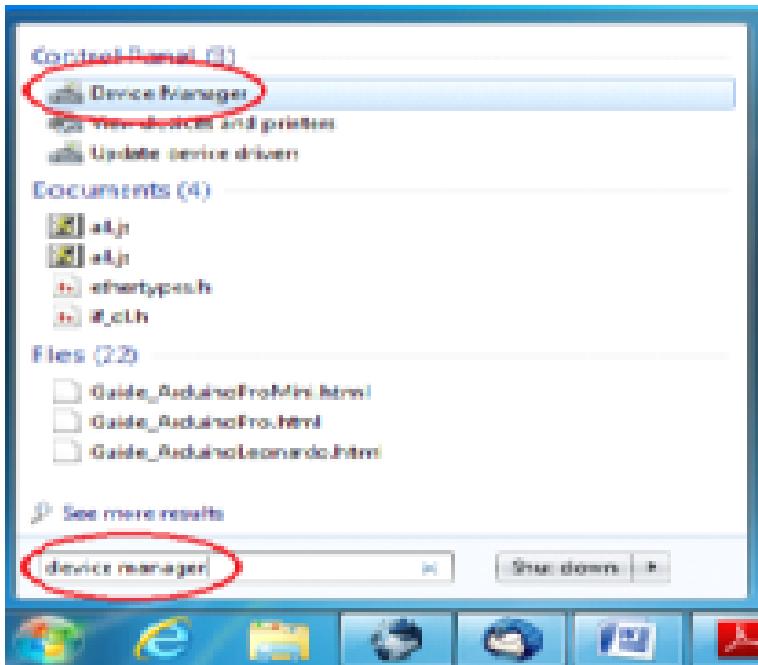
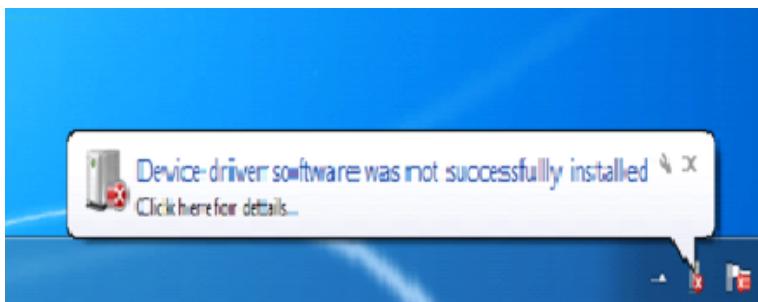
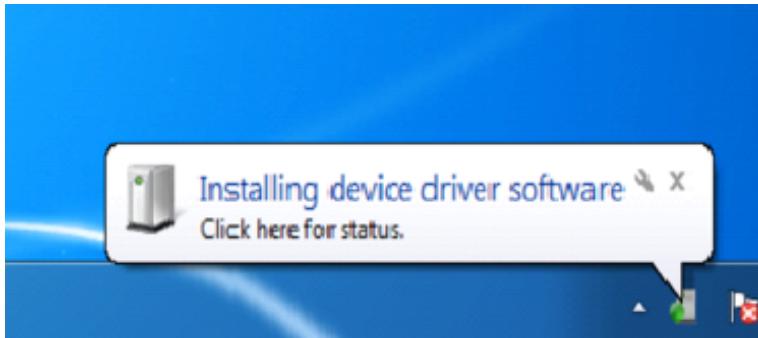
Membuat fungsi sama seperti sebelumnya namun method dari pyserial dimasukkan kedalam fungsi dan dipanggil fungsi yang kita buat tadi

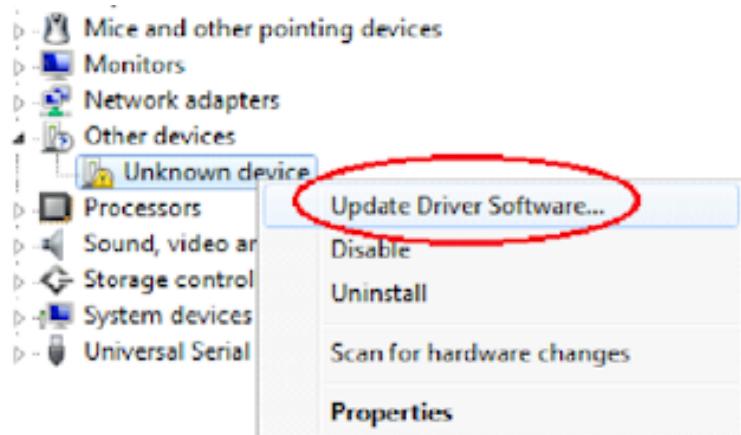
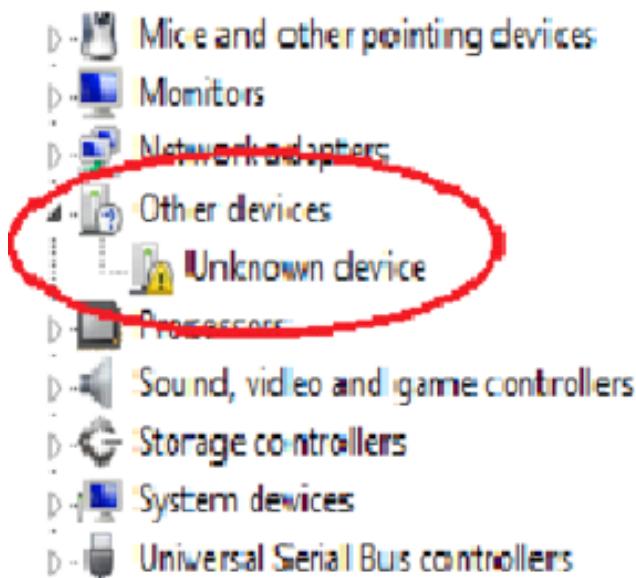
```

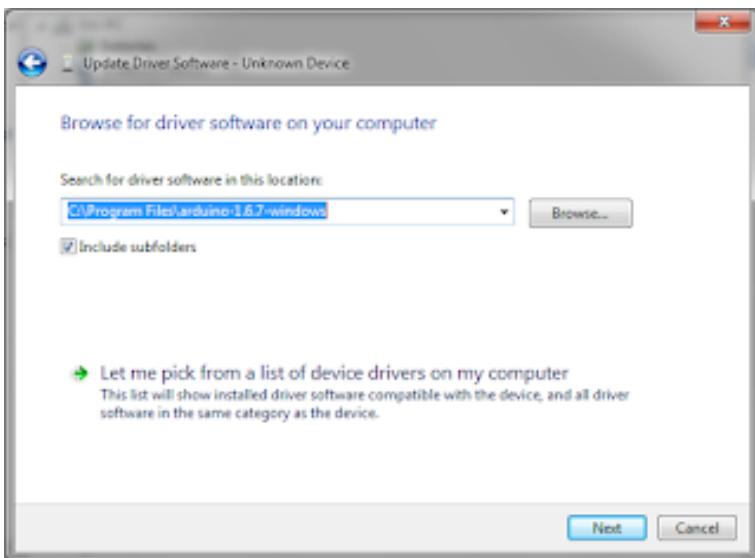
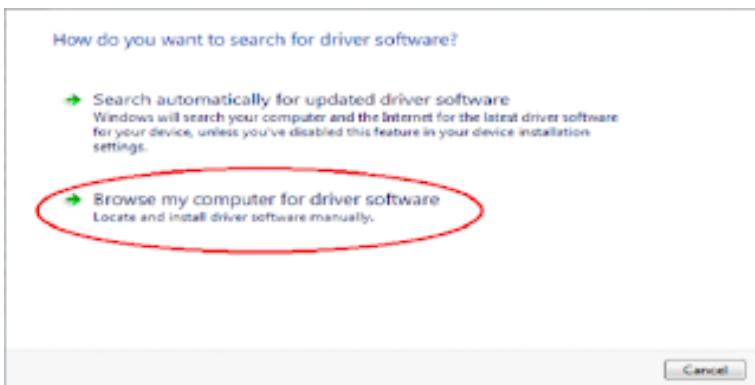
1 import serial
2
3 def bacaserialarduino():
4     variableserial = serial.Serial("COM1", 9600)
5     baca = variableserial.readline()
6     print(baca)
7
8 bacaserialarduino()

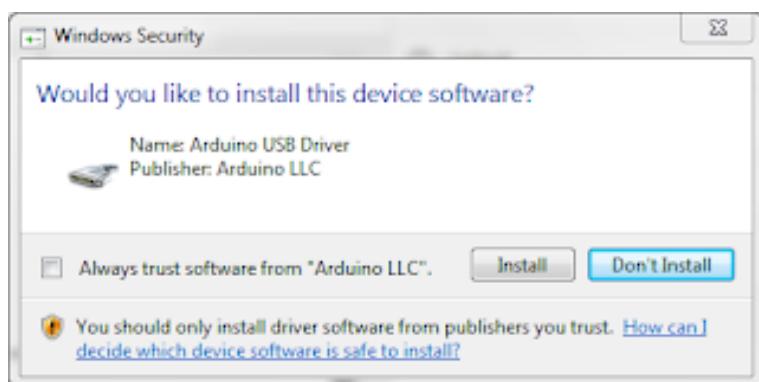
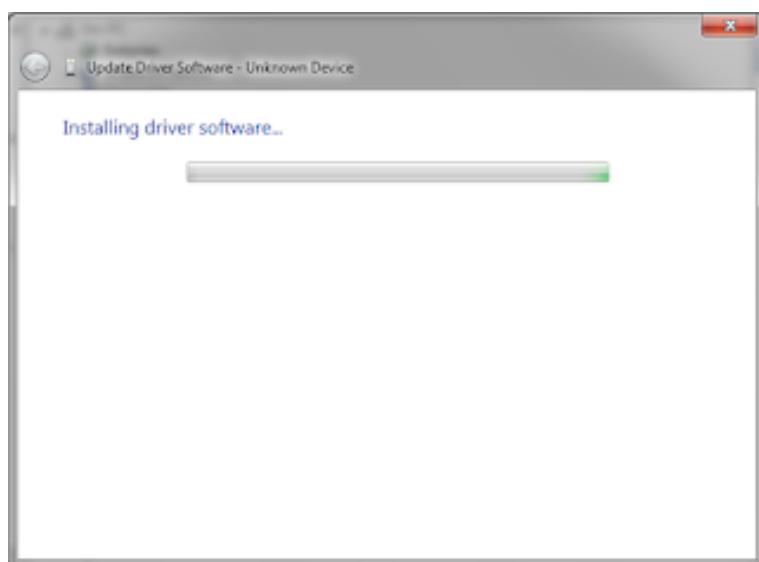
```

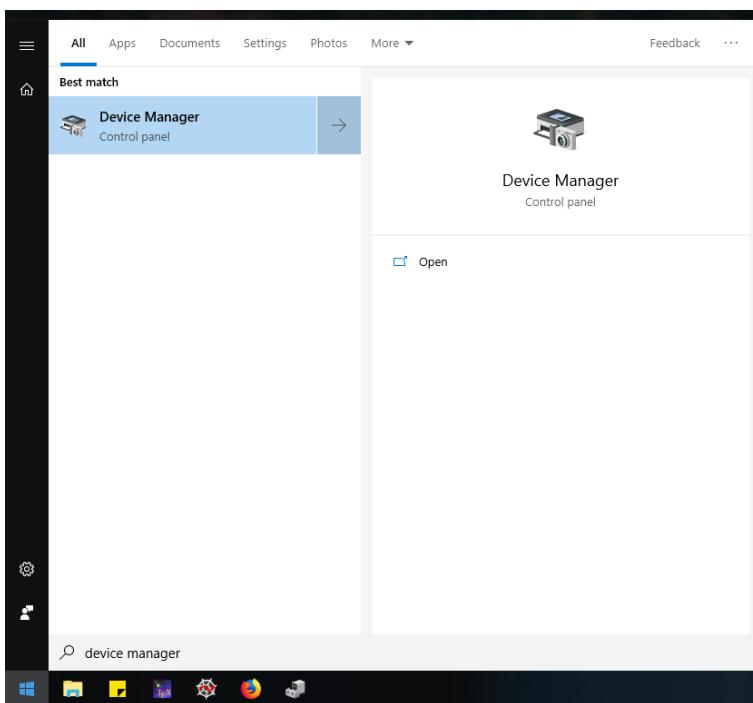
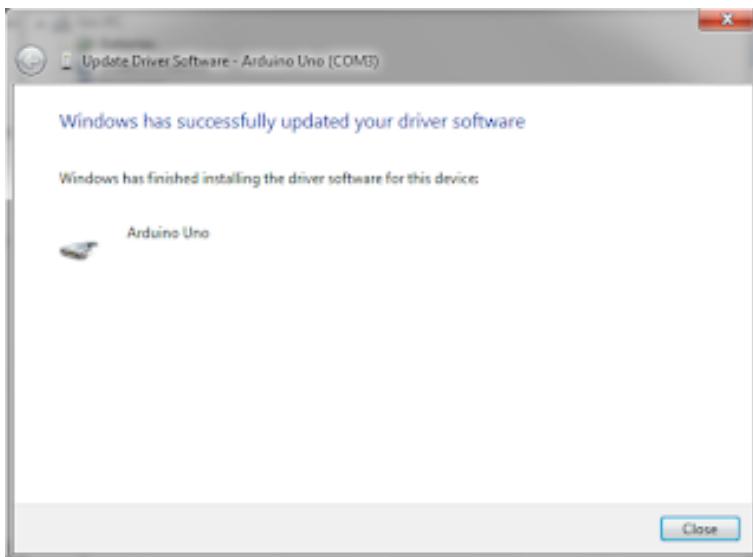
8. Scan Plagiarisme

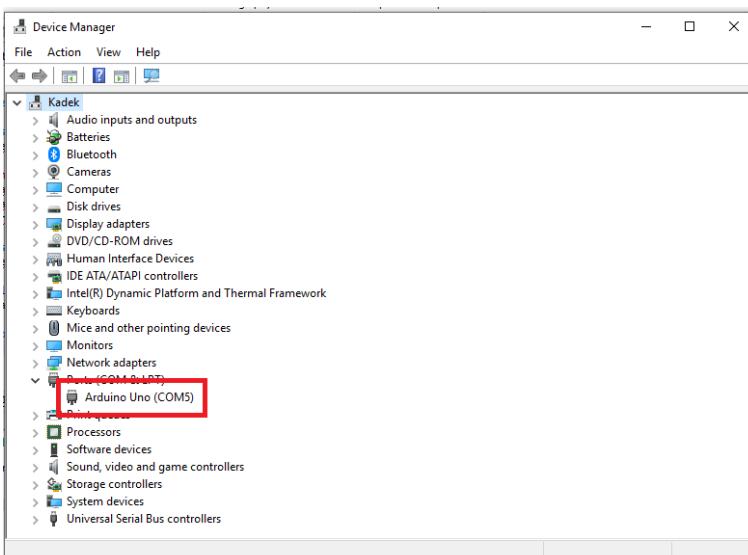
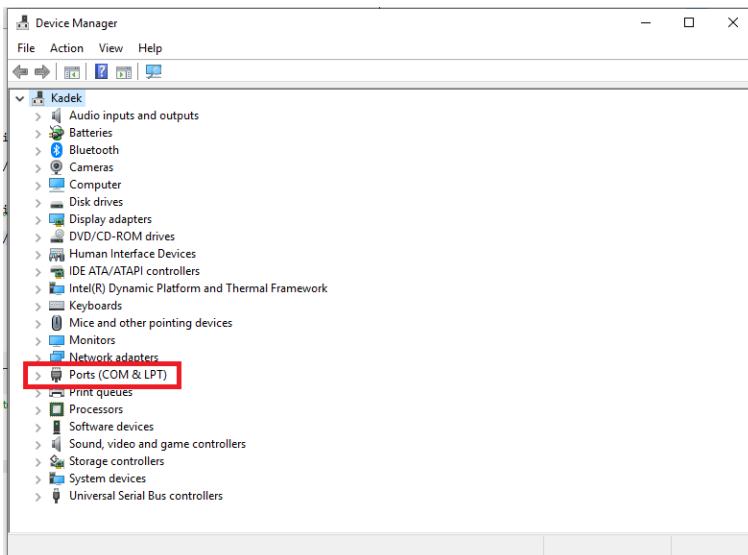


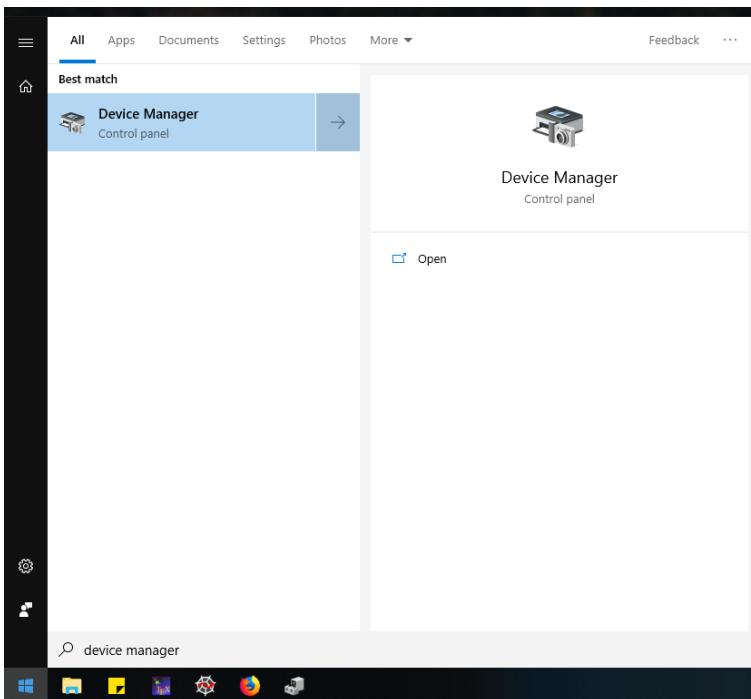
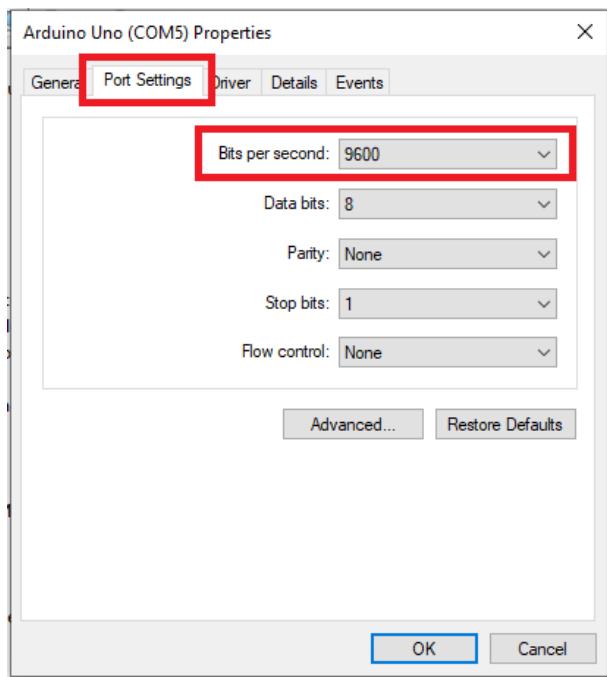


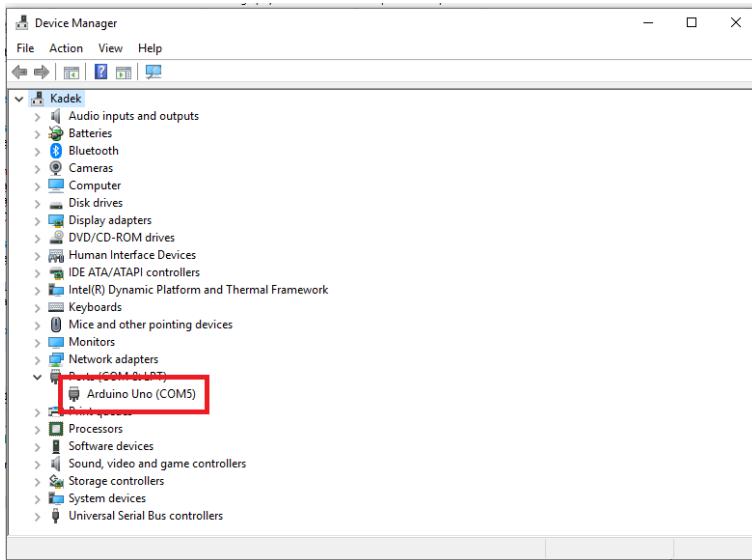
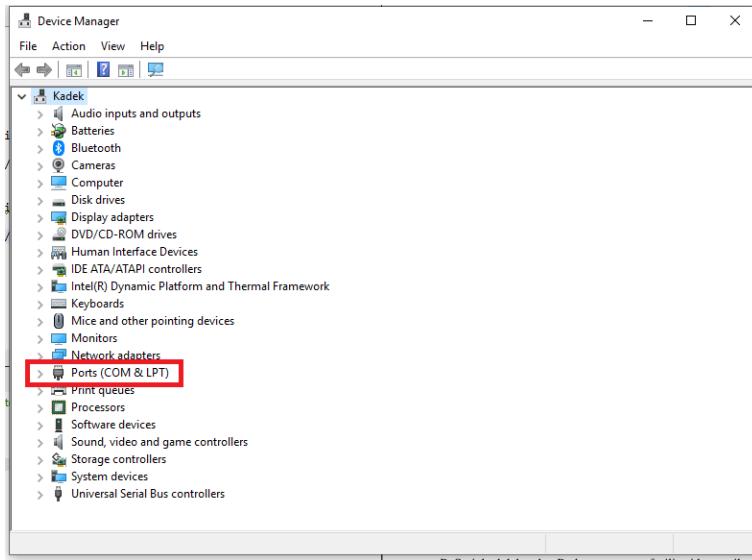






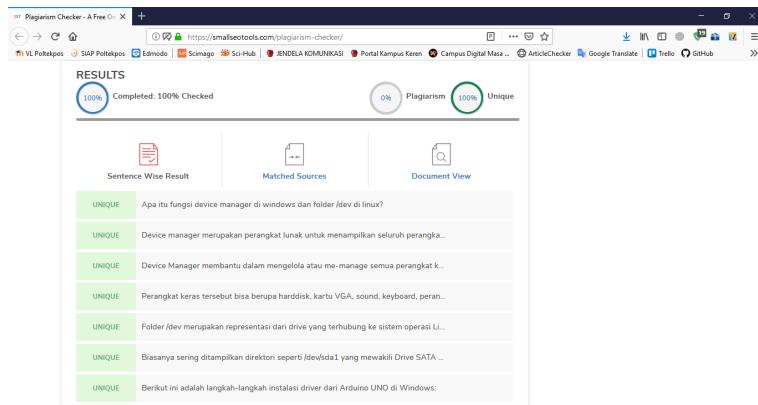






```
Console 2/A
In [3]: runfile('E:/Kuliah/Semester 4/Pemrograman 3/CobaGit/src/5/1174006/Teori/1174006.py', wdir='E:/Kuliah/Semester 4/Pemrograman 3/CobaGit/src/5/1174006/Teori')
COM5
```

Gambar 8.32 Hasil pembuatan fungsi pyserial.



Gambar 8.33 Hasil cek plagiat.

The screenshot shows the Spyder Python IDE interface. On the left, the code editor displays a file named '1174086.py' containing the following code:

```

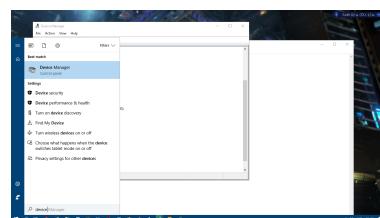
1 import serial
2
3 def testarduino():
4     ser = serial.Serial("COM5", 115200)
5     print(ser.name)
6
7 testarduino()
    
```

On the right, there are two terminal windows. The top one is titled 'Variable explorer' and is empty. The bottom one is titled 'Python console' and shows the output of running the script:

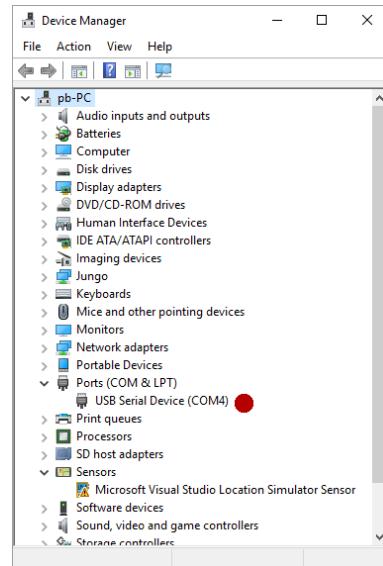
```

In [3]: runfile('E:/Kuliah/Semester 4/CobaGit/src/5/1174086/Teror/1174086.py', wdir='E:/Kuliah/Semester 4/Peogramman 3/CobaGit/src/5/1174086/Teror')
COM5
In [4]:
    
```

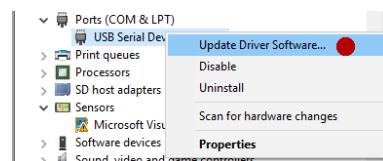
Gambar 8.34 Kode program file 1174086.py.



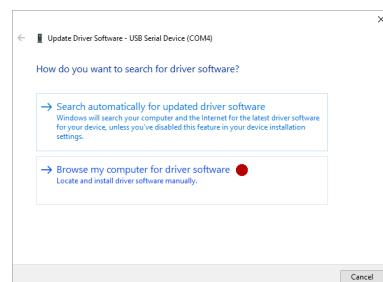
Gambar 8.35 Driver Manager



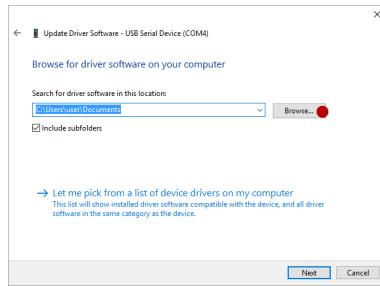
Gambar 8.36 Port



Gambar 8.37 Instalasi



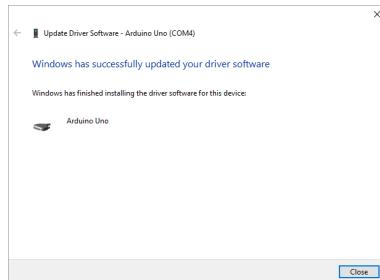
Gambar 8.38 browse my computer



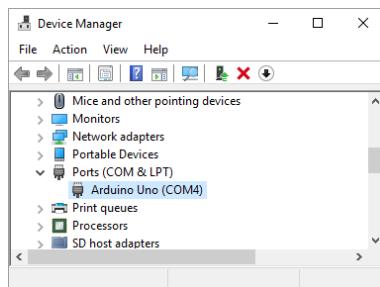
Gambar 8.39 Driver Arduino



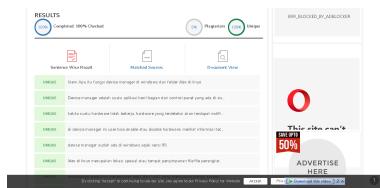
Gambar 8.40 Install driver



Gambar 8.41 Selesai



Gambar 8.42 Result



Gambar 8.43 Plagiarisme

BAB 9

PRAKTEK KOMUNIKASI PERANGKAT KERAS

9.1 D. Irga B. Naufal Fakhri

Kode Arduino untuk pembacaan temprature dan humidity menggunakan sensor DHT1:

```
1 #include <dht.h>
2
3 dht DHT;
4
5 #define DHT11_PIN 6
6
7 void setup(){
8     Serial.begin(9600);
9     Serial.println("Ready");
10 }
11
12 void loop()
13 {
14     int chk = DHT.read11(DHT11_PIN);
15     Serial.print("Temperature = ");
16     Serial.println(DHT.temperature);
17     Serial.print("Humidity = ");
18     Serial.println(DHT.humidity);
```

```

19     delay(2000);
20 }
```

9.1.1 Soal 1

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

```

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

Gambar 9.1 Membaca Serial tanpa loop

9.1.2 Soal 2

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

```

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

Gambar 9.2 Membaca Serial dengan loop

9.1.3 Soal 3

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

```

1 import csv
2 def write():
```

```

3   with open("1174066.csv", mode='w') as nama_file:
4       tulis_file = csv.writer(nama_file, delimiter=',', quotechar='''',
5           quoting=csv.QUOTE_MINIMAL)
6       ser = serial.Serial("COM3", 9600)
7       read = ser.readline().decode("utf-8").strip('\n').strip('\'r')
8       tulis_file.writerow([read])
9   write()

```

Listing 9.1 "Kode python"

```

1 Ready
2
3 Temperature = 30
4
5 Humidity = 66
6
7 Temperature = 30
8
9 Humidity = 66
10
11 Temperature = 30
12
13 Humidity = 66
14
15 Temperature = 30
16
17 Humidity = 66

```

Listing 9.2 "Data yang telah ditulis ke file csv"

9.1.4 Soal 4

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

```

1 import csv
2
3 def read(namafайл):
4     with open(namafайл, mode='r') as csv_file:
5         reader = csv.DictReader(csv_file)
6         for row in reader:
7             print(row['Ready'])
8
9 read("1174066.csv")

```

9.1.5 Ketrampilan Penanganan Error

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

- Syntax Errors

Syntax Errors adalah kesalahan pada penulisan syntax atau kode. Solusinya adalah memperbaiki penulisan syntax atau kode

- Zero Division Error

ZeroDivisionError adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0). Solusinya adalah tidak membagi suatu yang hasilnya nol.

- Name Error

NameError adalah exception saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi atau tidak ada. Solusinya adalah memastikan variabel atau function yang akan dipanggil ada didalam program atau tidak salah mengetikannya.

- Type Error

TypeError adalah exception saat melakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkoversi variabelnya sesuai dengan tipe data sesuai dengan yang akan digunakan.

```

1 import serial
2
3 def PenangananError():
4     try:
5         ser = serial.Serial('COM3', 9600)
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7     except SyntaxError:
8         print("Ada kesalahan dalam penulisan Syntax")
9     except NameError:
10        print("Variable yang dimasukkan tidak ada")
11    except TypeError:
12        print("Ada yang salah pada type data")
13    except:
14        print("Sedang terjadi sebuah kesalahan")
15
16 PenangananError()

```

9.2 Tia Nur Candida

Ketrampilan Pemrograman

9.2.1 No. 1

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendapatkan data langsung dari arduino!

```

1 import serial
2
3 def getData():

```

```

4     ser = serial.Serial('COM5',9600)
5     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
6
7 getData()

```

Listing 9.3 Fungsi untuk mendapatkan data dari Arduino.

9.2.2 No. 2

Buatlah fungsi (file terpisah/library dengan nama NPMSave.py) untuk mendapatkan data langsung dari arduino dengan looping!

```

1 import serial
2
3 def getDataLoop():
4     ser = serial.Serial('COM5',9600)
5     while (1):
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7
8 getDataLoop()

```

Listing 9.4 Fungsi untuk mendapatkan data langsung dari Arduino dengan looping.

9.2.3 No. 3

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.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('COM5',9600)
6     with open('praktek.csv', mode='w') as csv_file:
7         fieldnames = ['jarak']
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({'jarak': data})
14
15 writeCsv()

```

Listing 9.5 Fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.

9.2.4 No. 4

Buatlah fungsi (file terpisah/library dengan nama NPMcsv.py) untuk membaca file csv hasil arduino dan mengembalikan ke fungsi!

```

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

```

Listing 9.6 Fungsi untuk membaca file CSV hasil Arduino dan mengembalikan fungsi.

9.2.5 Kode Program Praktek

9.2.6 Cek Plagiat Praktek

Ketrampilan Penanganan Error

9.2.7 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 mengkoversi varibelnya 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('COM5', 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 9.7 Fungsi untuk menanggulangi error menggunakan Try Except.

9.2.8 Kode Program Penanganan Error

9.2.9 Cek Plagiat Penanganan Error

9.3 Fanny Shafira Damayanti | 1174069

9.3.1 Keterampilan Pemrograman

1. Fungsi untuk mendapatkan data langsung dari arduino

```

1 import serial
2
3 def getData():
4     ser = serial.Serial('COM5',9600)
5     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
6
7 getData()
8
9 import serial
10 import csv
11
12 def writeCsv():
13     ser = serial.Serial('COM5',9600)
14     with open('praktek.csv', mode='w') as csv_file:
15         fieldnames = ['jarak']
16         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
17
18         writer.writeheader()
19         while (1):
20             data = ser.readline().decode("utf-8").strip('\n').
21             strip('\r')
22             writer.writerow({'jarak': data})
23
24 writeCsv()

```

2. Fungsi untuk mendapatkan data langsung dari arduino dengan looping

```

1 import serial
2
3 def getDataLoop():
4     ser = serial.Serial('COM5',9600)
5     while (1):

```

```

6     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7
8 getDataLoop()

```

3. Fungsi 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('COM5', 9600)
6     with open('praktek.csv', mode='w') as csv_file:
7         fieldnames = ['jarak']
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({'jarak': data})
15
16 writeCsv()

```

4. Fungsi untuk membaca file csv hasil arduino dan mengembalikan ke fungsi

```

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

```

9.3.2 Keterampilan Penanganan Error

1. Peringatan error yang ada di praktek kali ini, dan fungsi try except untuk menanggulangi error tersebut

- 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 mengkoversi varibelnya sesuai dengan tipe data yang akan digunakan.

```

1 import serial
2
3 def tryExceptError():
4     try:
5         ser = serial.Serial('COM5', 9600)
6         print(ser.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()

```

9.4 Aulyardha Anindita | 1174054

9.4.1 Keterampilan Pemrograman

1. Jawaban Soal No. 1

```

1 import serial
2
3 def getData():
4     ser = serial.Serial('COM5', 9600)
5     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
6
7 getData()

```

2. Jawaban Soal No. 2

```

1 import serial
2
3 def getDataLoop():
4     ser = serial.Serial('COM5', 9600)
5     while (1):
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7
8 getDataLoop()

```

3. Jawaban Soal No. 3

```

1 import serial
2 import csv
3
4 def writeCsv():
5     ser = serial.Serial('COM5', 9600)
6     with open('praktek.csv', mode='w') as csv_file:
7         fieldnames = ['jarak']
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({'jarak': data})
15
16 writeCsv()

```

4. Jawaban Soal No. 4

```

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

```

9.4.2 Keterampilan Penanganan Error

Peringatan error di praktek keempat ini, yaitu:

- Syntax Errors Syntax Errors adalah keadaan dimana pada kode python mngalami kesalahan dalam penulisan. Untuk mengatasinya yaitu dengan memperbaiki penulisan kode yang salah
- Name Error NameError adalah suatu keadaan atau exception yang terjadi ketika kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Untuk mengatasinya yaitu dengan memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.
- Type Error TypeError adalah suatu keadaan atau exception yang akan terjadi apabila pada saat dilakukannya eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Untuk mengatasinya yaitu dengan mengoversi varibelnya sesuai dengan tipe data yang akan digunakan.

Fungsi yang menggunakan try except

```

1 import serial
2
3 def tryExceptError():

```

```

1 try :
2     ser = serial.Serial('COM5',9600)
3     print(sre.readline().decode("utf-8").strip('\n').strip('\r'))
4 except SyntaxError:
5     print("Kesalahan penulisan syntax")
6 except NameError:
7     print("Variable tersebut tidak ada")
8 except TypeError:
9     print("Tipe data salah")
10 except:
11     print("Terjadi sebuah kesalahan")
12
13 tryExceptError()
14
15
16

```

9.5 Nurul Izza Hamka | 1174062

9.5.1 Pemrograman

- Fungsi Untuk Mendapatkan Langsung Data Dari Arduino:

```

1 import serial
2
3 def getData():
4     ser = serial.Serial('COM5',9600)
5     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
6
7 getData()

```

- Fungsi Untuk Mendapatkan Langsung Data Dari Arduino Dengan Looping:

```

1 import serial
2
3 def getDataLoop():
4     ser = serial.Serial('COM5',9600)
5     while (1):
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7
8 getDataLoop()

```

- Fungsi Untuk Mendapatkan Data Dari Arduino Dan Langsung Dari Arduino Ditulis Dalam Csv :

```

1 import serial
2 import csv
3
4 def writeCsv():
5     ser = serial.Serial('COM5',9600)
6     with open('praktek.csv', mode='w') as csv_file:
7         fieldnames = ['jarak']
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({'jarak': data})
15 writeCsv()

```

4. Fungsi Untuk Membaca File Csv Hasil Arduino Dan Mengembalikan Fungsi:

```

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

```

9.5.2 Penanganan Error

5. Fungsi Penanganan Error Menggunakan Try Except:

```

1 import serial
2
3 def tryExceptError():
4     try:
5         ser = serial.Serial('COM5', 9600)
6         print(ser.readline().decode("utf-8").strip('\n').strip('\
7         r'))
8     except SyntaxError:
9         print("Kesalahan penulisan syntax")
10    except NameError:
11        print("Variable tersebut tidak ada")
12    except TypeError:
13        print("Tipe data salah")
14    except:
15        print("Terjadi sebuah kesalahan")
16 tryExceptError()

```

9.6 Chandra Kirana Poetra

9.6.1 Soal 1

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

```

1 import serial
2

```

```

3 def ambildata():
4     serX = serial.Serial('COM5',9600)
5     print(serX.readline().decode("utf-8").strip('\n').strip('\r'))
6
7 ambildata()

```

9.6.2 Soal 2

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

```

1 import serial
2
3 def ambildata():
4     serX = serial.Serial('COM5',9600)
5     while (1):
6         print(serX.readline().decode("utf-8").strip('\n').strip('\r'))
7
8 ambildata()

```

9.6.3 Soal 3

Buatlah fungsi (file terpisah/library 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 menuliscsv():
5     serX = serial.Serial('COM5',9600)
6     with open('praktek.csv', mode='w') as csv_file:
7         fieldnames = ['contoh']
8         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
9
10        writer.writeheader()
11        while (1):
12            data = serX.readline().decode("utf-8").strip('\n').strip(
13                '\r')
14            writer.writerow({'contoh': data})
15
menuliscsv()

```

9.6.4 Soal 4

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

```

1 import csv
2
3 def bacafilecsv():
4     with open('praktek.csv', mode='r') as csv_file:

```

```

5     reader = csv.DictReader(csv_file)
6     for row in reader:
7         print(row['contoh'])
8
9 bacafilecsv()

```

9.6.5 Ketampilan Penanganan Error

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.

- Syntax Errors

Syntax Errors adalah kesalahan yang paling umum yaitu terletak pada penulisan syntax atau kode di dalam python. Solusinya adalah dengan mengecek ulang dan memperbaiki penulisan syntax atau kode

- Name Error

NameError adalah suatu exception yang terjadi ketika program python memanggil sesuatu yang tidak ada atau bisa dikatakan belum didefinisikan. cara mengatasinya adalah dengan memastikan kembali bahwa variabel atau function yang akan dipanggil ada didalam program sudah didefinisikan dan mengecek ulang penulisan dalam pemanggilan function atau variable.

- Type Error

TypeError adalah exception muncul ketika saat program python mendeteksi adanya kesalahan penggunaan tipe data dalam program. Solusinya adalah konversi terdahulu variable yang akan digunakan ke format yang tepat .

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Apr  3 21:29:47 2019
4
5 @author: Chandra Kirana Poetra
6 """
7
8 import serial
9
10 def tryExceptError():
11     try:
12         ser = serial.Serial('COM5', 9600)
13         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
14     except SyntaxError:
15         print("Kesalahan penulisan syntax")
16     except NameError:
17         print("Variable tersebut tidak ada")
18     except TypeError:
19         print("Tipe data salah")
20     except:

```

```
21     print("Terjadi sebuah kesalahan")
22
23 tryExceptError()
```

9.7 Difa Al Fansha

9.7.1 Nomor 1

Buatlah fungsi (file terpisah/library dengan nama NPM_realtim.e.py) untuk mendapatkan data langsung dari arduino!

Jawab :

```
1 # Jawaban Nomor 1
2 import serial
3
4 def getData():
5     ser = serial.Serial('COM5',9600)
6     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7
8 getData()
```

9.7.2 Nomor 2

Buatlah fungsi (file terpisah/library dengan nama NPM_save.py) untuk mendapatkan data langsung dari arduino dengan looping!

Jawab :

```
1 # Jawaban 2
2 import serial
3
4 def getDataLoop():
5     ser = serial.Serial('COM5',9600)
6     while (1):
7         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
8
9 getDataLoop()
```

9.7.3 Nomor 3

Buatlah fungsi (file terpisah/library dengan nama NPM_realtim.e.py) untuk mendapatkan data dari arduino dan langsung ditulis kedalam file csv!

Jawab :

```
1 # Jawaban nomor 3
2 import csv
```

```

3 def writeCsv():
4     ser = serial.Serial('COM5', 9600)
5     with open('ujicoba.csv', mode='w') as csv_file:
6         fieldnames = ['angka']
7         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
8
9         writer.writeheader()
10        while (1):
11            data = ser.readline().decode("utf-8").strip('\n').strip(
12            '\r')
13            writer.writerow({'angka': data})
14
15 writeCsv()

```

9.7.4 Nomor 4

Buatlah fungsi (file terpisah/library dengan nama NPM_csv.py) untuk membaca file csv hasil arduino dan mengembalikan ke fungsi!

Jawab :

```

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

```

```
In [6]: runfile('E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek/1174006realtime.py', wdir='E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek')
241
```

Gambar 9.3 Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino.

```
In [7]: runfile('E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek/1174006save.py', wdir='E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek')
241
198
240
199
239
200
239
200
239
200
239
200
239
```

Gambar 9.4 Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dengan looping.

```
praktek.csv - Notepad
File Edit Format View Help
jarak
5
6
6
5
6
5
5
5
5
5
5
```

Gambar 9.5 Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.

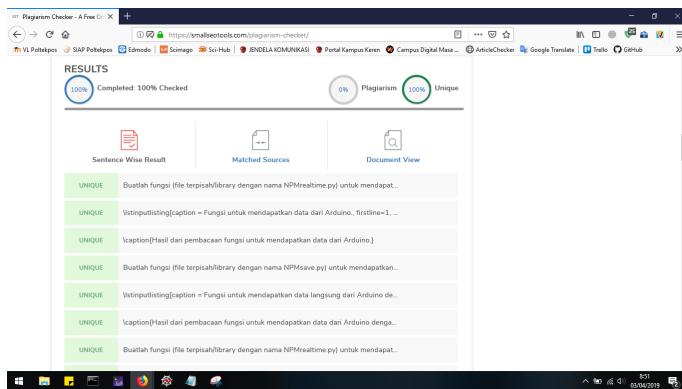
The screenshot shows the IPython console interface. In the top bar, it says "IPython console" and has tabs for "Console 1/A" and "Console 1/B". The main area displays the following Python code and its execution results:

```
In [12]: runfile('E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek/1174006csv.py', wdir='E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek')
[5
6
6
5
5
5
5
5
5
5]
```

Gambar 9.6 Hasil dari pembacaan fungsi untuk membaca file csv hasil arduino dan mengembalikan fungsi.

The screenshot shows three separate code cells in a Jupyter Notebook:

- Cell 1:** A function to read data from a serial port (COM5) and write it to a CSV file. It includes code to read the header and data, and to handle the CSV writer.
- Cell 2:** A function to read data from a serial port (COM5) and print the distance value.
- Cell 3:** A function to read data from a CSV file named "praktek.csv" and print the distance values.



The screenshot shows a Jupyter Notebook interface with a Python console tab. The console output shows the following error message:

```
1174006.py', wdir='E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek')
Terjadi sebuah kesalahan
```

In [14]:

Gambar 9.7 Hasil pembacaan fungsi untuk menanggulangi error menggunakan Try Except.

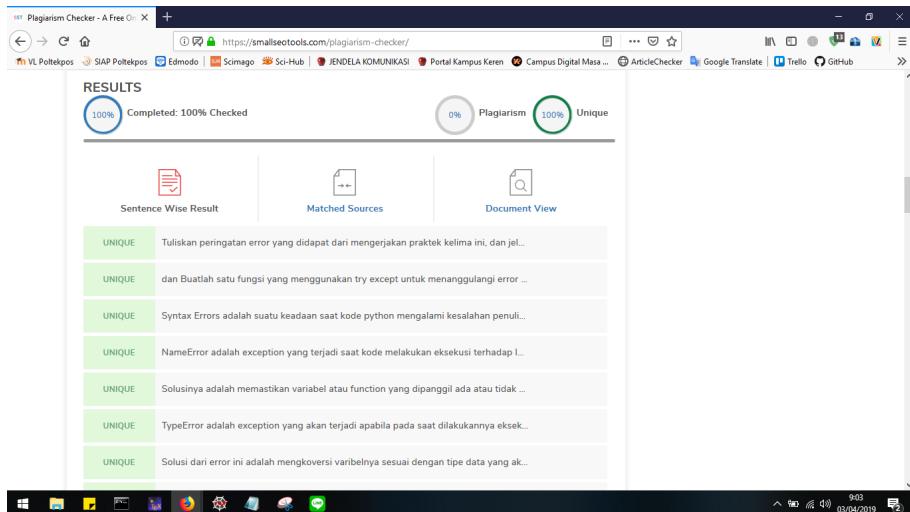
The screenshot shows a Jupyter Notebook interface with a code cell containing Python code. The code uses try-except blocks to handle errors. The code cell has several syntax errors highlighted with red triangles. The variable explorer table shows the following values:

Name	Type	Size	Value

The code cell output shows the same error message as in the previous screenshot:

```
1174006.py', wdir='E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek')
Terjadi sebuah kesalahan
```

In [14]:



Gambar 9.8 Cek Plagiarisme punya chandra

BAB 10

MATPLOTLIB

10.1 D. Irga B. Naufal Fakhri

10.1.1 Pemahaman Teori

1. Apa itu fungsi library matplotlib

Matplotlib adalah modul python untuk menggambar plot 2D dengan kualitas tinggi. matplotlib dapat digunakan dalam script python, interpreter python dan ipython, server, dan 6 GUI toolkit. matplotlib berusaha untuk membuat segalanya jadi mudah, dan yang tadinya seperti tidak menjadi mungkin untuk dilakukan. Dengan matplotlib, Anda dapat membuat plots, histograms, spectra, bar charts, errorcharts, scatterplots, dan masih banyak lagi.

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

- (a) pertama import dulu library matplotlib lalu beri alias plt

```
1 import matplotlib.pyplot as plt
```

- (b) Kemudian buat variable array x dengan isi terserah anda

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

- (c) Lalu buat variable array y juga dengan isi terserah anda yang penting jumlahnya sama dengan variable X

```
1 y = [13, 17, 19, 33]
```

- (d) Kemudian buat plot berisikan variable x dan y pada modul plt

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

- (e) Yang terakhir, munculkan plot yang telah kita buat dengan fungsi show()

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

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

Perbedaan pada fungsi plot adalah pada bentuk gambar grafik yang dihasilkan pada program dan jenis jenis grafik yang ada pada plot adalah:

- Plot

Grafik yang dihasilkan oleh plot ini adalah sebuah garis (line)

```
1 import matplotlib.pyplot as plt
2 x = [1, 2, 3, 4]
3 y = [13, 17, 19, 33]
4 plt.plot(x, y)
5 plt.show()
```

- Bar

Grafik yang dihasilkan oleh bar adalah sebuah bentuk grafik batang (bar) dan cara penggunaan bar adalah menggunakan fungsi bar(variable x, variable y)

```
1 import matplotlib.pyplot as plt
2 x = [1, 2, 3, 4]
3 y = [13, 17, 19, 33]
4 plt.bar(x, y)
5 plt.show()
```

- Histogram

Dalam penggunaan histogram yaitu menggunakan .hist(variable x, variable y) dengan variable x berisikan data dan variable y berisikan keliapan yang akan dimunculkan pada histogram

```
1 import matplotlib.pyplot as plt
2 x =
[21,22,23,4,5,6,77,8,9,10,31,32,33,34,35,36,37,18,49,50,100]
3 plt.hist(x, 5)
4 plt.show()
```

- Scatter

Grafik yang dihasilkan oleh scatter adalah diagram titik dan cara penggunaannya menggunakan .scatter(variable x, variable y)

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3 x = np.random.randn(100)
4 y = np.random.randn(100)
5 x1 = np.random.randn(100)
6 y1 = np.random.randn(100)
7 plt.scatter(x,y, label='Angka Random 1', color='R')
8 plt.scatter(x1,y1, label='Angka Random 2', color='B')
9 plt.xlabel('Random 1')
10 plt.ylabel('Random 2')
11 plt.title('Diagram Titik')
12 plt.legend()
13 plt.show()
```

- Stack Plot

Grafik yang dihasilkan oleh stack plot ini hampir sama seperti diagram line, hanya hasil datanya disatukan semua keatasnya dan cara penggunaannya adalah menggunakan .stackplot(variable, variable, variable)

```

1 hari = [1,2,3,4,5,6,7]
2 tidur =[7,8,6,8,7,9,10]
3 makan = [2,3,4,3,2,3,4]
4 kerja =[7,8,7,2,2,6,0]
5 main = [8,5,7,8,13,15,25]
6
7 plt.plot([],[],color='m', label='Tidur', linewidth=5)
8 plt.plot([],[],color='c', label='Makan', linewidth=5)
9 plt.plot([],[],color='r', label='Kerja', linewidth=5)
10 plt.plot([],[],color='k', label='Main', linewidth=5)
11
12 plt.stackplot(hari, tidur,makan,kerja,main, colors=['b','m','r',
   'c'])
13
14 plt.xlabel('Hari')
15 plt.ylabel('Lamanya')
16 plt.title('Stack Plot')
17 plt.legend()
18 plt.show()
```

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

Cara menggunakan legend adalah namaObject.legend() dan menambahkan labelnya seperti dibawah ini:

```

1 plt.xlabel('Hari')
2 plt.ylabel('Lamanya')
3 plt.title('Stack Plot')
4 plt.legend()
5 plt.show()
```

Penggunaan legend itu berfungsi untuk memudahkan kita ketika membaca grafik yang kita hasilkan karena kita memberi nama pada data yang ditampilkan sama seperti label kita memberikan nama kepada variable yang dimunculkan di grafik dan membedakan antara variable yang satu dengan yang lain, kita juga bisa menambahkan warna ke label

- 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

Subplot berfungsi untuk menampilkan grafik plot pada program yang sama, cara penggunaannya:

```

1 x = [1 ,2 ,3 ,4 ,5]
2 y = [6 ,7 ,8 ,9 ,10]
3 plt . subplot(331) #tinggi , lebar , urutan
4 plt . plot(x , y)
5 plt . subplot(332)
6 plt . bar(x , y)
7 plt . subplot(333)
8 plt . hist(x , 2)
9 plt . subplot(334)
10 plt . scatter(x , y)
11 plt . subplot(335) #tinggi , lebar , urutan

```

Cara penggunaannya sebagai contoh saya ambil plt.subplot(221), pada angka 2 yang pertama adalah pembagian keatas kalo kita mau bagi 3 keatas kita isi angka pertama dengan 3, angka 2 yang kedua adalah pembagian kesamping penggunaannya sama kaya angka pertama kalo kita mau ngebali kesamping 4 kita isi angka kedua 4, dan angka 1 pada angka ketiga itu tempat disimpannya grafik yang akan dimunculkan

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

Parameter warna yang bisa digunakan dibagi menjadi 2 tipe:

- RGB

Untuk keterangannya sebagai berikut R untuk warna Red atau Merah G untuk warna Green atau Hijau B untuk warna Blue atau Biru

- CMYK

Untuk keterangannya sebagai berikut C untuk warna Cyan atau Biru Muda M untuk warna Mangenta atau Merah Tua Y untuk warna Yellow Atau Kuning K untuk warna Black atau Hitam

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

Untuk histogram kita tidak boleh memiliki is variable x dan y yang sama. Misal x-nya ada 10 nilai sedangkan Y-nya ada 5 nilai, data tersebut tidak menjadi masalah karena pada histogram data yang dimunculkan adalah data rentang dari data variable y. Dan ini adalah contoh dari penggunaan histogram

```

1 import matplotlib.pyplot as plt
2 x = [21,22,23,4,5,6,77,8,9,10,31,32,33,34,35,36,37,18,49,50,100]
3 plt.hist(x, 5, histtype ='bar' , rwidth =0.8)
4 plt.xlabel('Angka')
5 plt.ylabel('Banyaknya angka rentang dari 5')
6 plt.show()

```

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

- Label

Label digunakan untuk mempermudah pembaca yaitu memberikan nama pada variable di grafik

- Color

Warna yang dimunculkan pada setiap data

- Startangle

Startangle digunakan untuk sudut awal pada diagram pie tersebut

- Shadow

Shadow(Bayangan) digunakan untuk membuat bayangan pada setiap diagram pie yang menonjol

- Explode

Explode digunakan untuk mengeluarkan suatu data agar data tersebut menjadi terlihat lebih menonjol

- Autopct

Autopct digunakan menyesuaikan berapa angka yang ada dibelakang koma

10.2 Fanny Shafira Damayanti | 1174069

10.2.1 Pemahaman Teori

1. Fungsi Library matplotlib

matplotlib adalah librari plotting 2D Python yang menghasilkan gambar publikasi bermutu di dalam berbagai format hardcopy dan lingkungan interaktif sepanjang platform. matplotlib dapat digunakan di dalam script Python, shell Python dan ipython.

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

caranya yaitu seperti contoh dibawah ini :

```

1 x=[2,4,6]
2 y=[1,3,5]

```

3. Perbedaan fungsi dan cara pakai untuk berbagai jenis (bar, histogram, scatter,dll) jenis plot di matplotlib

Perbedaannya adalah bentuk bentuk grafik yang akan di tampilkan sesuai dengan perintah yang digunakan pada pemogramannya.

cara pengguna plot tersebut sebagai berikut :

- line Perintah yang digunakan untuk membuat grafik line sebagai berikut.

```

1 #plt.plot(x,y)
2 #plt.show()
3 #line
```

- bar Memiliki koordinat X dan Koordinat Y

```

1 #plt.bar([1,3,5,7,9],[50,40,70,80,20],
2 #label="Lamborghini",color='Y',width=.5)
3 #plt.bar([2,4,6,8,10],[80,20,20,50,60],
4 #label="VW", color='C',width=.5)
5 #plt.legend()
6 #plt.xlabel('Days')
7 #plt.ylabel('Distance (kms)')
8 #plt.title('Information')
9 #plt.show()
10 #bar
```

- histogram Dalam penggunaan plot histogram titik x nya bisa tidak sama dengan titik Y.

```

1 #population_age =
2 [40,80,11,22,16,9,10,15,22,55,62,45,21,22,102,95,85,55,110,120,70,8]
3
4 #bins = [0,10,20,30,40,50,60,70,80,90,100]
5 #plt.hist(population_age , bins , histtype='bar' , rwidth=0.8)
6 #plt.xlabel('age groups')
7 #plt.ylabel('Number of people')
8 #plt.title('Histogram')
9 #plt.show()
10 #histogram
```

- scatter Penggunaan plot scatter atau bisa juga di bilang diagram titik.

```

1 #x = [1,1.5,2,2.5,3,3.5,3.6]
2 #y = [7.5,8,8.5,9,9.5,10,10.5]
3
4 #x1=[8,8.5,9,9.5,10,10.5,11]
5 #y1=[3,3.5,3.7,4,4.5,5,5.2]
6
7 #plt.scatter(x,y, label='Pendapatan Tinggi Tapi Penyimpanan
8 Rendah',color='C')
9 #plt.scatter(x1,y1,label='Pendapatan Rendah Tapi Penyimpanan
10 Tinggi',color='M')
11 #plt.xlabel('Pensimpanan dalam ratusan')
12 #plt.ylabel('Pendapatan dalam ribuan')
13 #plt.title('Diagram Titik')
14 #plt.legend()
```

```

13 #plt.show()
14 #scatter

```

- Stack plot Untuk penggunaan stack plot ini seperti diagram line, tapi ada fill colornya,jadi antar line itu bisa berdekatan.

```

1 plt.pie(slices ,
2         labels=activities ,
3         colors=cols ,
4         startangle=0,
5         shadow= True ,
6         explode=(0.1,0,0,0) ,
7         autopct='%1.1f%%')
8
9 plt.title('Pie Plot Keseharian Fanny')
10 plt.show()
11 #diagram Pie

```

4. Cara menggunakan legend dan label serta kaitannya dengan fungsi tersebut

Legend digunakan untuk mempermudah dalam membaca grafik, legend itu sendiri berisi info info dari grafik yang ada seperti nama, kemudian bentuk dan warna. kemudian untuk label itu sendiri digunakan untuk membedakan titik X dan titik Y.

Contohnya sebagai berikut :

```

1 #plt.legend()
2 #plt.xlabel('Days')
3 #plt.ylabel('Distance (kms)')

```

5. Fungsi dari subplot di matplotlib, dan cara kerja dari fungsi subplot sertakan contoh gambarnya

fungsi subplot di matplotlib yaitu untuk bisa membuat lebih dari 1 grafik dalam sebuah program. Contohnya :

```

1 #x = [1,2,3,4,5,6,7,8,9,10]
2 #y = [10,20,30,40,50,60,70,80,90,100]
3 #plt.subplot(221)#tinggi ,lebar ,urutan
4 #plt.plot(x, y)
5 #plt.subplot(222)
6 #plt.bar(x, y)
7 #plt.subplot(223)
8 #plt.hist(x, y)
9 #plt.subplot(224)
10 #plt.scatter(x, y)
11 #plt.show()

```

6. Parameter color yang bisa digunakan

- Tipe Warna RGB

R untuk warna Red atau Merah G untuk warna Green atau Hijau B untuk warna Blue atau Biru

- Tipe warna CMYK

C untuk warna Cyan atau Biru Muda M untuk warna Mangenta atau Merah Tua Y untuk warna Yellow Atau Kuning K untuk warna black atau Hitam

7. Cara kerja dari fungsi hist, sertakan ilustrasi dan gambar sendiri

Untuk fungsi histogram ini kedua titik koordinat boleh tidak sama. Misalnya x nya ada 10 nilai sedangkan Y nya ada 5 nilai, itu tidak akan jadi masalah karena diagram ini digunakan untuk mendata usia dari rentang tertentu atau kebutuhan lainnya. Contoh penggunaan histogram :

```

1 #population_age =
2     [40,80,11,22,16,9,10,15,22,55,62,45,21,22,102,95,85,55,110,120,70,80,7
3 #bins = [0,10,20,30,40,50,60,70,80,90,100]
4 #plt.hist(population_age , bins , histtype='bar' , rwidth=0.8)
5 #plt.xlabel('age groups')
6 #plt.ylabel('Number of people')
7 #plt.title('Histogram')
8 #plt.show()
9 #histogram
```

Contoh grafiknya :

8. Parameter dari fungsi pie diantaranya labels, colors, startangle, shadow, explode, autopct

- label Label digunakan untuk mempermudah pembaca dalam membaca dia-gram pie
- color warna digunakan untuk membedakan antar data
- startangle Digunakan untuk sudut yang digunakan untuk memulai diagram pie tersebut
- shadow bayangan digunakan untuk membuat bayangan dari setiap diagram pie yang menonjol
- explode explode digunakan untuk mengeluarkan suatu data agar data tersebut terlihat menonjol
- autopct Digunakan sesuai dengan berapa angka dibelakang koma yang kita inginkan

10.3 Tia Nur Candida | 1174086

10.3.1 Teori

10.3.1.1 Soal No. 1

Apa itu fungsi library matplotlib?

Matplotlib merupakan library plotting 2D Python yang menghasilkan gambar publikasi bermutu di dalam berbagai format hardcopy dan lingkungan interaktif di berbagai platform. Fungsi dari Matplotlib yaitu sebagai pembuat grafik di berbagai platform, seperti Python dan Jupyter. Grafik yang dibuat menggunakan Matplotlib dapat dibuat dalam berbagai bentuk, seperti grafik garis, batang, lingkaran, histogram, dan sebagainya.

10.3.1.2 Soal No. 2

Jelaskan langkah-langkah membuat sumbu X dan Y di matplotlib!

1. Pertama import library Matplotlib.

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

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

```
1 x=[3,4,2]  
2 y=[7,1,5]
```

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

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

4. Lalu panggil plot tadi dengan memanggil fungsi show.

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

10.3.1.3 Soal No. 3

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

1. 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.

2. 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.

3. 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.

4. 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.

5. 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.

6. Line Graph

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

10.3.1.4 Soal No. 4

Jelaskan bagaimana cara menggunakan legend dan label serta kaitannya dengan 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='Mahasiswa yang Tidak Hadir', linewidth=1)
2 plt.plot(x2,y2,'r',label='Mahasiswa yang Telat', linewidth=1)
```

Listing 10.1 Kode program menggunakan parameter label dengan Matplotlib.

- Kemudian panggil fungsi legend.

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

Listing 10.2 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 = [74,77,100,122,64,88]
5 x2 = [2014,2015,2016,2017,2018,2019]
6 y2 = [134,97,114,134,146,167]
7 plt.plot(x,y,'b',label='Mahasiswa yang Tidak Hadir', linewidth=1)
8 plt.plot(x2,y2,'r',label='Mahasiswa yang Telat', linewidth=1)
9 plt.title('Kehadiran Mahasiswa')
10 plt.ylabel('Jumlah Mahasiswa')
11 plt.xlabel('Tahun')
12 plt.legend()
13 plt.grid(True, color='k')
14 plt.show()
```

Listing 10.3 Kode program membuat diagram menggunakan Matplotlib.

Hasil Compile

10.3.1.5 Soal No. 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 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 10.4 Kode program membuat subplot menggunakan Matplotlib.

Hasil Compile

10.3.1.6 Soal No. 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)

10.3.1.7 Soal No. 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 =
3     [14,13,15,9,10,18,22,55,62,43,21,22,34,42,42,4,2,3,95,64,55,110,96,70,65,55]
4 umur = [0,10,20,30,40,50,60,70,80]
5 plt.hist(orang , umur , histtype='bar' , rwidth=0.5)
6 plt.xlabel('Umur')
7 plt.ylabel('Jumlah Kematian')
8 plt.title('Histogram')
9 plt.show()

```

Listing 10.5 Kode program membuat diagram menggunakan Matplotlib.

Hasil Compile

10.3.1.8 Soal No. 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.

10.4 Aulyardha Anindita | 1174054

10.4.1 Pemahaman Teori

1. Fungsi Library Matplotlib

Matplotlib adalah salah satu modul dalam python yang digunakan untuk menggambar plot 2D dengan kualitas yang tinggi. Dengan matplotlib, kita dapat membuat plots, histograms, spectra, bar charts, errorcharts, scatterplots, dan masih banyak lagi. Matplotlib adalah salah satu library paling banyak digunakan oleh data science untuk menyajikan datanya ke dalam visual yang lebih baik.

2. Langkah-langkah membuat sumbu x dan y di Matplotlib

- Buka spyder terlebih dahulu
- Import matplotlib
- Selanjutnya, buat x dan y. x merupakan sumbu horizontal dan y merupakan sumbu vertikal.
- Isi sumbu x dan y sesuai dengan data yang dinginkan
- Jangan lupa untuk memberikan perintah show untuk menampilkan grafiknya

Contoh :

```

1 #Langkah-langkah membuat sumbu x dan y
2 from matplotlib import pyplot as plt
3 x = [3,5,6,7]
4 y = [2,16,4,6]
5 plt.plot(x,y)
6 plt.title('Info Dita')
7 plt.xlabel('Sumbu X')
8 plt.ylabel('Sumbu Y')
9 plt.show()

```

3. Perbedaan fungsi dan cara pakai untuk berbagai jenis plot di Matplotlib

- Line, Line berfungsi untuk membuat grafik sederhana yang digambar menggunakan garis biasa. Cara penggunaannya yaitu dengan menggunakan plt.plot. Untuk lebih jelasnya seperti berikut:

```

1 #Fungsi Line
2 from matplotlib import pyplot as plt
3 x=[2,4,6]
4 y=[1,3,5]
5 plt.plot(x,y)
6 plt.show()

```

- Bar, Bar berfungsi untuk membuat grafik yang biasanya berbentuk kolom dan terkadang disusun baik secara vertikal maupun horizontal. Cara penggunaannya yaitu dengan menggunakan plt.bar seperti dibawah ini :

```

1 #Fungsi Bar
2 from matplotlib import pyplot as plt
3 plt.bar([0.25,1.25,2.25,3.25,4.25],[50,40,70,80,20],label="Sepatu",color="b",width=0.5)
4 plt.bar([0.75,1.75,2.75,3.75,4.75],[80,20,20,50,60],label="Tas",color="k",width=0.5)
5 plt.legend()
6 plt.xlabel("Harian")
7 plt.ylabel("Jumlah")
8 plt.title("Data Penjualan Dita")
9 plt.show()

```

- Histogram, Histogram berfungsi untuk membuat grafik yang biasanya berbentuk batang. Cara penggunaannya yaitu dengan menggunakan plt.hist seperti dibawah ini :

```

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

```

- Scatt, Scatt berfungsi untuk membuat grafik yang biasanya berbentuk titik yang menggambarkan hubungan atau korelasi antara dua pasangan/variabel. Cara penggunaannya yaitu dengan menggunakan plt.scatter seperti dibawah ini :

```

1 #Fungsi Scatter
2 from matplotlib import pyplot as plt
3 x = [1,1.5,2,2.5,3,3.5,3.6]
4 y = [7.5,8,8.5,9,9.5,10,10.5]
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 plt.scatter(x,y, label='Pengeluaran per bulan', color='b')
8 plt.scatter(x1,y1, label='Pemasukan per bulan', color='m')
9 plt.xlabel('Pengeluaran')
10 plt.ylabel('Pemasukan')
11 plt.title('Scatter Dita')
12 plt.legend()
13 plt.show()

```

- Stack Plot, berfungsi untuk membuat grafik garis yang hampir sama dengan line tetapi memiliki fill color sehingga antar line bisa berdekatan. Cara penggunaannya yaitu dengan menggunakan plt.stackplot seperti dibawah ini :

```

1 #Fungsi Stackplot
2 from matplotlib import pyplot as plt
3 tidur = [1,2,3,4,5,6,7]
4 ngampus =[7,8,6,8,7,9,10]
5 makan = [2,3,4,3,2,3,4]
6 jalan =[7,8,7,2,2,6,0]
7 main = [8,5,7,8,13,15,25]
8 plt.plot([],[],color='y', label='Tidur', linewidth=5)
9 plt.plot([],[],color='b', label='Ngampus', linewidth=5)
10 plt.plot([],[],color='c', label='Makan', linewidth=5)
11 plt.plot([],[],color='g', label='Jalan', linewidth=5)
12 plt.stackplot(tidur, ngampus,makan,jalan,main, colors=['y','b',
13 , 'c', 'g'])
14 plt.xlabel('Hari')
15 plt.ylabel('Aktivitas')

```

```

15 plt.title('Stack Plot Dita')
16 plt.legend()
17 plt.show()

```

- Pie, berfungsi untuk membuat grafik yang biasanya berbentuk lingkaran. Cara penggunaannya yaitu dengan menggunakan plt.pie seperti dibawah ini :

```

1 #Fungsi Pie
2 from matplotlib import pyplot as plt
3 Pemrograman = [1,2,3,4,5,6,7]
4 RPL =[7,8,6,11,7,8,9]
5 Jarkom = [2,3,4,3,2,7,5]
6 MB =[7,8,7,2,2,4,2]
7 SAP = [8,5,7,8,13,7,15]
8 slices = [7,2,2,13]
9 activities = ['Pemrograman', 'RPL', 'Jarkom', 'SAP']
10 cols = ['c', 'm', 'r', 'b']
11 plt.pie(slices,
12         labels=activities,
13         colors=cols,
14         startangle=0,
15         shadow= True,
16         explode=(0.1,0.1,0.1,0.1),
17         autopct='%1.1f%%')
18 plt.title('Pie Plot Dita')
19 plt.show()

```

4. Cara Menggunakan Legend dan Label

Legend biasanya digunakan untuk mempermudah dalam membaca suatu grafik karena berisi isi dari grafik tersebut seperti nama, dan lain sebagainya. Sedangkan pada label biasanya digunakan untuk membedakan nama titik X dan titik Y. Untuk cara penggunannya seperti dibawah ini :

```

1 plt.legend()
2 plt.xlabel("Harian")
3 plt.ylabel("Jumlah")

```

5. Fungsi, Cara Kerja, Ilustrasi dan Parameter Pada Subplot

- Fungsi dari Subplot adalah mampu membuat grafik lebih dari satu dalam sebuah program
- Cara Kerja dari Subplot bisa dilihat pada list berikut :

```

1 #Subplot
2 x = [1,2,3,4,5,6,7,8,9,10]
3 y = [10,20,30,40,50,60,70,80,90,100]
4 plt.subplot(221)#tinggi ,lebar ,urutan
5 plt.plot(x, y, color="r")
6 plt.subplot(222)

```

```
7 plt.bar(x, y, color="g")
8 plt.subplot(223)
9 plt.hist(x, y, color="m")
10 plt.subplot(224)
11 plt.scatter(x, y, color="c")
12 plt.show()
```

- Untuk parameternya sendiri disini saya menggunakan x dan y dimana x sebagai koordinat x dan y sebagai koordinat y.

6. Parameter Color

- R Untuk warna merah
- G Untuk warna hijau
- B Untuk warna biru
- W untuk warna putih
- Y Untuk warna kuning
- C Untuk warna cyan atau biru muda
- M Untuk warna magenta
- K Untuk warna hitam

7. Cara Kerja dan Ilustrasi Fungsi Hist

Cara Kerja fungsi hist ini memiliki kedua titik koordinat dimana jika nilai x misalnya adalah 9 sedangkan nilai y adalah 3 itu tidak akan terjadi masalah karena diagram ini digunakan untuk mendata data yang memiliki rentang waktu tertentu.

8. Parameter dari fungsi Pie

- Label, digunakan untuk mempermudah pembaca dalam membaca diagram pie
- Color, digunakan untuk memberikan style atau mempermudah membaca data karena data nya yang berbeda
- Startangle, digunakan sebagai sudut untuk memulai diagram pie
- Shadow, digunakan untuk membuat bayangan dari diagram pie sehingga datanya terlihat menonjol
- Explode, digunakan untuk mengeluarkan sebuah data sehingga data tersebut dapat terlihat menonjol
- Autopct, digunakan sesuai dengan berapa angka dibelakang koma yang kita inginkan.

10.5 Nurul Izza Hamka | 1174062 | Teori

1. Fungsi Library Matplotlib

Matplotlib adalah Library plotting dengan 2D pada Python yang dapat Menghasilkan sebuah gambar, matplotlib ini paling banyak dingunakan oleh data science karena idalamnya terdapat berbagai format hardcopy dan sangat interaktif di platformnya, serta mampu menyajikan visual data yang lebih baik. dengan matplotlib kita dapat membuat sebuah plots,histograms, bar charts, spectra, daln lainnya.

2. Langkah-langkah Membuat Sumbuh X dan Y di Matplotlib

- (a) Buat plot untuk sebuah garis
- (b) Lakukan Import

```
1 #no 2
2 x = (4,8,13,17,20)
3 y = (54, 67, 98, 78, 45)
4
5 import matplotlib.pyplot as plt
6 plt.plot([4,8,13,17,20],[54, 67, 98, 78, 45])
7 plt.xlabel('IZZA') #Sumbu -X
8 plt.ylabel('IZZA')#Sumbu -Y
9 plt.show()
```

3. Bagaimana Perbedaan Fungsi dan Cara Pakai Untuk Berbagai Jenis Plot Di Matplotlip

(a) Histogram

Histogram menunjukkan frekuensi pada sumbu vertikal dan sumbu horizontal adalah dimensi lain. Histogram Ini akan menampilkan sebuah frekuensi data dengan model batang , yang mana angka didalamnya di kelompokkan dalam rentang tertentu.

(b) Scatter

Matplotlip memiliki fungsi untuk membuat scatterplot yang disebut scatter () . Plot scatter adalah jenis plot yang menunjukkan data sebagai kumpulan poin. Posisi suatu titik tergantung pada nilai dua dimensi, di mana setiap nilai adalah posisi pada dimensi horizontal atau vertikal.

(c) Line

Garis dapat memiliki kedua gaya garis padat yang menghubungkan semua simpul, dan penanda di setiap titik.

4. Bagaimana Cara Menggunakan Legend Dan Label Serta Kaitannya Dengan Fungsinya

Formatting plot adalah suatu cara untuk memberikan sebuah informasi tentang label dan legend pada sebuah grafik dalam plot.

Tambahkan label = ke masing-masing plot Anda () panggilan, dan kemudian panggil legenda (loc = 'kiri atas'). dan kemudian memanggil kapak bukannya plt.gca (). Plot sederhana untuk kurva sinus dan cosinus dengan legend. Kemudian cukup tambahkan Pyplot.legend () ke bagian bawah skrip Anda dan legenda akan menampilkan label ini.

```
1 #no 4
2 #plt.legend()
3 #plt.xlabel('Days')
4 #plt.ylabel('Distance (kms)')
```

5. Apa fungsi Dari Subplot di Matplotlib, dan Bagaimana Cara Kerja Dari Fungsi Subplot, Sertakan Ilustrasi dan Gambaar sendiri dan Apa Parameternya Jika Ingin Menggambar dengan 9 Subplot di Dalamnya.

Fungsi Subplot pada Matplotlib adalah kita dapat memanggil () untuk memplot sebanyak dua kali ataupun lebih dalam satu gambar.

```
1 #no 5
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 x1 = np.linspace(1, 2)
6 x2 = np.linspace(1, 5)
7
8 plt.subplot(2,3,1)
9 plt.plot(x1, x2, '2-')
10 plt.title('Subplot A')
11 plt.subplot(2,3,2)
12 plt.plot(x1, x2, 'o-')
13 plt.title('Subplot B')
14 plt.subplot(2,3,3)
15 plt.title('Subplot C')
16 plt.subplot(2,3,4)
17 plt.title('Subplot D')
18 plt.subplot(2,3,5)
19 plt.title('Subplot E')
20 plt.subplot(2,3,6)
21 plt.title('Subplot F')
22
23 plt.show()
```

Ilustrasi 9 Gambar Subplot:

6. Sebutkan parameter color yang bisa digunakan: Matplotlib mengenali format berikut untuk menentukan warna:
 - tuple RGB yaitu 'r', 'g', 'b' - 'tab:red','tab:green','tab:blue'
 - Tuple CMYK yaitu 'c', 'm', 'y','k' - 'tab:cyan','tab:magenta','tab:yellow','tab:black';

7. Bagaimana Cara Kerja Dari Fungsi Hist, sertakan Ilustrasi Dan Gambar Sendiri.

Pada histogram kedua titik koordinat boleh tidak sama, Misalnya X mempunyai 10 nilai sedangkan Y nya ada 5 nilai, karena diagram ini digunakan untuk mendata usia dari rentang tertentu atau kebutuhan lainnya. Contoh dari penggunaan histogram :

```
1 #no 7
2 #population_age =
3 [40,80,11,22,16,9,10,15,22,55,62,45,21,22,102,95,85,55,110,120,70,80,7
4
5 #bins = [0,10,20,30,40,50,60,70,80,90,100]
6 #plt.hist(population_age , bins , histtype='bar' , rwidth=0.8)
7 #plt.xlabel('age groups')
8 #plt.ylabel('Number of people')
9 #plt.title('Histogram')
10 #plt.show()
11 #histogram
```

Ilustrasi Gambar :

8. Penjelasan tentang Parameter dari Fungsi Pie Diantaranya Labaels, Colors, Startangle, Shadow, Explode, Autopct.

- (a) Labels Urutan string yang memberikan label untuk setiap irisan,Jumlah label harus sama dengan jumlah elemen.
- (b) Colors
Urutan warna matplotlib berargumentasi di mana diagram lingkaran akan berputar. Jika Tidak Ada, akan menggunakan warna dalam siklus aktif saat ini.
- (c) Startangle
Jika tidak ada, apakah string atau fungsi yang digunakan untuk memberi label pada irisan dengan nilai numeriknya. Label akan ditempatkan di dalam irisan. Jika ini adalah string format, label akan menjadi fmt persen. Jika Fungsi itu akan dipanggil.
- (d) Shadow
Gambar bayangan di bawah pai.
- (e) Explode
mengimbangi irisan dari pie.explode adalah vektor atau matriks nol dan nonzeros yang sesuai dengan Fungsi pie mengimbangi irisan untuk elemen bukan nol hanya dalam explode .
- (f) Autopct
autopct memungkinkan Anda untuk menampilkan nilai persen menggunakan pemformatan string Python. Dengan Autopct dan Callable kita dapat melakukan hal lebih.

9. Plagiarisme

10.6 Difa Al Fansha

10.6.1 Pemahaman Teori

1. Apa itu fungsi library matplotlib

Jawab :

Python memiliki banyak library yang memudahkan para ilmuwan data / data scientist untuk menyelesaikan tugas-tugas rumit tanpa banyak gangguan pengkodean

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

Jawab :

```

1 # Jawaban Nomor 2
2 # 1. Mengimport library matplotlib & pyplot yang diberi nama pt
3 from matplotlib import pyplot as pt
4
5 # 2. Mendeklarasikan variabel x dan y
6 x = [1,2,3]
7 y = [4,5,2]
8 # 3. Deklarasikan pt.plot yang berisi variabel
9 pt.plot(x,y)
10 pt.show()
```

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

Jawab :

- Plot Hanya bisa menampilkan data yang berkelanjutan (continuous data). Jika Anda perlu mengetahui perubahan dari waktu ke waktu maka Line Chart sangat cocok untuk digunakan.

```

1 #Plot
2 from matplotlib import pyplot as garis
3 x = [9,3,2]
4 y = [4,3,1]
5 garis.plot(x,y, label = 'Garis', linewidth=5)
6 garis.legend()
7 garis.show()
```

- Bar Grafik batang menggambarkan perbandingan antar beberapa item.

```

1 # Bar
2 from matplotlib import pyplot as batang
3 x = [1]
4 y = [4]
5 batang.bar(x, label = 'Bar 1', height=1)
6 batang.bar(y, label = 'Bar 2',height=2)
7 batang.legend()
8 batang.show()
```

- Histogram sebuah representasi grafik yang menampilkan impresi visual dari distribusi data

```

1 # Histogram
2 from matplotlib import pyplot as kotak
3 x = [22,26,45,39,67,25,89,10,11,35,56,67,70,63,52,78]
4 binds = [0,20,40,60,80,100]
5 kotak.hist(x, label='Histogram', rwidth=0.8)
6 kotak.legend()
7 kotak.show()

```

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

Jawab :

```

1 # Jawaban Nomor 4
2 # 1. Mengimport pyplot yang diberi nama plt
3 from matplotlib import pyplot as plt
4 # 2. Mendeklarasikan variabel x dan y
5 x = [9,3,2]
6 y = [4,3,1]
7 # 3. Mendeklarasikan plt.plot dengan deklarasi variabel
8 plt.plot(x,y, label = 'Garis Legend', linewidth=5)
9 # 4. Tambahkan Legend yang berfungsi untuk menambahkan informasi
10 plt.legend()
11 # Memanggil plt
12 plt.show()

```

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

```

1 # Jawaban 5
2 # Subplot
3 # Deklrasi varibael
4 x = [1,2,3,4,5]
5 y = [6,7,8,9,10]
6 #tinggi ,lebar ,urutan
7 plt.subplot(331)
8 plt.plot(x, y)
9 plt.subplot(332)
10 plt.bar(x, y)
11 plt.subplot(333)
12 plt.hist(x, 2)
13 plt.subplot(334)
14 plt.scatter(x, y)
15 plt.subplot(335)
16 plt.plot(x, y)
17 plt.subplot(336)
18 plt.bar(x, y)
19 plt.subplot(337)
20 plt.hist(x, 2)
21 plt.subplot(338)
22 plt.scatter(x, y)
23 plt.subplot(339)
24 plt.hist(x, 2)
25 plt.show()

```

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

Jawab :

- 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 Mangenta 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

Jawab :

```

1 # Histogram
2 from matplotlib import pyplot as kotak
3 x = [22,26,45,39,67,25,89,10,11,35,56,67,70,63,52,78]
4 binds = [0,20,40,60,80,100]
5 kotak.hist(x, label='Histogram', rwidth=0.8)
6 kotak.legend()
7 kotak.show()
```

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

- Label berfungsi untuk mempermudah dalam hal pembacaan
- Colors warna setiap data
- Explode mengeluarkan suatu data agar data tersebut menjadi terlihat lebih menonjol
- Shadow untuk membuat bayangan pada setiap diagram pie yang menonjol

10.7 Chandra Kirana Poetra

10.7.1 Pemahaman Teori

1. Apa itu fungsi library matplotlib? Matplotlib adalah suatu library plotting 2d berbasis python yang membuat publikasi figure yang berkualitas berbentuk hard-copy dengan interaktivitas di berbagai platforms. Matplotlib bisa digunakan di script python, iPython, jupyter, web application servers, dan toolkit graphical. matplotlib bisa membuat plot, histograms, power spectra, bar charts, error-charts, scatterplots dan masih banyak lagi dengan hanya beberapa baris kode.
2. Jelaskan langkah-langkah membuat sumbu X dan Y di matplotlib?

- (a) Untuk pertama-tama Import terdahulu library matplotlib dan berikan alias librarymatplotlib agar mudah diingat

```
1 import matplotlib.pyplot as librarymatplotlib
```

- (b) Kemudian buat variable array dengan nama X, isi sesuai keinginan

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

- (c) Kemudian buat variable array dengan nama X, isi sesuai keinginan dengan jumlah yang sama dengan variable array X

```
1 y = [10, 12, 30, 46, 55]
```

- (d) Kemudian kita panggil function plot yang akan diisi variable X dan Y

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

- (e) Terakhir, gunakan function show() untuk menampilkan apa yang telah kita buat

```
1 librarymatplotlib.show()
```

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

Perbedaan diagram diagram yang ada itu hanya pada bentuknya lalu jenis jenis grafik yang ada pada plot adalah:

- Plot

Merupakan Grafik berbentuk garis

```
1 import matplotlib.pyplot as librarymatplotlib
2 x = [1, 2, 3, 4, 5]
3 y = [10, 12, 30, 46, 55]
4 librarymatplotlib.plot(x, y)
5 librarymatplotlib.show()
```

- Bar

Bar adalah sebuah grafik berbentuk batang (bar) dan berikut cara penggunaanya

```
1 import matplotlib.pyplot as librarymatplotlib
2 x = [1, 2, 3, 4, 5]
3 y = [10, 12, 30, 46, 55]
4 librarymatplotlib.bar(x, y)
5 librarymatplotlib.show()
```

- Histogram

Histogram merupakan diagram yang mirip bentuknya seperti batang namun perbedaannya adalah hist(variableX,variableY) dimana variableY merupakan kelipatan yang akan dimunculkan

```

1 import matplotlib.pyplot as librarymatplotlib
2 x = [10,20,30,25,22,55,11,33,22,55,88,44,100]
3 librarymatplotlib.hist(x, 5)
4 librarymatplotlib.show()

```

▪ Scatter

Scatter merupakan diagram grafik yang menghasilkan oleh scatter/penyebaran titik dan cara penggunaannya menggunakan .scatter(variable x, variable y)

```

1 import matplotlib.pyplot as librarymatplotlib
2 import numpy as npy
3 x = npy.random.randn(100)
4 y = npy.random.randn(100)
5 x1 = npy.random.randn(100)
6 y1 = npy.random.randn(100)
7 librarymatplotlib.scatter(x,y, label='Contoh 1', color='m')
8 librarymatplotlib.scatter(x1,y1, label='Contoh 2', color='B')
9 librarymatplotlib.xlabel('Contoh 1')
10 librarymatplotlib.ylabel('Contoh 2')
11 librarymatplotlib.title('Diagram Swarm')
12 librarymatplotlib.legend()
13 librarymatplotlib.show()

```

▪ Stack Plot

Stackplot merupakan diagram grafik yang mirip seperti diagram line, hanya hasil datanya disatukan semua keatasnya dan cara penggunaannya adalah menggunakan .stackplot(variable, variable, variable)

```

1 hari = [1,2,3,4,5,6,7]
2 Gaming =[9,5,3,5,7,3,2]
3 Belajar = [1,2,3,4,5,6,7]
4 kerja =[1,2,3,4,3,4,6]
5 Freetime = [8,5,7,8,13,15,25]
6
7 librarymatplotlib.plot([],[],color='m', label='Gaming',
   linewidth=5)
8 librarymatplotlib.plot([],[],color='c', label='Belajar',
   linewidth=5)
9 librarymatplotlib.plot([],[],color='r', label='Kerja',
   linewidth=5)
10 librarymatplotlib.plot([],[],color='k', label='Freetime',
   linewidth=5)
11
12 librarymatplotlib.stackplot(hari , Gaming , Belajar , kerja ,
   Freetime , colors=['b','m','c','r','k'])
13
14 librarymatplotlib.xlabel('Hari')
15 librarymatplotlib.ylabel('Durasi')
16 librarymatplotlib.title('Stack Plot Diagram')
17 librarymatplotlib.legend()
18 librarymatplotlib.show()

```

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

Cara menggunakan legend dan label adalah dengan memanggil function xlabel, ylabel, title, dan legend yang ada di library matplotlib

```

1 librarymatplotlib.xlabel('Hari')
2 librarymatplotlib.ylabel('Durasi')
3 librarymatplotlib.title('Stack Plot Diagram')
4 librarymatplotlib.legend()
5 librarymatplotlib.show()
```

Pastikan legend yang dibuat berisi informasi yang informatif dan dapat membantu siapapun yang membaca grafiknya menjadi lebih mudah paham

- Jelaskan apa fungsi dari subplot di matplotlib, dan bagaimana cara kerja dari fungsi subplot, sertakan ilustrasi dan gambar sendiri dan apa parameteranya jika ingin menggambar plot dengan 9 subplot di dalamnya

Subplot berfungsi untuk menampilkan grafik plot pada satu tempat, namun peletakannya seperti table, contoh kode:

```

1 x = [1,2,3,4,5,6,7,8,9,10]
2 y = [1.5,2,3,7,9,10,4,5,7]
3 librarymatplotlib.subplot(331)
4 librarymatplotlib.plot(x, y)
5 librarymatplotlib.subplot(332)
6 librarymatplotlib.bar(x, y)
7 librarymatplotlib.subplot(333)
8 librarymatplotlib.hist(x, 2)
9 librarymatplotlib.subplot(334)
10 librarymatplotlib.scatter(x, y)
11 librarymatplotlib.subplot(335)
12 librarymatplotlib.plot(x, y)
13 librarymatplotlib.subplot(336)
14 librarymatplotlib.bar(x, y)
15 librarymatplotlib.subplot(337)
16 librarymatplotlib.hist(x, 2)
17 librarymatplotlib.subplot(338)
18 librarymatplotlib.scatter(x, y)
19 librarymatplotlib.subplot(339)
20 librarymatplotlib.hist(x, 2)
21 librarymatplotlib.show()
```

contoh saya akan menggunakan (331) 3 pertama merupakan X berarti dia ada ruang ke bawah 3, 3 ke dua merupakan Y jadi dia ada ruang 3 ke samping maka akan membentuk seperti table 3x3 yang dimana angka 1 merupakan letak dia akan disimpan

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

Ada 2 tipe parameter warna yang dapat digunakan diantaranya yaitu :

- RGB Red, Green, Blue
- CMYK Cyan,Magenta,Yellow,Black

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

Grafik histogram menampilkan nilai x dan y, namun Y disini berlaku sebagai nilai rentang, apabila menggunakan cara lain nilai X dan Y harus sama jumlahnya, nah di histogram ini tidak perlu dikarenakan Y menjadi rentang. Berikut contoh histogram

```
1 import matplotlib.pyplot as librarymatplotlib  
2 x = [10,20,30,15,23,12,66,44,33,22,33,11,23,55,11,22,77,99,22]  
3 librarymatplotlib.hist(x, 5, histtype ='bar' , rwidth =0.8)  
4 librarymatplotlib.xlabel('Angka')  
5 librarymatplotlib.ylabel('Rentang dari 5')  
6 librarymatplotlib.show()
```

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

- Label

Label digunakan untuk memberi keterangan sehingga dapat mempermudah pembaca dengan menambahkan beberapa informasi mengenai grafik

- Color

Warna

- Startangle

Startangle digunakan untuk sudut awal pada diagram pie tersebut

- Shadow

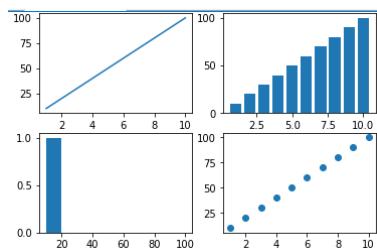
Shadow(Bayangan) digunakan untuk membuat efek bayangan

- Explode

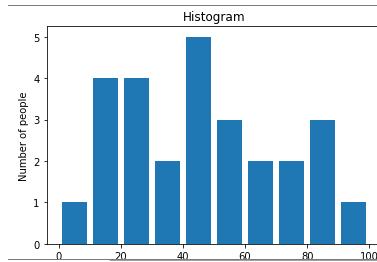
Gunakan Explode untuk membuat suatu data menjadi lebih menonjol

- Autopct

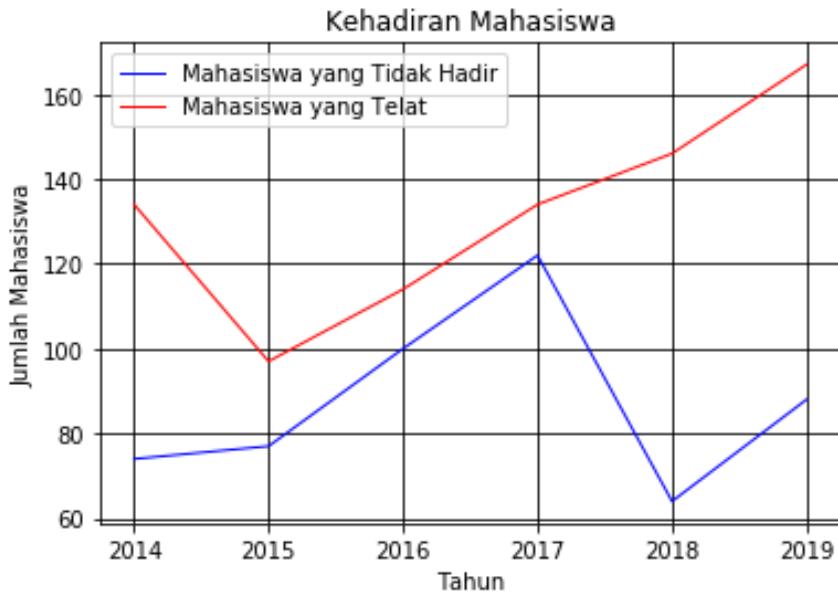
Autopct digunakan untuk menyesuaikan berapa angka yang ada dibelakang koma seperti penggunaan uang



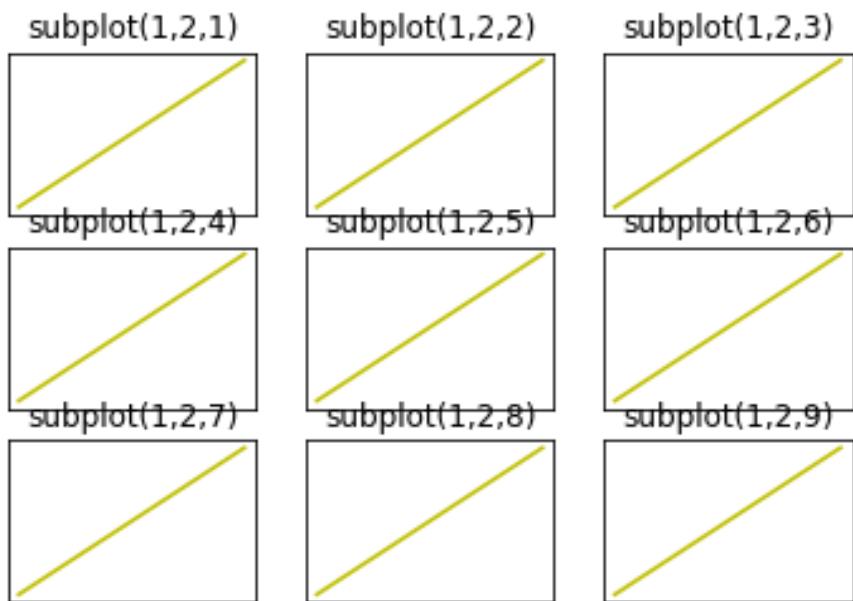
Gambar 10.1 SubPlot



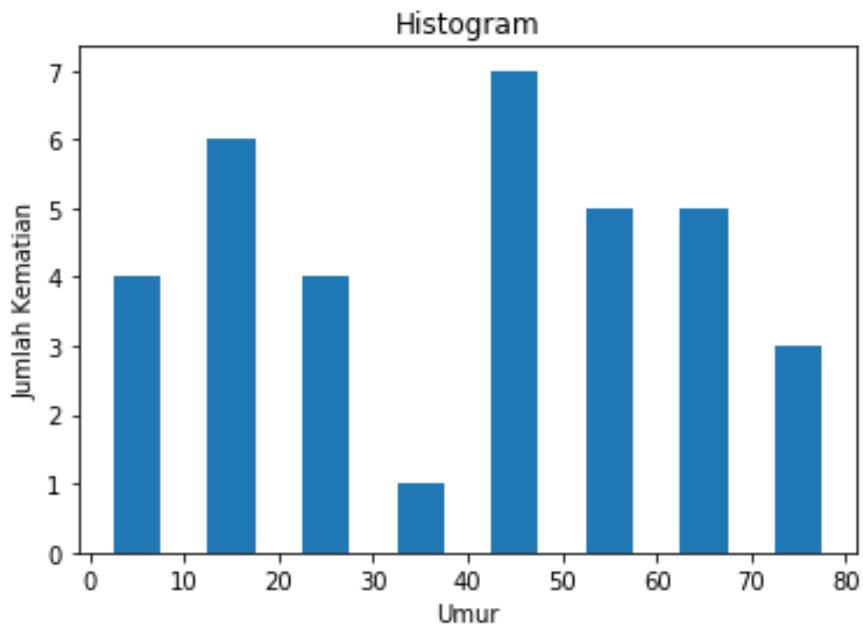
Gambar 10.2 Diagram Histogram



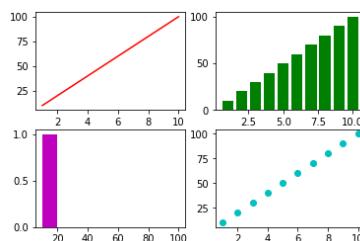
Gambar 10.3 Hasil compile membuat diagram menggunakan Matplotlib.



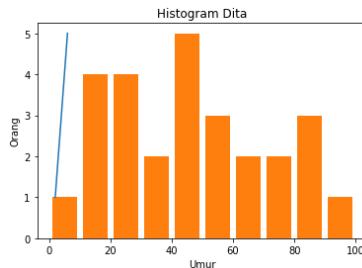
Gambar 10.4 Hasil compile subplot menggunakan Matplotlib.



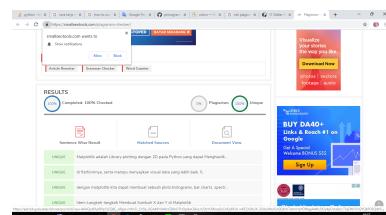
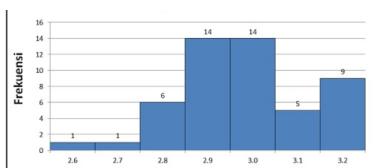
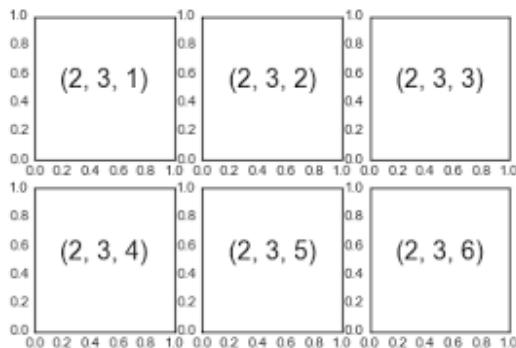
Gambar 10.5 Hasil compile membuat diagram menggunakan Matplotlib.



Gambar 10.6 Subplot Dita



Gambar 10.7 Histogram Dita



BAB 11

PRAKTEK MATPLOTLIB

11.1 D. Irga B. Naufal Fakhri

11.1.1 Buatlah library fungsi (file terpisah/library dengan nama NPM-/textunderscore bar.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2

Ini adalah fungsi untuk membuat plot bar sesuai dengan hasil modulus:

```
1 from matplotlib import pyplot as plt
2 import numpy as np
3 def bar(mod):
4     if mod == 3:
5         x = [1,2,3,4,5]
6         y = [2,3,6,5,7]
7         x1 = [10,23,22,11,25,22,17,12,15,18]
8         y1 = [23,22,25,24,21,22,26,29,11,16]
9         x2 = [1,2,3,4,5,6,7,8,9,10]
10        y2 = [11,12,13,14,15,16,17,18,19,20]
11        plt.subplot(221)
12        plt.bar(x,y)
13        plt.subplot(222)
```

```

14     plt.bar(x1,y1)
15     plt.subplot(223)
16     plt.bar(x2,y2)
17     plt.show()
18 elif mod == 2:
19     x = np.random.randint(0, 10, 10)
20     y = np.random.randint(0, 10, 10)
21     x1 = np.random.randint(10, 20, 10)
22     y1 = np.random.randint(10, 20, 10)
23     plt.subplot(211)
24     plt.bar(x,y)
25     plt.subplot(212)
26     plt.bar(x1,y1)
27     plt.show()
28 elif mod == 1:
29     x = np.random.randint(0, 10, 10)
30     y = np.random.randint(0, 10, 10)
31     x1 = np.random.randint(10, 20, 10)
32     y1 = np.random.randint(10, 20, 10)
33     plt.subplot(111)
34     plt.bar(x,y)
35     plt.show()
36 else:
37     print("Data tidak Valid")

```

Dan ini adalah cara memanggilnya:

```

1 %% Pemanggilan Bar
2 import 11174066_bar as bar
3
4 npm = input("Masukkan NPM Anda: ")
5 modnpm = int(npm)%3+2
6 print(modnpm)
7
8 if modnpm == 3:
9     bar.bar(3)
10 elif modnpm == 2:
11     bar.bar(2)
12 elif modnpm == 1:
13     bar.bar(1)
14 else:
15     print("Data tidak Valid")

```

11.1.2 Buatlah library fungsi (file terpisah/library dengan nama NPM-/textunderscore scatter.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2

Ini adalah fungsi untuk membuat plot scatter sesuai dengan hasil modulus:

```

1 from matplotlib import pyplot as plt
2 import numpy as np
3 def scatter(mod):
4     if mod == 3:
5         x = [1,2,3,4,5]
6         y = [2,3,6,5,7]

```

```

7      x1 = [10,23,22,11,25,22,17,12,15,18]
8      y1 = [23,22,25,24,21,22,26,29,11,16]
9      x2 = [1,2,3,4,5,6,7,8,9,10]
10     y2 = [11,12,13,14,15,16,17,18,19,20]
11     plt.subplot(221)
12     plt.scatter(x,y)
13     plt.subplot(222)
14     plt.scatter(x1,y1)
15     plt.subplot(223)
16     plt.scatter(x2,y2)
17     plt.show()
18 elif mod == 2:
19     x = np.random.randint(0, 10, 10)
20     y = np.random.randint(0, 10, 10)
21     x1 = np.random.randint(10, 20, 10)
22     y1 = np.random.randint(10, 20, 10)
23     plt.subplot(211)
24     plt.scatter(x,y)
25     plt.subplot(212)
26     plt.scatter(x1,y1)
27     plt.show()
28 elif mod == 1:
29     x = np.random.randint(0, 10, 10)
30     y = np.random.randint(0, 10, 10)
31     x1 = np.random.randint(10, 20, 10)
32     y1 = np.random.randint(10, 20, 10)
33     plt.subplot(111)
34     plt.scatter(x,y)
35     plt.show()
36 else:
37     print("Data tidak Valid")

```

Dan ini adalah cara memanggilnya:

```

1 %% Pemanggilan Scatter
2 import 11174066_scatter as scatter
3
4 npm = input("Masukkan NPM Anda: ")
5 modnpm = int(npm)%3+2
6 print(modnpm)
7
8 if modnpm == 3:
9     scatter.scatter(3)
10 elif modnpm == 2:
11     scatter.scatter(2)
12 elif modnpm == 1:
13     scatter.scatter(1)
14 else:
15     print("Data tidak Valid")

```

11.1.3 Buatlah library fungsi (file terpisah/library dengan nama NPM-/textunderscore pie.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2

Ini adalah fungsi untuk membuat plot pie sesuai dengan hasil modulus:

```

1 from matplotlib import pyplot as plt
2
3 def pie(mod):
4     if mod == 3:
5         aktivitas= [8,1,14,2]
6         game = [10,18,4]
7         prog = [7,5,6,1]
8         activities = ['Tidur','Makan','Kerja','Main']
9         games = ['Minecraft','osu','CSGO']
10        programming = ['C#','Java','PHP','HTML']
11        cols = ['m','r','c','b']
12        plt.subplot(221)
13        plt.pie(aktivitas,
14                 labels=activities ,
15                 colors=cols ,
16                 startangle=0,
17                 shadow=True ,
18                 explode=(0.2,0,0,0) ,
19                 autopct='%.1f%%')
20        plt.title('Pie Aktivitas')
21
22        plt.subplot(222)
23        plt.pie(game,
24                 labels=games ,
25                 colors=cols ,
26                 startangle=90,
27                 shadow=True ,
28                 explode=(.3 ,0.1 ,0) ,
29                 autopct='%.1f%%')
30        plt.title('Pie Gaming')
31
32        plt.subplot(223)
33        plt.pie(prog ,
34                 labels=programming ,
35                 colors=cols ,
36                 startangle=90,
37                 shadow=True ,
38                 explode=(.1 ,0 ,0 ,0) ,
39                 autopct='%.1f%%')
40        plt.title('Pie Programming')
41        plt.show()
42    else:
43        print("Data belum dimasukkan")

```

Dan ini adalah cara memanggilnya:

```

1 %% Pemanggilan Pie
2 import 11174066_pie as pie
3
4 npm = input("Masukkan NPM Anda: ")
5 modnpm = int(npm)%3+2
6 print(modnpm)
7
8 if modnpm == 3:
9     pie.pie(3)
10 elif modnpm == 2:
11     pie.pie(2)

```

```

12 elif modnpm == 1:
13     pie.pie(1)
14 else:
15     print("Data tidak Valid")

```

11.1.4 Buatlah library fungsi (file terpisah/library dengan nama NPM-/textunderscore pie.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2

Ini adalah fungsi untuk membuat subplot sesuai dengan hasil modulus:

```

1 from matplotlib import pyplot as plt
2 import numpy as np
3 def plot(mod):
4     if mod == 3:
5         x = [1,2,3,4,5]
6         y = [2,3,6,5,7]
7         x1 = [10,23,22,11,25,22,17,12,15,18]
8         y1 = [23,22,25,24,21,22,26,29,11,16]
9         x2 = [1,2,3,4,5,6,7,8,9,10]
10        y2 = [11,12,13,14,15,16,17,18,19,20]
11        plt.subplot(221)
12        plt.plot(x,y)
13        plt.subplot(222)
14        plt.plot(x1,y1)
15        plt.subplot(223)
16        plt.plot(x2,y2)
17        plt.show()
18    elif mod == 2:
19        x = np.random.randint(0, 10, 10)
20        y = np.random.randint(0, 10, 10)
21        x1 = np.random.randint(10, 20, 10)
22        y1 = np.random.randint(10, 20, 10)
23        plt.subplot(211)
24        plt.plot(x,y)
25        plt.subplot(212)
26        plt.plot(x1,y1)
27        plt.show()
28    elif mod == 1:
29        x = np.random.randint(0, 10, 10)
30        y = np.random.randint(0, 10, 10)
31        x1 = np.random.randint(10, 20, 10)
32        y1 = np.random.randint(10, 20, 10)
33        plt.subplot(111)
34        plt.plot(x,y)
35        plt.show()
36    else:
37        print("Data tidak Valid")

```

Dan ini adalah cara memanggilnya:

```

1 %% Pemanggilan Plot
2 import 11174066_plot as plot
3
4 npm = input("Masukkan NPM Anda: ")

```

```

5 modnpm = int(npm)%3+2
6 print(modnpm)
7
8 if modnpm == 3:
9     plot.plot(3)
10 elif modnpm == 2:
11     plot.plot(2)
12 elif modnpm == 1:
13     plot.plot(1)
14 else:
15     print("Data tidak Valid")

```

11.1.5 Keterampilan Penanganan Error

```

1 def PenangananError():
2     try:
3         from matplotlib import pyplot as plt
4     except SyntaxError:
5         print("Ada kesalahan dalam penulisan Syntax")
6     except NameError:
7         print("Variable yang dimasukkan tidak ada")
8     except TypeError:
9         print("Ada yang salah pada type data")
10    except:
11        print("Sedang terjadi sebuah kesalahan")
12
13 PenangananError()

```

11.2 Chandra Kirana Poetra

11.2.1 Buatlah librari fungsi (file terpisah/library 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     hasil = 1174079 % 3 + 2
6
7     for i in range(1, hasil):
8         plt.subplot(2,2,i)
9         plt.bar
([2005,2006,2007,2008,2009,2010],[2000000000,2200000000,2300000000,2400000000]
10
11         label="Indonesia",color='b',width=.3)
12         plt.bar
([2005,2006,2007,2008,2009,2010],[1000000000,1200000000,1300000000,1400000000]
13
14         label="Malaysia",color='r',width=.3)
15         plt.bar
([2005,2006,2007,2008,2009,2010],[2000000000,3000000000,4000000000,4200000000,4400000000])

```

```

14     label="Singapore",color='g',width=.3)
15     plt.legend()
16     plt.xlabel('Tahun')
17     plt.ylabel('Populasi')
18     plt.title('Populasi Negara')
19     plt.subplots_adjust(wspace=1, hspace=.7)
20
21     plt.show()

```

Cara memanggilnya:

```

1 contohbar = __import__('1174079_Bar')
2 contohscat = __import__('1174079_Scatter')
3 contohpie = __import__('1174079_Pie')
4 contohplot = __import__('1174079_Plot')
5
6 contohbar.bar()#memanggil 1174079_Bar
7 contohscat.scatter()#memanggil 1174079_Scatter
8 contohpie.pie()#memanggil 1174079_Pie
9 contohplot.plot()#memanggil 1174079_Plot

```

11.2.2 Buatlah librari fungsi (file terpisah/library 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     hasil = 1174079 % 3 + 2
6
7     x = [1,2,5,2,6,8,2,4,7,10]
8     y = [5,5,5,7,1,3,6,8,6,5]
9
10    x1=[5,5,5,7,1,3,6,8,6,5]
11    y1=[1,2,5,2,6,8,2,4,7,10]
12
13    for i in range(1, hasil):
14        plt.subplot(2,2,i)
15        plt.scatter(x,y, label='pendapatan tinggi simpanan rendah',
16 color='r')
16        plt.scatter(x1,y1,label='pendapatan rendah simpanan tinggi',
17 color='g')
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
23    plt.show()

```

Cara memanggilnya:

```

1 contohbar = __import__('1174079_Bar')
2 contohscat = __import__('1174079_Scatter')
3 contohpie = __import__('1174079_Pie')
4 contohplot = __import__('1174079_Plot')
5

```

```

6 contohbar.bar()#memanggil 1174079_Bar
7 contohscat.scatter()#memanggil 1174079_Scatter
8 contohpie.pie()#memanggil 1174079_Pie
9 contohplot.plot()#memanggil 1174079_Plot

```

11.2.3 Buatlah librari fungsi (file terpisah/library 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     hasil = 1174079 % 3 + 2
6
7     potong = [7,4,2,12]
8     activity = [ 'Game' , 'MeTime' , 'Kuliah' , 'Bebas' ]
9     kolom = [ 'c' , 'm' , 'y' , 'g' ]
10
11    for i in range(1, hasil):
12        plt.subplot(2,2,i)
13        plt.pie(potong ,
14                 labels=activity ,
15                 colors=kolom ,
16                 startangle=90,
17                 explode=(0,0,0.2,0),
18                 autopct='%.1f%%')
19        plt.title('Aktivitas Sehari hari')
20        plt.subplots_adjust(hspace=.4)
21
22    plt.show()

```

Cara memanggilnya:

```

1 contohbar = __import__('1174079_Bar')
2 contohscat = __import__('1174079_Scatter')
3 contohpie = __import__('1174079_Pie')
4 contohplot = __import__('1174079_Plot')
5
6 contohbar.bar()#memanggil 1174079_Bar
7 contohscat.scatter()#memanggil 1174079_Scatter
8 contohpie.pie()#memanggil 1174079_Pie
9 contohplot.plot()#memanggil 1174079_Plot

```

11.2.4 Buatlah librari fungsi (file terpisah/library 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     hasil = 1174079 % 3 + 2
6
7     x = [2001,2002,2003,2004,2005,2006]
8     y = [10,20,50,70,90,110]
9     x2 = [2001,2002,2003,2004,2005,2006]

```

```
10 y2 = [30,20,30,50,90,130]
11
12 for i in range(1, hasil):
13     plt.subplot(2,2,i)
14     plt.plot(x,y,'b',label='Sample A', linewidth=1)
15     plt.plot(x2,y2,'r',label='Sample B', linewidth=1)
16     plt.title('Sample')
17     plt.ylabel('Jumlah Mikroba')
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()
```

Cara memanggilnya:

```
1 contohbar = __import__('1174079_Bar')
2 contohscat = __import__('1174079_Scatter')
3 contohpie = __import__('1174079_Pie')
4 contohplot = __import__('1174079_Plot')
5
6 contohbar.bar()#memanggil 1174079_Bar
7 contohscat.scatter()#memanggil 1174079_Scatter
8 contohpie.pie()#memanggil 1174079_Pie
9 contohplot.plot()#memanggil 1174079_Plot
```

11.2.5 Keterampilan Penanganan Error

```
1 from matplotlib import pyplot as plt
2
3 def tryExceptError():
4     try:
5         a=[1,2,3]
6         b=[5,2,4]
7         plt.plot(x,y)
8         plt.show()
9     except SyntaxError:
10         print("Kesalahan pada penulisan syntax")
11     except NameError:
12         print("Variable yang anda maksud tidak ada")
13     except TypeError:
14         print("Tipe data salah")
15     except:
16         print("Terjadi error")
17
18 tryExceptError()
```


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.

Index

disruptif, xli
modern, xli