

# **CERDAS MENGUASAI PYTHON**



---

# CERDAS MENGUASAI PYTHON

## Dalam 24 Jam

---

**Rolly M. Awangga**  
Informatics Research Center



**Kreatif Industri Nusantara**

***Penulis:***

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

***Editor:***

M. Yusril Helmi Setyawan

***Penyunting:***

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

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

Deza Martha Akbar

***Penerbit:***

Kreatif Industri Nusantara

***Redaksi:***

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

***Distributor:***

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

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

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

# CONTRIBUTORS

---

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



# CONTENTS IN BRIEF

---

<b>1</b>	<b>SEJARAH DAN KARAKTERISTIK PYTHON</b>	<b>1</b>
<b>2</b>	<b>Pemrograman Dasar</b>	<b>35</b>
<b>3</b>	<b>Fungsi dan Kelas</b>	<b>163</b>
<b>4</b>	<b>Pengelolaan File CSV</b>	<b>297</b>





# DAFTAR ISI

---

Daftar Gambar	xix
Daftar Tabel	xxi
Foreword	xliii
Kata Pengantar	xlvi
Acknowledgments	xlvi
Acronyms	xlix
Glossary	li
List of Symbols	liii
Introduction	lv
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
<b>1 SEJARAH DAN KARAKTERISTIK PYTHON</b>	<b>1</b>
1.1 Chandra Kirana Poetra	1
1.1.1 Sejarah python	1
1.1.2 Instalasi Anaconda	1
1.1.3 Spyder	2
	<b>ix</b>

1.2	Chapter 1   D. Irga B. Naufal Fakhri D4 TI 2C	2
1.2.1	Sejarah Python	2
1.2.2	Tanggal Rilis Python	3
1.2.3	Perbedaan Python 2 dengan Python 3	4
1.2.4	Penggunaan Python di perusahaan dunia	5
1.2.5	Cara menginstall Anaconda	6
1.2.6	Cara menggunakan Spyder pada Anaconda	6
1.2.7	Membuat Hello World di Spyder	7
1.3	Bakti Qilan Mufid	7
1.3.1	Resume Sejarah Python	7
1.3.2	Perbedaan Python 2 dan Python 3	7
1.3.3	Implementasi dan penggunaan Python pada Perusahaan	8
1.4	Instalasi	9
1.4.1	Cara Pemakaian Script dan interpreter python	9
1.4.2	Cara Pemakaian spyder termasuk variable explorer	9
1.5	Mencoba Python	9
1.6	Identasi	9
1.7	Resume   Kaka Kamaludin	9
1.8	Instalasi	10
1.8.1	Install Anacaconda	10
1.8.2	Install Python	10
1.8.3	Install Python pip	10
1.9	MuhammadRezaSyachrani/1174084	10
1.9.1	Background	10
1.9.2	Problems	11
1.9.3	Objective and Contribution	11
1.9.4	Scoop and Environtment	11
1.10	Instalasi	11
1.10.1	Cara Pemakaian Script dan interpreter python	11
1.10.2	Cara Pemakaian spyder termasuk variable explorer	11
1.11	Mencoba Python	11
1.12	Identasi	12
1.13	AlvanAlvanzah/1174077	12
1.13.1	Background	12
1.13.2	Problems	13
1.13.3	Objective and Contribution	13
1.13.4	Scoop and Environtment	13
1.14	Ilham Muhammad Ariq	13

1.14.1	Mengenal Python dan Anaconda	13
1.15	Cara Pemakaian Python	14
1.16	Instalasi Python	14
1.17	Indentasi	14
1.18	Difa Al Fansha	14
1.18.1	Resume Sejarah Python	14
1.18.2	Perbedaan Python 2 dan Python 3	15
1.18.3	Implementasi dan penggunaan Python pada Perusahaan	15
1.19	Instalasi	16
1.19.1	Cara Pemakaian Script dan interpreter python	16
1.19.2	Cara Pemakaian spyder termasuk variable explorer	16
1.20	Mencoba Python	16
1.21	Identasi	16
1.22	Nurul Izza Hamka	17
1.23	Instalasi	18
1.24	Mencoba Python	19
1.25	Identitas	19
1.26	Aulyardha Anindita	19
1.26.1	Sejarah Python	19
1.26.2	Perbedaan Python 2 dan Python 3	20
1.26.3	Implementasi dan Penggunaan Python di Perusahaan Dunia	20
1.26.4	Instalasi	21
1.26.5	Proses Instalasi Python	21
1.26.6	Proses Instalasi Anaconda	21
1.26.7	Cara Pemakaian Script dan Interpreter Python	21
1.26.8	Script	21
1.26.9	Interpreter Python	22
1.26.10	Cara Pemakaian Spyder Termasuk Variable Explorer	22
1.26.11	Mencoba Python	23
1.26.12	Identasi	23
1.27	Dini Permata Putri 1174053	23
1.27.1	Background	23

## **2 Pemrograman Dasar 35**

2.1	Chandra Kirana Poetra	35
2.1.1	Teori	35
2.1.2	Praktek	38

2.1.3	Keterampilan dan penanganan error	40
2.2	Chapter 2   D. Irga B. Naufal Fakhri D4 TI 2C	40
2.2.1	Teori Praktikum	40
2.2.2	Ketrampilan Pemrograman	46
2.2.3	Ketrampilan Penanganan Error	49
2.3	Bakti Qilan Mufid	50
2.3.1	Teori	50
2.4	Fanny Shafira Damayanti	55
2.4.1	Teori	55
2.4.2	Jenis-Jenis Variable	55
2.4.3	Cara Menampilkan Kode untuk meminta Input dan output nya	56
2.4.4	Operator Dasar Aritmatika	56
2.4.5	Sintax Perulangan	57
2.4.6	Syntax Kondisi	57
2.4.7	Jenis error di Python	58
2.4.8	Cara menggunakan Try Except	58
2.4.9	Keterampilan Pemrograman	59
2.4.10	Keterampilan Penanganan Error	60
2.5	Tugas 2 Alfadian Owen	61
2.5.1	Teori	61
2.6	Muhammad Abdul Gani Wijaya	61
2.6.1	Variable Phyton	61
2.6.2	Input Phyton	62
2.6.3	Operator Aritmatika Phyton	62
2.5.2	Keterampilan Pemrograman	65
2.6.4	Pengulangan Phyton	70
2.6.5	Kondisi Phyton	70
2.6.6	Try Except Phyton	71
2.6.7	Keterampilan Pemograman	71
2.6.8	Program	71
2.6.9	Keterampilan Penanganan Error	72
2.6.10	Program	72
2.7	Tia Nur Candida	72
2.7.1	Jenis Variabel pada Python	72
2.7.2	Cara Menampilkan Output ke Layar	73
2.7.3	Operator Dasar Aritmatika	73
2.7.4	Jenis dan Syntax Perulangan	74

2.7.5	Cara Memakai Syntax untuk Memilih Kondisi	74
2.7.6	Jenis Error yang sering ditemui	75
2.7.7	TryExcept	75
2.8	Nurul Izza Hamka	76
2.8.1	Jenis-Jenis Variabel dan Pemakaian Variable di Kode Pyhton :	76
2.8.2	Tipe String	76
2.8.3	Tipe List	76
2.8.4	Tipe Tuple	77
2.8.5	Kode Untuk Meminta Input Dari User dan Bagaimana Melakukan Output Ke Layar Pada Python ?	77
2.8.6	Operator Dasar Aritmatika, Tambah, Kali, Kurang Bagi dan Bagaimana Mengubah String Ke Integer dan Integer Ke String ?	77
2.8.7	Sintak Perulangan , Jenis-Jenisnya Contoh dan Cara Pakainya Di Python.	78
2.8.8	Cara Pakai Sintak Untuk Memilih Kondisi dan Bagaimana Contoh Sintak Kondisi Di Dalam Kondisi	79
2.8.9	Jenis Error Yang Sering Di Temui Di Python Dalam Mengerjakan Sintak Diatas , dan Bagaimana Cara Mengatasinya ?	79
2.8.10	Cara Memakai Try Except	80
2.8.11	Keterampilan Pemrograman	80
2.8.12	Ketrampilan Penanganan Error	82
2.9	Aulyardha Anindita	82
2.9.1	Jenis-jenis Variabel	82
2.9.2	Bilangan (Number)	82
2.9.3	String	83
2.9.4	List	83
2.9.5	Tuple	84
2.9.6	Set	84
2.9.7	Dictionary	84
2.9.8	Cara Meminta Input Kepada User dan Bagaimana Outputnya	85
2.9.9	Input	85
2.9.10	Output	85
2.9.11	Operator Dasar Aritmetika dan Cara Mengubah String ke Int Begitu Juga Sebaliknya	85

2.9.12	Operator Dasar Aritmetika	85
2.9.13	Cara Mengubah String Ke Int Sebaliknya	86
2.9.14	Perulangan	86
2.9.15	While loop	86
2.9.16	For Loop	87
2.9.17	Kondisi	87
2.9.18	If	87
2.9.19	If else	87
2.9.20	Elif	88
2.9.21	Jenis Error	88
2.9.22	Syntax errors	88
2.9.23	Runtime errors	88
2.9.24	Semantic errors	89
2.9.25	Debugging	89
2.9.26	Exceptions	89
2.9.27	Cara Memakai Try Except	89
2.9.28	Keterampilan Pemrograman	90
2.9.29	Keterampilan Penanganan Error	92
2.10	Difa Al Fansha	92
2.10.1	Teori	92
2.10.2	Ketrampilan Pemrograman	95
2.10.3	Keterampilan Penanganan Error	96
2.11	Ilham Muhammad Ariq	97
2.11.1	Teori	97
2.11.2	Ketrampilan Pemrograman	101
2.11.3	Keterampilan Penanganan Error	102
2.12	Muhammad Reza Syachrani / 1174084	103
2.12.1	Teori	103
2.12.2	Keterampilan Pemrograman	105
2.12.3	Ketrampilan Penanganan Error	107
2.13	Handi Hermawan	107
2.13.1	Teori	107
2.13.2	Keterampilan Pemrograman	108
2.13.3	Keterampilan dan penanganan error	109
2.14	Mochamad Arifqi Ramadhan	110
2.14.1	Variable, pemakaian Variable dan Jenis-Jenis Type data	110
2.14.2	Ketrampilan Pemrograman	112
2.14.3	Ketrampilan Penanganan Error	113

2.15	Fernando Lorencius Sihite	114
2.15.1	Variable, pemakaian Variable dan Jenis-Jenis Type data	114
2.15.2	Ketrampilan Pemrograman	119
2.15.3	Ketrampilan Penanganan Error	121
2.16	Arrizal Furqona Gifary	122
2.16.1	Teori	122
2.16.2	Praktek	124
2.16.3	Keterampilan dan penanganan error	126
2.17	Alvan Alvanzah/1174077	126
2.17.1	Teori	126
2.17.2	Ketrampilan Pemrograman	129
2.17.3	Ketrampilan Penanganan Error	132
2.18	Kaka Kamaludin	132
2.19	Praktek	136
2.20	Praktek	137
2.21	Advent Nopele Olansi Damiahan Sihite	137
2.21.1	Variable, pemakaian Variable dan Jenis-Jenis Type data	138
2.21.2	Ketrampilan Pemrograman	141
2.21.3	Ketrampilan Penanganan Error	143
2.22	Advent nopele olansi damiahan sihite	143
2.22.1	Teori	143
2.22.2	Praktek	145
2.22.3	Keterampilan dan penanganan error	147
2.23	Ainul Filiaini	147
2.23.1	Keterampilan pemrograman	152
2.24	Sekar Jasmine	155
2.24.1	Keterampilan Pemrograman	157
2.25	Dini Permata Putri	157
2.25.1	sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut dikode phyton	157
2.25.2	tuliskan bagaimana kode untuk meminta input dari user dan tuliskan bagaimana melakukan output ke layar	159
2.25.3	tuliskan operator dasar aritmatika, tambah, kali, kurang bagi, dan bagaimana mengubah string ke integer dan integer ke string	159
2.25.4	tuliskan dan jelaskan sintak untuk perulangan, jenis-jenisnya contoh kode dan cara pakainya di phyton	160



2.25.5	tuliskan dan jelaskan cara pakai sintak untuk memilih kondisi, dan bagaimana contoh sintak kondisi didalam kondisi	160
2.25.6	tuliskan apa saja jenis error yang sering ditemui di python dalam mengerjakan sintak diatas dan bagaimana cara mengatasinya	161
2.25.7	tuliskan dan jelaskan cara memakai Try Except	161

### **3 Fungsi dan Kelas 163**

3.1	D. Irga B. Naufal Fakhri	163
3.1.1	Pemahaman Teori	163
3.1.2	Keterampilan Pemograman	166
3.1.3	Ketrampilan Penanganan Error	169
3.2	Nurul Izza Hamka   1174062   Teori	169
3.3	Fanny Shafira Damayanti   1174069	174
3.3.1	Teori	174
3.3.2	keterampilan pemrograman	176
3.3.3	penanganan Error	179
3.4	1174054   Aulyardha Anindita	179
3.4.1	Pemahaman Teori	179
3.4.2	Fungsi	179
3.4.3	Paket	180
3.4.4	Kelas, Objek, Atribut, dan Method	180
3.4.5	Cara Pemanggilan Library Kelas dari Instansiansi	181
3.4.6	Pemakaian Paket dengan Perintah from dan import	181
3.4.7	Pemakaian Paket Fungsi	181
3.4.8	Pemakaian Paket Kelas	181
3.4.9	Keterampilan Pemrograman	182
3.4.10	Penanganan Error	185
3.5	Dini Permata Putri	185
3.6	ainul filiani	189
3.7	Chandra Kirana Poetra	192
3.7.1	Pemahaman Teori	192
3.7.2	Praktek	195
3.7.3	Praktek	198
3.8	Sekar Jasmine	198
3.8.1	Fungsi	198
3.8.2	Kelas	200

3.8.3	Pemahaman Teori	201
3.8.4	Keterampilan Pemrograman	204
3.9	Bakti Qilan Mufid	206
3.9.1	Teori	206
3.9.2	Ketrampilan Pemrograman	208
3.9.3	Ketrampilan Penanganan Error	211
3.10	Advent Nopele Olansi Damiahian Sihite 1174089	212
3.11	Arrizal Furqona Gifary	218
3.12	Muhammad Reza Syachrani / 1174084	225
3.12.1	Pemahaman Teori	225
3.12.2	Ketrampilan Pemrograman	228
3.12.3	Ketrampilan Penanganan Error	231
3.13	Difa Al Fansha	232
3.13.1	Jawaban Teori	232
3.13.2	Jawaban Praktek	233
3.14	Alfadian Owen	237
3.14.1	Teori	237
3.14.2	Praktek	239
3.15	Muhammad Abdul Gani Wijaya	241
3.16	Ilham Muhammad Ariq D4TI2C 1174087	247
3.16.1	Pemahaman Teori	247
3.16.2	Ketrampilan Pemrograman	249
3.16.3	Keterampilan Penanganan Error	254
3.17	Alvan Alvanzah/1174077	254
3.17.1	Pemahaman Teori	254
3.17.2	Ketrampilan Pemrograman	257
3.17.3	Ketrampilan Penanganan Error	260
3.18	Engelbertus Adiputra Mau Leto/1174078	261
3.19	Tia Nur Candida	267
3.20	Mochamad Arifqi Ramadhan	273
3.21	Chapter 3   Kaka Kamaludin	276
3.21.1	Teori	276
3.21.2	Praktek	278
3.21.3	Praktek	282
3.22	Fernando Lorencius Sihite	282
3.23	Handi Hermawan	289
3.23.1	Pemahaman Teori	289
3.23.2	Ketrampilan Pemrograman	290

3.23.3	Ketrampilan Penanganan Error	294
<b>4</b>	<b>Pengelolaan File CSV</b>	<b>297</b>
4.1	D. Irga B. Naufal Fakhri	297
4.1.1	Pemahaman Teori	297
Index		301

# DAFTAR GAMBAR

---

1.1	Klik Next	24
1.2	Klik I Agree	24
1.3	Pilih just me saja	25
1.4	Directory tempat anaconda akan diinstall	25
1.5	Opsi Register	26
1.6	Tunggu hingga selesai	26
1.7	Opsi Tambahan	27
1.8	Finish	27
1.9	Tampilan awal anaconda	28
1.10	IDE Spyder	28
1.11	Menu Console	29
1.12	Tampilan Instalasi 1	29
1.13	Tampilan Instalasi 2	29

1.14	Tampilan Instalasi 3	30
1.15	Tampilan Instalasi 4	30
1.16	Tampilan Instalasi 5	30
1.17	Tampilan Instalasi 6	30
1.18	Tampilan Instalasi 7	31
1.19	Tampilan Instalasi 8	31
1.20	Tampilan Instalasi 9	31
1.21	Tampilan awal Anaconda	31
1.22	Tampilan spider	32
1.23	Tampilan new file pada spider	32
1.24	<code>print("Hello World")</code>	32
1.25	Dialog simpan file	32
1.26	Hasil Program	33
2.1	input dan outputnya	61
2.2	operasi aritmatika	62
2.3	contoh loop/pengulangan	63
2.4	contoh syntax kondisi	64
2.5	contoh try except	65

# DAFTAR TABEL

---



# Listings

---

src/Teori.py	35
src/Teori.py	36
src/Teori.py	36
src/teori.py	37
src/teori.py	37
src/teori.py	38
src/1174079.py	38
src/1174079.py	38
src/1174079.py	39
src/1174079.py	39
src/1174079.py	39
src/1174079.py	39
src/1174079.py	39
src/1174079.py	39
src/1174079.py	40
src/1174079.py	40
src/error.py	40



[illegible]

src/1174066.py	49
src/2err_1174066.py	49
src/1174083.py	53
src/1174083.py	53
src/1174083.py	53
src/1174083.py	54
src/1174083.py	54
src/1174083.py	54
src/1174083.py	54
src/1174083.py	54
src/1174083.py	54
src/1174083.py	54
src/1174083_2err.py	55
src/1174069.py	59
src/1174069.py	59
src/1174069.py	59
src/1174069.py	59
src/1174069.py	60
src/1174069.py	60
src/1174069.py	60
src/1174069.py	60
src/1174069.py	60
src/1174069.py	60
src/err2_1174069.py	60
src/1174071.py	71
src/Gani2.py	72
src/1174062.py	80
src/1174062.py	81
src/1174062.py	81
src/1174062.py	81
src/1174062.py	81
src/1174062.py	81
src/1174062.py	81
src/1174062.py	82
src/1174062.py	82
src/1174062.py	82

src/1174054.py	90
src/1174054.py	91
src/1174054.py	91
src/1174054.py	91
src/1174054.py	91
src/1174054.py	92
src/1174054.py	92
src/1174054.py	92
src/1174054.py	92
src/1174054.py	92
src/1174054.py	92
src/2err_1174054.py	92
src/1174076.py	95
src/1174076.py	95
src/1174076.py	95
src/1174076.py	95
src/1174076.py	95
src/1174076.py	96
src/1174076.py	96
src/1174076.py	96
src/1174076.py	96
src/1174076.py	96
src/1174076.py	96
src/1174076.py	96
src/1174087.py	97
src/1174087.py	98
src/1174087.py	98
src/1174087.py	98
src/1174087.py	99
src/1174087.py	99
src/1174087.py	99
src/1174087.py	100
src/1174087.py	101
src/1174087.py	101
src/1174087.py	101
src/1174087.py	101
src/1174087.py	101
src/1174087.py	102
src/1174087.py	102

src/1174087.py	102
src/1174087.py	102
src/1174087.py	102
src/1174087.py	102
src/2erry_1174087.py	102
src/1174084.py	103
src/1174084.py	104
src/1174084.py	104
src/1174084.py	104
src/1174084.py	104
src/1174084.py	104
src/1174084.py	104
src/1174084.py	104
src/1174084.py	105
src/1174084.py	105
src/1174084.py	106
src/1174084.py	106
src/1174084.py	106
src/1174084.py	106
src/1174084.py	106
src/1174084.py	106
src/1174084.py	106
src/1174084.py	106
src/1174084.py	106
src/1174084.py	107
src/1174084.py	107
src/2err_1174084.py	107
src/1174074.py	112
src/1174074.py	112
src/1174074.py	112
src/1174074.py	113
src/1174074.py	113
src/1174074.py	113
src/1174074.py	113
src/1174074.py	113
src/1174074.py	113
src/1174074.py	113
src/1174074.py	113
src/1174074(2err).py	114
src/1174072.py	119

src/1174072.py	120
src/1174072.py	120
src/1174072.py	120
src/1174072.py	120
src/1174072.py	121
src/1174072.py	121
src/1174072.py	121
src/1174072.py	121
src/1174072.py	121
src/1174072.py	121
src/2err_1174072.py	121
src/teori.py	122
src/teori.py	123
src/teori.py	123
src/teori.py	123
src/teori.py	123
src/teori_1174070.py	124
src/1174070.py	124
src/1174070.py	124
src/1174070.py	125
src/1174070.py	125
src/1174070.py	125
src/1174070.py	125
src/1174070.py	125
src/1174070.py	125
src/1174070.py	125
src/1174070.py	126
src/1174070.py	126
src/1174070.py	126
src/error_1174070.py	126
src/1174077.py	127
src/1174077.py	127
src/1174077.py	127
src/1174077.py	127
src/1174077.py	127
src/1174077.py	127
src/1174077.py	127
src/1174077.py	128
src/1174077.py	128
src/1174077.py	128

src/1174077.py	128
src/1174077.py	128
src/1174077.py	128
src/1174077.py	129
src/1174077.py	129
src/1174077.py	129
src/1174077.py	130
src/1174077.py	130
src/1174077.py	130
src/1174077.py	130
src/1174077.py	130
src/1174077.py	131
src/1174077.py	131
src/1174077.py	131
src/1174077.py	131
src/1174077.py	132
src/2err1174077.py	132
src/1174067_teor1.py	133
src/1174067_teor1.py	133
src/1174067_teor1.py	133
src/1174067_teor1.py	134
src/1174067_teor1.py	135
src/1174067_teor1.py	135
src/1174067_praktek.py	136
src/1174067_praktek.py	136
src/1174067_praktek.py	136
src/1174067_praktek.py	136
src/1174067_praktek.py	136
src/1174067_praktek.py	137
src/1174067_praktek.py	137
src/1174067_praktek.py	137
src/1174067_praktek.py	137
src/1174067_praktek.py	137
src/1174067_2err.py	137
src/1174083.py	141
src/1174083.py	141
src/1174083.py	141

src/1174083.py	141
src/1174083.py	142
src/1174083.py	142
src/1174083.py	142
src/1174083.py	142
src/1174083.py	142
src/1174083.py	142
src/1174083_2err.py	143
src/Teorii.py	143
src/Teorii.py	144
src/Teorii.py	144
src/Teorii.py	144
src/Teorii.py	145
src/1174089_Teorii.py	145
src/1174089.py	145
src/1174089.py	145
src/1174089.py	145
src/1174089.py	146
src/1174089.py	146
src/1174089.py	146
src/1174089.py	146
src/1174089.py	146
src/1174089.py	146
src/1174089.py	146
src/1174089_errorr.py	147
src/1174073.py	152
src/1174073.py	153
src/1174073.py	153
src/1174073.py	153
src/1174073.py	153
src/1174073.py	154
src/1174073.py	154
src/1174073.py	154
src/1174073.py	154
src/1174073.py	154

3.1	Contoh kode fungsi inputan ke fungsi.	163
3.2	Contoh kode fungsi outputan ke fungsi.	163
3.3	Contoh import package atau library.	164
3.4	Contoh import package atau library.	164
3.5	Contoh package atau library.	165
3.6	Contoh import package atau library.	165
3.7	Contoh import package atau library.	165
3.8	Contoh import package atau library didalam folder.	165
3.9	Contoh import package atau library didalam folder.	165
	src/1174066.py	166
	src/1174066.py	166
	src/1174066.py	166
	src/1174066.py	167
	src/1174066.py	167
	src/1174066.py	167
	src/1174066.py	167
	src/1174066.py	167
	src/1174066.py	167
	src/1174066.py	168
	src/main_1174066.py	168
	src/main_1174066.py	168
	src/main_1174066.py	169
	src/1174062.py	170
	src/kelas3lib.py	170
	src/1174062.py	170
	src/1174062.py	171
	src/1174062.py	171
	src/1174062.py	171
	src/1174062.py	171
	src/1174062.py	172
	src/1174062.py	173
	src/1174062.py	173
	src/1174062.py	173
	src/1174062.py	173
	src/1174062.py	174
	src/1174062.py	174
	src/1174062.py	174
	src/1174062.py	174



src/1174069/1174069.py	175
src/1174069/1174069.py	175
src/1174069/1174069.py	175
src/1174069/1174069.py	175
src/1174069/1174069.py	176
src/1174069/main_1174069.py	176
src/1174069/main_1174069.py	176
src/1174069/1174069.py	176
src/1174069/1174069.py	177
src/1174069/1174069.py	177
src/1174069/1174069.py	178
src/1174069/1174069.py	178
src/1174069/1174069.py	178
src/1174069/1174069.py	178
src/1174069/1174069.py	178
src/1174069/1174069.py	178
src/1174069/main_1174069.py	179
src/1174069/main_1174069.py	179
src/1174069/1174069.py	179
src/1174054/1174054.py	179
src/1174054/1174054.py	180
src/1174054/1174054.py	180
src/1174054/1174054.py	180
src/1174054/1174054.py	180
src/1174054/1174054.py	181
src/1174054/1174054.py	181
src/1174054/main_1174054.py	181
src/1174054/main_1174054.py	181
src/1174054/1174054.py	182
src/1174054/1174054.py	183
src/1174054/1174054.py	183
src/1174054/1174054.py	183
src/1174054/1174054.py	184
src/1174054/1174054.py	184
src/1174054/1174054.py	184
src/1174054/1174054.py	184
src/1174054/1174054.py	184

src/1174054/1174054.py	184
src/1174054/main_1174054.py	185
src/1174054/main_1174054.py	185
src/1174054/1174054.py	185
src/1174079/1174079.py	192
src/1174079/1174079.py	192
src/1174079/1174079.py	193
src/1174079/1174079.py	193
src/1174079/1174079.py	193
src/1174079/1174079.py	194
src/1174079/1174079.py	194
src/1174079/1174079.py	194
src/1174079/1174079.py	194
src/1174079/1174079.py	195
src/1174079/1174079.py	195
src/1174079/1174079.py	196
src/1174079/1174079.py	196
src/1174079/1174079.py	196
src/1174079/1174079.py	196
src/1174079/1174079.py	196
src/1174079/1174079.py	197
src/1174079/1174079.py	197
src/1174079/1174079.py	197
src/1174079/main.py	197
src/1174079/main.py	198
src/1174079/1174079.py	198
src/1174075.py	204
src/1174075.py	204
src/1174075.py	205
src/1174075.py	205
src/1174075.py	205
src/1174075.py	205
src/1174075.py	205
src/1174075.py	205
src/1174075.py	205
src/1174083.py	206
src/1174083.py	206
src/1174083.py	207

src/1174083.py	207
src/1174083.py	208
src/1174083.py	208
src/1174083.py	208
src/1174083.py	209
src/1174083.py	209
src/1174083.py	209
src/1174083.py	209
src/1174083.py	210
src/1174083.py	210
src/1174083.py	210
src/1174083.py	210
src/1174083.py	210
src/1174083/main.py	211
src/1174083/main.py	211
src/1174083.py	212
src/1174089.py	212
src/1174089.py	212
src/1174089.py	213
src/1174089.py	213
src/1174089.py	213
src/1174089.py	214
src/1174089.py	214
src/1174089.py	214
src/1174089.py	214
src/1174089.py	215
src/1174089.py	216
src/1174089.py	216
src/1174089.py	216
src/1174089.py	216
src/1174089.py	217
src/1174089.py	217
src/1174089.py	217
src/1174089.py	217
src/main_advent.py	218
src/main_advent.py	218
src/1174089.py	218
src/1174070.py	218

src/1174070.py	218
src/1174070.py	219
src/1174070.py	219
src/1174070.py	219
src/1174070.py	220
src/1174070.py	220
src/1174070.py	220
src/1174070.py	221
src/1174070.py	222
src/1174070.py	222
src/1174070.py	222
src/1174070.py	223
src/1174070.py	223
src/1174070.py	223
src/1174070.py	223
src/1174070.py	224
src/1174070.py	224
src/main_1174070.py	224
src/main_1174070.py	224
src/1174070.py	224
3.10 Fungsi sederhana	225
3.11 Library atau paket luas	225
3.12 Cara penggunaan Library luas	225
3.13 Kelas Library Menghitung	226
3.14 Cara penggunaan kelas Library Menghitung	226
3.15 Contoh kode lain pemakaian paket fungsi from import	227
3.16 Contoh kode pemakaian paket fungsi dimana file library ada di dalam folder	227
3.17 Contoh kode pemakaian paket kelas dimana file library ada di dalam folder	227
src/1174084/1174084.py	228
src/1174084/1174084.py	229
src/1174084/1174084.py	229
src/1174084/1174084.py	229
src/1174084/1174084.py	229
src/1174084/1174084.py	229
src/1174084/1174084.py	230
src/1174084/1174084.py	230

src/1174084/1174084.py	230
src/1174084/1174084.py	230
src/1174084/main.py	230
src/1174084/main.py	231
src/1174084/1174084.py	232
src/1174076/kalkulator.py	232
src/1174076/1174076.py	233
src/1174076/1174076.py	233
src/1174076/1174076.py	234
src/1174076/1174076.py	234
src/1174076/1174076.py	234
src/1174076/1174076.py	235
src/1174076/1174076.py	235
src/1174076/1174076.py	235
src/1174076/1174076.py	236
src/1174076/1174076.py	236
src/1174076/1174076.py	236
src/t_1174091.py	237
src/p_1174091.py	237
src/m_1174091.py	237
src/c_1174091.py	237
src/c_1174091.py	238
src/c2_1174091.py	238
src/t_1174091.py	238
src/kalkulator_1174091.py	239
src/t_1174091.py	239
src/c_1174091.py	239
src/c2_1174091.py	239
src/1174091.py	239
src/1174091.py	240
src/1174091.py	240
src/1174091.py	240
src/1174091.py	240
src/1174091.py	240
src/1174091.py	241
src/1174091.py	241
src/1174091.py	241
src/1174091.py	241

src/1174091.py	241
src/1174071.py	241
src/1174071.py	242
src/1174071.py	242
src/1174071.py	242
src/1174071.py	242
src/1174071.py	243
src/1174071.py	243
src/1174071.py	243
src/1174071.py	243
src/1174071.py	245
src/1174071.py	245
src/1174071.py	245
src/1174071.py	245
src/1174071.py	245
src/1174071.py	246
src/1174071.py	246
src/1174071.py	246
src/1174071.py	246
src/main_gani.py	247
src/main_gani.py	247
src/1174071.py	247
src/1174087/3lib.py	247
src/1174087/main.py	248
src/1174087/kelas3lib.py	248
src/1174087/main.py	248
src/1174087/main.py	249
src/1174087/main.py	249
src/1174087/main.py	249
src/1174087/3lib.py	250
src/1174087/3lib.py	250
src/1174087/3lib.py	251
src/1174087/3lib.py	251
src/1174087/3lib.py	251
src/1174087/3lib.py	251
src/1174087/3lib.py	252
src/1174087/3lib.py	252
src/1174087/3lib.py	252

src/1174087/3lib.py	252
src/1174087/main.py	253
src/1174087/kelas3lib.py	254
src/1174087/3lib.py	254
src/1174077/1174077.py	254
src/1174077/1174077.py	254
src/1174077.py	254
src/1174077/1174077.py	255
src/1174077/1174077.py	255
src/1174077/1174077.py	255
src/1174077/1174077.py	255
src/1174077/1174077.py	255
src/1174077/1174077.py	256
src/1174077/1174077.py	256
src/1174077/1174077.py	256
src/1174077/1174077.py	257
src/1174077/1174077.py	257
src/1174077/1174077.py	258
src/1174077/1174077.py	258
src/1174077/1174077.py	258
src/1174077/1174077.py	258
src/1174077/1174077.py	258
src/1174077/1174077.py	259
src/1174077/1174077.py	259
src/1174077/1174077.py	259
src/1174077/1174077.py	259
src/1174077/main.py	260
src/1174077/main.py	260
src/1174077/1174077.py	261
src/1174078.py	261
src/1174078.py	261
src/1174078.py	261
src/1174078.py	262
src/1174078.py	262
src/1174078.py	262
src/1174078.py	263
src/1174078.py	263
src/1174078.py	263

src/1174078.py	264
src/1174078.py	264
src/1174078.py	265
src/1174078.py	265
src/1174078.py	265
src/1174078.py	265
src/1174078.py	266
src/1174078.py	266
src/1174078.py	266
src/main_engel.py	266
src/main_engel.py	267
src/1174078.py	267
src/1174086.py	267
src/1174086.py	267
src/1174086.py	267
src/1174086.py	268
src/1174086.py	268
src/1174086.py	268
src/1174086.py	269
src/1174086.py	269
src/1174086.py	269
src/1174086.py	270
src/1174086.py	271
src/1174086.py	271
src/1174086.py	271
src/1174086.py	271
src/1174086.py	272
src/1174086.py	272
src/1174086.py	272
src/1174086.py	272
src/1174086.py	272
src/1174086.py	273
src/1174074/1174074.py	273
src/1174074.py	273
src/1174074.py	274
src/1174074.py	274
src/1174074.py	274
src/1174074.py	274
src/1174074.py	275



src/1174074.py	275
src/1174074.py	275
src/1174074.py	275
src/1174074.py	275
src/1174074.py	275
src/1174074.py	275
src/1174074.py	275
src/1174074.py	275
src/1174074.py	276
src/main.py	276
src/main.py	276
src/1174074.py	276
src/1174067/1174067PT.py	276
src/1174067/1174067PT.py	277
src/1174067/1174067PT.py	277
src/1174067/1174067PT.py	277
src/1174067/1174067PT.py	277
src/1174067/1174067PT.py	278
src/1174067/1174067PT.py	278
src/1174067/3lib.py	278
src/1174067/3lib.py	279
src/1174067/3lib.py	279
src/1174067/3lib.py	280
src/1174067/3lib.py	280
src/1174067/3lib.py	280
src/1174067/3lib.py	280
src/1174067/3lib.py	281
src/1174067/3lib.py	281
src/1174067/3lib.py	281
src/1174067/main.py	281
src/1174067/main.py	282
src/1174067/main.py	282
src/1174072/1174072.py	282
src/1174072/1174072.py	282
src/1174072/1174072.py	283
3.18 Contoh penggunaan kelas di python.	283
3.19 Contoh penggunaan Objek di python.	283

3.20 Contoh penggunaan Atribut di python.	283
3.21 Contoh penggunaan Metode di python.	283
src/1174072/1174072.py	284
src/1174072/1174072.py	284
src/1174072/1174072.py	284
src/1174072/1174072.py	284
src/1174072/1174072.py	285
src/1174072/1174072.py	286
src/1174072/1174072.py	286
src/1174072/1174072.py	286
src/1174072/1174072.py	286
src/1174072/1174072.py	286
src/1174072/1174072.py	287
src/1174072/1174072.py	287
src/1174072/1174072.py	287
src/1174072/1174072.py	287
src/1174072/main_1174072.py	288
src/1174072/main_1174072.py	288
src/1174072/1174072.py	289
3.22 Jawaban soal No. 1 Ketrampilan Pemrograman.	290
3.23 Jawaban soal No. 2 Ketrampilan Pemrograman.	291
3.24 Jawaban soal No. 3 Ketrampilan Pemrograman.	291
3.25 Jawaban soal No. 4 Ketrampilan Pemrograman.	291
3.26 Jawaban soal No. 5 Ketrampilan Pemrograman.	292
3.27 Jawaban soal No. 6 Ketrampilan Pemrograman.	292
3.28 Jawaban soal No. 7 Ketrampilan Pemrograman.	292
3.29 Jawaban soal No. 8 Ketrampilan Pemrograman.	292
3.30 Jawaban soal No. 9 Ketrampilan Pemrograman.	293
3.31 Jawaban soal No. 10 Ketrampilan Pemrograman.	293
3.32 Jawaban soal No. 11 Ketrampilan Pemrograman.	293
3.33 Jawaban soal No. 12 Ketrampilan Pemrograman.	294
src/1174066.csv	297
src/1174066.csv	298
src/1174066_csv.py	298
src/1174066_csv.py	299
src/1174066_csv.py	299
src/1174066_csv.py	300
src/1174066_csv.py	300



# FOREWORD

---

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa



# KATA PENGANTAR

---

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat  
Februari, 2019*



# ACKNOWLEDGMENTS

---

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

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

R. M. A.





# ACRONYMS

---

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



# GLOSSARY

---

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



# SYMBOLS

---

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



# INTRODUCTION

---

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

Informatics Research Center  
Bandung, Jawa Barat, Indonesia

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

Rumusnya cuman sebagai contoh aja biar keren[?].

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





# BAB 1

---

## SEJARAH DAN KARAKTERISTIK PYTHON

---

### 1.1 Chandra Kirana Poetra

#### 1.1.1 Sejarah python

Python merupakan suatu bahasa pemrograman yang terinspirasi dari bahasa pemrograman ABC, bahasa pemrograman ABC inilah yang memengaruhi design dan juga pengembangan dari python. Dibuat oleh Guido Van Rossum pada tahun 1989, python pada awalnya dikembangkan pada tahun 1980an pada saat Guido bekerja di CWI (Centrum voor Wiskunde en Informatica) sebagai programmer yang mengimplementasikan bahasa pemrograman bernama ABC, di sana dia mulai mencari bahasa seperti ABC tapi dengan akses mirip seperti AMOEBA, jadi Guido membuat sendiri bahasa pemrograman sederhana yang bisa menutup celah di ABC. Dan akhirnya pada tahun 1991, versi pertama dari python release ke publik

#### 1.1.2 Instalasi Anaconda

1. Pertama anda perlu mendownload terlebih dahulu anacondanya.
2. Visi link ini di <https://www.anaconda.com/distribution/download-section>

3. Setelah download anda selesai, buka file yang anda download tadi
4. Klik next
5. Klik I Agree
6. Pilih Just me dan klik next
7. Pilih directory tempat anaconda akan diinstal lalu next
8. Pilih hanya opsi yang bawah saja
9. Tunggu hingga proses selesai lalu next
10. Opsi tambahan untuk instal visual studio code, skip saja
11. Klik saja finish

### 1.1.3 Spyder

1. Setelah tadi install anaconda, buka aplikasinya.
2. Biasanya, spyder sudah terinstall bersamaan dengan anaconda, klik launch pada spyder
3. ketika di menu kiri, print("Hellow World") untuk percobaan pertama lalu klik simbo panah hijau untuk run, maka anda akan melihat hasilnya di console
4. Output akan dihasilkan di sini

## 1.2 Chapter 1 | D. Irga B. Naufal Fakhri D4 TI 2C

### 1.2.1 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.

### 1.2.2 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

## 1.2.3 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 Integer

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.

### 1.2.4 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 kodenya 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/](http://www.python.org/about/success/)

### 1.2.5 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](http://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
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'
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'
13. Tekan 'Finish' untuk menyelesaikan instalasi

### 1.2.6 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

### 1.2.7 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

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

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

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

## 1.3 Bakti Qilan Mufid

### 1.3.1 Resume Sejarah Python

Bahasa pemrograman Python dirilis pertama kali oleh Guido van Rossum di tahun 1991, yang sudah dikembangkan sejak tahun 1989. Awal pemilihan nama Python tidak secara langsung berasal dari nama ular piton, tapi sebuah acara humor di BBC pada era 1980an dengan judul "Monty Python's Flying Circus". Monty Python adalah kelompok lawak yang membawakan acara tersebut. Kebetulan Guido van Rossum adalah penggemar dari acara ini. Pada tahun 1994, Python 1.0 dirilis, yang diikuti dengan Python 2.0 pada tahun 2000. Python 3.0 keluar pada tahun 2008.

### 1.3.2 Perbedaan Python 2 dan Python 3

#### 1.3.2.1 Python 2

Dipublikasikan pada akhir tahun 2000, Python 2 dinilai lebih transparan dan inklusif untuk pengembangan software ketimbang versi sebelumnya. Hal ini didukung dengan adanya PEP – Python Enhancement Proposal, sebuah spesifikasi teknis yang menjadi tuntunan informasi untuk penggunaanya dan menggambarkan fitur baru pada Python itu sendiri. Sebagai tambahan, Python 2 dilengkapi dengan berbagai fitur programatikal seperti cycle-detecting garbage collector untuk mengotomasi manajemen memori, peningkatan dukungan untuk Unicode, list comprehension untuk membuat sebuah list berdasarkan list yang sudah ada. Unifikasi pada tipe data Python dan class ke satu hirarki terjadi pada rilis Python 2.2

#### 1.3.2.2 Python 3

Python 3 diharapkan sebagai masa depan Python dan merupakan versi yang saat tulisan ini dibuat masih aktif dikembangkan. Python 3 sendiri adalah versi dengan



banyak perubahan yang dirilis akhir tahun 2008. Fokus dari Python 3 itu sendiri adalah untuk melakukan perapian pada codebase dan menghapuskan duplikasi (redundancy). Perubahan terbesar pada Python 3 termasuk memasukkan statemen print ke dalam built-in function. Awalnya, Python 3 mengalami hambatan pada pengadopsiannya. Itu akibat dari tidak adanya backwards compatibility dengan Python 2. Hal ini membuat pengguna Python sangat berat hati untuk pindah ke versi 3 ini. Tambahnya, banyak sekali library yang hanya tersedia untuk Python 2., tapi setelah tim pengembangan di balik Python 3 telah berulang kali menjelaskan bahwa dukungan terhadap Python 2 akan segera dihentikan, dan semakin banyak library disalin ke Python 3, maka penerapan Python 3 semakin lama semakin meningkat.

### 1.3.3 Implementasi dan penggunaan Python pada Perusahaan

daftar berikut adalah beberapa perusahaan yang menggunakan Python, diantaranya:

1. Google adalah perusahaan besar yang menggunakan banyak kode Python di dalam mesin pencariannya. Dan mesin pencari google adalah yang paling terkenal di dunia.
2. Youtube, situs video terbesar dan terpopuler di dunia, sebagian besar kodenya 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.

## 1.4 Instalasi

### 1.4.1 Cara Pemakaian Script dan interpreter python

### 1.4.2 Cara Pemakaian spyder termasuk variable explorer

## 1.5 Mencoba Python

Untuk memulai suatu pemrograman, kita akan awali dengan membuat sebuah hello world. Di Python, cukup mudah untuk membuat sebuah hello world. Silahkan buat sebuah file dengan nama helloworld.py kemudian buat kode berikut di dalam file tersebut:

```
print "Hello world..."
```

Sekarang mari kita eksekusi file tersebut di konsol dengan perintah berikut: python helloworld.py

```
Hello world...
```

## 1.6 Identasi

Ketika menulis kode program Python perlu memperhatikan indentasi, karena kode program Python distrukturkan berdasarkan indentasi. Kode program yang berada pada sisi kiri yang sama maka dibaca sebagai satu blok, untuk membuat sub blok maka cukup dengan memberikan jarak spasi atau tab ke kanan. Soal indentasi ini akan lebih jelas ketika pembahasan tentang pencabangan, perulangan, fungsi, class, dan materi yang lain yang membutuhkan penulisan kode program bersarang. Contohnya adalah sebagai berikut: import sys

```
if len(sys.argv) < 2:

    print("Harap memasukkan argumen.")

    sys.exit(1)
```

## 1.7 Resume | Kaka Kamaludin

Python dibuat oleh Guido van Rossum yang dirilis pertama kali pada tahun 1992, 'python 1.0' dirilis pada januari 1994 dan versi terakhir yang ini dirilis saat ini adalah 'Python 3.7' pada 27 Juni 2018. Nama Python sendiri diambil dari acara televisi Monty Python's Flying Circus. Python mendukung multi paradigma pemrograman. pengembangan Python masih dilakukan oleh Team Guido dan

Python Software Foundation sebagai pemegang hak cipta intelektual Python sejak versi 2.1 . perbedaan antara python2 dan python3 contohnya pada statement print, pada python2 :  
`print "Hello World"`  
sedangkan pada python3 :  
`print ("Hello World")`

## 1.8 Instalasi

### 1.8.1 Install Anacanda

1. Download Anaconda3 <https://www.anaconda.com/distribution/>
2. `'bash Anaconda3-2018.12-Linux-x86-64.sh`
3. review license agreement
4. agree license agreement, text 'yes'
5. lokasi install default di /root/anaconda3
6. tambahkan lokasi di /root/.bashrc, text 'yes'
7. selesai

### 1.8.2 Install Python

1. `apt-get install python3`
2. selesai

### 1.8.3 Install Python pip

1. `apt-get install python3-pip`
2. selesai

## 1.9 MuhammadRezaSyachrani/1174084

### 1.9.1 Background

Python merupakan salah satu bahasa pemrograman tingkat tinggi (high level language) yang dikembangkan oleh Guido van Rossum pada tahun 1989 dan diperkenalkan untuk pertama kalinya pada tahun 1991 di Scitchting Mathematisch Centrum (CWI). Python dirancang untuk memberikan kemudahan bagi programmer melalui segi efisiensi waktu, kemudahan dalam pengembangan dan kompatibilitas dengan

sistem. Python bisa digunakan untuk membuat aplikasi standalone (berdiri sendiri) dan pemrograman script (scripting programming).

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. Sisi utama yang membedakan Python dengan bahasa pemrograman lain adalah dalam hal aturan penulisan kode program. Programmer yang menggunakan selain python dapat dibingungkan dengan aturan indentasi, tipe data, tuple, dan dictionary. Python memiliki kelebihan tersendiri dibandingkan dengan bahasa lain terutama dalam hal penanganan modul, ini yang membuat beberapa programmer menyukai python. Selain itu python merupakan salah satu produk yang opensource, free, dan multiplatform.

### **1.9.2 Problems**

- Bagaimana cara mengimplementasikan bahasa pemrograman python

### **1.9.3 Objective and Contribution**

#### **1.9.3.1 Objective**

- Dapat mengimplementasikan bahasa pemrograman python

#### **1.9.3.2 Contribution**

- Dapat membangun suatu aplikasi dan alat yang mengimplementasikan bahasa pemrograman python

### **1.9.4 Scoop and Environment**

- Mengimplementasikan Python dalam pemrograman

## **1.10 Instalasi**

### **1.10.1 Cara Pemakaian Script dan interpreter python**

### **1.10.2 Cara Pemakaian spyder termasuk variable explorer**

## **1.11 Mencoba Python**

Untuk memulai suatu pemrograman, kita akan awali dengan membuat sebuah hello world. Di Python, cukup mudah untuk membuat sebuah hello world. Silahkan buat sebuah file dengan nama helloworld.py kemudian buat kode berikut di dalam file tersebut:

```
print "Hello world..."
```

Sekarang mari kita eksekusi file tersebut di konsol dengan perintah berikut: `python helloworld.py`

```
Hello world...
```

## 1.12 Identasi

Ketika menulis kode program Python perlu memperhatikan indentasi, karena kode program Python distrukturkan berdasarkan indentasi. Kode program yang berada pada sisi kiri yang sama maka dibaca sebagai satu blok, untuk membuat sub blok maka cukup dengan memberikan jarak spasi atau tab ke kanan. Soal indentasi ini akan lebih jelas ketika pembahasan tentang pencabangan, perulangan, fungsi, class, dan materi yang lain yang membutuhkan penulisan kode program bersarang. Contohnya adalah sebagai berikut: `import sys`

```
if len(sys.argv) < 2:

    print("Harap memasukkan argumen.")

    sys.exit(1)
```

## 1.13 AlvanAlvanzah/1174077

### 1.13.1 Background

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

Python adalah bahasa pemrograman yang bersifat open source. Bahasa pemrograman ini dioptimalisasikan untuk software quality, developer productivity, program portability, dan component integration. Python telah digunakan untuk mengembangkan berbagai macam perangkat lunak, seperti internet scripting, systems programming, user interfaces, product customization, numeric programming dll. Python saat ini telah menduduki posisi 4 atau 5 bahasa pemrograman paling sering digunakan di seluruh dunia. Menggunakan alat pihak ketiga, kode Python dapat dikemas ke dalam program yang dapat dieksekusi mandiri. Penerjemah python tersedia untuk banyak sistem operasi.

### 1.13.2 Problems

- Bagaimana cara agar memahami bahasa pemrograman python

### 1.13.3 Objective and Contribution

#### 1.13.3.1 Objective

- Dapat memahami bahasa pemrograman Python

#### 1.13.3.2 Contribution

- Dapat mengimplementasikan bahasa pemrograman python

### 1.13.4 Scoop and Environment

- Mempelajari tentang bahasa pemrograman python

## 1.14 Ilham Muhammad Ariq

### 1.14.1 Mengenal Python dan Anaconda

#### 1. SEJARAH PYTHON

Python dikembangkan pada tahun 1990 oleh Guido van Rossum di CWI Amsterdam sebagai kelanjutan dari bahasa pemrograman ABC.

Tahun 1995, Guido pindah ke CNRI di Virginia Amerika sambil terus melanjutkan pengembangan Python. Versi terakhir yang dikeluarkan adalah 1.6. Tahun 2000, Guido dan para pengembang inti Python pindah ke BeOpen.com yang merupakan sebuah perusahaan komersial dan membentuk BeOpen PythonLabs. Python 2.0 dikeluarkan oleh BeOpen. Setelah mengeluarkan Python 2.0, Guido dan beberapa anggota tim PythonLabs pindah ke Digital Creations.

Saat ini pengembangan Python terus dilakukan oleh sekumpulan pemrogram yang dikoordinir Guido dan Python Software Foundation. Python Software Foundation adalah sebuah organisasi non-profit yang dibentuk sebagai pemegang hak cipta intelektual Python sejak versi 2.1 dan dengan demikian mencegah Python dimiliki oleh perusahaan komersial. Saat ini distribusi Python sudah mencapai versi 2.7.14 dan versi 3.6.3

Nama Python dipilih oleh Guido sebagai nama bahasa ciptaannya karena kecintaan Guido pada acara televisi Monty Python's Flying Circus.

#### 2. PERBEDAAN PYTHON 2 DAN 3

Python versi 2 merupakan versi yang banyak digunakan saat ini, baik dilingkungan produksi dan pengembangan. Sementara Python versi 3 adalah pengembangan lanjutan dari versi 2. Python 3 memiliki lebih banyak fitur dibandingkan Python 2. Untuk membuka Python 2 kita hanya menggunakan perintah `python` saja, sedangkan Python 3 menggunakan perintah `python3`.

## 1.15 Cara Pemakaian Python

Untuk menuliskan script Python, cukup buka terminal yang ingin digunakan misalkan cmd dan ketikan pyhton contoh syntax dasar Hello word

```
ketikan :
print('hello word')
outputnya :
hello word
```

## 1.16 Instalasi Python

1. Download file python terlebih dahulu
2. Kemudian install file yg telah didownload
3. Pilih saja 'Install for all users' agar bisa dipakai untuk semua user di komputernya
4. Tentukan lokasi python akan diinstal, kemudian klik next.
5. Pada tahapan ini, kita akan menentukan fitur-fitur yang akan diinstal. Jangan lupa untuk mengaktifkan 'Add python.exe to path' agar perintah python dikenali pada CMD (Command Prompt).

## 1.17 Indentasi

Python memanfaatkan indentasi untuk membuka/menutup fungsi tersebut. jika melakukan coding dengan notepad++. Pada notepad++ setting perintah Tab menjadi indentasi 4 karakter spasi, dengan memilih Setting -> Preferences ceklist box Replace by Space dengan Tab Size = 4. Dengan begitu yang tinggal mengklik 'Tab' pada keyboard untuk melakukan indentasi kedalam, tanpa harus mengisi dengan spasi sebanyak 4 kali.

## 1.18 Difa Al Fansha

### 1.18.1 Resume Sejarah Python

Bahasa pemrograman Python dirilis pertama kali oleh Guido van Rossum di tahun 1991, yang sudah dikembangkan sejak tahun 1989. Awal pemilihan nama Python tidak secara langsung berasal dari nama ular piton, tapi sebuah acara humor di BBC pada era 1980an dengan judul "Monty Python's Flying Circus". Monty Python adalah kelompok lawak yang membawakan acara tersebut. Kebetulan Guido van Rossum adalah penggemar dari acara ini. Pada tahun 1994, Python 1.0 dirilis, yang diikuti dengan Python 2.0 pada tahun 2000. Python 3.0 keluar pada tahun 2008.

## 1.18.2 Perbedaan Python 2 dan Python 3

### 1.18.2.1 *Python 2*

Dipublikasikan pada akhir tahun 2000, Python 2 dinilai lebih transparan dan inklusif untuk pengembangan software ketimbang versi sebelumnya. Hal ini didukung dengan adanya PEP – Python Enhancement Proposal, sebuah spesifikasi teknis yang menjadi tuntunan informasi untuk penggunaanya dan menggambarkan fitur baru pada Python itu sendiri. Sebagai tambahan, Python 2 dilengkapi dengan berbagai fitur programatikal seperti cycle-detecting garbage collector untuk mengotomasi manajemen memori, peningkatan dukungan untuk Unicode, list comprehension untuk membuat sebuah list berdasarkan list yang sudah ada. Unifikasi pada tipe data Python dan class ke satu hirarki terjadi pada rilis Python 2.2

### 1.18.2.2 *Python 3*

Python 3 diharapkan sebagai masa depan Python dan merupakan versi yang saat tulisan ini dibuat masih aktif dikembangkan. Python 3 sendiri adalah versi dengan banyak perubahan yang dirilis akhir tahun 2008. Fokus dari Python 3 itu sendiri adalah untuk melakukan perapian pada codebase dan menghapuskan duplikasi (redundancy). Perubahan terbesar pada Python 3 termasuk memasukkan statemen print ke dalam built-in function. Awalnya, Python 3 mengalami hambatan pada pengadopsiannya. Itu akibat dari tidak adanya backwards compatibility dengan Python 2. Hal ini membuat pengguna Python sangat berat hati untuk pindah ke versi 3 ini. Tambahannya, banyak sekali library yang hanya tersedia untuk Python 2., tapi setelah tim pengembangan di balik Python 3 telah berulang kali menjelaskan bahwa dukungan terhadap Python 2 akan segera dihentikan, dan semakin banyak library disalin ke Python 3, maka penerapan Python 3 semakin lama semakin meningkat.

## 1.18.3 Implementasi dan penggunaan Python pada Perusahaan

daftar berikut adalah beberapa perusahaan yang menggunakan Python, diantaranya:

1. Google adalah perusahaan besar yang menggunakan banyak kode Python di dalam mesin pencariinya. Dan mesin pencari google adalah yang paling terkenal di dunia.
2. Youtube, situs video terbesar dan terpopuler di dunia, sebagian besar kodenya 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 mikrokontroler, menggunakan Python sebagai bahasa utamanya.

## 1.19 Instalasi

### 1.19.1 Cara Pemakaian Script dan interpreter python

### 1.19.2 Cara Pemakaian spyder termasuk variable explorer

## 1.20 Mencoba Python

Untuk memulai suatu pemrograman, kita akan awali dengan membuat sebuah hello world. Di Python, cukup mudah untuk membuat sebuah hello world. Silahkan buat sebuah file dengan nama helloworld.py kemudian buat kode berikut di dalam file tersebut:

```
print "Hello world..."
```

Sekarang mari kita eksekusi file tersebut di konsol dengan perintah berikut: python helloworld.py

```
Hello world...
```

## 1.21 Identasi

Ketika menulis kode program Python perlu memperhatikan indentasi, karena kode program Python distrukturkan berdasarkan indentasi. Kode program yang berada pada sisi kiri yang sama maka dibaca sebagai satu blok, untuk membuat sub blok maka cukup dengan memberikan jarak spasi atau tab ke kanan. Soal indentasi ini

akan lebih jelas ketika pembahasan tentang pencabangan, perulangan, fungsi, class, dan materi yang lain yang membutuhkan penulisan kode program bersarang. Contohnya adalah sebagai berikut: `import sys`

```
if len(sys.argv) < 2:

    print("Harap memasukkan argumen.")

    sys.exit(1)
```

## 1.22 Nurul Izza Hamka

**1.22.0.1 Resume Sejarah Python** Python merupakan bahasa pemrograman yang bersifat interpretative. Python dikembangkan oleh Guido Van Rossum pada tahun 1990 di CWI, Amsterdam sebagai kelanjutan dari Bahasa pemrograman ABC. Nama Python dipilih oleh Guido sebagai nama bahasa ciptaanya karena kecintaannya pada televisi Monthly Python's Flying Circus. Sekarang, distribusi python sudah mencapai versi 2.6.1 sampai dengan 3.0.

Pada tahun 1995 Guido pindah ke CNRI di Virginia Amerika sambil terus melanjutkan pengembangan Python. Versi terakhir yang dikeluarkan adalah 1.6. Tahun 2000, setelah itu Guido dan timnya berpindah lagi ke BeOpen.com dan dari sini mereka mengeluarkan Python versi 2.0.

Sampai sekarang pengembangan python terus dilakukan oleh pada pemrogram yang diambil alih oleh Guido dan juga Python Software Foundation.

**1.22.0.2 Perbedaan Python 2 dan 3** Python versi 2 merupakan versi yang dikembangkan pada tahun 2000 dan yang paling banyak digunakan saat ini, baik dilingkungan produksi dan pengembangan, dan untuk membuka python 2 ini tinggal ketik python.

Sementara Python versi 3 adalah pengembangan lanjutan dari versi 2, yang terakhir rilis pada tahun 2008. Python 3 memiliki lebih banyak fitur dibandingkan Python 2, versi 3 ini ketika akan dibuka maka akan menggunakan perintah `python3`. Perubahan terbesar pada Python 3 termasuk memasukkan statemen `print` ke dalam built-in function.

Adapun perbedaan dari segi `print`:

Untuk Python 2 ketikkan `input` tidak menggunakan kurung biasa, namun pakek kurung juga bias dan dia menghasilkan atau mencetak satu baris. Sedangkan untuk python 3 harus menggunakan tanda kurung dan juga akan menghasilkan atau mencetak satu baris.

**1.22.0.3 Implementasi Python Di Perusahaan Dunia** Ada beberapa perusahaan terkenal dunia yang menggunakan bahasa Python, yaitu :

1. Google adalah perusahaan besar yang menggunakan banyak kode Python di dalam mesin pencariannya.
2. Youtube, situs video terbesar dan terpopuler di dunia, sebagian besar kodenya ditulis dalam bahasa Python.
3. Dropbox, menggunakan python baik di sisi server maupun di sisi pengguna layanannya.
4. ESRI, produsen terkenal pembuat software pemetaan GIS banyak menggunakan Python di produknya.
5. NASA, badan antariksa Amerika ini menggunakan Python untuk bidang sainsnya.

## 1.23 Instalasi

**1.23.0.1 Instalasi Python** Cara Install Python di Windows

1. Unduh Python versi 3.7,
2. Buka file Python yang sudah di unduh,
3. Sebelum install pilih atau centang 'Add Python to PATH' di ujung kiri bawah
4. Pilih user, ada baiknya pilih 'Install For All Users' agak dapat di gunakan oleh semua user komputer,
5. Pilih Lokasi untuk menyimpan aplikasi Python,
6. Kemudian Next untuk melanjutkan,
7. Install untuk aplikasi Python Finish.

**1.23.0.2 Instalasi Anaconda** Cara Install Python di Windows

1. Download aplikasi Anaconda di Windows,
2. Buka file Anaconda yang sudah di unduh,
3. Klik 'I Agree' pada perjanjian Lisensi,
4. Pilih user, ada baiknya pilih 'All Users' agak dapat di gunakan oleh semua user komputer,
5. Pilih Lokasi untuk menyimpan aplikasi Anaconda,
6. Ketika muncul dua pilihan, cukup centang 'Register Anaconda',
7. Install untuk Anaconda Complete.

**1.23.0.3 Script dan Interpreter Python** Menggunakan Interpreter Python:

1. Jalankan interpreter. Buka Command Prompt, ketik python pada prompt dan tekan Enter. jika Python telah berhasil akan muncul tanda seperti ini (`>>>`).
2. Untuk menampilkan bantuan informasi kita dapat menggunakan perintah `help()` dan dapat dilakukan dengan 2 cara, yaitu dengan `help(int)`. Kedua adalah dengan mengetikkan perintah `help()` didalam interpreter yang akan merubah mode interpreter '`>>>`' menjadi mode '`help_`'.

**1.23.0.4 Pemakaian Spyder Variable explorer** Explorer Variabel menunjukkan konten namespace, Variable Explorer Spyder menawarkan dukungan bawaan untuk mengedit daftar, string, dan kamus. Variable Explorer memiliki editor khusus,

seperti :

- Integers
- Floats
- Complex numbers
- Strings

## 1.24 Mencoba Python

Membuat file `lat1.py` menggunakan teks Editor, kemudian ketikkan 'Hello Word!'. setelah itu simpan file `lat1.py` dan run menggunakan terminal.

## 1.25 Identitas

**1.25.0.1 Identitas dan Cara Menanganinya** Excepetions berbeda dengan syntax error. Ketika tidak menangani exceptions dengan tepat, program-nya akan keluar secara paksa karena dia tidak tahu apa yang perlu dilakukan dalam kasus tersebut. Menangani banyak exception menggunakan satu klausa `except` dengan melwatkan exception tersebut ke klausa sebagai sebuah tuple.

## 1.26 Aulyardha Anindita

### 1.26.1 Sejarah Python

Python adalah bahasa pemrograman yang mendukung multi paradigma pemrograman seperti pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Python juga pada umumnya digunakan sebagai bahasa skrip walaupun pada praktiknya penggunaan python lebih luas.

Bahasa pemrograman python dikembangkan oleh Guido van Rossum pada tahun 1990 di Stichting Mathematisch Centrum (CWI), Amsterdam yang merupakan kelanjutan dari bahasa pemrograman ABC dimana merupakan versi terakhir yang dikeluarkan oleh CWI adalah 1.2. Nama Python sendiri dipilih oleh Guido sebagai nama bahasa ciptaannya karena rasa cintanya pada acara televisi Monty Python's Flying Circus.

Sekitar tahun 1995, Guido pindah ke CNRI di Virginia Amerika sambil dia melanjutkan pengembangan Python. Disinilah dia merilis beberapa versi dari python. Versi terakhir yang dikeluarkan saat itu adalah 1.6. Dan sekitar tahun 2000, Guido dan timnya pindah ke BeOpen.com. BeOpen.com merupakan sebuah perusahaan komersial yang membentuk BeOpen PythonLabs. Python 2.0 dikeluarkan oleh BeOpen. Setelah mereka mengeluarkan Python 2.0, Guido dan beberapa anggota tim PythonLabs pindah ke DigitalCreations. Dan pada tahun 2001, dibentuklah Organisasi Python yaitu Python Software Foundation(PSF). PSF adalah suatu organisasi nirlaba yang dibuat untuk semua hal yang berkaitan dengan hak intelektual Python.

### 1.26.2 Perbedaan Python 2 dan Python 3

Pada python 2 kita bisa menggunakan tanda kurung atau tidak sedangkan pada python3 kita wajib menggunakan tanda kurung, jika tidak maka kita akan mendapatkan hasil error.

Pada python 2 dalam melakukan sebuah inputan kita harus menggunakan input(teks), sedangkan untuk python 3 kita hanya perlu menggunakan syntax input(teks) saja.

Pada python2 dilengkapi dengan berbagai fitur programatikal seperti sysle-detecting garbage collector untuk mengotomasi manajemen memori, peningkatan dukungan untuk Unicode, list comprehension untuk membuat list berdasarkan dari list yang sudah ada. Sedangkan pada python 3 adalah melakukan perapian pada codebase dan menghapuskan duplikat atau redundansi.

### 1.26.3 Implementasi dan Penggunaan Python di Perusahaan Dunia

1. Google adalah perusahaan besar yang menggunakan banyak kode Python di dalam mesin pencariinya. Dan mesin pencari google adalah yang paling terkenal di dunia.
2. Youtube, situs video terbesar dan terpopuler di dunia, sebagian besar kodenya 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. Pixar, banyak menggunakan Python dalam animasi movie.
11. Blender, Maya, software pembuat animasi 3D terkenal, menggunakan Python sebagai salah satu bahasa skrip pemrogramannya.
12. Raspberry Pi, komputer mini yang banyak digunakan sebagai mikrokontroller, menggunakan Python sebagai bahasa utamanya.
13. ESRI, produsen terkenal pembuat software pemetaan GIS banyak menggunakan Python di produknya.

### 1.26.4 Instalasi

#### 1.26.5 Proses Instalasi Python

1. Download terlebih dahulu file Python nya. jika sudah didownload, maka kita masuk ke proses instalasi dengan mendouble klik pada file python nya.
2. Centang Install launcher for all user untuk mengaktifkan python pada semua user Windows dan centang Python 3.6 to PATH untuk menambah path command Python. Kemudian klik Install Now. Klik Yes saat muncul notifikasi User Account Control.
3. Tunggu sampai proses instalasi selesai
4. Instalasi python berhasil
5. Untuk mengetahui apakah python nya sudah berjalan apa tidak yaitu dengan masuk ke Command Prompt dan ketikkan Python, jika ada berarti sudah tersambung

#### 1.26.6 Proses Instalasi Anaconda

1. Download terlebih dahulu file Anaconda. Jika sudah didownload, maka langsung saja masuk ke proses instalasi dengan mendouble klik pada file Anacondanya.
2. Akan muncul tampilan pertama, lalu pilih next
3. Kemudian read lisensi dan klik I Agree
4. Kemudian pilih tempat penyimpanan nya, bagusnya yang default saja lalu pilih next
5. Kemudian pilih add anaconda to PATH atau tidak. Pilih apakah akan mendaftarkan Anaconda sebagai default Python 3.7?. Kecuali kita berencana menginstal dan menjalankan beberapa versi Anaconda, atau beberapa versi Python, biarkan default dan biarkan kotak ini dicentang. kemudian klik next.
6. Klik tombol Install. Jika Kita ingin melihat packages Anaconda yang sedang dipasang, klik Show Details, lalu pilih Next.
7. Untuk menginstal VS Code, klik tombol Install Microsoft VS Code. Setelah instalasi selesai, klik tombol Next Atau untuk menginstal Anaconda tanpa VS code, klik tombol skip. Memasang VS code dengan pemasang Anaconda membutuhkan koneksi internet. Pengguna offline mungkin dapat menemukan pemasang offline VS Code dari Microsoft.
8. Setelah proses instalasi berhasil, Kita akan melihat kotak dialog "Thanks for installing Anaconda3" lalu pilih Finish

#### 1.26.7 Cara Pemakaian Script dan Interpreter Python

##### 1.26.8 Script

1. Gunakan teks editor untuk menulis skrip
2. kemudian simpan dengan nama yang kalian inginkan
3. Kemudian untuk menjalankan skripnya, gunakan perintah berikut: `python nama_krip.py`

4. Skrip python diterjemahkan ke dalam kode biner oleh (intepreter) python, sehingga komputer dapat mengerti arti perintah tersebut sehingga komputer mengerjakan perintah tersebut.

### 1.26.9 Interpreter Python

1. Membuka interpreter python pada submenu dari Aplikasi Python yang terdapat pada All Programs. Untuk keluar dari interpreter, ketik Ctrl+D / Ctrl+Q atau menggunakan perintah quit().
2. help() untuk menampilkan bantuan informasi kita dapat menggunakan perintah help(). Perintah help() dapat digunakan dengan 2 cara, yaitu dengan menggunakannya beserta object yang diinginkan, contohnya help(int). Kedua adalah dengan mengetikan perintah help() didalam interpreter yang akan merubah mode interpreter 'i' menjadi mode 'help:'.
3. Setelah kita berada dalam mode 'help:' kita dapat langsung menggunakannya dengan memasukan keywords atau object yang diinginkan. Contohnya adalah keywords.
4. Jika kita mengetik salah satu keywords, maka interpreter akan memberikan informasi yang bersangkutan dengan keywords tersebut. Contohnya adalah if.
5. Disamping keywords kita juga dapat mendapatkan informasi tentang topics. Untuk mengetahui macam-macam topics, cukup dengan mengetikan perintah topics kedalam mode 'help:'
6. Perintah topics memberikan informasi yang berguna kepada kita mengenai bahasa pemrograman python. Selain cara help() diatas, kita juga dapat menggunakan cara yang kedua yaitu langsung bersama object yang diinginkan misalnya adalah string. Untuk mencobanya ketik help(str)
7. Dengan menggunakan perintah help(str) kita dapat mengexplore object string beserta atribut dan method-method yang dimilikinya. Dan hal tersebut berlaku untuk semua keywords python. Tanda titik dua diatas menandakan informasi yang disampaikan masih bersambung, untuk mengetahui informasi selanjutnya tekan space sampai muncul kata (END)

### 1.26.10 Cara Pemakaian Spyder Termasuk Variable Explorer

1. Variable Explorer Spyder menawarkan dukungan bawaan untuk mengedit daftar, string, kamus, array NumPy, Pandas DataFrames, dan banyak lagi, dan dapat juga histogram, plot, atau bahkan menampilkan beberapa di antaranya sebagai gambar RGB.
2. Variable Explorer memiliki editor khusus untuk serangkaian objek Python internal dan pihak ketiga yang umum, dan dapat melihat, mengedit, dan mengintrospeksi objek paling arbitrer secara mendalam melalui Penjelajah Objek yang lebih umum

### 1.26.11 Mencoba Phyton

```
print ("Hello World Python!")
```

### 1.26.12 Identasi

Indentasi merupakan keluarnya suatu teks/naskah dari batas kiri, batas kanan atau keduanya. Kita dapat mengatur indentasi hanya pada baris pertama (first line indent) dari paragraf atau mengatur indentasi mengandung (hanging).

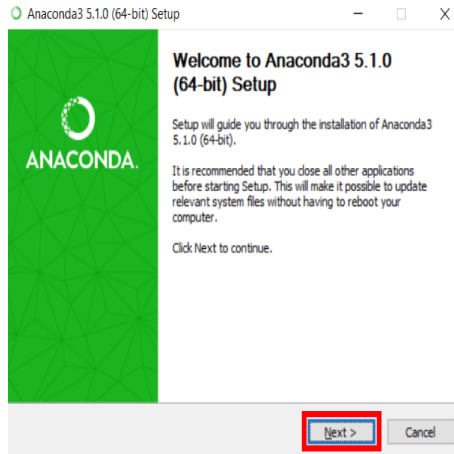
Cara mengatasinya yaitu dengan mensetting ulang indent nya, mengatur panjang indent. sehingga ruas kiri dan kanan nya menjadi rata.

## 1.27 Dini Permata Putri 1174053

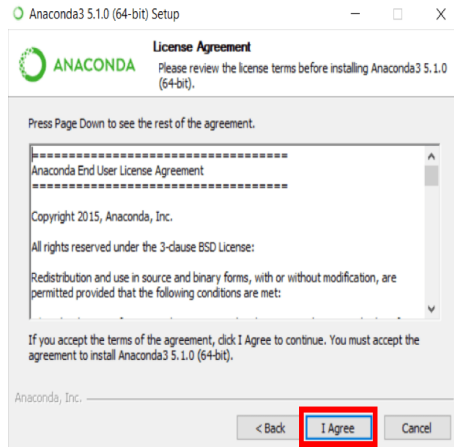
### 1.27.1 Background

Bahasa pemrograman saat ini jumlahnya sangat banyak. Python merupakan salah satu bahasa pemrograman populer yang digunakan oleh banyak developer. Menurut survei bahasa pemrograman versi [www.tiobe.com](http://www.tiobe.com), Python berada di peringkat ke-5 pada tahun 2016. Selain itu, Python juga bisa digunakan untuk enterprise. Dalam tingkatan bahasa pemrograman, Python termasuk high level language. Python menjadi salah satu bahasa pemrograman yang dapat digunakan untuk membangun aplikasi, baik itu berbasis desktop, web ataupun berbasis mobile.

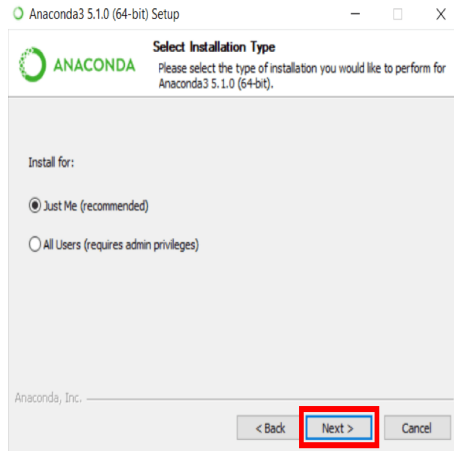




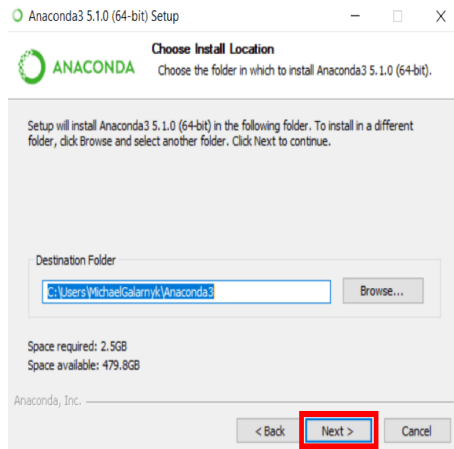
**Gambar 1.1** Klik Next



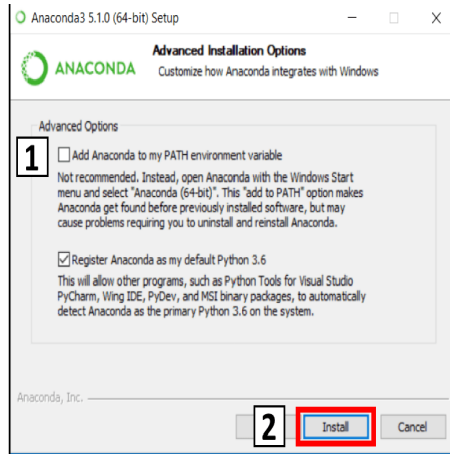
**Gambar 1.2** Klik I Agree



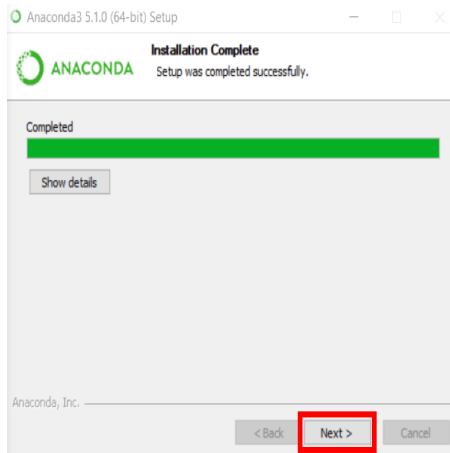
**Gambar 1.3** Pilih just me saja



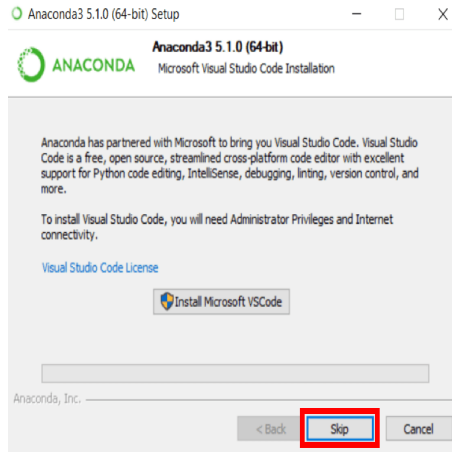
**Gambar 1.4** Directory tempat anaconda akan diinstall



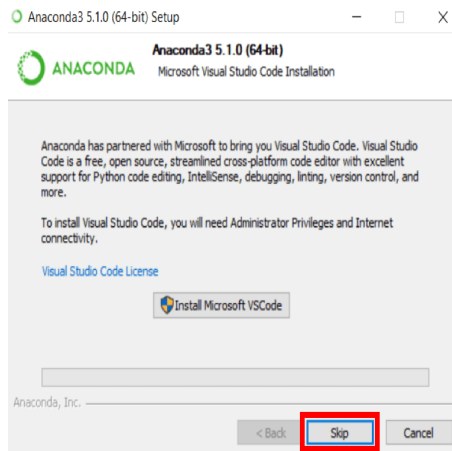
**Gambar 1.5** Opsi Register



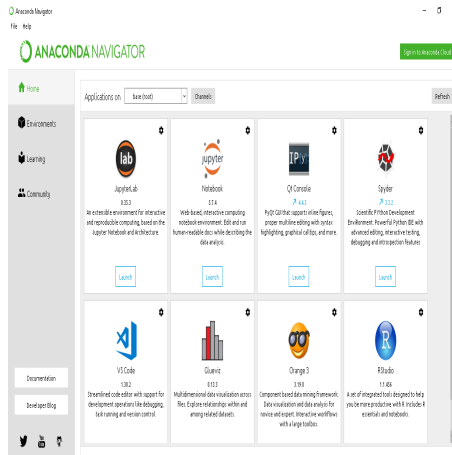
**Gambar 1.6** Tunggu hingga selesai



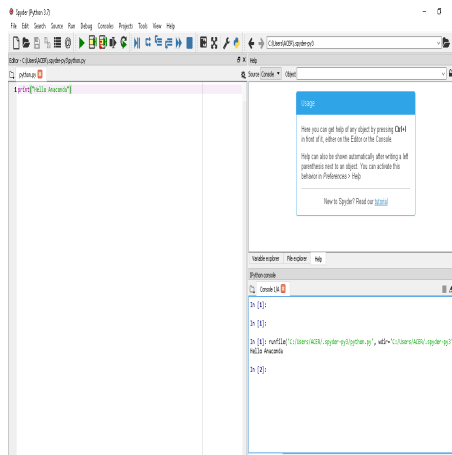
**Gambar 1.7** Opsi Tambahan



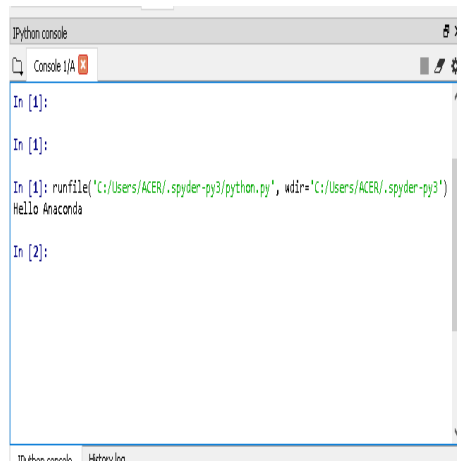
**Gambar 1.8** Finish



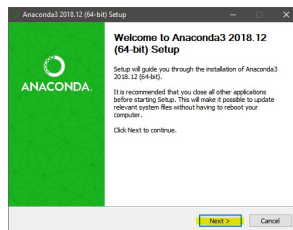
**Gambar 1.9** Tampilan awal anaconda



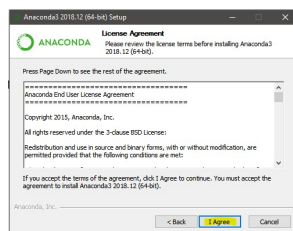
**Gambar 1.10** IDE Spyder



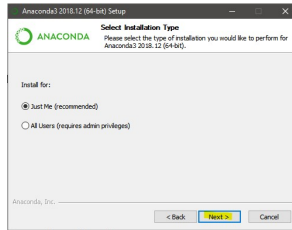
**Gambar 1.11** Menu Console



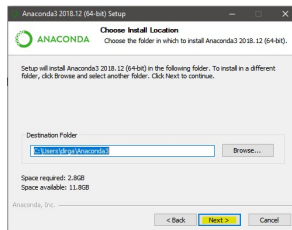
**Gambar 1.12** Tampilan Instalasi 1



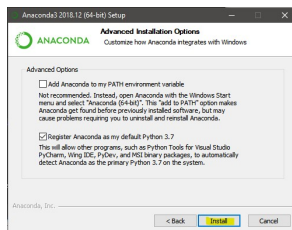
**Gambar 1.13** Tampilan Instalasi 2



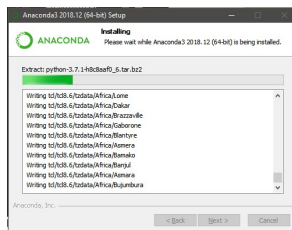
**Gambar 1.14** Tampilan Instalasi 3



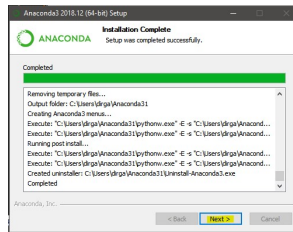
**Gambar 1.15** Tampilan Instalasi 4



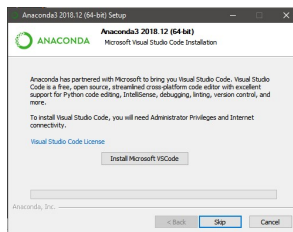
**Gambar 1.16** Tampilan Instalasi 5



**Gambar 1.17** Tampilan Instalasi 6



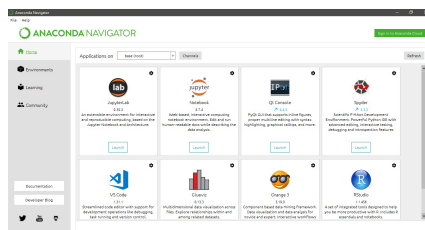
Gambar 1.18 Tampilan Instalasi 7



Gambar 1.19 Tampilan Instalasi 8

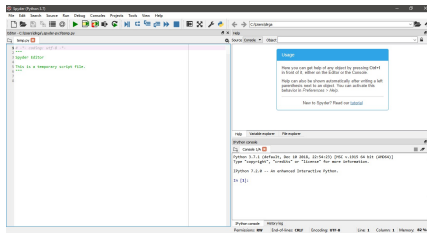


Gambar 1.20 Tampilan Instalasi 9

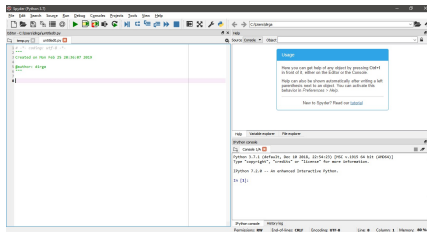


Gambar 1.21 Tampilan awal Anaconda





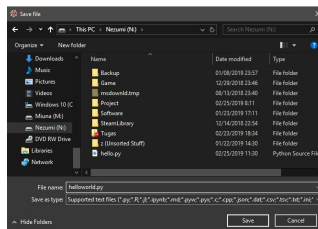
**Gambar 1.22** Tampilan spider



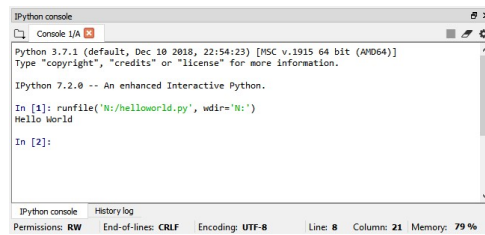
**Gambar 1.23** Tampilan new file pada spider

```
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 1.24** `print("Hello World")`



**Gambar 1.25** Dialog simpan file



The image shows a screenshot of an IPython console window. The window title is "IPython console". The console output shows the following:

```
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 is a status bar with the following information:

- IPython console
- History log
- Permissions: RW
- End-of-lines: CRLF
- Encoding: UTF-8
- Line: 8
- Column: 21
- Memory: 79 %

**Gambar 1.26** Hasil Program



## BAB 2

---

# PEMROGRAMAN DASAR

---

## 2.1 Chandra Kirana Poetra

### 2.1.1 Teori

1. sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut di kode Python Variable merupakan tempat yang dapat digunakan untuk menyimpan data, dalam python kita bisa membuat variable dengan cara berikut

```
1
2 #Soal no 1 Tipe data
3
4
5 #contoh integer
6 a = 1
7 b = 1.1
8 z = 2+1j
9
10 print(type(a))
11 print(type(b))
12 print(type(z))
13
14 # contoh string
```

```

15
16 a = "Chandra Kirana Poetra"
17 print(a[1])
18 #print ini digunakan untuk mencetak huruf yang dipilih indexnya
19
20 a = "Chandra Kirana Poetra"
21 print(a[2:5])
22 #print ini digunakan untuk mencetak huruf yang dipilih indexnya
    dengan mengisi
23 #nilai index tempat kita memilih index untuk memulai mengambil,
    dan juga nilai index
24 #akhirnya
25
26 a = "Chandra Kirana Poetra"
27 print(len(a))
28 #print ini digunakan untuk menjumlahkan length yang artinya
    panjang dari string a
29
30 a = "Chandra Kirana Poetra"
31 print(a.lower())
32 #print ini digunakan untuk mencetak huruf yang kecil semua
33
34 a = "Chandra Kirana Poetra"
35 print(a.upper())
36 #print ini digunakan untuk mencetak huruf yang besar semua
37
38 #boolean
39
40 a = 10
41 b = 8
42 if (a>b):
43     print("Variabel A lebih besar dibandingkan dengan Variable B")
44
45 elif (a<b):
46     print("Variabel A lebih kecil dibandingkan dengan Variable B")
47
48 else:
49     print("Variabel A Nilainya sama dengan Variable B")

```

2. Kode untuk meminta input dari user dan bagaimana melakukan output ke layar seperti pada gambar2

```

1 #soal no 2
2 namalengkap = input("Masukan Nama lengkap kamu : ")
3 print("Halo " + str(namalengkap))

```

3. Operator dasar aritmatika Terdapat penambahan, pengurangan perkalian, perkalian, pembagian, modulus

```

1 #soal no 3
2
3 a = 1
4 b = 2
5 print(a+b)# ini merupakan contoh penjumlahan
6 print(a-b)# ini merupakan contoh pengurangan

```

```

7 print(a*b)# ini merupakan contoh perkalian
8 print(a/b)# ini merupakan contoh pembagian
9 print(a%b)# ini merupakan contoh modulus

```

4. Perulangan ada dua jenis perulangan di dalam python mereka adalah perulangan while dan perulangan for

```

1 #soal no 4
2 #While looping
3
4 a = 1
5 while a < 10:
6     print(a)
7     a +=1
8
9 # for looping
10 benda = ["Piring", "Sendok", "Garpu"]
11 for x in benda:
12     print(x)

```

5. sintak Untuk memilih kondisi Kondisi IF digunakan ketika ingin menentukan tindakan apa yang harus digunakan sesuai dengan kondisi yang telah diatur

```

1 #soal no 5
2 #contoh if statement
3
4 a = 5
5 b = 10
6 if b > a:
7     print("b ternyata lebih besar dari a")
8
9 #contoh elif statement
10 a = 5
11 b = 10
12 if a > b:
13     print("a ternyata lebih besar dari b")
14 elif a < b:
15     print("a ternyata lebih kecil dari b")
16
17 #contoh else statement
18 a = 5
19 b = 5
20 if a > b:
21     print("a ternyata lebih besar dari b")
22 elif a < b:
23     print("a ternyata lebih kecil dari b")
24 else:
25     print("a dan b sama")
26

```

6. Jenis-jenis error pada python Syntax Errors adalah keadaan dimana kode python mengalami kesalahan penulisan. IndentationError adalah error yang terjadi saat indentasi error. SystemError adalah error yang terjadi ketika interpreter mendeteksi error internal TypeError adalah error yang terjadi saat dilakukan eksekusi

pada suatu operasi atau fungsi dengan type object yang tidak sesuai. ValueError adalah error ketika value yang dimasukan tidak sesuai UnicodeTranslateError adalah error yang muncul ketika mentranslate unicode UnicodeDecodeError adalah error yang muncul ketika proses decode unicode UnicodeEncodeError adalah error yang muncul ketika proses encode unicode UnicodeError adalah error yang muncul ketika error terkait unicode terdeteksi

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

```
1 #soal no 7
2 try:
3     print(g)
4 except:
5     print("Ada error")
```

### 2.1.2 Praktek

1. Jawaban soal no 1

```

1 #soal no 1
2 print("++ ++ ++++++++ ++ ++ ++++++ ++++++++
  ++++++++")
3 print("++ ++ ++++++++ ++ ++ + + ++++++++
  ++++++++")
4 print("++ ++ ++ ++ ++ + + ++ +
  ")
5 print("++ ++ ++ ++ ++ + + ++ +
  ")
6 print("++ ++ ++ ++++++ + + ++
  ++++++++")
7 print("++ ++ ++ ++ ++ + + ++
  ")
8 print("++ ++ ++ ++ ++ + + ++
  ")
9 print("++ ++ ++ ++ ++ + + ++
  ")
10 print("++ ++ ++ ++ ++ + + ++
  ++++++++")
11 print("++ ++ ++ ++ ++ ++++++ ++
  ++++++++")

```

- ## 2. Jawaban soal no 2

```
1 # soal no 2
2
3 contohnpm = input("Masukan NPM : ")
4 angka = 0
5 while(angka < 79):
6     print("Halo, " + str(contohnpm) + " Apa Kabar ?")
7     angka = angka + 1
```

- ### 3. Jawaban soal no 3

```
1 #soal no 3
2 contohtmlpm = input("Masukan NPM : ")
3 angka = 0
4 while (angka < 79):
5     print("Halo, " + str(contohtmlpm[4:7]) + " Apa Kabar ?")
6     angka = angka + 1
```

#### 4. Jawaban soal no 4

```
1 #soal no 4
2 contohtmlpm = input("Masukan NPM : ")
3 print("Halo, " + str(contohtmlpm[-3]) + " Apa Kabar ?")
```

#### 5. Jawaban soal no 5

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

#### 6. Jawaban soal no 6

```
1 #soal no 6
2 print(h)
```

#### 7. Jawaban soal no 7

```
1 #soal no 7
2 print(i)
```

#### 8. Jawaban soal no 8

```
1 #soal no 8
2 print(a)
3 print(b)
4 print(c)
5 print(d)
6 print(e)
7 print(f)
8 print(g)
```

#### 9. Jawaban soal no 9

```
1 #soal no 9
2 print(d)
```

#### 10. Jawaban soal no 10



```
1 #soal no 10
2
3 print(a,b,c,f,g)
```

#### 11. Jawaban soal no 11

```
1 #soal no 11
2 print(c,f)
```

### 2.1.3 Keterampilan dan penanganan error

```
1 a = "10"
2 b = 5
3
4 try:
5     print(a+b)
6 except TypeError:
7     print("Beda Tipe Data")
```

## 2.2 Chapter 2 | D. Irga B. Naufal Fakhri D4 TI 2C

### 2.2.1 Teori Praktikum

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 penambahan 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** 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.

## 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.

## 2.2.2 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="UI")[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("Halo, "+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("Halo, " + 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
```

## 2.2.3 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 mengkonversi variabelnya 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")

```

## 2.3 Bakti Qilan Mufid

### 2.3.1 Teori

Praktek teori penunjang yang dikerjakan :

**2.3.1.1 Variable, pemakaian Variable dan Jenis-Jenis Type data** Variabel merupakan tempat menyimpan data, sedangkan tipe data adalah jenis data yang tersimpan dalam variabel. Variabel bersifat mutable, artinya nilainya bisa berubah-ubah.

#### 1. Pemakaian Variabel

Variabel di python dapat dibuat dengan format seperti ini:

NamaVariabel = (nilai)

Contoh:

VariabelKu = "ini isi variabel"

variabel2 = 20

Kemudian untuk melihat isi variabel, kita dapat menggunakan fungsi print.

print VariabelKu

print variabel2

#### (a) Aturan Penulisan Variabel

- Nama variabel boleh diawali menggunakan huruf atau garis bawah (`_`), contoh: `nama`, `_nama`, `namaKu`, `nama_variabel`.
- Karakter selanjutnya dapat berupa huruf, garis bawah (`_`) atau angka, contoh: `__nama`, `n2`, `nilai1`.
- Karakter pada nama variabel bersifat sensitif (case-sensitif). Artinya huruf besar dan kecil dibedakan. Misalnya, `variabel_Ku` dan `variabel_ku`, keduanya adalah variabel yang berbeda.
- Nama variabel tidak boleh menggunakan kata kunci yang sudah ada dalam python seperti `if`, `while`, `for`, dsb.

#### (b) Tipe Data

Cara mengisi nilai variabel ditentukan dengan jenis datanya, misalkan untuk tipe data teks (string) maka harus diapit dengan tanda petik ("`...`"). Sedangkan untuk angka (integer) dan boolean tidak perlu diapit dengan tanda petik.

#### (c) Jenis-Jenis Tipe Data

- Boolean, Contoh *true* atau *false*
- String, Contoh "Belajar Python"
- Integer, Contoh 15 atau 1234
- Float, Contoh 2.5 atau 0.55
- List, Contoh ['abcd', 123, 1.5]

## 2. Meminta input dan melakukan output

```
x = input("masukan nama: ")
print('Hallo, ' + x) #dengan perintah ini, akan menampilkan output
```

## 3. Operator dan Konvert

- Tambah contoh  $x + y$
- Kurang contoh  $x - y$
- Bagi contoh  $x / y$
- Kali contoh  $x * y$
- Modulus contoh  $x$
- Pangkat  $x ** y$
- equal contoh  $x == y$
- not equal contoh  $x != y$
- lebih besar dari contoh  $x > y$
- kurang dari  $x < y$
- Konvert string ke integer, contoh  $x = \text{int}("123")$
- Konvert integer ke string, contoh  $x = \text{str}(456)$

## 4. Perulangan di Python

- Perulangan for  
 contoh :  
 ulang = 2  
 for i in range(ulang):  
     print ("Perulangan ke-" + str(i))  
 Hasil :  
 Perulangan ke-0  
 Perulangan ke-1
- Perulangan While  
 contoh :  
 jawab = 'ya'  
 hitung = 0  
 while(jawab == 'ya'):

```
hitung += 1
jawab = input("Ulang lagi tidak? ")
print ("Total perulangan: " + str(hitung))
```

## 5. Kodisi di Python

### (a) Kondisi **If**

Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar True. Jika kondisi bernilai salah False maka statement/kondisi if tidak akan di-eksekusi. Dibawah ini adalah contoh penggunaan kondisi if pada Python

```
a = 33
b = 200
if b > a:
    print("b lebih besar dari a")
```

### (b) Kondisi **If Else** Kondisi if else adalah kondisi dimana jika pernyataan benar True maka kode dalam if akan dieksekusi, tetapi jika bernilai salah False maka akan mengeksekusi kode di dalam else. Dibawah ini adalah contoh penggunaan kondisi if else pada Python

```
a = 200
b = 33
if b > a:
    print("b lebih besar dari a")
else:
    print("b bukan lebih besar dari a")
```

### (c) Kondisi **Elif** Pengambilan keputusan (kondisi if elif) merupakan lanjutan/percabangan logika dari "kondisi if". Dengan elif kita bisa membuat kode program yang akan menyeleksi beberapa kemungkinan yang bisa terjadi. Hampir sama dengan kondisi "else", bedanya kondisi "elif" bisa banyak dan tidak hanya satu. Dibawah ini adalah contoh penggunaan kondisi elif pada Python

```
a = 33
b = 33
if b > a:
    print("b lebih besar dari a")
elif a == b:
    print("a sama dengan b")
```

## 6. Error yang sering dialami

### (a) Syntax Error, Cara mengatasinya dengan cara melihat kode dan mengecek kesalahan dalam penulisan.

- (b) Run-time Error, Cara mengatasinya mengecek file pada directory nya, dan memastikan file nya tidak ada yang terhapus.
- (c) Logical Error, Cara mengatasinya mengecek kode secara manual karena error tidak akan ternotice, tetapi akan terasa karena keluaran berbeda dengan yang diharapkan.

## 7. Cara memakai Try Except

Python menyediakan metode penanganan eksepsi dengan menggunakan pernyataan try dan except. Di dalam blok try kita meletakkan baris program yang kemungkinan akan terjadi error. Bila terjadi error, maka penanganannya diserahkan kepada blok except.

contoh :

try:

```
print(x)
```

except:

```
print("terjadi error bre ")
```

### 2.3.1.2 Ketramplan Pemrograman

Buat program di python dengan ketentuan :

#### 1. Jawaban

```
1 print(" ** ** ***** ***)
2 print(" **** ***** ***)
3 print(" *** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** )
4 print(" *** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** )
5 print(" *** ** ** ***** ***)
6 print(" *** ** ** ***** ***)
7 print(" *** ** ** ***** ***)
8 print(" *** ** ** ***** ***)
```

#### 2. Jawaban

```
1 npm = input("Masukkan NPM : ")
2 ulang = 1
3 while (ulang <= 83):
4     print("Halo , "+str (npm)+" apa kabar?")
5     ulang += 1
```

#### 3. Jawaban

```
1 npm = input("Masukkan NPM : ")
2 ulang = 1
3 while (ulang <= 83):
4     print("Halo , "+str (npm[4:7])+" apa kabar?")
5     ulang += 1
```

#### 4. Jawaban

```

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

```

## 5. Jawaban

```

1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 8
7 g = 3
8
9 npm = [a,b,c,d,e,f,g]
10
11 for n in npm:
12     print(n, end = "")
13
14 print()

```

## 6. Jawaban

```

1 print(a+b+c+d+e+f+g)

```

## 7. Jawaban

```

1 print(a*b*c*d*e*f*g)

```

## 8. Jawaban

```

1 for n in npm:
2     print(n)

```

## 9. Jawaban

```

1 for n in npm:
2     if (n % 2 == 0):
3         if (n != 0):
4             print(n, end = "")

```

## 10. Jawaban

```

1 for n in npm:
2     if (n % 2 != 0):
3         print(n, end = "")

```

## 11. Jawaban

```

1 for n in npm:
2     if (n % 2 != 0):
3         print(n, end = "")
4
5 print()

```

### 2.3.1.3 *Ketrampilan Penanganan Error* Bagian Penanganan error dari script python.

#### 1. Jawaban

- (a) Syntax Errors, adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- (b) Zero Division Error, adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0). Solusinya adalah tidak membagi suatu yang hasilnya nol.
- (c) Name Error, 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.
- (d) Type Error, adalah exception yang terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

#### 2. Jawaban

```

1 a = "7"
2 b = 7
3
4 try :
5     print(a+b)
6 except TypeError:
7     print("Beda tipe data")

```

## 2.4 Fanny Shafira Damayanti

### 2.4.1 Teori

#### 2.4.2 Jenis-Jenis Variable

1. Bilangan (Number) Tipe data bilangan yang umum ada 2 yaitu, integer dan float. Integer adalah bilangan bulat, sedangkan float adalah bilangan pecahan. Selain itu ada tipe bilangan lain, yaitu bilangan kompleks yaitu bilangan yang memiliki bagian real dan imajiner. Integer, float, dan kompleks masing-masing di Python diwakili oleh kelas int, float, dan complex.
2. String String adalah satu atau serangkaian karakter yang diletakkan diantara tanda kutip, baik tanda kutip tunggal ( ' ) maupun ganda ( " ). Huruf, angka, maupun karakter lainnya yang digabung menjadi teks adalah contoh string.  
String adalah tipe data yang anggotanya berurutan dan memiliki indeks. Indeks dimulai dari angka 0 bila dimulai dari depan dan -1 bila diindeks dari belakang. Tiap karakter bisa diakses menggunakan indeksnya dengan format namastring[indeks] . Pada string juga bisa dilakukan slicing atau mengakses sekelompok substring dengan format namastring[awal:akhir]

3. List List adalah tipe data yang berisi item yang berurut. Seperti halnya tipe data string, tiap item (anggota) list memiliki indeks sesuai dengan urutannya. Indeks dimulai dari 0 dan bukan dari 1.

List bisa berisi anggota dengan tipe yang sama maupun berbeda. Untuk mendeklarasikan list, digunakan tanda kurung [ ] dan masing-masing anggotanya dipisahkan oleh tanda koma.

4. Tuple Tuple adalah jenis data lain yang mirip dengan list. Perbedaanannya dengan list adalah anggotanya tidak bisa diubah (immutable). List bersifat mutable, sedangkan tuple bersifat immutable. Sekali tuple dibuat, maka isinya tidak bisa dimodifikasi lagi.

Tuple dideklarasikan dengan menggunakan tanda kurung. dan anggotanya dipisahkan oleh tanda koma. Tuple berguna untuk data yang dimaksudkan tidak diubah isinya. Misalnya tuple komposisi warna untuk putih adalah (255,255,255).

5. Set Set adalah salah satu tipe data di Python yang tidak berurut (unordered). Set memiliki anggota yang unik (tidak ada duplikasi). Jadi misalnya kalau kita meletakkan dua anggota yang sama di dalam set, maka otomatis set akan menghilangkan yang salah satunya.

Set dibuat dengan meletakkan anggota – anggotanya di dalam tanda kurung kurawal , dipisahkan menggunakan tanda koma. Kita juga bisa membuat set dari list dengan memasukkan list ke dalam fungsi set()

6. Dictionary Dictionary adalah tipe data yang tiap anggotanya terdiri dari pasangan kunci-nilai (key-value). Mirip dengan kamus dimana ada kata ada arti. Dictionary umumnya dipakai untuk data yang besar dan untuk mengakses anggota data secara acak. Anggota dictionary tidak memiliki indeks.

Dictionary dideklarasikan dengan menggunakan tanda kurung kurawal , dimana anggotanya memiliki bentuk kunci:nilai atau key:value dan tiap anggota dipisah tanda koma. Kunci dan nilainya bisa memiliki tipe sembarang.

### 2.4.3 Cara Menampilkan Kode untuk meminta Input dan output nya

Untuk menampilkan Python sudah menyediakan fungsi input() dan rawinput() untuk mengambil inputan dari keyboard. Cara pakainya: Namavariabel = input (“Sebuah teks”) Artinya, teks yang kita inputkan dari keyboard akan disimpan ke dalam namavariabel. Untuk menampilkan output teks, kita menggunakan fungsi print().

### 2.4.4 Operator Dasar Aritmatika

1. Penjumlahan +
2. Pengurangan -
3. Perkalian \*



4. Pembagian /
5. Sisa Bagi percent
6. Pemangkatan \*\*

Integer = int(a) untuk konversi string ke integer

String = str(a) untuk konversi integer ke string

## 2.4.5 Sintax Perulangan

### 1. The while Loop

Dengan while loop, kita dapat menjalankan serangkaian pernyataan selama suatu kondisi benar.

Contoh : Print i sepanjang i adalah kurang dari 6:

```
i = 1
while i < 6:
    print(i)
    i += 1
```

### 2. Python For Loops

For loop digunakan untuk mengulangi urutan (baik daftar, tuple, dictionary, set, atau string).

Ini kurang seperti kata kunci untuk dalam bahasa pemrograman lain, dan berfungsi lebih seperti metode iterator seperti yang ditemukan dalam bahasa pemrograman berorientasi objek lainnya. Dengan for loop kita dapat mengeksekusi seperangkat pernyataan, satu kali untuk setiap item dalam list, tuple, set dll.

Contoh :

Print each fruit in a fruit list:

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
```

## 2.4.6 Syntax Kondisi

### Kondisi If Else

Pengambilan keputusan (kondisi if else) tidak hanya digunakan untuk menentukan tindakan apa yang akan diambil sesuai dengan kondisi, tetapi juga digunakan untuk menentukan tindakan apa yang akan diambil/dijalankan jika kondisi tidak sesuai. Pada python ada beberapa statement/kondisi diantaranya adalah if, else dan elif Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar.

Kondisi `if else` adalah kondisi dimana jika pernyataan benar `true` maka kode dalam `if` akan dieksekusi, tetapi jika bernilai salah `false` maka akan mengeksekusi kode di dalam `else`.

Dibawah ini adalah contoh penggunaan kondisi `if else` pada Python

```
nilai = 3
if (nilai < 7):
    print ("Selamat anda lulus")
else :
    print ("Maaf anda tidak lulus")
```

## 2.4.7 Jenis error di Python

### 1. Syntax Errors

Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Python interpreter dapat mendeteksi kesalahan ini saat kode dieksekusi.

```
print "Hello World"
```

SyntaxError: invalid syntax

Output dari program yang dieksekusi akan menampilkan pesan “invalid syntax”. Penanganan Syntax Errors dilakukan dengan memperbaiki penulisan kode yang salah tersebut.

untuk menanganinya cukup tambahkan tanda kurung () pada :

```
print ("Hello World")
```

### 2. Exceptions

Exceptions adalah suatu keadaan saat penulisan syntax sudah benar, namun kesalahan terjadi karena syntax tidak bisa dieksekusi. Banyak hal yang menyebabkan terjadinya Exceptions, mulai dari kesalahan matematika, kesalahan nama function, kesalahan library, dan lain-lain.

## 2.4.8 Cara menggunakan Try Except

Di kode ini kita akan mencoba menangkap dua error pada kode yang dikurung oleh `try..except`. Terdapat sebuah dictionary yang berisi key nama, kota, dan umur. Kemudian kita membuka sebuah file yang bernama `contact.txt`. Walaupun ada kode error setelahnya yang akan mengakibatkan error pengaksesan indeks, yang akan ditangkap terlebih dahulu adalah error yang diakibatkan galalnya membaca file.

contoh :

```
orang = "nama": "fanny", "kota": "bandung", "umur": "19"
```

```
try:
```

```
contact = open("contact.txt", 'r')
```

```
print orang["pekerjaan"]
```

```
except IOError, e:
```

```
print "Terjadi error IO: ", e
except KeyError, e:
print "Terjadi kesalahan pada akses list/dict/
tuple:", e
print orang
```

#### 2.4.9 Keterampilan Pemrograman

### 1. Jawaban Soal 1

```

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

```

## 2. Jawaban Soal 2

```
1 npm = input("Masukan NPM anda :")
2 hitung = 0
3 while(hitung < 69):
4     print("Halo, " + str(npm) + " Apa kabar?")
5     hitung = hitung + 1
```

### 3. Jawaban Soal 3

```
1 npm = input("Masukan NPM anda :")
2 hitung = 0
3 while(hitung < 9):
4     print("Halo, " + str(npm[4:7]) + " Apa kabar?")
5     hitung = hitung + 1
```

#### 4. Jawaban Soal 4

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

### 5. Jawaban Soal 5

```
1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 6
7 g = 9
8 h = a+b+c+d+e+f+g
9 i = a*b*c*d*e*f*g
10
11 print(a,b,c,d,e,f,g)
```

#### 6. Jawaban Soal 6

```
1 print(h)
```

#### 7. Jawaban Soal 7

```
1 print(i)
```

#### 8. Jawaban Soal 8

```
1 print(a)
2 print(b)
3 print(c)
4 print(d)
5 print(e)
6 print(f)
7 print(g)
```

#### 9. Jawaban Soal 9

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

#### 10. Jawaban Soal 10

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

#### 11. Jawaban Soal 11

```
1 print(c)
```

### 2.4.10 Keterampilan Penanganan Error

#### 1. Jawaban Soal

```
1 a = 6
2 b = "9"
3
4 try :
5     c = a + b
6     print(c)
7 except TypeError :
8     print("Ini Error sayangku! Coba perbaiki lagi !")
```

```

1 #- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8 print("berapa umur anda?")
9 a=input()
10 print("jadi umur anda " + a)

```

2. input()

```

In [6]: runfile('C:/Users/User/.spyder-py3/temp.py', wdir='C:/Users/User/.spyder-py3')
berapa umur anda?
12
jadi umur anda 12

```

**Gambar 2.1** input dan outputnya

## 2.5 Tugas 2 Alfadian Owen

### 2.5.1 Teori

1. Integer, Boolean, Char, Float, String

## 2.6 Muhammad Abdul Gani Wijaya

### 2.6.1 Variable Python

Ada beberapa jenis type data pada Python, contohnya integer dan string. Penulisan variable seperti berikut : `namavariabel = type data`, contoh :

```

8 a=2
9 b=6
10 c=a+b
11 d=a-b
12 e=a/b
13 f=a*b
14 g="3"
15 h="5"
16 i=int(g)+int(h)
17 j=str(a)+str(b)
18 print(c,d,e,f,i,j)

```

3. asd

**Gambar 2.2** operasi aritmatika

```

x = 5
y = "John"
print(x)
print(y)

```

### 2.6.2 Input Python

Untuk menampilkan input menggunakan perintah `input()`. Perintah `input()` dapat dimasukkan ke dalam variable, sehingga value dari input akan menjadi value dari variable tersebut. Contohnya :

```

print("Enter your name:")
x = input()
print("Hello," + x)

```

### 2.6.3 Operator Aritmatika Python

Operator aritmatika dapat digunakan pada Python, seperti + (tambah), - (kurang), \* (kali), / (bagi). Contohnya :

```

x9 = 5
y9 = 3
print(x9 + y9)

```

```

x10 = 5
y10 = 3

```

4. loop digunakan untuk mengulang sampai memenuhi sebuah condition, contoh :

```
x='botak'
for x in range(6):
    print("botak")
```

In [17]: runfile('C:/Users/User/.spyder-py3/temp.py', wdir='C:/User  
botak  
botak  
botak  
botak  
botak  
botak

**Gambar 2.3** contoh loop/pengulangan

5. memilih kondisi menggunakan syntax if else, cara menggunakannya yaitu jika sebuah kondisi tidak terpenuhi maka akan meng-execute kondisi lain. contoh:

```

8 print("masukan x:")
9 x=input()
10 print("masukan y:")
11 y=input()
12 if x>y:
13     print(x + " Lebih besar dari " + y)
14 elif x==y:
15     print(x + " sama dengan " + y)
16 else:
17     print(x + " Lebih kecil dari " + y)

```

In [19]: runfile('C:/Users/User/.spyder-py3/temp.py', wdir='C:/Us  
 masukan x:  
 2  
 masukan y:  
 5  
 2 lebih kecil dari 5

**Gambar 2.4** contoh syntax kondisi



7. try berfungsi untuk menguji error, except berfungsi untuk menangani error. cara

```

8 print("masukan angka:")
9 x=input()
10 print("masukan angka ke 2 :")
11 y=input()
12 try:
13     if x>y:
14         print(int(x) + " lebih kecil dari " + int(y))
15     elif x==y:
16         print(int(x) + " sama dengan " + int(y))
17     else:
18         print(int(x) + " lebih besar dari " + int(y))
19 except ValueError:
20     print("masukan angka bukan huruf")

```

memakainya :

```
In [27]: runfile('C:/Users/User/.spy
```

```
masukan angka:
```

a

```
masukan angka ke 2 :
```

b

```
masukan angka bukan huruf
```

**Gambar 2.5** contoh try except

## 2.5.2 Keterampilan Pemrograman

1. a

```
7 print("input : ")
8 a=input()
9 for x in range(87):
10     print("hello " + a + " apa kabar?")
```

2.

```
In [31]: runfile('C:/Users/User/.spy
input :

1174091
hello 1174091 apa kabar?
hello 1174091 apa kabar?
hello 1174091 apa kabar?
hello 1174091 apa kabar?
hello 1174091 apa kabar?
hello 1174091 apa kabar?
```

```

7 print("input : ")
8 a=input()
9 for x in range(87):
10  print("hello " + a[4:7] + " apa kabar?")

```

3.

```

hello 091 apa kabar?
hello 091 apa kabar?
hello 091 apa kabar?
hello 091 apa kabar?
hello 091 apa kabar?
hello 091 apa kabar?
hello 091 apa kabar?
hello 091 apa kabar?
hello 091 apa kabar?
hello 091 apa kabar?
hello 091 apa kabar?
hello 091 apa kabar?

```

In [35]: ]

---

```

7 a=1
8 b=1
9 c=7
10 d=4
11 e=0
12 f=9
13 g=1
14 print(str(a)+str(b)+str(c)+str(d)+str(e)+str(f)+str(g))

```

4.

```
7 a=1
8 b=1
9 c=7
10 d=4
11 e=0
12 f=9
13 g=1
14 print(a+b+c+d+e+f+g)
```

5.

```
a=1
b=1
c=7
d=4
e=0
f=9
g=1
print(a*b*c*d*e*f*g)
```

6.

```
a=1
b=1
c=7
d=4
e=0
f=9
g=1
h = [a, b, c, d, e, f, g]
for x in range(len(h)):
    print (h[x])
```

7.

```

7 a=1
8 b=1
9 c=7
10 d=4
11 e=0
12 f=9
13 g=1
14 h = [a, b, c, d ,e ,f ,g]
15 i=(str(a)+str(b)+str(c)+str(d)+str(e)+str(f)+str(g))
16 ganjil = [num for num in h if num % 2 == 1]
17 genap = [num for num in h if num % 2 == 0]
18 print(genap)

```

8.

```

7 a=1
8 b=1
9 c=7
10 d=4
11 e=0
12 f=9
13 g=1
14 h = [a, b, c, d ,e ,f ,g]
15 i=(str(a)+str(b)+str(c)+str(d)+str(e)+str(f)+str(g))
16 ganjil = [num for num in h if num % 2 == 1]
17 genap = [num for num in h if num % 2 == 0]
18 print(ganjil)

```

9.

```

7 a=1
8 b=1
9 c=7
10 d=4
11 e=0
12 f=9
13 g=1
14 h = [a, b, c, d ,e ,f ,g]
15 i=(str(a)+str(b)+str(c)+str(d)+str(e)+str(f)+str(g))
16 ganjil = [num for num in h if num % 2 == 1]
17 genap = [num for num in h if num % 2 == 0]
18 prima = [num for num in h if num % 2 != 0]
19 print(prima)

```

10.

```
print(x10 - y10)
```

```
    x11 = 5
y11 = 3
print(x11 * y11)
```

```
    x12 = 12
y12 = 3
print(x12 / y12)
```

Untuk mengubah type data menggunakan perintah `namavariabel = typedata(namavariabel yang akan diubah type datanya)`. Contoh :

```
    x13 = 12
y13 = "3"
x14 = str(x13)
y14 = int(y13)
print(x14)
print(y14)
```

## 2.6.4 Pengulangan Phyton

Untuk perulangan pada phyton dapat meggunakan perintah `while` dan `for`. Contoh

```
: i = 0
while i < 6:
i += 1
if i == 3:
continue
print(i)
```

```
    fruits = ["apple", "banana", "cherry"]
for x in fruits:
print(x)
```

## 2.6.5 Kondisi Phyton

Kondisi juga dapat diterapkan pada phyton, seperti jika `a` kurang dari `b` maka `b` lebih besar dari `a`. Contohnya :

```
a = 33
b = 200
if b > a:
print("b is greater than a")
```

### 2.6.6 Try Except Phyton

Pada phyton seringkali kita menemukan error. Untuk menangkap error dapat menggunakan try except. Contoh :

```
try:
print(x15)
except NameError:
print("Variable x15 is not defined")
except:
print("Something else went wrong")
```

Pada kode tersebut tidak terdapat variable x15 sehingga saat mengeksekusi perintah print(x15) terjadi error. Jika terjadi error karena tidak terdapat variable x15 maka outputnya akan menampilkan "Variable x15 is not defined".

### 2.6.7 Keterampilan Pemograman

#### 2.6.8 Program

```
1 print (1174071%3)
2 print("*** ***)")
3 print("*** ***)")
4 print("*** ***)")
5 print("*** ***)")
6 print("*** ***)")
7 print("*** ***)")
8 print("*** ***)")
9 print("*** ***)")
10 print("*** ***)")
11
12 #No.2
13 npm = input("Masukan NPM :")
14 hitung = 0
15 while (hitung < 71):
16     print("Halo, " + str(npm) + " Apa Kabar?")
17     hitung = hitung + 1
18
19 #No.3
20 npm = input("Masukan NPM :")
21 hitung = 0
22 while (hitung < 8):
23     print("Halo, " + str(npm[4:7]) + " Apa Kabar?")
24     hitung = hitung + 1
25
26 #No.4
27 npm = input("Masukan NPM :")
28 print("Halo, " + str(npm[-3]) + " Apa Kabar?")
29
30 #No.5
31 a = 1
32 b = 1
```

```

33 c = 7
34 d = 4
35 e = 0
36 f = 2
37 g = 6
38 h = a+b+c+d+e+f+g
39 i = a*b*c*d*e*f*g
40
41 print(a,b,c,d,e,f,g)
42
43 #No.6
44 print(h)
45
46 #No.7
47 print(i)
48
49 #No.8
50 print(a)
51 print(b)
52 print(c)
53 print(d)
54 print(e)
55 print(f)

```

## 2.6.9 Keterampilan Penanganan Error

### 2.6.10 Program

```

1 a = 1
2 b = "2"
3
4 try:
5     c = a + b
6     print(c)
7 except TypeError:
8     print("Perbedaan Tipe Data")

```

## 2.7 Tia Nur Candida

### 2.7.1 Jenis Variabel pada Python

1. Bilangan (Number) Tipe data bilangan umumnya ada 2 yaitu, integer dan float. Integer merupakan bilangan bulat, sedangkan float adalah bilangan pecahan. Terdapat bilangan lain yaitu bilangan kompleks yaitu bilangan yang memiliki bagian real dan imajiner. Integer, float, dan kompleks masing-masing di Python diwakili oleh kelas int, float, dan complex.
2. String String merupakan satu atau serangkaian karakter yang diletakkan diantara tanda kutip, baik tanda kutip tunggal ( ' ) maupun ganda ( " ). Huruf, angka, maupun karakter lainnya yang digabung menjadi teks adalah contoh string. String adalah tipe data yang anggotanya berurut dan memiliki indeks. Indeks dimulai dari angka 0 bila dimulai dari depan dan -1 bila diindeks dari



belakang. Tiap karakter bisa diakses menggunakan indeksnya dengan format nama string[indeks] . Pada string juga bisa dilakukan slicing atau mengakses sekelompok substring dengan format namastring[awal:akhir].

3. List List merupakan tipe data yang berisi item yang berurut. Seperti halnya tipe data string, tiap item (anggota) list memiliki indeks sesuai dengan urutannya. Indeks dimulai dari 0 dan bukan dari 1. List bisa berisi anggota dengan tipe yang sama maupun berbeda. Untuk mendeklarasikan list, digunakan tanda kurung [ ] dan masing-masing anggotanya dipisahkan oleh tanda koma.
4. Tuple Tuple merupakan jenis data lain yang mirip dengan list. Perbedaannya dengan list adalah anggotanya tidak dapat diubah (immutable). List bersifat mutable, sedangkan tuple bersifat immutable. Sekali tuple dibuat, maka isinya tidak bisa dimodifikasi lagi. Tuple dideklarasikan dengan menggunakan tanda kurung ( ). dan anggotanya dipisahkan oleh tanda koma. Tuple berguna untuk data yang dimaksudkan tidak diubah isinya. Misalnya tuple komposisi warna untuk putih adalah (255,255,255).
5. Set Set merupakan salah satu tipe data pada Python yang tidak berurut (unordered). Set memiliki anggota yang unik (tidak ada duplikasi). Jadi misalnya kalau kita meletakkan dua anggota yang sama di dalam set, maka otomatis set akan menghilangkan yang salah satunya. Set bisa digunakan untuk melakukan operasi himpunan matematika seperti gabungan, irisan, selisih, dan komplemen. Set dibuat dengan meletakkan anggota – anggotanya di dalam tanda kurung kurawal , dipisahkan menggunakan tanda koma. Kita juga bisa membuat set dari list dengan memasukkan list ke dalam fungsi set() Set bisa berisi data campuran, baik integer, float, string, dan lain sebagainya. Akan tetapi set tidak bisa berisi list, set, dan dictionary.
6. Dictionary Dictionary merupakan tipe data yang tiap anggotanya terdiri dari pasangan kunci-nilai (key-value). Mirip dengan kamus dimana ada kata ada arti. Dictionary umumnya dipakai untuk data yang besar dan untuk mengakses anggota data secara acak. Anggota dictionary tidak memiliki indeks. Dictionary dideklarasikan dengan menggunakan tanda kurung kurawal , dimana anggotanya memiliki bentuk kunci:nilai atau key:value dan tiap anggota dipisah tanda koma. Kunci dan nilainya bisa memiliki tipe sembarang.

## 2.7.2 Cara Menampilkan Output ke Layar

Untuk menampilkan Python sudah menyediakan fungsi input() dan rawinput() untuk mengambil inputan dari keyboard. Cara pakainya: Nama variable = input (“Sebuah teks”) Artinya, teks yang kita inputkan dari keyboard akan disimpan ke dalam nama variabel. Untuk menampilkan output teks, kita menggunakan fungsi print().

## 2.7.3 Operator Dasar Aritmatika

1. Penjumlahan +

2. Pengurangan -
3. Perkalian \*
4. Pembagian /
5. Sisa Bagi (persen)
6. Pemangkatan \*\*

## 2.7.4 Jenis dan Sintax Perulangan

### 1. While Loop

While loop dapat mengeksekusi statement selama kondisi nya true Contoh: i = 1

```
while i < 6:
    print(i)
    i += 1
```

### 2. For Loop

For loop digunakan untuk mengulangi urutan (list, tuple, dictionary, set, atau string). Menggunakan for loop dapat mengeksekusi seperangkat statement, satu kali untuk setiap item dalam list, tuple, set dll.

Contoh : fruits = ["apple", "banana", "cherry"]

```
for x in fruits:
    print(x)
```

## 2.7.5 Cara Memakai Sintax untuk Memilih Kondisi

**2.7.5.1 Kondisi If Else** Pengambilan keputusan (kondisi if else) tidak hanya digunakan untuk menentukan tindakan apa yang akan diambil sesuai dengan kondisi, tetapi juga digunakan untuk menentukan tindakan apa yang akan diambil/dijalankan jika kondisi tidak sesuai. Pada python ada beberapa statement/kondisi diantaranya adalah if, else dan elif Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar. Kondisi if else adalah kondisi dimana jika pernyataan benar true maka kode dalam if akan dieksekusi, tetapi jika bernilai salah false maka akan mengeksekusi kode di dalam else. Dibawah ini adalah contoh penggunaan kondisi if else pada Python

Kondisi if else adalah jika kondisi bernilai TRUE maka akan dieksekusi pada if, tetapi jika bernilai FALSE maka akan dieksekusi kode pada else

```
nilai = 3
```

Jika pernyataan pada if bernilai TRUE maka if akan dieksekusi, tetapi jika FALSE kode pada else yang akan dieksekusi.

```
if(nilai < 7):
```

```
print("Selamat Anda Lulus")
else:
print("Maaf Anda Tidak Lulus")
```

### 2.7.6 Jenis Error yang sering ditemui

1. Syntax Errors Syntax Errors merupakan suatu keadaan saat kode python mengalami kesalahan penulisan. Python interpreter dapat mendeteksi kesalahan ini saat kode dieksekusi.

```
1!!! print"Hello World"
```

2 SyntaxError: invalid syntax

Output dari program yang dieksekusi akan menampilkan pesan “invalid syntax“. Penanganan Syntax Errors dilakukan dengan memperbaiki penulisan kode yang salah tersebut.

2. Exceptions Exceptions merupakan suatu keadaan saat penulisan syntax sudah benar, namun kesalahan terjadi karena syntax tidak bisa dieksekusi. Banyak hal yang menyebabkan terjadinya Exceptions, mulai dari kesalahan matematika, kesalahan nama function, kesalahan library, dan lain-lain.

Berikut beberapa contoh Exceptions:

- (a) ZeroDivisionError ZeroDivisonError adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0).
- (b) NameError NameError adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Misalnya saat menjumlahkan variable yang tidak didefinisikan, memanggil function yang tidak ada, dan lain-lain.
- (c) TypeError TypeError adalah exception yang terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai.

### 2.7.7 TryExcept

Pada kode ini akan mencoba menangkap dua error pada kode yang dikurung oleh try..except. Terdapat sebuah dictionary yang berisi key nama, kota, dan umur. Kemudian kita membuka sebuah file yang bernama contact.txt. Walaupun ada kode error setelahnya yang akan mengakibatkan error pengaksesan indeks, yang akan ditangkap terlebih dahulu adalah error yang diakibatkanagalnya membaca file.

```
orang = "nama":"tia", "kota":"sumedang", "umur":20"
```

```
try:
```

```
contact = open("contact.txt", 'r')
```

```
print orang["pekerjaan"]
except IOError, e:
    print "Terjadi error IO: ", e
except KeyError, e:
    print "Terjadi kesalahan pada akses list/dict/tuple:", e

    print orang
```

## 2.8 Nurul Izza Hamka

### 2.8.1 Jenis-Jenis Variabel dan Pemakaian Variable di Kode Python :

#### 2.8.2 Tipe String

Dalam Python, string direpresentasikan oleh tipe 'str'. Tipe data string dituliskan di antara dua tanda petik tunggal atau ganda atau kutip ganda tiga kali.

Contoh Pemakaian String pada python :

```
x = "Assalamualaikum"
print(x)
```

Contoh lain Pada penjumlahan yaitu :

```
a = "Selamat "
b = "Datang"
c = a+b
print (c)
```

#### 2.8.3 Tipe List

Terdapat dua macam objek list dalam Python, yaitu tuple dan list. Perbedaanannya adalah, pada tuple daftar yang dituliskan bersifat tetap dan tidak dapat diubah, sedangkan pada list kita dapat menambah dan menghapus daftar; melakukan penyusunan; dan beberapa jenis modifikasi lainnya. Pemrosesan pada tuple umumnya lebih cepat dibandingkan pemrosesan list, namun karena kemudahan pada proses modifikasi, aturan praktis yang umumnya dianut oleh para programmer Python adalah selalu gunakan list.

Contoh :

```
Benda = ["Buku", "Pulpen", "Meja", "Sisir"]
print benda [2]
```

### 2.8.4 Tipe Tuple

Tuple pada dasarnya mirip dengan list, namun bersifat tetap, tuple mirip dengan array konstan. Ketika kita sudah membuatnya, kita tidak dapat melakukan perubahan pada elemen – elemen di dalamnya. Tuple ditulis dengan menggunakan notasi “()”.

Contoh :

```
tup1 = ('Aljabar', 'Logistik', 1993, 2017)
```

```
tup2 = (1, 2, 3, 4, 5)
```

```
tup3 = "a", "b", "c", "d"
```

**2.8.4.1 Tipe Dictionary** Dictionary adalah salah satu tipe kumpulan objek yang mirip dengan list dan tuple. Namun, perbedaannya adalah dictionary memuat objek yang tidak tersusun. Sebagai gantinya, dictionary menghubungkan relasi satu ke satu (pemetaan) antara kunci (key) ke suatu nilai. Dalam bahasa pemrograman lainnya dikenal sebagai array asosiatif yang indeksnya berupa nama tertentu.

Contoh :

```
d = "Nama" : "Nurul Izza Hamka", "Prodi" : "D IV Teknik Informatika", ... "NPM" : "1174062"
```

### 2.8.5 Kode Untuk Meminta Input Dari User dan Bagaimana Melakukan Output Ke Layar Pada Python ?

Meminta input atau masukan dari user, prompt sebagai string yang kita ingin tampilkan di layar. Sintaksnya adalah seperti berikut:

```
input([prompt])
```

Gunakan fungsi print() untuk menampilkan data ke perangkat keluaran standar (layar).

```
panjang = input('Masukkan nilai panjang: ')
Masukkan nilai panjang: 10
lebar = input('Masukkan nilai lebar: ')
Masukkan nilai lebar: 5
luas = int(panjang) * int(lebar)
print("Luas =", luas)
Luas = 50
```

### 2.8.6 Operator Dasar Aritmatika, Tambah, Kali, Kurang Bagi dan Bagaimana Mengubah String Ke Integer dan Integer Ke String ?

Macam-Macam Operator Aritmatika :

**2.8.6.1 Operator Penjumlahan** Tambah adalah variabel yang digunakan untuk menyimpan hasil penjumlahan. Contoh : variabel a ditambah variabel b (  $a + b$  ).

**2.8.6.2 Operator Perkalian** Kali adalah variabel yang digunakan untuk menyimpan hasil perkalian variabel a dikali variabel b (  $a * b$  ).

**2.8.6.3 Operator Pengurangan** Kurang adalah variabel yang digunakan untuk menyimpan hasil pengurangan. Contoh : variabel a dikurangkan variabel b (  $a - b$  ).

**2.8.6.4 Operator Pembagian** Bagi adalah variabel yang digunakan untuk menyimpan hasil pembagian. Contoh : variabel a dibagi variabel b (  $a / b$  ).

Cara mengubah Integer ke String :

integer ini merupakan tipe data bilangan bulat. Namun kita bisa mengkonversinya ke tipe data string:

```
a=100
```

konversi integer ke string :

```
string = str(a)
```

Cara mengubah String ke Integer :

Tipe data string ini merupakan tipe data yang menampung sebuah teks. Namun, kita juga bisa mengkonversinya ke tipe integer :

```
a = '1212'
```

konversi string ke integer

```
integer = int(a)
```

## 2.8.7 Sintak Perulangan , Jenis-Jenisnya Contoh dan Cara Pakainya Di Python.

Terdapat dua jenis perulangan dalam bahasa pemrograman python, yaitu perulangan dengan for dan while. Perulangan for disebut counted loop (perulangan yang terhitung), sementara perulangan while disebut uncounted loop (perulangan yang tak terhitung). Perbedaannya adalah perulangan for biasanya digunakan untuk mengulangi kode yang sudah diketahui banyak perulangannya. Sementara while untuk perulangan yang memiliki syarat dan tidak tentu berapa banyak perulangannya.

**2.8.7.1 Perulangan For** ulang = 5

```
for i in range(ulang):
```

```
print "Perulangan ke-" + str(i)
```

**2.8.7.2 Perulangan While** jawab = 'ya'

```
hitung = 0
```

```

while(jawab == 'ya'):
    hitung += 1
jawab = rawinput("Ulang lagi tidak? ")
print "Total perulangan: " + str(hitung)

```

### 2.8.8 Cara Pakai Sintak Untuk Memilih Kondisi dan Bagaimana Contoh Sintak Kondisi Di Dalam Kondisi

struktur if dalam Python dijalankan untuk memeriksa apakah kondisi ini adalah bernilai benar atau salah. Jika kondisi ini bernilai true, maka python akan menjalankan statemen didalam blok kondisi tersebut dan sebaliknya jika kondisi bernilai false maka statemen didalam blok tersebut tidak akan dijalankan.

#### 2.8.8.1 If $x = 1$

```

if x lebih dari 0:
    print("Nilai persennx adalah besar dari 0" persen x )

```

#### 2.8.8.2 If-else If-else

```

umur = 37
if umur lebih dari 18 and umur kurang dari 30:
    print "Sudah beranjak dewasa"
elif umur lebih dari 30 and umur kurang dari 45:
    print "Masa - masa emas"
elif umur lebih dari 45 and umur kurang dari 55:
    print "Memasuki masa paruh baya"
elif umur lebih dari 55:
    print "Masa - masa manula"
else:
    print "Masih dibawah umur"

```

### 2.8.9 Jenis Error Yang Sering Di Temui Di Python Dalam Mengerjakan Sintak Diatas , dan Bagaimana Cara Mengatasinya ?

Terdapat 2 jenis error pada Bahasa pemrograman Python yaitu Errors dan Exceptions.

**2.8.9.1 Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Python interpreter dapat mendeteksi kesalahan ini saat kode dieksekusi.

Output dari program yang dieksekusi akan menampilkan pesan "invalid syntax". Penanganan Syntax Errors dilakukan dengan memperbaiki penulisan kode yang salah tersebut.

**2.8.9.2 Exceptions** Exceptions adalah suatu keadaan saat penulisan syntax sudah benar, namun kesalahan terjadi karena syntax tidak bisa dieksekusi. Banyak hal

yang menyebabkan terjadinya Exceptions, mulai dari kesalahan matematika, kesalahan nama function, kesalahan library, dan lain-lain.

Berikut beberapa contoh Exceptions:

1. `NameError`, adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Misalnya saat menjumlahkan variable yang tidak didefinisikan, memanggil function yang tidak ada, dan lain-lain.
2. `TypeError`, adalah exception yang terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai.
3. `ZeroDivisonError`, adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0).

Cara penanganan Exceptions yaitu, memasukkan kode yang kamu anggap bisa menimbulkan exception di dalam klausa `try`. Langkah selanjutnya adalah menggunakan keyword `except` untuk menangani exception yang terjadi pada kode berikutnya.

Selain itu juga untuk menangani banyak exception menggunakan satu klausa `except` dengan melewati exception tersebut ke klausa sebagai sebuah tuple.

## 2.8.10 Cara Memakai Try Except

Blok `try` memungkinkan Anda menguji blok kode untuk kesalahan.

Blok `except` memungkinkan Anda menangani kesalahan.

Jika kesalahan terjadi, eksekusi kode blok `try` dihentikan dan ditransfer sampai ke blok `except`.

Selain menggunakan blok `except` setelah blok `try`, Anda juga dapat menggunakan blok `Finally`.

Kode di blok `finally` akan dieksekusi terlepas dari apakah `except` terjadi.

## 2.8.11 Keterampilan Pemrograman

### 1. No 1

```

1
2 #contoh fungsi
3 def Izzah() :
4     print ("Apa Kabar Izzah")
5 Izzah()
6
7 #class Employee:
8 #     'Common base class for all employees'
9 #     empCount = 0
10
11 #     def __init__(self, name, salary):
12 #         self.name = name

```



## 2. No 2

```

1 #         self.salary = salary
2 #         Employee.empCount += 1
3
4 #         def displayCount(self):
5 #             print ("Total Employee %d" % Employee.empCount)
6
7 #         def displayEmployee(self):

```

## 3. No 3

```

1 #         print ("Name : ", self.name, ", Salary: ", self.salary)
2
3
4 #This would create first object of Employee class"
5 #emp1 = Employee("Izza", 2000)
6 #This would create second object of Employee class"
7 #emp2 = Employee("Manni", 5000)

```

## 4. No 4

```

1 #emp1.displayEmployee()
2 #emp2.displayEmployee()
3 #print ("Total Employee %d" % Employee.empCount)

```

## 5. No 5

```

1
2 #import belajar
3 #a = 10
4 #b = 5
5
6 #c = belajar.penambahan(a,b)
7 #print(c)
8
9 #def penanganan_error(a,b):
10 #     try :
11 #         c = a+b
12 #         print(c)
13 #     except TypeError:

```

## 6. No 6

```

1 #         print("We Are Different")

```

## 7. No 7

```

1 #Chapter 3
2 #No 1
3 def penulisan(npm):
4     npm = list(str(npm))

```

## 8. No 8

```

1  angka1 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
2  ##### ", "4": " ###  ### ", "5": " ##### ", "6": " ##### ", "7": "
   ##### ", "8": " ##### ", "9": " ##### "}
3  angka2 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
   ##### ", "4": " ###  ### ", "5": " ##### ", "6": " ##### ", "7": "
   ##### ", "8": " ##### ", "9": " ##### "}
4  angka3 = {"0": " ##  ## ", "1": " ### ", "2": " ##### ", "3": "
   ##### ", "4": " ###  ### ", "5": " ##### ", "6": " ##### ", "7": "
   ##### ", "8": " ##### ", "9": " ##### "}
5  angka4 = {"0": " ##  ## ", "1": " ### ", "2": " ##### ", "3": "
   ##### ", "4": " ###  ### ", "5": " ##### ", "6": " ##### ", "7": "
   ##### ", "8": " ##### ", "9": " ##### "}
6  angka5 = {"0": " ##  ## ", "1": " ### ", "2": " ##### ", "3": "
   ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
   ##### ", "8": " ##### ", "9": " ##### "}
7  angka6 = {"0": " ##  ## ", "1": " ### ", "2": " ##### ", "3": "
   ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
   ##### ", "8": " ##### ", "9": " ##### "}
8  angka7 = {"0": " ##  ## ", "1": " ### ", "2": " ##### ", "3": "
   ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
   ##### ", "8": " ##### ", "9": " ##### "}
9  angka8 = {"0": " ##  ## ", "1": " ### ", "2": " ##### ", "3": "
   ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
   ##### ", "8": " ##### ", "9": " ##### "}

```

## 9. No 9

```

1  angka9 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
   ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
   ##### ", "8": " ##### ", "9": " ##### "}
2  angka10 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
   ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
   ##### ", "8": " ##### ", "9": " ##### "}

```

## 10. No 10

```

1  hasil1 = []
2  hasil2 = []
3  hasil3 = []

```

## 2.8.12 Ketrampilan Penanganan Error

## 2.9 Aulyardha Anindita

### 2.9.1 Jenis-jenis Variabel

### 2.9.2 Bilangan (Number)

Pada bilangan (Number) ada dua yang sering dipakai, yaitu : integer dan float. Integer adalah bilangan bulat, sedangkan float adalah bilangan pecahan. Selain itu ada tipe bilangan lain, yaitu bilangan kompleks yaitu bilangan yang memiliki bagian real dan

imaginer. Integer, float, dan kompleks masing-masing di Python diwakili oleh kelas `int`, `float`, dan `complex`.

Contoh kode python :

```
x = 5
print(x, "angkanya adalah ", type(x))
x = 2.5
print(x, "angkanya adalah ", type(x))
```

## 2.9.3 String

String adalah satu atau serangkaian karakter yang diletakkan diantara tanda kutip, baik tanda kutip tunggal ( `'` ) maupun ganda ( `"` ). Huruf, angka, maupun karakter lainnya yang digabung menjadi teks adalah contoh string. String adalah tipe data yang anggotanya berurut dan memiliki indeks. Indeks dimulai dari angka 0 bila dimulai dari depan dan -1 bila diindeks dari belakang. Tiap karakter bisa diakses menggunakan indeksnya dengan `formatnamastring[indeks]` . Pada string juga bisa dilakukan slicing atau mengakses sekelompok substring dengan format `namastring[awal:akhir]`

Contoh kode python :

```
kalimat = "Nama saya Aulyardha Anindita"
print(kalimat)
print(kalimat[0])
print(kalimat[-1])
print(kalimat[4:7])
print(kalimat[:4])
```

## 2.9.4 List

List adalah tipe data yang berisi item yang berurut. Seperti halnya tipe data string, tiap item (anggota) list memiliki indeks sesuai dengan urutannya. Indeks dimulai dari 0 dan bukan dari 1. List bisa berisi anggota dengan tipe yang sama maupun berbeda. Untuk mendeklarasikan list, digunakan tanda kurung `[]` dan masing-masing anggotanya dipisahkan oleh tanda koma.

Contoh kode python :

```
a = [5,10,15,20,25,30,35,40]
a[2] = 15
print("a[2] = ", a[2])
a[0:3] = [5, 10, 15]
print("a[0:3] = ", a[0:3])
a[5:] = [30, 35, 40]
print("a[5:] = ", a[5:])
```

### 2.9.5 Tuple

Tuple adalah jenis data lain yang mirip dengan list. Perbedaannya dengan list adalah anggotanya tidak bisa diubah (immutable). List bersifat mutable, sedangkan tuple bersifat immutable. Sekali tuple dibuat, maka isinya tidak bisa dimodifikasi lagi. Tuple dideklarasikan dengan menggunakan tanda kurung ( ). dan anggotanya dipisahkan oleh tanda koma. Tuple berguna untuk data yang dimaksudkan tidak diubah isinya. Misalnya tuple komposisi warna untuk putih adalah

(255,255,255). Contoh kode python :

```
white = (255,255, 255)
```

```
red = (255,0,0)
```

```
print(white)
```

```
print(red[0])
```

```
print(red[1])
```

```
red[0] = 128
```

### 2.9.6 Set

Set adalah salah satu tipe data di Python yang tidak berurut (unordered). Set memiliki anggota yang unik (tidak ada duplikasi). Jadi misalnya kalau kita meletakkan dua anggota yang sama di dalam set, maka otomatis set akan menghilangkan yang salah satunya. Set bisa digunakan untuk melakukan operasi himpunan matematika seperti gabungan, irisan, selisih, dan komplemen. Set dibuat dengan meletakkan anggota – anggotanya di dalam tanda kurung kurawal , dipisahkan menggunakan tanda koma. Kita juga bisa membuat set dari list dengan memasukkan list ke dalam fungsi set() Set bisa berisi data campuran, baik integer, float, string, dan lain sebagainya. Akan tetapi set tidak bisa berisi list, set, dan dictionary.

Contoh Kode Python :

```
myset = 1,2,3
```

```
print(myset)
```

### 2.9.7 Dictionary

Dictionary adalah tipe data yang tiap anggotanya terdiri dari pasangan kunci-nilai (key-value). Mirip dengan kamus dimana ada kata ada arti. Dictionary umumnya dipakai untuk data yang besar dan untuk mengakses anggota data secara acak. Anggota dictionary tidak memiliki indeks. Dictionary dideklarasikan dengan menggunakan tanda kurung kurawal , dimana anggotanya memiliki bentuk kunci:nilai atau key:value dan tiap anggota dipisah tanda koma. Kunci dan nilainya bisa memiliki tipe sembarang.

Contoh Kode Python :

```
d = 1:'satu', 2:'dua', 'tiga':3
```

```
print(tipe(d))
```

```
print("d[1] = ", d[1])  
print("d['tiga'] = ", d['tiga'])
```

### 2.9.8 Cara Meminta Input Kepada User dan Bagaimana Outputnya

Input adalah masukan yang kita berikan ke program. Program akan memprosesnya dan menampilkan hasil outputnya. Input, proses, dan output adalah inti dari semua program komputer.

### 2.9.9 Input

Python sudah menyediakan fungsi `input()` dan `rawinput()` untuk mengambil inputan dari keyboard.

Contoh :

```
nama = rawinput("Nama Kamu Siapa?: ")  
umur = input("Umur Kamu Berapa: ")
```

Akan menampilkan output :

```
print "Hello",nama,"umur kamu adalah",umur,"tahun"  
"Hello Dita, umur kamu adalah 19 Tahun"
```

### 2.9.10 Output

Untuk menampilkan output teks, kita menggunakan fungsi `print()`.

Contoh :

```
nama = "Aulyardha Anindita"  
print "Hello",nama
```

Hasil :

```
"Hello Aulyardha Anindita"
```

### 2.9.11 Operator Dasar Aritmetika dan Cara Mengubah String ke Int Begitu Juga Sebaliknya

### 2.9.12 Operator Dasar Aritmetika

1. `+` = Penjumlahan, Contoh : `x+y`

2. `-` = Pengurangan, Contoh : `x-y`

3.  $*$  = Perkalian, Contoh :  $x*y$
4.  $/$  = Pembagian, Contoh :  $x/y$
5.  $\text{Persen}$  = Modulus, Contoh:  $x \text{ persen } y$
6.  $**$  = Exponentiation, Contoh :  $x ** y$
7.  $//$  = Floor division, Contoh :  $x // y$

### 2.9.13 Cara Mengubah String Ke Int Sebaliknya

**2.9.13.1 String ke Int** Tipe data string merupakan tipe data yang menampung sebuah teks. tipe data string juga bisa dikonversi ke int. Tidak boleh ada satu karakter pun yang berupa huruf. Jika ada hurufnya, maka kita akan mendapatkan error.

Contoh Kode :

```
a = '1234'
integer = int(a)
print(integer)
```

**2.9.13.2 Int ke String** Tipe data integer ini merupakan tipe data bilangan bulat. Namun kita bisa mengkonversinya ke tipe data yang lain. Seperti string, float, complex, dan long.

Contoh Kode :

```
a=40
string = str(a)
print(string)
```

### 2.9.14 Perulangan

#### 2.9.15 While loop

Dengan while loop , kita dapat menjalankan serangkaian pernyataan selama suatu kondisi benar.

Contoh Kode :

```
i = 1
while i < 6:
    print(i)
    if i == 3:
```

```
break  
i += 1
```

### 2.9.16 For Loop

For loop digunakan untuk mengulangi urutan (baik dalam bentuk list, tuple, dictionary, set atau string. Dengan for loop kita dapat mengeksekusi seperangkat pernyataan, satu kali untuk setiap item dalam daftar, tuple, set dll.

Contoh Kode :

```
hewan = ["Kelinci", "Serangga", "Kupu-kupu"]  
for x in hewan:  
    if x == "Kelinci":  
        break  
    print(x)
```

### 2.9.17 Kondisi

#### 2.9.18 If

If digunakan untuk mengantisipasi kondisi yang terjadi saat jalanya program dan menentukan tindakan apa yang akan diambil sesuai dengan kondisi. Jika kondisi bernilai salah False maka statement/kondisi if tidak akan di-eksekusi.

Contoh Kode :

```
nilai = 8  
if(nilai < 7):  
    print("Selamat Anda Lulus")  
if(nilai < 10):  
    print("Selamat Anda Lulus")
```

#### 2.9.19 If else

Pengambilan keputusan (kondisi if else) tidak hanya digunakan untuk menentukan tindakan apa yang akan diambil sesuai dengan kondisi, tetapi juga digunakan untuk menentukan tindakan apa yang akan diambil/dijalankan jika kondisi tidak sesuai.

Kondisi if else adalah kondisi dimana jika pernyataan benar True maka kode dalam if akan dieksekusi, tetapi jika bernilai salah False maka akan mengeksekusi kode di dalam else.

Contoh :

```
nilai = 3  
if(nilai < 7):  
    print("Selamat Anda Lulus")  
else:
```

```
print("Maaf Anda Tidak Lulus")
```

### 2.9.20 Elif

Pengambilan keputusan (kondisi if elif) merupakan lanjutan/percabangan logika dari "kondisi if". Dengan elif kita bisa membuat kode program yang akan menyeleksi beberapa kemungkinan yang bisa terjadi. Hampir sama dengan kondisi "else", bedanya kondisi "elif" bisa banyak dan tidak hanya satu.

Contoh Kode :

```
hariini = "Minggu"
if(hariini == "Senin"):
    print("Saya akan kuliah")
elif(hariini == "Selasa"):
    print("Saya akan kuliah")
elif(hariini == "Rabu"):
    print("Saya akan kuliah")
elif(hariini == "Kamis"):
    print("Saya akan kuliah")
elif(hariini == "Jumat"):
    print("Saya akan kuliah")
elif(hariini == "Sabtu"):
    print("Saya akan kuliah")
elif(hariini == "Minggu"):
    print("Saya akan libur")
```

### 2.9.21 Jenis Error

#### 2.9.22 Syntax errors

Python hanya dapat mengeksekusi sebuah program hanya jika program tersebut berisi baris - baris perintah dengan sintaks yang benar. Kalau dalam program tersebut terdapat kesalahan sintaks maka proses akan berhenti dan menampilkan pesan kesalahan, yang kemudian dikenal sebagai Syntax errors . Sintaks merujuk ke sebuah struktur program dan aturan - aturan yang berperan dalam struktur tersebut, kalimat tersebut akan mempunyai kesalahan sintaks jika penulisan kalimat tidak sesuai dengan aturan yang berlaku.

#### 2.9.23 Runtime errors

Jenis kesalahan kali ini disebut dengan runtime errors, disebut begitu karena kesalahan tidak akan muncul sampai Anda menjalankan program tersebut. Kesalahan ini juga dikenal dengan exceptions atau pengecualian karena biasanya mengindikasikan



sesuatu pengecualian yang buruk telah terjadi. Runtime errors sangat jarang terjadi pada program sederhana pada contoh beberapa bab pertama.

### **2.9.24 Semantic errors**

Jika terdapat kesalahan jenis ini dalam program Anda, program masih akan berjalan dengan lancar dan tidak mengeluarkan pesan kesalahan, tetapi tidak akan sesuai dengan harapan, karna akan terjadi penyimpangan dan berbeda dengan apa yang diinginkan. Karena program tersebut tidak sesuai dengan yang diharapkan dan akan meminta Anda untuk menelusuri kembali program tersebut dari awal untuk memperbaiki algoritmanya, kesalahan ini akan sering muncul pada saat Anda mulai berpengalaman dengan suatu bahasa pemrograman.

### **2.9.25 Debugging**

Debugging merupakan kekayaan intelektual seseorang yang paling tinggi, menantang dan bagian yang paling menarik dari pemrograman. Menurut pendapat beberapa orang, pemrograman dan debugging adalah hal yang sama. Jadi pemrograman adalah sebuah proses yang harus melalui proses beberapa kali debugging untuk mendapatkan hasil yang Anda inginkan. Dalam proses debugging, suatu komentar instruksi program sangat berguna sekali dalam pembacaan suatu kode. Pada umumnya komentar berisi keterangan tentang kegunaan suatu fungsi itu. Sintaksnya adalah tanda kres atau tanda pagar. Setelah meletakkan tanda tersebut, kita dapat mengetikkan kalimat apa saja yang berhubungan dengan suatu instruksi perintah, sebab apapun kalimat tersebut tidak akan di proses oleh interpreter.

### **2.9.26 Exceptions**

Jika terjadi kesalahan pada saat program dijalankan (run-time errors), program tersebut membuat sebuah pengecualian (exceptions). Biasanya program terhenti dan menampilkan pesan kesalahan. Pada setiap kasus, pesan kesalahan dibagi menjadi dua bagian: Jenis kesalahan sebelum titik dua, dan menjelaskan secara spesifik tentang kesalahan tersebut dibagian setelah titik dua. Pada umumnya, interpreter Python juga menampilkan sebuah penelusuran kembali dimana kesalahan pada program tersebut, yang dapat kita lihat sebelumnya.

### **2.9.27 Cara Memakai Try Except**

Kode di awal artikel meminta pengguna untuk memasukkan integer sebagai sebuah input. Jika pengguna tidak menyediakan masukkan integer, program-nya akan menghentikan eksekusi dan memunculkan nilai error exception-nya. Pada bagian ini, kita akan menulis beberapa kode untuk memberi tahu pengguna bahwa masukkan mereka bukanlah nilai integer yang valid.

Langkah pertama dari prosesnya adalah memasukkan kode yang kamu anggap bisa menimbulkan exception di dalam klausa try. Langkah selanjutnya adalah meng-

gunakan keyword `except` untuk menangani exception yang terjadi pada kode di atas. Kode yang dimodifikasi untuk masukkan penggunaanya akan tampak seperti berikut:

```

1
2
3
4
5
6
7
8
keepasking = True
while keepasking:
try:
x = int(input("Please enter a number: "))
print("Dividing 50 by", x,"will give you :","50/x)
except ValueError:
print("The input was not an integer. Please try again...")

```

Apa yang terjadi di sini adalah program-nya mencoba untuk mengeksekusi kodenya di dalam klausa `try`. Jika tidak ada exception yang muncul, programnya akan melewati klausa `except` dan sisa kode-nya akan dieksekusi secara normal. Jika sebuah exception muncul, program-nya akan melewatkan sisa kode di dalam klausa `try` dan tipe dari exception-nya akan dicocokkan dengan nama exception setelah keyword `except`. Dalam kasus yang cocok, kode di dalam klausa `except` akan dieksekusi terlebih dahulu dan sisa setelah klausa `try` akan dieksekusi secara normal.

Ketika kamu memasukkan sebuah integer sebagai masukkannya, program-nya memberikanmu hasil akhir dari pembagian. Ketika nilai non-integral disediakan, program-nya akan mencetak sebuah pesan memintamu untuk mencoba dan memasukkan sebuah integer lagi. Ingat kali ini, programnya tidak akan keluar secara paksa saat kamu memasukkan nilai yang tidak valid.

Kamu bisa memiliki banyak klausa `except` untuk menangani aneka exceptions. Tolong diingat bahwa handler hanya akan menangani exception yang terjadi dan berkoordinasi dengan klausa `try`. Mereka tidak akan menangani exception apapun yang muncul di handler exception yang lainnya.

## 2.9.28 Keterampilan Pemrograman

### 1. Jawaban soal no 1

```

1 #No 1
2 print("+++  +++  ++++++++  +++  +++  ++++++++  ++++++++  +++
    +++")
3 print("+++  +++  ++++++++  +++  +++  ++++++++  ++++++++  +++
    +++")
4 print("+++  +++          +++  +++  +++  +++  +++  +++  +++
    +++")

```

```

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

```

## 2. Jawaban soal no 2

```

1 #No 2
2 npm = input("Masukkan NPM :")
3 hitung = 0
4 while(hitung < 54):
5     print("Halo, " + str(npm) + " Apa kabar?")
6     hitung = hitung +1

```

## 3. Jawaban soal no 3

```

1 #No 3
2 npm = input("Masukkan NPM :")
3 hitung = 0
4 while(hitung < 9):
5     print("Halo, " + str(npm[4:7]) + " Apa kabar?")
6     hitung = hitung +1

```

## 4. Jawaban soal no 4

```

1 #No 4
2 npm = input("Masukkan NPM :")
3 print("Halo, " + str(npm[-3]) + " Apa kabar?")

```

## 5. Jawaban soal no 5

```

1 #No 5
2 a = 1
3 b = 1
4 c = 7
5 d = 4
6 e = 0
7 f = 5
8 g = 4
9 h = a+b+c+d+e+f+g
10 i = a*b*c*d*e*f*g
11
12 print(a,b,c,d,e,f,g)

```

## 6. Jawaban soal no 6

```
1 #No 6
2 print(h)
```

#### 7. Jawaban soal no 7

```
1 #No 7
2 print(i)
```

#### 8. Jawaban soal no 8

```
1 #No 8
2 print(a)
3 print(b)
4 print(c)
5 print(d)
6 print(e)
7 print(f)
8 print(g)
```

#### 9. Jawaban soal no 9

```
1 #No 9
2 print(d,g)
```

#### 10. Jawaban soal no 10

```
1 #No 10
2 print(a,b,c,f)
```

#### 11. Jawaban soal no 11

```
1 #No 11
2 print(c,f)
```

## 2.9.29 Keterampilan Penanganan Error

```
1 #Keterampilan Penanganan Error
2
3 a = 1
4 b = "2"
5
6 try:
7     c = a + b
8     print(c)
9 except TypeError:
10    print("Tipe datanya beda, anda harus mengubah tipe data sehingga
    bisa dijumlahkan!")
```

## 2.10 Difa Al Fansha

### 2.10.1 Teori

1. Variabel merupakan tempat menyimpan data Aturan Penulisan Variabel :

- Nama variabel boleh diawali menggunakan huruf atau garis bawah.
- Karakter selanjutnya dapat berupa huruf, garis bawah atau angka.
- Karakter pada nama variabel bersifat sensitif (case-sensitif). Artinya huruf besar dan kecil dibedakan.
- Nama variabel tidak boleh menggunakan kata kunci yang sudah ada dalam python seperti if, while, for, dsb.

'Contoh :'

```
x = 21
y = "Difa Al Fansha"
print(x)
print(y)
```

2. Input adalah masukan yang kita berikan ke program dan program akan memproses dan menampilkan hasil outputnya.

'Contoh :'

```
# Mengambil input
nama = input("Siapa nama kamu : ")
umur = input("Berapa umur kamu : ")

# Menampilkan output
print ("Hello",nama,"umur kamu adalah",umur,"tahun")
```

3. Operator Dasar Aritmatika.

'Contoh :'

```
a = 21
b = 7
c = "Difa Al Fansha"
d = "17"
# Tambah
print (a + b)
# Kurang
print (a - b)
# Kali
print (a * b)
# Bagi
print (a / b)
# String ke int
print ("Nama :",c, "Dan", "Umur :", int(d))
# Int ke String
print (str(a) + str(b))
```

#### 4. Perulangan

'Contoh :'

```
# For Loops
Nama = ["Difa Al Fansha"]
for x in Nama :
    print (x)

# While Loops
e = 1
while e < 6 :
    print (e)
    e += 1
```

5. Kondisi if ini bisa digunakan beberapa cara, kondisi if digunakan jika user memerlukan kondisi yang memerlukan pernyataan.

'Contoh :'

```
f = 50
g = 200
if g > f :
    print ("G Lebih besar dari F")
    if g == 200:
        print ("Nilai G adalah 200")
```

#### 6. Error yang sering ditemui

SyntaxError: invalid syntax

Cara mengatasi

```
f = 50
g = 200
if g > f :
    print ("G Lebih besar dari F")
    if g == 200:
        print ("Nilai G adalah 200")
else :
    print ("F Lebih besar dari G")
```

7. Ketika sebuah error ditemukan, untuk mengatasi itu, pernyataan try dan except. Try berfungsi untuk menangkap error, dan except berguna untuk mengatasi error tersebut.

Apabila try benar, sistem akan menjalankan try dan tidak mengeksekusi except, apabila try teradpat error , sistem akan menjalankan except.

```
try :
    print (Hello)
except :
    print ("Ada yang salah")
```

## 2.10.2 Ketrampilan Pemrograman

### 1. Jawaban Nomor 1

```
1 print (1174076%3)
2 # mod 3 = 1
3
4 print(" ## ## #####      ##      #####      #####")
5 print("#### #####      ##### ##      #####      ##")
6 print("### ##      ## ## ## ##      ##      ##")
7 print("### ##      ## ##### ##      ##      ##      #####")
8 print("### ##      ##      ## ##      ##      ##      ##")
9 print("### ##      ##      ##      #####      ##      #####")
```

### 2. Jawaban Nomor 2

```
1 Npm = input("Input : ")
2 sebanyak = 1
3 while(sebanyak <= 76):
4     print("Halo , "+str(Npm)+" apa kabar?")
5     sebanyak += 1
```

### 3. Jawaban Nomor 3

```
1 Npm = input("Input NPM : ")
2 sebanyak = 1
3 while(sebanyak <= 13):
4     print("Halo , "+str(Npm[4:7])+" Apa kabar?")
5     sebanyak += 1
```

### 4. Jawaban Nomor 4

```
1 Npm = input("Input Npm : ")
2 print("Halo , "+str(Npm[-3])+" Apa kabar?")
```

### 5. Jawaban Nomor 5

```
1 h = 1
2 i = 1
3 j = 7
4 k = 4
5 l = 0
6 m = 7
```

```

7 n = 6
8
9 Npm = [h,i,j,k,l,m,n]
10
11 for y in Npm:
12     print(y, end = "")
13 print("")

```

## 6. Jawaban Nomor 6

```

1 print(h+i+j+k+l+m+n)

```

## 7. Jawaban Nomor 7

```

1 print(h*i*j*k*l*m*n)

```

## 8. Jawaban Nomor 8

```

1 for z in Npm:
2     print(z)

```

## 9. Jawaban Nomor 9

```

1 for n in Npm:
2     if(n % 2 == 0):
3         if(n != 0):
4             print(n, end = "")

```

## 10. Jawaban Nomor 10

```

1 for n in Npm:
2     if(n % 2 != 0):
3         print(n, end = "")
4 print()

```

## 11. Jawaban Nomor 11

```

1 for n in Npm:
2     if(n % 2 == 0):
3         if(n != 0):
4             print(n, end = "")

```

## 2.10.3 Keterampilan Penanganan Error

### 1. Jawaban Nomor 1 : Error

```

a = 100
b = "50"
c = a + b
print("Hasil :", c)

```

### 2. Jawaban Nomor 2 : Solusi



```
a = 100
b = "50"

try:
    c = a + int(b)
    print("Hasil :", c)
except TypeError :
    print("Tipe data tidak sama")
```

## 2.11 Ilham Muhammad Ariq

### 2.11.1 Teori

1. sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut di kode Python Variabel merupakan tempat menyimpan data. Kemudian didalam python kita dapat membuat variable dengan berbagai cara, yakni sebagai berikut

```
1 #No 1
2 #Boolean
3 boolean1 = True
4 print(boolean1)
5
6 #String
7 str1 = "Ariq"
8 print(str1)
9
10 #Integer
11 int1 = 1174087
12 print(int1)
13
14 #Float
15 float1 = 5.5
16 print(float1)
17
18 #Hexadecimal
19 hexadecimal1 = 0x5
20 print(hexadecimal1)
21
22 #Complex
23 complex1 = 5j
24 print(complex1)
25
26 #List
27 list1 = [1,2,3]
28 print(list1)
29 print(list1[0])
30
31 #Tuple
32 tuple1 = (1,2,3)
33 print(tuple1)
34 print(tuple1[0])
35
```

```

36 #Set
37 set1 = {1,2,3}
38 print(set1)
39
40
41 #Dictionary
42 dictionary1 = {1:'satu', 2:'dua', 'tiga':3}
43 print(dictionary1)
44 print(dictionary1[1])

```

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

```

1 #No 2
2 #Input dan Output
3 npm = input('Masukan Npm anda : ')
4 print(npm, "adalah Npm kamu")

```

3. Tuliskan operator dasar aritmatika, tambah, kali, kurang bagi, dan bagaimana mengubah string ke integer dan integer ke string Operator aritmatika adalah operator yang digunakan untuk melakukan suatu perhitungan pada program

```

1 #No 3
2 a = 2 #int
3 b = 3 #int
4 c = "1" #str
5
6 print(a+b) #tambah
7 print(a-b) #kurang
8 print(a*b) #kali
9 print(a/b) #bagi
10 print(int(c)) # string ke integer
11 print(str(a)) # int ke string

```

4. Tuliskan dan jelaskan sintak untuk perulangan, jenis-jenisnya contoh kode dan cara pakainya di python Untuk Perulangan Pada Python ada For dan While, contoh ada dibawah ini :

```

1 #No 4
2 #While Loop
3 hitung = 1
4 while (hitung < 5):
5     print (hitung)
6     hitung += 1
7
8 #For Loop
9 angka = [1,2,3]
10 for x in angka:
11     print(x)

```

5. Tuliskan jelaskan cara pakai sintak untuk memilih kondisi, dan bagaimana contoh sintak kondisi di dalam kondisi. Pengambilan kondisi If yang digunakan untuk mengantisipasi kondisi yang terjadi saat program dijalankan dan menentukan tindakan apa yang akan diambil sesuai dengan kondisi. If statement

```

1 #If
2 nilai = 70
3 if nilai > 70:
4     print("Grade A")

```

### If Else

```

1 #If Else
2 nilai = 50
3 if angka > 70:
4     print("Grade A")
5 else:
6     print("Grade B")

```

### Elif

```

1 #Elif
2 urutan = input("Masukan Urutan:")
3 jenis = input("Masukan Jenis:")
4 if urutan == 1:
5     print("Kamu anak pertama")
6 elif jenis == "perempuan":
7     print("Benar")
8 else:
9     print("Salah")

```

6. Tuliskan apa saja jenis error yang sering ditemui di python dalam mengerjakan sintak diatas. dan bagaimana cara mengatasinya

- Exception Kelas dasar untuk semua pengecualian / exception
- StopIteration Dibesarkan ketika metode (iterator) berikutnya dari iterator tidak mengarah ke objek apa pun.
- SystemExit Dibesarkan oleh fungsi sys.exit ().
- StandardError Kelas dasar untuk semua pengecualian built-in kecuali StopIteration dan SystemExit.
- ArithmeticError Kelas dasar untuk semua kesalahan yang terjadi untuk perhitungan numerik.
- OverflowError Dibesarkan saat perhitungan melebihi batas maksimum untuk tipe numerik.
- FloatingPointError Dibesarkan saat perhitungan floating point gagal.
- ZeroDivisonError Dibesarkan saat pembagian atau modulo nol dilakukan untuk semua tipe numerik.
- AssertionError Dibesarkan jika terjadi kegagalan pernyataan Assert.
- AttributeError Dibesarkan jika terjadi kegagalan referensi atribut atau penugasan.
- EOFError Dibesarkan bila tidak ada input dari fungsi rawinput () atau input () dan akhir file tercapai.

- `ImportError` Dibesarkan saat sebuah pernyataan impor gagal.
- `KeyboardInterrupt` Dibesarkan saat pengguna menyela eksekusi program, biasanya dengan menekan `Ctrl + c`.
- `LookupError` Kelas dasar untuk semua kesalahan pencarian.
- `IndexError` Dibesarkan saat sebuah indeks tidak ditemukan secara berurutan.
- `KeyError` Dibesarkan saat kunci yang ditentukan tidak ditemukan dalam kamus.
- `NameError` Dibesarkan saat pengenalan tidak ditemukan di namespace lokal atau global.
- `UnboundLocalError` Dibesarkan saat mencoba mengakses variabel lokal dalam suatu fungsi atau metode namun tidak ada nilai yang ditugaskan padanya.
- `EnvironmentError` Kelas dasar untuk semua pengecualian yang terjadi di luar lingkungan Python.
- `IOError` Dibesarkan saat operasi input / output gagal, seperti pernyataan cetak atau fungsi `open()` saat mencoba membuka file yang tidak ada.
- `OSError` Dibangkitkan untuk kesalahan terkait sistem operasi.
- `SyntaxError` Dibesarkan saat ada kesalahan dengan sintaks Python.
- `IndentationError` Dibesarkan saat indentasi tidak ditentukan dengan benar.
- `SystemError` Dibesarkan saat penafsir menemukan masalah internal, namun bila kesalahan ini ditemui juru bahasa Python tidak keluar.
- `SystemExit` Dibesarkan saat juru bahasa Python berhenti dengan menggunakan fungsi `sys.exit()`. Jika tidak ditangani dalam kode, menyebabkan penafsir untuk keluar.
- `TypeError` Dibesarkan saat operasi atau fungsi dicoba yang tidak valid untuk tipe data yang ditentukan.
- `ValueError` Dibesarkan ketika fungsi bawaan untuk tipe data memiliki jenis argumen yang valid, namun argumen tersebut memiliki nilai yang tidak valid yang ditentukan.
- `RuntimeError` Dibesarkan saat kesalahan yang dihasilkan tidak termasuk dalam kategori apa pun.
- `NotImplementedError` Dibesarkan ketika metode abstrak yang perlu diimplementasikan di kelas warisan sebenarnya tidak dilaksanakan.

## 7. Tuliskan dan jelaskan cara memakai Try Except.

```

1 #No 7
2 try:
3     print(ter)
4 except NameError:
5     print("Variable x is not defined")
6 except:
7     print("Something else went wrong")

```

## 2.11.2 Ketrampilan Pemrograman

1. Buatlah luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk NPM mod 3=0, tanda pagar untuk NPM mod 3 =1, tanda plus untuk NPM mod3=2.

```

1 #No 1
2 print(1174087%3)
3
4 print(" ##  ## #####   ##  ##  #####  #####")
5 print("###  ##   ##  ##  ##  ##  ##  ##  ##")
6 print(" ##  ##   ##  #####  ##  ##  #####  ##")
7 print(" ##  ##  ##   ##  ##  ##  ##  ##  ##")
8 print(" ##  ##  ##   ##  ##  #####  #####  ##")

```

2. Buatlah program hello word dengan input NPM yang disimpan dalam sebuah variabel string bernama NPM dan output sebanyak dua digit belakang NPM, contoh NPM : 113040087 maka akan ada output sebanyak 87 dengan tulisan 'Hallo, 113040087 apa kabar?'

```

1 #No 2
2 i = 1
3 npm = input("Masukan Npm:")
4 while i < 87:
5     print("Hallo ,",npm, "apa kabar?" )
6     i += 1

```

3. Buatlah program hello word dengan input nama yang disimpan dalam sebuah variabel string bernama NPM dan beri luaran output berupa tiga karakter belakang dari NPM sebanyak penjumlahan tiga digit tersebut.

```

1 #No 3
2 i = 1
3 npm = input("Masukan Npm:")
4 while i < 15:
5     print("Hallo ,",npm[4:8], "apa kabar?" )
6     i += 1

```

4. Buatlah program hello word dengan input nama yang disimpan dalam sebuah variabel string bernama NPM dan beri luaran output berupa digit ketiga dari belakang dari variabel NPM,

```

1 #No 4
2 npm = input("Masukan Npm:")
3 print("Hallo ,",npm[4], "apa kabar?" )

```

5. (untuk soal no 5 dan selanjutnya) buat program dengan mengisi variabel alfabet dengan nomor npm satu persatu berurut.

```

1 #No 5
2 a = 1
3 b = 1
4 c = 7

```

```

5 d = 4
6 e = 0
7 f = 8
8 g = 7
9 print(str(a)+str(b)+str(c)+str(d)+str(e)+str(f)+str(g))

```

6. Dari soal no 5, Lakukan penjumlahan dari seluruh variabel tersebut.

```

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

```

7. Dari soal no 5, Lakukan perkalian dari seluruh variabel tersebut.

```

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

```

8. Dari soal no 5, Lakukan print secara vertikal dari NPM anda menggunakan variabel diatas.

```

1 #No 8
2 npm = [a,b,c,d,e,f,g]
3 for x in npm:
4     print(x)

```

9. Dari soal no 5, Lakukan print NPM anda tapi hanya digit genap saja.

```

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

```

10. Dari soal no 5, Lakukan print NPM anda tapi hanya digit ganjil saja.

```

1 #No 10
2 for n in npm:
3     if(n % 2 != 0):
4         print(n, end = "")

```

11. Dari soal no 5, Lakukan print NPM anda tapi hanya digit yang termasuk bilangan prima saja.

```

1 #No 11
2 print(str(c)+str(g))

```

### 2.11.3 Keterampilan Penanganan Error

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar  5 00:25:12 2019
4
5 @author: PandA23

```

```

6 """
7
8 #Keterampilan Penanganan Error
9 a = 1
10 b = "2"
11
12 try :
13     c = a + b
14     print(c)
15 except TypeError:
16     print("Tipe data nya berbeda")

```

## 2.12 Muhammad Reza Syachrani / 1174084

### 2.12.1 Teori

- Variable dibuat saat anda pertama kali memberikan nilai padanya yang merupakan tempat penyimpanan untuk teks dan angka, variable juga tidak perlu dideklarasikan dengan tipe tertentu dan dapat mengubah jenisnya setelah sudah ditetapkan. Variabel selalu ditugaskan dengan tanda sama dengan, diikuti oleh nilai variable dan harus memiliki nama sehingga kita dapat menemukannya lagi. Aturan untuk variabel Python:
  - Nama variable harus dimulai dengan huruf atau karakter garis bawah
  - Nama variable tidak dapat dimulai dengan angka
  - Nama variable hanya dapat berisi karakter alfanumerik dan garis bawah (A-z, 0-9, dan "-")
  - Nama variable tidak bisa menggunakan kata yang disediakan untuk Python.
- Kode untuk meminta inputan dari pengguna yaitu menggunakan perintah "input()" dan apabila ingin menampilkan output pada layer menggunakan perintah "print()".
- Operator dasar aritmatika :
  - Tambah (+)  
Contoh : x + y
  - Kurang (-)  
Contoh : x - y
  - kali (\*)  
Contoh : x \* y
  - Bagi (/)  
Contoh : x / y

Cara mengubah tipe data string ke integer :

```

1 a = '1174084' #variabel/angka yang akan di konversi.
2 integer = int(a) #konversi string ke integer

```

Cara mengubah tipe data integer ke string :

```
1 a = 1174084 #variabel/angka yang akan di konversi.
2 string = str(a) #konversi integer ke string
```

4. Pada perulangan terdapat dua jenis di dalam python yaitu perulangan while dan perulangan for

▪ Perulangan while :

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

▪ Perulangan for :

```
1 fruits = ["apple", "banana", "cherry"]
2 for x in fruits:
3     print(x)
```

5. sintak Untuk memilih kondisi, dan kondisi didalam kondisi

▪ If Kondisi ini dipergunakan jika penkondisiannya hanya satu.

```
1 a = 33
2 b = 200
3 if b > a:
4     print("b lebih besar dari a")
```

▪ Elif Kondisi ini dipergunakan jika pengkondisiannya lebih dari dua dan jika kondisi sebelumnya tidak benar, maka coba kondisi ini akan dijalankan

```
1 a = 33
2 b = 33
3 if b > a:
4     print("b lebih besar dari a")
5 elif a == b:
6     print("a dan b sama")
```

▪ Kondisi di dalam kondisi Kondisi ini dipergunakan jika pengkondisiannya memerlukan pengkondisian di dalamnya.

```
1 gaji = 10000000
2 berkeluarga = True
3
4 if gaji > 3000000:
5     print("Gaji sudah diatas UMR")
6     if berkeluarga:
7         print("Wajib ikutan asuransi dan menabung untuk pensiun")
8     else:
9         print("Tidak perlu ikutan asuransi")
10 else:
11     print("Gaji belum UMR")
```



## 6. Jenis-jenis error yang terjadi dan cara mengatasinya pada Python

### ▪ Type Error

`TypeError` merupakan exception yang terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

### ▪ Syntax Error

`Syntax Error` adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.

### ▪ Name Error

`Name Error` 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.

## 7. Try Except pada Python merupakan cara untuk menguji kesalahan kode agar kita dapat menangani kesalahannya. Contoh penggunaan Try Except :

```
1 try :
2     print(x)
3 except:
4     print("An exception occurred")
```

## 2.12.2 Keterampilan Pemrograman

### 1. Jawaban Soal 1

```
1 print (1174084%3)
2
3 print ("*** ***)
4 print ("*** ***)
5 print ("*** ***)
6 print ("*** ***)
7 print ("*** ***)
8 print ("*** ***)
9 print ("*** ***)
10 print ("*** ***)
11 print ("*** ***)
```

### 2. Jawaban Soal 2

```

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

```

### 3. Jawaban Soal 3

```

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

```

### 4. Jawaban Soal 4

```

1 npm = input("Masukan NPM :")
2 print("Halo, " + str(npm[-3]) + " Apa Kabar?")

```

### 5. Jawaban Soal 5

```

1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 8
7 g = 4
8 npm = [a,b,c,d,e,f,g]
9
10 for a in npm:
11     print(a, end = "")

```

### 6. Jawaban Soal 6

```

1 print(a+b+c+d+e+f+g)

```

### 7. Jawaban Soal 7

```

1 print(a*b*c*d*e*f*g)

```

### 8. Jawaban Soal 8

```

1 for a in npm:
2     print(a)

```

### 9. Jawaban Soal 9

```

1 for n in npm:
2     if(n % 2 == 0):
3         if(n != 0):
4             print(n, end = "")

```

## 10. Jawaban Soal 10

```
1 for n in npm:
2     if (n % 2 != 0):
3         print(n, end = "")
```

## 11. Jawaban Soal 11

```
1 print(c)
```

### 2.12.3 Ketrampilan Penanganan Error

#### 1. Jenis- jenis error dan penanganan error

##### ▪ Type Error

TypeError merupakan exception yang terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

##### ▪ Syntax Error

Syntax Error adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.

##### ▪ Name Error

Name Error 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.

#### 2. Jawaban :

```
1 a = 1
2 b = "2"
3
4 try:
5     c = a + b
6     print(c)
7 except TypeError:
8     print("Tipe Data Berbeda!")
```

## 2.13 Handi Hermawan

### 2.13.1 Teori

1. sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut di kode Python Variable merupakan tempat yang dapat digunakan untuk menyimpan data, dalam python kita bisa membuat variable dengan cara berikut
2. Kode untuk meminta input dari user dan bagaimana melakukan output ke layar seperti pada gambar2

3. Operator dasar aritmatika Terdapat penambahan, pengurangan perkalian, perkalian, pembagian, modulus
4. Perulangan ada dua jenis perulangan di dalam python mereka adalah perulangan while dan perulangan for
5. sintak Untuk memilih kondisi Kondisi IF digunakan ketika ingin menentukan tindakan apa yang harus digunakan sesuai dengan kondisi yang telah diatur
6. Jenis-jenis error pada python Syntax Errors adalah keadaan dimana kode python mengalami kesalahan penulisan. IndentationError adalah error yang terjadi saat indentasi error. SystemError adalah error yang terjadi ketika interpreter mendeteksi error internal TypeError adalah error yang terjadi saat dilakukan eksekusi pada suatu operasi atau fungsi dengan type object yang tidak sesuai. ValueError adalah error ketika value yang dimasukan tidak sesuai UnicodeTranslateError adalah error yang muncul ketika mentranslate unicode UnicodeDecodeError adalah error yang muncul ketika proses decode unicode UnicodeEncodeError adalah error yang muncul ketika proses encode unicode UnicodeError adalah error yang muncul ketika error terkait unicode terdeteksi

## 2.13.2 Keterampilan Pemrograman

```

1  Spyder Editor
2
3  This is a temporary script file .
4  """
5
6  #No.1
7  print(1174080%3)
8
9  print("***      *****      ***      *****      *****
10         *****")
11 print("***      *****      ***      ***      ***      ***      ***
12         *****")
13 print("***      ***      ***      ***      ***      ***      ***      ***
14         *****")
15 print("***      ***      ***      ***      ***      ***      ***      ***
16         *****")
17 print("***      ***      ***      ***      ***      ***      ***      ***
18         *****")
19 #No.2
20 npm = input("Masukan NPM :")
21 hitung = 0

```

```
22 while(hitung < 2):
23     print("Halo, " + str(npm) + " Apa Kabar?")
24     hitung = hitung +1
25
26 #No.3
27 npm = input("Masukan NPM : ")
28 hitung = 0
29 while(hitung < 2):
30     print("Halo, " + str(npm[4:7]) + " Apa Kabar?")
31     hitung = hitung +1
32
33 #No.4
34 npm = input("Masukan NPM : ")
35 print("Halo, " + str(npm[-3]) + " Apa Kabar?")
36
37 #No.5
38 a = 1
39 b = 1
40 c = 7
41 d = 4
42 e = 0
43 f = 8
44 g = 0
45 h = a+b+c+d+e+f+g
46 i = a*b*c*d*e*f*g
47
48 print (a,b,c,d,e,f,g)
49
50 #No.6
51 print(h)
52
53 #No.7
54 print(i)
55
56 #No.8
57 print(a)
58 print(b)
59 print(c)
60 print(d)
61 print(e)
62 print(f)
63 print(g)
64
65 #No.9
66 print(d,f)
67
68 #No.10
69 print(a,c)
70
71 #No.11
72 print(2)
```

### 2.13.3 Keterampilan dan penanganan error

1. Tuliskan peringatan error yang didapat dari mengerjakan praktek kedua ini, dan jelaskan cara penanganan 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 mengkonversi variabelnya sesuai dengan tipe data sesuai dengan yang akan digunakan.

## 2.14 Mochamad Arifqi Ramadhan

Praktek teori penunjang yang dikerjakan :

### 2.14.1 Variable, pemakaian Variable dan Jenis-Jenis Type data

Variabel merupakan tempat menyimpan data, sedangkan tipe data adalah jenis data yang tersimpan dalam variabel. Variabel bersifat mutable, artinya nilainya bisa berubah-ubah.

#### 1. Pemakaian Variabel

Variabel di python dapat dibuat dengan format seperti ini:

NamaVariabel = (nilai)

Contoh:

Variabel01 = "isi variabelnya"

variabel022 = 10

Kemudian untuk melihat isi variabel, kita dapat menggunakan fungsi print.

print Variabel01

print variabe02

#### 2. Aturan Penulisan Variabel

- Nama variabel boleh diawali menggunakan huruf atau garis bawah (`_`), contoh: `nama`, `_nama`, `namaKu`, `nama_variabel`.
- Karakter selanjutnya dapat berupa huruf, garis bawah (`_`) atau angka, contoh: `__nama`, `n2`, `nilai1`.
- Karakter pada nama variabel bersifat sensitif (case-sensitif). Artinya huruf besar dan kecil dibedakan. Misalnya, `variabel_Ku` dan `variabel_ku`, keduanya adalah variabel yang berbeda.

- Nama variabel tidak boleh menggunakan kata kunci yang sudah ada dalam python seperti if, while, for, dsb.

### 3. Tipe Data

Cara mengisi nilai variabel ditentukan dengan jenis datanya, misalkan untuk tipe data teks (string) maka harus diapit dengan tanda petik ("..."). Sedangkan untuk angka (integer) dan boolean tidak perlu diapit dengan tanda petik.

### 4. Jenis-Jenis Tipe Data

- Boolean, Contoh *true* atau *false*
- String, Contoh "Belajar Python"
- Integer, Contoh 15 atau 1234
- Float, Contoh 2.5 atau 0.55
- List, Contoh ['abcd', 123, 1.5]

### 5. Meminta input dan melakukan output

`x = input("masukan nama: ") print('Hallo, ' + x)` #dengan perintah ini, akan menampilkan output

### 6. Operator dan Konversi

- Tambah contoh  $x + y$
- Kurang contoh  $x - y$
- Bagi contoh  $x / y$
- Kali contoh  $x * y$
- Modulus contoh  $x$
- Pangkat  $x ** y$
- equal contoh  $x == y$
- not equal contoh  $x != y$
- lebih besar dari contoh  $x > y$
- kurang dari  $x < y$

### 7. Perulangan di Python

- **syntak Pengulangan**
- `i = 0`
- `while i < 6:`
- `i += 1`
- `if i == 3:`

- continue
- print(i)
- Try Except
- try: print(x15)
- except NameError: print("Variable x15 tidak sesuai")
- except: print("Terjadi Error")

### 2.14.2 Ketrampilan Pemrograman

Buat program di python dengan ketentuan :

(a) Jawaban 1

```

1 print (1174074%3)
2
3 print("***      *** *****      ***      *** *****")
4 print("*****      ***      ***")
5 print("*****      ***      ***      ***      ***")
6 print("*****      ***      ***      ***      ***")
7 print("*****      ***      *****      ***      ***")
8 print("*****      ***      ***      ***      ***")
9 print("*****      ***      ***      ***      ***")
10 print("*****      ***      ***      ***      *****")
11 print("*****      ***      ***      ***      ***")

```

(b) Jawaban 2

```

1 npm = input("Masukan NPM :")
2 hitung = 0
3 while(hitung < 74):
4     print("Halo , " + str(npm) + " Apa Kabar?")
5     hitung = hitung +1

```

(c) Jawaban 3

```

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

```

(d) Jawaban 4



```

1 npm = input("Masukan NPM :")
2 print("Halo, " + str(npm[-3]) + " Apa Kabar?")

```

(e) Jawaban 5

```

1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 7
7 g = 4
8 h = a+b+c+d+e+f+g
9 i = a*b*c*d*e*f*g
10 print(a,b,c,d,e,f,g)

```

(f) Jawaban 6

```

1 #No.6
2 print(h)

```

(g) Jawaban 7

```

1 #No.7
2 print(i)

```

(h) Jawaban 8

```

1 print(a)
2 print(b)
3 print(c)
4 print(d)
5 print(e)
6 print(f)
7 print(g)

```

(i) Jawaban 9

```

1 #No.9
2 print(d,g)

```

(j) Jawaban 10

```

1 #No.10
2 print(a,b,c,f)

```

(k) Jawaban 11

```

1 #No.11
2 print(c,g)

```

### 2.14.3 Ketrampilan Penanganan Error

Bagian Penanganan error dari script python.

(a) Jawaban

- i. Syntax Errors, adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.

(b) Jawaban

```

1 a = 1
2 b = "2"
3
4 try :
5     c = a + b
6     print(c)
7 except TypeError :
8     print("Tipe Datanya Berbeda Sehingga Tidak Bisa Bersatu")

```

## 2.15 Fernando Lorencius Sihite

Praktek teori penunjang yang dikerjakan :

### 2.15.1 Variable, pemakaian Variable dan Jenis-Jenis Type data

Variabel merupakan tempat menyimpan data, sedangkan tipe data adalah jenis data yang tersimpan dalam variabel.

#### 1. Pemakaian Variabel

Variabel di python dapat dibuat dengan format seperti ini:

NamaVariabel = (nilai)

Contoh:

a = "Fernando Lorencius S"

b = "renang"

Kemudian untuk melihat isi variabel, kita dapat menggunakan fungsi print.

print (a)

print (b)

Aturan Penulisan Variabel

- Nama variabel boleh diawali menggunakan huruf atau garis bawah (`_`), contoh: `nama`, `_nama`, `namaKu`, `nama_variabel`.
- Karakter selanjutnya dapat berupa huruf, garis bawah (`_`) atau angka, contoh: `__nama`, `n2`, `nilai1`.
- Karakter pada nama variabel bersifat sensitif (case-sensitif). Artinya huruf besar dan kecil dibedakan. Misalnya, `variabel_Ku` dan `variabel_ku`, keduanya adalah variabel yang berbeda.
- Nama variabel tidak boleh menggunakan kata kunci yang sudah ada dalam python seperti `if`, `while`, `for`, dsb.

### Tipe Data

Cara mengisi nilai variabel ditentukan dengan jenis datanya, misalkan untuk tipe data teks (string) maka harus diapit dengan tanda petik ("..."). Sedangkan untuk angka (integer) dan boolean tidak perlu diapit dengan tanda petik.

#### Jenis-Jenis Tipe Data

- Boolean, Contoh *true* atau *false*
- String, Contoh "Belajar Python"
- Integer, Contoh 15 atau 1234
- Float, Contoh 2.5 atau 0.55
- List, Contoh ['abcd', 123, 1.5]

### 2. Meminta input dan melakukan output

Untuk menampilkan output teks, kita menggunakan fungsi `print()`.

Contoh menggunakan input :

`a = "Fernando Lorencius S"`

`b = "renang"`

```
print("Nama Saya :",a)
print("Hobby Saya :",b)
```

### 3. Operator dan Konversi

- Tambah contoh `x + y`
- Kurang contoh `x - y`
- Bagi contoh `x / y`
- Kali contoh `x * y`

Penggunaan Operasi aritmatika

Contoh penggunaan Aritmatika

`a = 10`

`b = 2`

perintah pembagian

```
print ( a / b )
```

perintah pertambahan

```
print ( a + b )
```

perintah pengurangan

```
print ( a - b )
```

Perintah perkalian

```
print ( a * b )
```

#### 4. Perulangan di Python

Perulangan dalam bahasa pemrograman berfungsi menyuruh komputer melakukan sesuatu secara berulang-ulang. Terdapat dua jenis perulangan dalam bahasa pemrograman python, yaitu perulangan dengan for dan while.

Perulangan for disebut counted loop (perulangan yang terhitung), sementara perulangan while disebut uncounted loop (perulangan yang tak terhitung). Perbedaannya adalah perulangan for biasanya digunakan untuk mengulangi kode yang sudah diketahui banyak perulangannya. Sementara while untuk perulangan yang memiliki syarat dan tidak tentu berapa banyak perulangannya.

Contoh Penggunaan For :

```
hobby = ["renang", "tidur", "nonton"]
for x in hobby:
    print(x)
```

Contoh Penggunaan While :

```
i = 1\\
while i < 6:\\
    print(i)\\
    i += 1\\
```

#### 5. Penggunaan Kondisi

Penggunaan Kondisi IF

Pengambilan keputusan (kondisi if) digunakan untuk mengantisipasi kondisi yang terjadi saat jalannya program dan menentukan tindakan apa yang akan diambil sesuai dengan kondisi.

Pada python ada beberapa statement/kondisi diantaranya adalah if, else dan elif. Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar True. Jika kondisi bernilai salah False maka statement/kondisi if tidak akan dieksekusi. Dibawah ini adalah contoh penggunaan kondisi if pada Python

Kondisi if adalah kondisi yang akan dieksekusi oleh program jika bernilai benar atau TRUE

```
nilai = 8\\
```

```
jika kondisi benar/TRUE maka program akan mengeksekusi perintah
if(nilai > 7):\\
    print("Selamat Anda Lulus")\\
```

```
jika kondisi salah/FALSE maka program tidak akan mengeksekusi
if(nilai > 10):\\
    print("Selamat Anda Lulus")\\
```

### Penggunaan KOndisi IF else

Pengambilan keputusan (kondisi if else) tidak hanya digunakan untuk menentukan tindakan apa yang akan diambil sesuai dengan kondisi, tetapi juga digunakan untuk menentukan tindakan apa yang akan diambil/dijalankan jika kondisi tidak sesuai.

Pada python ada beberapa statement/kondisi diantaranya adalah if, else dan elif. Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar.

Kondisi if else adalah kondisi dimana jika pernyataan benar True maka kode dalam if akan dieksekusi, tetapi jika bernilai salah False maka akan mengeksekusi kode di dalam else.

Dibawah ini adalah contoh penggunaan kondisi if else pada Python

Kondisi if else adalah jika kondisi bernilai TRUE maka akan dieksekusi pada if, tetapi jika bernilai FALSE maka akan dieksekusi kode pada else

```
nilai = 6\\
Jika pernyataan pada if bernilai TRUE maka if akan dieksekusi
if(nilai > 7):\\
    print("Selamat Anda Lulus")\\
else:\\
    print("Maaf Anda Tidak Lulus")\\
```

### Penggunaan Kondisi Elif

Pengambilan keputusan (kondisi if elif) merupakan lanjutan/percabangan logika dari “kondisi if”. Dengan elif kita bisa membuat kode program yang akan menyeleksi beberapa kemungkinan yang bisa terjadi. Hampir sama dengan kondisi “else”, bedanya kondisi “elif” bisa banyak dan tidak hanya satu.

```
hari = "Minggu"
```

```
if(hari == "Senin"):
    print("Saya akan kuliah")
elif(hari == "Selasa"):
    print("Saya akan kuliah")
elif(hari == "Rabu"):
    print("Saya akan kuliah")
```

```

elif(hari == "Kamis"):
print("Saya akan kuliah")
elif(hari == "Jumat"):
print("Saya akan kuliah")
elif(hari == "Sabtu"):
print("Saya akan kuliah")
elif(hari == "Minggu"):
print("Saya akan libur")

```

## 6. Kesalahan sintaks

Python hanya dapat mengeksekusi sebuah program hanya jika program tersebut berisi baris - baris perintah dengan sintaks yang benar. Kalau dalam program tersebut terdapat kesalahan sintaks maka proses akan berhenti dan menampilkan pesan kesalahan, yang kemudian dikenal sebagai Syntax errors . Sintaks merujuk ke sebuah struktur program dan aturan - aturan yang berperan dalam struktur tersebut. Sebagai contohnya, dalam bahasa Indonesia, sebuah kalimat harus diawali dengan huruf kapital dan diakhiri dengan tanda titik (.), kalimat tersebut akan mempunyai kesalahan sintaks jika penulisan kalimat tidak sesuai dengan aturan yang berlaku. Hal ini juga berlaku di dalam bahasa pemrograman komputer.

Pada kebanyakan pembaca, beberapa kesalahan sintaks bukanlah sebuah masalah yang serius, seperti penulisan puisi, pantun dan yang lainnya. Tetapi bahasa pemrograman Python bukanlah pemaaf yang baik dalam hal tersebut, jika terdapat satu kesalahan sintaks, maka program akan langsung memberikan pesan kesalahan dan keluar dari program. Pada waktu Anda baru mulai memprogram, mungkin Anda akan banyak menemui kesalahan sintaks tersebut, tetapi lain halnya jika anda sudah terbiasa memprogram, Anda hanya akan menemui beberapa kesalahan dan menemukan kesalahan tersebut dengan cepat.

Kesalahan sintaks, dapat juga disebut dengan kesalahan dalam memarsing kode python yang salah, umumnya ditemui pada saat Anda baru memulai belajar bahasa pemrograman python.

Contohnya :

```

>>> while 1 print 'Hello world'\

File "<stdin>", line 1\\

    while 1 print 'Hello world'\

SyntaxError: invalid syntax\\

```

Pada contoh diatas, interpreter memberitahukan bahwa pada perintah terdapat kesalahan sintaks, interpreter akan menampilkan baris yang salah dan menunjukkan posisi kode yang salah dengan tanda panah kecil, contoh di atas pada penggunaan while seharusnya memberi tanda titik dua ":" setelah kondisi while.

## 7. Cara memakai try except

Salah satu bentuk penanganan error di Python adalah dengan menggunakan statement try..except. Mungkin kita pernah mendeteksi error dengan memanfaatkan kondisional biasa menggunakan if..else, namun hal tersebut akan lebih praktis ditangani bila dengan menggunakan try..except. Anda dapat mengurung suatu blok kode dengan try..except untuk menangani error yang mungkin kita sendiri tidak mengetahuinya. Biasanya try..except ini digunakan untuk menangani error saat penggunaan IO, operasi database, atau pengaksesan indeks suatu list atau dictionary, dan berbagai kasus lainnya.

Sekarang kita akan mengenal beberapa kasus sederhana yang menggunakan try..except.

contoh :

```
try:
    print(x)
except:
    print("An exception occurred")
```

```
try:
    print(x)
except NameError:
    print("Variable x is not defined")
except:
    print("Variabel X tidak dapat di artikan")
```

## 2.15.2 Ketrampilan Pemrograman

Jawaban No.1

```
1 print (1174072%3)
2 print("+++  +++  ++++++++  +++  +++  ++++++++  ++++++++
   ++++++++")
3 print("+++  +++  ++++++++  +++  +++  +++  +++  ++++++++
   ++++++++")
4 print("+++  +++  +++  +++  +++  +++  +++  +++
   +++")
5 print("+++  +++  +++  +++  +++  +++  +++  +++
   +++")
```

```

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

```

Jawaban no.2

```

1 npm = input("Masukan NPM :")
2 hitung = 0
3 while(hitung <=72):
4     print("Halo, " + str(npm) + " Apa Kabar?")
5     hitung += 1

```

Jawaban no.3

```

1 npm = input("masukan NPM : ")
2 ulang = 1
3 while(ulang <= 72):
4     print("Halo, "+str(npm[4:7])+ " apa kabar?")
5     ulang += 1

```

jawaban no.4

```

1 npm = input("Input : ")
2 print("Halo, "+str(npm[-3])+ " apa kabar?")

```

jawaban no.5

```

1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 7
7 g = 2
8
9 npm = [a,b,c,d,e,f,g]
10
11 for n in npm:
12     print(n, end = "")
13     print()

```

jawaban no.6



```
1  
2 print(a+b+c+d+e+f+g)
```

Jawaban no.7

```
1  
2 print(a*b*c*d*e*f*g)
```

jawaban no.8

```
1  
2 for n in npm:  
3     print(n)
```

jawaban no.9

```
1 for n in npm:  
2     if(n % 2 == 0):  
3         if(n != 0):  
4             print(n, end = "")  
5  
6 print()
```

jawaban no.10

```
1 for n in npm:  
2     if(n % 2 != 0):  
3         print(n, end = "")  
4  
5 print()
```

jawaban no.11

```
1 for n in npm:  
2     if(n % 1 != 0):  
3         print(n, end = "")
```

### 2.15.3 Ketrampilan Penanganan Error

```
1 c = 1  
2 d = "4"  
3  
4 try:  
5     a = c - d  
6     print(a)  
7 except:  
8     print("Tolong Ubah Error Tersebut")
```

## 2.16 Arrizal Furqona Gifary

### 2.16.1 Teori

1. sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut di kode Python Variable merupakan tempat yang dapat digunakan untuk menyimpan data, dalam python kita bisa membuat variable dengan cara berikut

```

1
2 #Soal no 1 Tipe data
3
4
5 #contoh integer
6 a = 1
7 b = 1.1
8 z = 2+1j
9
10 print(type(a))
11 print(type(b))
12 print(type(z))
13
14 # contoh string
15
16 a = "Chandra Kirana Poetra"
17 print(a[1])
18 #print ini digunakan untuk mencetak huruf yang dipilih indexnya
19
20 a = "Chandra Kirana Poetra"
21 print(a[2:5])
22 #print ini digunakan untuk mencetak huruf yang dipilih indexnya
    dengan mengisi
23 #nilai index tempat kita memilih index untuk memulai mengambil,
    dan juga nilai index
24 #akhirnya
25
26 a = "Chandra Kirana Poetra"
27 print(len(a))
28 #print ini digunakan untuk menjumlahkan length yang artinya
    panjang dari string a
29
30 a = "Chandra Kirana Poetra"
31 print(a.lower())
32 #print ini digunakan untuk mencetak huruf yang kecil semua
33
34 a = "Chandra Kirana Poetra"
35 print(a.upper())
36 #print ini digunakan untuk mencetak huruf yang besar semua
37
38 #boolean
39
40 a = 10
41 b = 8
42 if (a>b):
43     print("Variabel A lebih besar dibandingkan dengan Variable B")
    )

```

```

44 elif (a<b) :
45     print("Variabel A lebih kecil dibandingkan dengan Variable B")
46 else :
47     print("Variabel A Nilainya sama dengan Variable B")

```

2. Kode untuk meminta input dari user dan bagaimana melakukan output ke layar seperti pada gambar2

```

1 #soal no 2
2 namalengkap = input("Masukan Nama lengkap kamu : ")
3 print("Halo " + str(namalengkap))

```

3. Operator dasar aritmatika Terdapat penambahan, pengurangan perkalian, perkalian, pembagian, modulus

```

1 #soal no 3
2
3 a = 1
4 b = 2
5 print(a+b)# ini merupakan contoh penjumlahan
6 print(a-b)# ini merupakan contoh pengurangan
7 print(a*b)# ini merupakan contoh perkalian
8 print(a/b)# ini merupakan contoh pembagian
9 print(a%b)# ini merupakan contoh modulus

```

4. Perulangan ada dua jenis perulangan di dalam python mereka adalah perulangan while dan perulangan for

```

1 #soal no 4
2 #While looping
3
4 a = 1
5 while a < 10:
6     print(a)
7     a +=1
8
9 # for looping
10 benda = ["Piring", "Sendok", "Garpu"]
11 for x in benda:
12     print(x)

```

5. sintak Untuk memilih kondisi Kondisi IF digunakan ketika ingin menentukan tindakan apa yang harus digunakan sesuai dengan kondisi yang telah diatur

```

1 #soal no 5
2 #contoh if statement
3
4 a = 5
5 b = 10
6 if b > a:
7     print("b ternyata lebih besar dari a")
8
9 #contoh elif statement

```

```

10 a = 5
11 b = 10
12 if a > b:
13     print("a ternyata lebih besar dari b")
14 elif a < b:
15     print("a ternyata lebih kecil dari b")
16
17
18 #contoh else statement
19 a = 5
20 b = 5
21 if a > b:
22     print("a ternyata lebih besar dari b")
23 elif a < b:
24     print("a ternyata lebih kecil dari b")
25 else:
26     print("a dan b sama")

```

6. Jenis-jenis error pada python Syntax Errors adalah keadaan dimana kode python mengalami kesalahan penulisan. IndentationError adalah error yang terjadi saat indentasi error. SystemError adalah error yang terjadi ketika interpreter mendeteksi error internal TypeError adalah error yang terjadi saat dilakukan eksekusi pada suatu operasi atau fungsi dengan type object yang tidak sesuai. ValueError adalah error ketika value yang dimasukan tidak sesuai UnicodeTranslateError adalah error yang muncul ketika mentranslate unicode UnicodeDecodeError adalah error yang muncul ketika proses decode unicode UnicodeEncodeError adalah error yang muncul ketika proses encode unicode UnicodeError adalah error yang muncul ketika error terkait unicode terdeteksi
7. Cara memakai try except Cara pemakaian try except adalah sebagai berikut :

## 2.16.2 Praktek

1. Jawaban soal no 1

```

1 uji ()
2
3
4 def uji_param(nama):
5     print("Nama saya :"+str(nama))
6
7 uji_param(input("Masukan Nama Kamu : "))
8
9 def uji_return(a,b):
10     r = a + b
11     return r

```

2. Jawaban soal no 2

```

1 a = 12
2 b = 70

```

```

3 c = uji_return(a,b)
4 print(c)
5
6 #from fungsi_izal import *
7 #print(penulisan(int(input("Masukan NPM kamu : "))))

```

### 3. Jawaban soal no 3

```

1 #class Employee:
2 #    'Common base class for all employees'
3 #    empCount = 0
4
5 #    def __init__(self, name, salary):
6 #        self.name = name
7 #        self.salary = salary

```

### 4. Jawaban soal no 4

```

1 #        Employee.empCount += 1
2
3 #    def displayCount(self):
4 #        print ("Total Employee %d" % Employee.empCount)

```

### 5. Jawaban soal no 5

```

1 #    def displayEmployee(self):
2 #        print ("Name : ", self.name, " , Salary: ", self.salary)
3
4
5 #This would create first object of Employee class"
6 #emp1 = Employee("Zara", 2000)
7 #This would create second object of Employee class"
8 #emp2 = Employee("Manni", 5000)
9 #emp1.displayEmployee()
10 #emp2.displayEmployee()
11 #print ("Total Employee %d" % Employee.empCount)
12
13 #import belajar

```

### 6. Jawaban soal no 6

```

1 #a = 100
2 #b = 50

```

### 7. Jawaban soal no 7

```

1 #c = belajar.penambahan(a,b)
2 #print(c)

```

### 8. Jawaban soal no 8

```

1 #Ketrampilan Pemrograman
2 #No.1
3 def jawabanNo1():
4

```

```

5  npm = input("Masukan NPM :")
6  npm = list(str(npm))
7
8  angka1 = {"0": "##### ", "1": "  ##", "2": " ##### ", "3": "
          ##### ", "4": "  #####", "5": "##### ", "6": " #####",
9  "7": "##### ", "8": "  ##### ", "9": "  ##### "}
10 angka2 = {"0": "###  ##", "1": "#####", "2": "##  ## ", "3": "
          ##  ## ", "4": "  ##  ##", "5": "##  ## ", "6": "###  ## ",
11 "7": "##### ", "8": "##  ## ", "9": "##  ## "}

```

## 9. Jawaban soal no 9

```

1  angka3 = {"0": "###  ##", "1": "  ##", "2": "  ## ", "3": "
          ##  ## ", "4": "  ##  ##", "5": "##  ## ", "6": "###  ## ",
2  "7": "  ##  ## ", "8": "  ##### ", "9": "##  ## "}
3  angka4 = {"0": "###  ##", "1": "  ##", "2": "  ## ", "3": "
          ##  ## ", "4": "##### ", "5": "##### ", "6": "##### ",
4  "7": "  ##  ## ", "8": "##### ", "9": "##### "}
5  angka5 = {"0": "###  ##", "1": "  ##", "2": "  ## ", "3": "
          ##  ## ", "4": "  ##  ##", "5": "  ##  ## ", "6": "###  ## ",
6  "7": "  ##  ## ", "8": "##  ## ", "9": "  ##  ## "}

```

## 10. Jawaban soal no 10

```

1  angka6 = {"0": "##### ", "1": "  ##", "2": "##### ", "3": "
          ##### ", "4": "  ## ", "5": "##### ", "6": "  ##### ",
2  "7": "###  ## ", "8": "  ##### ", "9": "##### "}
3  hasil1 = []
4  hasil2 = []

```

## 11. Jawaban soal no 11

```

1  hasil3 = []
2  hasil4 = []

```

## 2.16.3 Keterampilan dan penanganan error

```

1  a = "10"
2  b = 5
3
4  try:
5      print(a+b)
6  except TypeError:
7      print("Beda Tipe Data")

```

## 2.17 Alvan Alvanzah/1174077

### 2.17.1 Teori

#### 1. Jenis-jenis variabel dan cara pemakaian pada kode Python.

- Variabel

```
1 x = "Python is "  
2 y = "awesome"  
3 z = x + y  
4 print(z)
```

- Cara Pemakaian, variabel dapat memiliki nama pendek (seperti x dan y) atau nama yang lebih deskriptif. Aturan untuk variabel Python:
  - (a) Nama variabel harus dimulai dengan huruf atau karakter garis bawah
  - (b) Nama variabel tidak dapat dimulai dengan angka
  - (c) Nama variabel hanya dapat berisi karakter alfanumerik dan garis bawah (A-z, 0-9, dan underscore)
  - (d) Nama variabel peka huruf besar-kecil (usia, Usia dan AGE adalah tiga variabel berbeda)

## 2. Kode untuk meminta input dari user dan melakukan output ke layar pada Python.

### ▪ Meminta input

Fungsi input () memungkinkan input pengguna.

```
1 print('Enter your name:')  
2 x = input()  
3 print('Hello , ' + x)
```

### ▪ Output ke layar

Fungsi print () mencetak pesan yang ditentukan ke layar.

```
1 print("Hello World")
```

## 3. Operator dasar aritmatika pada Python.

### ▪ Tambah

```
1 x = 5  
2 y = 3  
3  
4 print(x + y)
```

### ▪ Kurang

```
1 x = 5  
2 y = 3  
3  
4 print(x - y)
```

### ▪ Kali

```
1 x = 5  
2 y = 3  
3  
4 print(x * y)
```

### ▪ Bagi

```

1 x = 12
2 y = 3
3
4 print(x / y)

```

#### ▪ String ke Integer

```

1 a = '1212' #variabel/angka yang akan di konversi.
2
3 integer = int(a) #konversi string ke integer

```

#### ▪ Integer ke String

```

1 a=100 #variabel/angka yang akan di konversi.
2
3 string = str(a) #konversi integer ke string

```

### 4. Sintak perulangan, jenis-jenisnya, dan cara penggunaannya pada Python.

#### ▪ While Loop

Dengan loop sementara, kita dapat menjalankan serangkaian pernyataan selama suatu kondisi benar.

```

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

```

#### ▪ For Loop

Dengan for loop kita dapat mengeksekusi seperangkat pernyataan, satu kali untuk setiap item dalam daftar, tuple, set dll.

```

1 fruits = ["apple", "banana", "cherry"]
2 for x in fruits:
3     print(x)

```

### 5. Sintak pengkodisian dan contoh penggunaannya kondisi di dalam kondisi pada Python.

#### ▪ Kondisi

Kondisi ini dapat digunakan dalam beberapa cara, paling umum di "jika pernyataan" dan loop.

"Jika pernyataan" ditulis dengan menggunakan kata kunci if.

```

1 fruits = ["apple", "banana", "cherry"]
2 for x in fruits:
3     print(x)

```

#### ▪ Kondisi dalam kondisi

Kata kunci elif adalah cara python mengatakan "jika kondisi sebelumnya tidak benar, maka coba kondisi ini".



```

1 a = 33
2 b = 33
3 if b > a:
4     print("b is greater than a")
5 elif a == b:
6     print("a and b are equal")

```

## 6. Jenis-jenis error dan cara mengatasinya di Python.

- **Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- **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 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 terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

## 7. Cara pemakaian Try Except di Python.

- Berikut ini adalah contoh penggunaan try except.

```

1 try:
2     print("Hello")
3 except:
4     print("Something went wrong")
5 else:
6     print("Nothing went wrong")

```

## 2.17.2 Ketrampilan Pemrograman

### 1. Jawaban Soal 1

```

1 print (1174077%3)
2
3 print("***      *****      ***      *****      *****")
4 print("*****")
5 print("***      *****      ***      ***      ***      ***")
6 print("*****")
7 print("***      ***      ***      ***      ***      ***")
8 print("*****")

```

```

7 print("***  ***      ***      *****  ***  ***      ***
      ***")
8 print("***  ***      ***      ***  ***  ***      ***
      ***")
9 print("***  ***      ***      ***  ***  ***      ***
      ***")
10 print("***  ***      ***      ***  *****  ***
      ***")
11 print("***  ***      ***      ***  *****  ***
      ***")

```

## 2. Jawaban Soal 2

```

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

```

## 3. Jawaban Soal 3

```

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

```

## 4. Jawaban Soal 4

```

1 npm = input("Masukan NPM :")
2 print("Halo, " + str(npm[-3]) + " Apa Kabar?")

```

## 5. Jawaban Soal 5

```

1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 7
7 g = 7
8 npm = [a,b,c,d,e,f,g]
9
10 for a in npm:
11     print(a, end = "")

```

## 6. Jawaban Soal 6

```

1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0

```

```
6 f = 7
7 g = 7
8 print(a+b+c+d+e+f+g)
```

## 7. Jawaban Soal 7

```
1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 7
7 g = 7
8 print(a*b*c*d*e*f*g)
```

## 8. Jawaban Soal 8

```
1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 7
7 g = 7
8 for a in npm:
9     print(a)
```

## 9. Jawaban Soal 9

```
1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 7
7 g = 7
8 for n in npm:
9     if(n % 2 == 0):
10         if(n != 0):
11             print(n, end = "")
```

## 10. Jawaban Soal 10

```
1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 7
7 g = 7
8 for n in npm:
9     if(n % 2 != 0):
10         print(n, end = "")
```

## 11. Jawaban Soal 11

```

1 #soal no 11
2 a = 1
3 b = 1
4 c = 7
5 d = 4
6 e = 0
7 f = 7
8 g = 7
9 print(c)

```

## 2.17.3 Ketrampilan Penanganan Error

### 1. Jawaban Soal No. 1

- Syntax Errors

Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.

- 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 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 terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

### 2. Jawaban Soal No. 2

```

1 a = "5"
2 b = 5
3
4 try:
5     print(a+b)
6 except TypeError:
7     print("Beda tipe data")

```

## 2.18 Kaka Kamaludin

### 1. Aturan Penulisan Variabel

Nama variable bersifat case-sensitive, huruf besar kecil sangat berpengaruh dan nama variable bisa menggunakan ( \_ ) diawal dan ditengah

kalimat. nama variable tidak boleh menggunakan syntax yang ada di python seperti if, else, print dll.

Contoh penamaan variable:

- name12
- name\_variable
- \_variable1
- nameVariab1e
- NameVariable

Contoh sytanx :

```
1 _variableName0 = "value"
```

## 2. Meminta input dari user dan menampilkannya

di python3 menangkap inputan pada keyboard cukup menggunakan fungsi input() untuk mengambil angka atau text. dan untuk outputnya menggunakan fungsi print().

contoh syntax :

```
1 _variableName1 = input("Enter your name:")
2 print("Hello , " + _variableName1)
```

## 3. penggunaan aritmatika dan convert type data

aritmatika sering dipakai dalam pemrograman terutama di python. //

Contoh aritmatika:

```
1 #Addition :
2 x = 5
3 y = 3
4 print(x + y)
5
6 #Subtraction :
7 x = 5
8 y = 3
9 print(x - y)
10
11 #Multiplication :
12 x = 5
13 y = 3
14 print(x * y)
15
16 #Division :
17 x = 5
18 y = 3
```

```

19 print(x / y)
20
21 #Modulus :
22 x = 5
23 y = 3
24 print(x % y)
25
26 #Equal :
27 x = 5
28 y = 3
29 print(x == y)
30
31 ##Non Equal :
32 x = 5
33 y = 3
34 print(x != y)
35
36 ##Greater than :
37 x = 5
38 y = 3
39 print(x > y)
40
41 ##Less than :
42 x = 5
43 y = 3
44 print(x < y)
45
46 ## Greater than or equal to :
47 x = 5
48 y = 3
49 print(x >= y)
50
51 ## Less than or equal to :
52 x = 5
53 y = 3
54 print(x <= y)

```

#### 4. Perulangan

perulangan di python bisa menggunakan "for" dan "while". for dipakai untuk perulangan terhitung, sedangkan while dipakai untuk perulangan tidak terhitung.

```

1 ##while :
2 i = 1
3 while i < 6:
4     print(i)
5     i += 1
6
7 ##for :
8 fruits = ["apple", "banana", "cherry"]
9 for x in fruits:
10    print(x)

```

## 5. kondisi, kondisi didalam kondisi

kondisi di python menggunakan if dan else.

```

1  ##Contoh kondisi didalam kondisi :
2  gaji = 10000000
3  berkeluarga = True
4  punyaRumah = True
5
6  if gaji > 3000000:
7      print ("Gaji sudah diatas UMR")
8      if berkeluarga:
9          print ("Wajib ikutan asuransi dan menabung untuk pensiun")
10     )
11     else:
12         print ("Tidak perlu ikutan asuransi")
13         if punyaRumah:
14             print ("wajib bayar pajak rumah")
15         else:
16             print ("tidak wajib bayar pajak rumah")
17     else:
18         print ("Gaji belum UMR")
19
20 ##Contoh kondisi :
21 umur = 20
22 if umur > 18:
23     print ("Sudah beranjak dewasa")
24 else:
25     print ("Masih dibawah umur")

```

## 6. error yang sering ditemu di python

error yang biasanya di temui ada dalam penulisan syntax. Pengurutan syntax jika menggunakan 'space' gunakan 'space' semua, jangan di acak misal menggunakan 'tab', jika menggunakan 'tab' semua. cara mengatasinya dengan membaca tutorial penggunaan python sesuai versinya

## 7. Try Except

try except merupakan cara untuk mengatasi error selain menggunakan if else. try except mempunyai kelebihan dapat memblok syntax yang error.

```

1  x = 0
2  try:
3      x = 1 / 0
4  except Exception as e:
5      print (e)
6
7  print(x + 1)
8
9  #maka akan muncuk :
10 #division by zero
11 #!

```

## 2.19 Praktek

### 1. jawaban

```

1 print(1174067%3)
2 #mod3 = 1
3
4 print("  ++  ++  ++++++++  +++  ++++++  ++++++
  ++++++++")
5 print("+++++ +++++ ++++++++  ++++++  +++  +++  +++
  ++++++++ ")
6 print("  +++  +++  +++  +++  +++  +++  +++  +++
  ")
7 print("  +++  +++  +++  ++++++++  +++  +++  ++++++++
  ")
8 print("  +++  +++  +++  +++  +++  +++  +++  +++
  ")
9 print("  +++  +++  +++  +++  ++++++  ++++++++  +++
  ")

```

### 2. jawaban

```

1 NPM = input("NPM : ")
2 jumlah = 0
3 while(jumlah <= 67):
4     print("Hallo , " + str(NPM) + " Apa kabar?")
5     jumlah = jumlah +1

```

### 3. jawaban

```

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

```

### 4. jawaban

```

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

```

### 5. jawaban

```

1 a = int(NPM[0])
2 b = int(NPM[1])
3 c = int(NPM[2])
4 d = int(NPM[3])
5 e = int(NPM[4])
6 f = int(NPM[5])
7 g = int(NPM[6])
8 NPM = [a,b,c,d,e,f,g]
9
10 for x in NPM:
11     print(x, end="")
12 print()

```



6. jawaban

```
1 print(a + b + c + d + e + f + g)
```

7. jawaban

```
1 print(a * b * c * d * e * f * g)
```

8. jawaban

```
1 for x in NPM:
2     print(x)
```

9. jawaban

```
1 for x in NPM:
2     if (x % 2 == 0):
3         if (x != 0):
4             print(x, end = "")
5 print()
```

10. jawaban

```
1     if (x % 2 != 0):
2         print(x, end = "")
3
4 print()
```

11. jawaban

## 2.20 Praktek

1. syntax error, penulisan syntax salah, cek kembali syntaxnya. division by zero, tidak membagi angka 0.

```
21 x = "6"
2 y = 5
3
4 try:
5     print(x*y)
6 except TypeError:
7     print("tipe data berbeda")
```

## 2.21 Advent Nopele Olansi Damiahn Sihite

Praktek teori penunjang yang dikerjakan :

### 2.21.1 Variable, pemakaian Variable dan Jenis-Jenis Type data

Variabel merupakan tempat menyimpan data, sedangkan tipe data adalah jenis data yang tersimpan dalam variabel. Variabel bersifat mutable, artinya nilainya bisa berubah-ubah.

#### 1. Pemakaian Variabel

Variabel di python dapat dibuat dengan format seperti ini:

NamaVariabel = (nilai)

Contoh:

VariabelKu = "ini isi variabel"

variabel2 = 20

Kemudian untuk melihat isi variabel, kita dapat menggunakan fungsi print.

print VariabelKu

print variabel2

#### (a) Aturan Penulisan Variabel

- Nama variabel boleh diawali menggunakan huruf atau garis bawah (`_`), contoh: `nama`, `_nama`, `namaKu`, `nama_variabel`.
- Karakter selanjutnya dapat berupa huruf, garis bawah (`_`) atau angka, contoh: `__nama`, `n2`, `nilai1`.
- Karakter pada nama variabel bersifat sensitif (case-sensitif). Artinya huruf besar dan kecil dibedakan. Misalnya, `variabel_Ku` dan `variabel_ku`, keduanya adalah variabel yang berbeda.
- Nama variabel tidak boleh menggunakan kata kunci yang sudah ada dalam python seperti `if`, `while`, `for`, dsb.

#### (b) Tipe Data

Cara mengisi nilai variabel ditentukan dengan jenis datanya, misalkan untuk tipe data teks (string) maka harus diapit dengan tanda petik ("`...`"). Sedangkan untuk angka (integer) dan boolean tidak perlu diapit dengan tanda petik.

#### (c) Jenis-Jenis Tipe Data

- Boolean, Contoh `true` atau `false`
- String, Contoh `"Belajar Python"`
- Integer, Contoh `15` atau `1234`
- Float, Contoh `2.5` atau `0.55`
- List, Contoh `['abcd', 123, 1.5]`

#### 2. Meminta input dan melakukan output

`x = input("masukan nama: ")`

`print('Hallo, ' + x)` #dengan perintah ini, akan menampilkan output

### 3. Operator dan Konvert

- Tambah contoh  $x + y$
- Kurang contoh  $x - y$
- Bagi contoh  $x / y$
- Kali contoh  $x * y$
- Modulus contoh  $x$
- Pangkat  $x ** y$
- equal contoh  $x == y$
- not equal contoh  $x != y$
- lebih besar dari contoh  $x > y$
- kurang dari  $x < y$
- Konvert string ke integer, contoh `x = int("123")`
- Konvert integer ke string, contoh `x = str(456)`

### 4. Perulangan di Python

- Perulangan for  
 contoh :  
`ulang = 2`  
`for i in range(ulang):`  
`print ("Perulangan ke-" +str(i))`  
 Hasil :  
 Perulangan ke-0  
 Perulangan ke-1
- Perulangan While  
 contoh :  
`jawab = 'ya'`  
`hitung = 0`  
`while(jawab == 'ya'):`  
`hitung += 1`  
`jawab = input("Ulang lagi tidak? ")`  
`print ("Total perulangan: " + str(hitung))`

### 5. Kodisi di Python

#### (a) Kondisi **If**

Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar

True. Jika kondisi bernilai salah False maka statement/kondisi if tidak akan di-eksekusi. Dibawah ini adalah contoh penggunaan kondisi if pada Python

```
a = 33
b = 200
if b > a:
    print("b lebih besar dari a")
```

- (b) Kondisi **If Else** Kondisi if else adalah kondisi dimana jika pernyataan benar True maka kode dalam if akan dieksekusi, tetapi jika bernilai salah False maka akan mengeksekusi kode di dalam else. Dibawah ini adalah contoh penggunaan kondisi if else pada Python

```
a = 200
b = 33
if b > a:
    print("b lebih besar dari a")
else:
    print("b bukan lebih besar dari a")
```

- (c) Kondisi **Elif** Pengambilan keputusan (kondisi if elif) merupakan lanjutan/percabangan logika dari "kondisi if". Dengan elif kita bisa membuat kode program yang akan menyeleksi beberapa kemungkinan yang bisa terjadi. Hampir sama dengan kondisi "else", bedanya kondisi "elif" bisa banyak dan tidak hanya satu. Dibawah ini adalah contoh penggunaan kondisi elif pada Python

```
a = 33
b = 33
if b > a:
    print("b lebih besar dari a")
elif a == b:
    print("a sama dengan b")
```

## 6. Error yang sering dialami

- (a) Syntax Error, Cara mengatasinya dengan cara melihat kode dan mengecek kesalahan dalam penulisan.
- (b) Run-time Error, Cara mengatasinya mengecek file pada directory nya, dan memastikan file nya tidak ada yang terhapus.
- (c) Logical Error, Cara mengatasinya mengecek kode secara manual karena error tidak akan ternotice, tetapi akan terasa karena keluaran berbeda dengan yang diharapkan.

## 7. Cara memakai Try Except

Python menyediakan metode penanganan eksepsi dengan menggunakan pernyataan try dan except. Di dalam blok try kita meletakkan baris program yang kemungkinan akan terjadi error. Bila terjadi error, maka penanganannya diserahkan kepada blok except.

contoh :

try:

```
print(x)
```

except:

```
print("terjadi error bre ")
```

## 2.21.2 Ketrampilan Pemrograman

Buat program di python dengan ketentuan :

### 1. Jawaban

```
1 print(" **      **      *****      *****      *****      *****      *****")
2 print("*****      *****      *****      *****      *****      *****      *****")
3 print(" ***      ***      ***      **      **      **      ***      ***      ***      ***")
4 print(" ***      ***      ***      ***      **      **      ***      *****      *****")
5 print(" ***      ***      ***      *****      *****      *****      *****      *****")
6 print(" ***      ***      ***      *****      *****      *****      *****      ***")
7 print(" ***      ***      ***      ***      ***      ***      *****      *****")
8 print(" ***      ***      ***      ***      *****      *****      *****      *****")
```

### 2. Jawaban

```
1 npm = input("Masukkan NPM : ")
2 ulang = 1
3 while(ulang <= 83):
4     print("Halo , "+str(npm)+" apa kabar?")
5     ulang += 1
```

### 3. Jawaban

```
1 npm = input("Masukkan NPM : ")
2 ulang = 1
3 while(ulang <= 83):
4     print("Halo , "+str(npm[4:7])+" apa kabar?")
5     ulang += 1
```

### 4. Jawaban

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

### 5. Jawaban

```
1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 8
7 g = 3
8
9 npm = [a,b,c,d,e,f,g]
10
11 for n in npm:
12     print(n, end = "")
13
14 print()
```

## 6. Jawaban

```
1 print(a+b+c+d+e+f+g)
```

## 7. Jawaban

```
1 print(a*b*c*d*e*f*g)
```

## 8. Jawaban

```
1 for n in npm:
2     print(n)
```

## 9. Jawaban

```
1 for n in npm:
2     if(n % 2 == 0):
3         if(n != 0):
4             print(n, end = "")
```

## 10. Jawaban

```
1 for n in npm:
2     if(n % 2 != 0):
3         print(n, end = "")
```

## 11. Jawaban

```
1 for n in npm:
2     if(n % 2 != 0):
3         print(n, end = "")
4
5 print()
```

## 2.21.3 Ketrampilan Penanganan Error

Bagian Penanganan error dari script python.

### 1. Jawaban

- (a) Syntax Errors, adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- (b) Zero Division Error, adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0). Solusinya adalah tidak membagi suatu yang hasilnya nol.
- (c) Name Error, 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.
- (d) Type Error, adalah exception yang terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

### 2. Jawaban

```
1 a = "7"  
2 b = 7  
3  
4 try:  
5     print(a+b)  
6 except TypeError:  
7     print("Beda tipe data")
```

## 2.22 Advent nopele olansi damiahan sihite

### 2.22.1 Teori

- 1. sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut di kode Python Variable merupakan tempat yang dapat digunakan untuk menyimpan data, dalam python kita bisa membuat variable dengan cara berikut

```
1 #1. Variabel di python dapat dibuat dengan format seperti ini:  
2 x = 21  
3 y = "Advent Nopele Sihite"  
4 print(x)  
5 print(y)
```

- 2. Kode untuk meminta input dari user dan bagaimana melakukan output ke layar seperti pada gambar2

```

1 #2.
2 # Mengambil input
3 nama = input("Siapa nama kamu : ")
4 umur = input("Berapa umur kamu : ")
5
6 # Menampilkan output
7 print ("Hello",nama,"umur kamu adalah",umur,"tahun")

```

3. Operator dasar aritmatika Terdapat penambahan, pengurangan perkalian, perkalian, pembagian, modulus

```

1 #3 Operator dasar aritmatika
2 a = 21
3 b = 7
4 c = "Advent Nopele Sihite"
5 d = "17"
6 # Tambah
7 print (a + b)
8 # Kurang
9 print (a - b)
10 # Kali
11 print (a * b)
12 # Bagi
13 print (a / b)
14 # String ke int
15 print ("Nama :",c, "Dan", "Umur :", int(d))
16 # Int ke String
17 print (str(a) + str(b))

```

4. Perulangan ada dua jenis perulangan di dalam python mereka adalah perulangan while dan perulangan for

```

1 #4.4. Perulangan
2 # For Loops
3 Nama = ["Advent Nopele Sihite"]
4 for x in Nama :
5     print (x)
6
7 # While Loops
8 a = 1
9 while a < 6 :
10     print (a)
11     a += 1
12
13 #5
14 f = 50
15 g = 200
16 if g > f :
17     print ("G Lebih besar dari F")
18     if g == 200:
19         print ("Nilai G adalah 200")

```

5. sintak Untuk memilih kondisi Kondisi IF digunakan ketika ingin menentukan tindakan apa yang harus digunakan sesuai dengan kondisi yang telah diatur



```

1 #7
2 try :
3     print (Hello)
4 except :
5     print ("Ada yang salah")

```

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

```

1 #7
2 try :
3     print (Hello)
4 except :
5     print ("Ada yang salah")

```

## 2.22.2 Praktek

1. Jawaban soal no 1

```

1 """
2 Spyder Editor
3
4 This is a temporary script file.
5 """
6
7 #No.1
8 print (1174089%3)
9 print("***      ***      *****      ***      ***      *****      *****
10      *****")
11 print("***      ***      *****      ***      ***      *****      *****
12      *****")
13 print("***      ***      ***      ***      ***      ***      ***      ***
14      ***")

```

2. Jawaban soal no 2

```

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

```

3. Jawaban soal no 3

```

1 npm = input("Masukan NPM :")
2 hitung = 0
3 while(hitung < 89):
4     print("Halo , " + str(npm) + " Apa Kabar?")
5     hitung = hitung +1

```

## 4. Jawaban soal no 4

```

1 npm = input("Masukan NPM :")
2 hitung = 0
3 while(hitung < 17):

```

## 5. Jawaban soal no 5

```

1
2 #No.4
3 npm = input("Masukan NPM :")
4 print("Halo, " + str(npm[-3]) + " Apa Kabar?")
5
6 #No.5
7 a = 1
8 b = 1
9 c = 7
10 d = 4
11 e = 0
12 f = 2

```

## 6. Jawaban soal no 6

```

1 h = a+b+c+d+e+f+g
2 i = a*b*c*d*e*f*g

```

## 7. Jawaban soal no 7

```

1 print(a,b,c,d,e,f,g)

```

## 8. Jawaban soal no 8

```

1 print(h)
2
3 #No.7
4 print(i)
5
6 #No.8
7 print(a)
8 print(b)

```

## 9. Jawaban soal no 9

```

1 print(d)
2 print(e)

```

## 10. Jawaban soal no 10

```

1 print(g)

```

## 11. Jawaban soal no 11

```

1 print(d,e,g)

```

## 2.22.3 Keterampilan dan penanganan error

```

1 a = "10"
2 b = 5
3
4 try :
5     print(a+b)
6 except TypeError:
7     print("Beda Tipe Data")

```

## 2.23 Ainul Filiaini

1. Jenis-jenis variable dan cara pemakaian variable tersebut di kode python.

1. Variable Gabungan

menggabungkan tipe data yang berbeda seperti string dan angka secara bersamaan. Sebagai contoh, kita akan menggabungkan "Guru" dengan angka "99".

Tidak seperti java yang menggabungkan angka dengan string tanpa mendeklarasi angka sebagai string, python mengharuskan pendeklarasian angka tersebut sebagai string, jika tidak ia akan menampilkan type error.

Contoh :

a = "Guru"

b = 99

cetak a + b

Setelah variable dideklarasikan sebagai string, ia dapat menggabungkan kedua "Guru" + str("99") = "Guru99" di output.

a = "Guru"

b = 99

print(a + str(b))

2. Variable Local dan Global

Dalam python ketika kita ingin menggunakan variable yang sama untuk sisa program atau modul kita, kita mendeklarasikannya sebagai variable global, apabila jika kita ingin menggunakan variable dalam fungsi atau metode tertentu, kita menggunakan variable lokal

perbedaan antara variable lokal dengan variable global :

1. variable "f" adalah lingkup global dan diberi nilai 101 yang di cetak dalam outputnya.
2. Variable f lagi dinyatakan dalam fungsi dan mengasumsi ruang lokal. ini diberikan nilai "saya sedang belajar python." yang di cetak sebagai outputnya. Variable global yang didefinisikan "f" sebelumnya.
3. setelah pemanggilan fungsi selesai, variable lokal f di hancurkan. pada baris 12, ketika kita kembali mencetak nilai

”f” apakah ini menampilkan nilai variable global f = 101

Contoh :

```
f = 101;
print(f)
# Variable global vs local in functions
def someFunction ()
global f
print(f)
f = "changing global variable"
someFunction()
print(f)
```

Dengan menggunakan kata kunci global, Anda dapat referensi variable global didalam satu fungsi.

1. Variable ”f” adalah lingkup global dan diberi nilai 101 yang dicetak dalam output
2. Variable f dinyatakan menggunakan kata kunci global. Ini bukan bukan variable local, tapi variable global yang sama dinyatakan sebelumnya. krenanya ketika mencetak nilainya, ouputnya adalah 101
3. Kami mengubah nilai ”f” didalam fungsi. setelah pemanggilan fungsi selesai, nilai yang diubah dari variable ”f” tetap ad Pada baris 12, ketika kita kembali mencetak nilai ”f” apakah ini menampilkan nilai”mengubah variable global”

### 3. Hapus Sebuah variable

Kita juga bisa menghapus sebuah variable dengan perintah del” nama variable”.

2. kode input dari user  
nama\_variable = input (“Sebuah Teks”)  
artinya teks yang kita inputkan dari keyboard akan disimpan kedalam nama\_variable.

contoh :

```
# mengambil input
nama = raw_input(”siapa nama kamu : ”)
# menampilkan output
print ”Hello”,nama,”umur kamu adalah”,umur, ”tahun”
```

Dan Hasilnya :

Siapa naman kamu :

Berapa umur kamu :

Cara menampilkan Output:

Untuk menampilkan output teks , kita menggunakan fungsi print().

contoh :

```
print ” Hello World!”
```

```
print nama_variable  
print "Gabung dengan", variable
```

Menampilkan Variable dan Teks :

```
nama = "PetaniKode"  
print "Hello", nama
```

Hasilnya :

```
Hello PetaniKode  
Antara kata hello
```

#### 4. Operator dasar aritmatika

Tambah (+)

```
x = 4  
y = 5  
print (x + y)  
Jawabannya : 9
```

Kurang(-)

```
x = 4  
y = 2  
Print (x - y)  
Jawabannya : 2
```

Kali(x)

```
x = 4  
y = 2  
print (x * y)  
Jawabannya : 8
```

Bagi(/)

```
x = 8  
y = 2  
print (x / y)  
Jawabannya : 4
```

```
Public class String ToInteger  
Public static void main(String[]args)  
System.out.println("Mengubah String ke Integer");
```

```
String str = "1234";  
int a = 5;  
//mengubah variable str menjadi int  
int num = Integer.parseInt(str);
```

```
int b = num + a;  
System.out.println(b);
```

Sebaliknya, untuk mengubah Integer ke String menggunakan fungsi berikut:

```
/*
To change this template, choose Tools— Templates
/
package Latihan StringInt;
/**

/
public class IntegerToString
Public static void main(String[] args)
int a = 1000;
String b = String.valueOf(a);
```

#### 5. cara pakai sintak untuk memilih kondisi

Ada tiga kondisi di python yang dapat anda gunakan untuk membangun alur logika untuk program anda. Python memiliki statement if, if..else, dan if..elif.. else. berikut adalah contoh penggunaan if di python;

```
umur = 20
if umur /> 18:
print "sudah beranjak dewasa"
```

Hasil Eksekusi di konsol:

```
$ python if.py
sudah beranjak dewasa
```

untuk memeriksa kondisi yang tidak memenuhi kondisi utama. maka else digunakan untuk menangani semua kondisi selain kondisi yang telah dituliskan. Berikut adalah contoh penggunaan else di dalam kondisi Python :

```
umur = 20
if umur /> 18:
print "Sudah beranjak dewasa"
else:
```

```
printl "Masih dibawah umur"
```

hasil eksekusi di konsol:

```
$ python if-else.py
Sudah beranjak dewasa
```

Bila kondisi yang akan didefinisikan cukup banyak, anda menambahkan kondisi lain dengan menggunakan elif di bawah statement

```

    if dan sebelum statement else:
umur = 37
if umur < 18 and umur > 30:
print "Sudah beranjak dewasa"
elif umur < 30 and umur > 45:
print "Masa - masa emas"
elif umur < 45 and umur > 55:
print "Memasuki masa paruh baya"
elif umur < 55:
print "Masa - masa manula"
else:
print "Masih dibawah umur"

```

Hasil eksekusi konsol :

```

$python if-elif.py
masa - masa emas

```

dan tentu saja, sebuah kondisional dapat disimpan di dalam if lain. berikut ini adalah contoh kode if bersarang di python :

```

gaji = 10000000
berkeluarga = True
punya_rumah = True

```

```

    if gaji < 3000000:
print "Gaji sudah diatas UMR"
if berkeluarga:
print "Wajib ikutan asuransi dan menabung untuk pensiun"
else
print "Tidak perlu ikutan asuransi"

```

```

    if punya_rumah:
print "wajib bayar pajak rumah"
else:
print "tidak wajib bayar pajak rumah"
else:
print "Gaji belum UMR"

```

Hasil eksekusi konsol :

```

$ python nested-if.py
Gaji sudah diatas UMR
Wajib ikutan asuransi dan menabung untuk pensiun
wajib bayar pajak rumah

```

6. jenis error di python :

Tidak mengenali variable :

```

$$$ d = [1, 2, 3, 4]
$$$ d[5]

```

Traceback (most recent call last):

File "stdin", line 1, in <module>  
 IndexError: list index out of range

'''

### 1. kesalahan syntax

Kesalahan ini biasanya disebabkan oleh cara mengganti program yang tidak dapat diakses kaidah menerima kode yang benar-benar sesuai dengan bahasa pemrograman yang digunakan atau memang kurang atribut. Contoh di pythor

### 2. Kesalahan Eksekusi

Kesalahan ini menyebabkan eksekusi dieksekusi. Contoh dari kesalahan ini diterbitkan adalah memasukan data yang

salah (bukan bilangan) pada fungsi input(), atau contoh lain yang disediakan indeks di luar daftar indeks daftar, atau membuka pernyataan dekorasi yang tidak ada di dalam disk kita.

### 3. Kesalahan Logika

Kesalahan ini diterapkan saat skrip sedang dalam eksekusi dieksekusi tetapi tidak dilakukan berhentinya eksekusi skrip tersebut. Kesalahan seperti ini pada umumnya karena logika dalam naskah yang salah. Programmer misalkan yang menerjemahkan kode bilangan Phi 3.14 tetapi pada kenyataanya dia menerima 31 sehingga skrip yang diperlukan mampu menghitung luas dengan benar menghasilkan angka G + sar dan salah.

### 7. Try Except

Salah satu bentuk penanganan error di Python adalah dengan menggunakan statement try..except. Mungkin kita pernah mendeteksi error dengan memanfaatkan kondisional biasa menggunakan if..else, namun hal tersebut akan lebih praktis ditangani bila dengan menggunakan try..except. Anda dapat mengurung suatu blok kode dengan try..except untuk menangani error yang mungkin kita sendiri tidak mengetahuinya. Biasanya try..except ini digunakan untuk menangani error saat penggunaan IO, operasi database, atau pengaksesan indeks suatu list atau dictionary, dan berbagai kasus lainnya.

contoh :

```
try: print("lins")
except :
print("lins kamu pintar")
finaly:
    print("memang pintar")
```

## 2.23.1 Keterampilan pemrograman

### 1. jawaban soal 1

```
1 print(1174073%3)
```



```

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

```

## 2. jawaban soal 2

```

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

```

## 3. jawaban soal 3

```

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

```

## 4. jawaban soal 4

```

1 npm = input("Masukan NPM :")
2 print("Halo, " + str(npm[-3]) + " Apa Kabar?")

```

## 5. jawaban soal 5

```

1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 7
7 g = 3

```

```
8 h = a+b+c+d+e+f+g
9 i = a*b*c*d*e*f*g
10 print(a,b,c,d,e,f,g)
```

## 6. jawab an soal 6

```
1 print(h)
```

## 7. jawaban soal 7

```
1 print(i)
```

## 8. jawaban soal 8

```
1 print(a)
2 print(b)
3 print(c)
4 print(d)
5 print(e)
6 print(f)
7 print(g)
```

## 9. jawaban soal 9

```
1 a = 1
2 b = 1
3 c = 7
4 d = 3
5 e = 0
6 f = 7
7 g = 3
8 for n in npm:
9     if(n % 2 == 0):
10         if(n != 0):
11             print(n, end = "")
```

## 10. jawaban soal 10

```
1 a = 1
2 b = 1
3 c = 7
4 d = 3
5 e = 0
6 f = 7
7 g = 3
8 for n in npm:
9     if(n % 2 != 0):
10         print(n, end = "")
```

## 11. jawaban soal 11

```
1 a = 1
```

## 2.24 Sekar Jasmine

1. Jenis-Jenis variabel dan cara pemakaian variabel tersebut di kode python

1. Variabel Gabungan

menggabungkan tipe data yang berbeda seperti string dan angka secara bersamaan

. contoh , kita menggabungkan antara "guru" dengan angka "200".

tidak seperti java yang menggabungkan angka dengan string tanpa mendeklarasi angka sebagai string , python mengharuskan pendeklarasikan angka tersebut sebagai string , jika tidak ia akan menampilkan type error

Contoh :

```
a = "guru"
```

```
b = "200"
```

```
cetak a+b
```

setelah variabel dideklarasikan sebagai string , ia dapat menggabungkan kedua "siswa"

```
+ str("200") = "siswa200" di output
```

```
a = "siswa"
```

```
b = 200
```

```
print(a+ str(b))
```

2. Beberapa fitur yang dimiliki Python

a. memiliki aturan layout kode sumber yang memudahkan pengecekan, pembacaan kembali dan penulisan ulang kode sumber. berorientasi obyek. b. memiliki sistem pengelolaan memori otomatis (garbage collection, seperti java), mudah dikembangkan dengan menciptakan modul-modul baru; modul-modul tersebut dapat dibangun dengan bahasa Python maupun C/C++.

3. Tipe Data

A. Number Tipe data Number merepresentasikan nilai-nilai berupa angka. Python menggolongkan beberapa tipe data umum seperti, Integer (bilangan bulat) dan Floating-point (bilangan desimal) ke dalam tipe data Number. Contoh :

```
a = "siswa"
```

```
b = 200;
```

```
print(g)
```

B. String

Selain angka, python juga mampu melakukan manipulasi string, yang dapat diekspresikan dengan beberapa cara. Penulisan nilai string pada python menggunakan tanda petik satu ( ' ) atau tanda petik dua ( " ). Contoh :

```
'''Selamat Datang Siswa Baru''' "Selamat Datang Siswa Baru" '''selamat datang siswa baru''' "selamat datang siswa baru"
```

Untuk menggabungkan dua buah string atau lebih dapat dilakukan dengan dua cara. Pertama, dengan menulis langsung dua buah string yang diapit dengan tanda kutip atau dengan penggunaan operator (+) Contoh :

'''Menggenal''' 'python' 'Menggenalpython' '''institusi' + 'pertanian bogor'  
'institutipertanianbogor'

### C. Dictionary

dictionary memakai key untuk merujuk pada isi variabelnya. Sifat kedua jenis data ini hanya berbeda dalam beberapa hal saja. Untuk mendeklarasikan sebuah dictionary, Python memakai tanda Contoh :

'''D = 'drink': 'spam', 'time': 8, 'color': 'purple' '''D 'drink': 'spam', 'color': 'purple', 'time': 8 '''D['color'] 'purple'

4. Operator dalam Python dibagi menjadi 3 bagian, yaitu :

```

1  A. operator aritmatika : +, - , *, / , %
2      >>> x = 7 + 3 * 6
3      >>> x
4      25
5      >>> y = 200/6*5
6      >>> y
7      166
8
9
10     >>> x = ( 8 + 7 ) * 9
11     >>> x
12     71
13     >>> y = 300 / ( 5 * 4 )
14     >>> y
15     240
16
17 Menggunakan operator %
18     >>> 8 % 6
19     1
20     >>> 9 % 7
21     1
22
23 B. operator perbandingan : >= , <= , != , > , < , ==
24     Contoh :\\
25     a = input ( "Inputkan nilai a: " )
26     b = input ( "Inputkan nilai b: " )
27
28     #apakah a sama dengan b?
29     c = a == b
30     print "apakah %d == %d: :r" % (a,b,c)
31
32     #apakah a < b ?
33     c = a < b
34     print "apakah %d < %d: %r" % (a,b,c)
35
36     #apakah a > b ?
37     c = a > b
38     print "apakah %d > %d: %r" % (a,b,c)
39
40     #apakah a <= b ?
41     c
42     <= b ?
43     print "apakah %d <= %d: %r" (a,b,c)
44
45     #apakah a >= b ?

```

```

46     c = a >= b
47     print "apakah %d >= %d: %r" (a,b,c)
48 C. o= a perator penugasan : *= , /= , %= , +=, -=
49     a = input ("Inputkan nilai a: ")
50     print "Nilai a = %d" % a
51
52     # Coba kita jumlahkan nilai a dengan operator penugasan
53     a += 5
54
55     # setelah nilai a ditambah 5 , coba kita lihat isinya
56     print "Nilai setelah ditambah 5:"
57     print "a = %d" % a

```

### 2.24.1 Keterampilan Pemrograman

1. jawab soal 1
2. jawab soal 2
3. jawab soal 3
4. jawab soal 4
5. jawab soal 5
6. jawab soal 6
7. jawab soal 7
8. jawab soal 8
9. jawab soal 9
10. jawab soal 10
11. jawab soal 11

## 2.25 Dini Permata Putri

### 2.25.1 sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut dikode python

proses memasukan data ke dalam variabel nama = "John Doe" proses mencetak variabel print(name)

nilai dan tipe data dalam variabel dapat diubah umur = 20 nilai awal print(umur) mencetak nilai umur type(umur) mengecek tipe data umur umur = "dua puluh satu" nilai setelah diubah print(umur) mencetak nilai umur type(umur) mengecek tipe data umur

```

1 namaDepan = "Budi"
2 namaBelakang = "Susanto"
3 nama = namaDepan + " " + namaBelakang
4 umur = 19
5 hobi = "Nyanyi"
6 print("Biodata\n", nama, "\n", umur, "\n", hobi)
7
8 contoh variabel lainnya
9 inivariabel = "Halo"
10 ini_juga_variabel = "Hai"
11 _inivariabeljuga = "Hi"
12 inivariabel222 = "bye"
13
14 panjang = 8
15 lebar = 4
16 luas = panjang * lebar
17 print(luas)
18
19 Membuat variabel di python
20 nama_variabel = <nilai>
21 contoh :
22 variabel_ku = "ini isi variabel"
23 variabel2 = 20
24 print variabel_ku
25 print variabel2
26
27 Tipe data
28 contoh :
29 nama_mu = "Petani Kode"
30 uumur =20
31 tinggi = 183.22
32 contoh :
33 >>> usia = 19
34 >>> type(usia)
35 <type 'int'>
36 >>> usia = "19"
37 >>> type(usia)
38 <type 'str'>
39 >>> usia = '19'
40 >>> type (usia)
41 <type 'str'>
42 >>> usia = 21.5
43 >>> type(usia)
44 <type 'float'>
45 >>>usia = true
46 Traceback (most recent cal last)
47 File "<stdin>", line 1, in <module>
48 NameError: name 'true' is not defined
49 >>> usia = true
50 >>> type(usia)
51 <type 'bool'>

```

## 2.25.2 tuliskan bagaimana kode untuk meminta input dari user dan tuliskan bagaimana melakukan output ke layar

```

1 cara mengambil input dari keyboard
2 nama_variabel = input("Sebuah Teks")
3
4 #mengambil input
5 nama + raw_input(Siapa nama kamu: ")
6
7 #menampilkan output
8 print "Hello", nama, "umur kamu adalah",umur,"tahun"
9
10 cara menampilkan output
11 print "Hello World!"
12 print nama_variabel
13 print "Gabung dengan", variabel
14
15 menampilkan variabel dan teks
16 nama = "Petanikode"
17 print "Hello", nama
18
19 menggunakan string formatting cara lama
20 menggunakan simbol %
21 nama = raw_input("inputkan nama: ")
22 print "Selamat datang %s" % nama
23 tanda % akan otomatis diganti dengan nilai yang kita inputkan ke
    variabel nama.
24 nama = raw_input("Inputkan nama: ")
25 umur = input("inputkan umur: ")
26 tinggi = input(inputkan tinggi badan: ")
27 print "Hello %5, saat ini usiamu %d tahun dan tinggi badanmu %f cm" %
    (nama, umur, tinggi)

```

## 2.25.3 tuliskan operator dasar aritmatika, tambah, kali, kurang bagi, dan bagaimana mengubah string ke integer dan integer ke string

```

1 #penjumlahan
2 print(13-2)
3 apel=7
4 jeruk=9
5 buah=apel+jeruk
6 print(buah)
7
8 pengurangan
9 hutang=8000
10 bayar=4000
11 sisahutang=hutang-bayar
12 print(*sisahutang anda adalah", sisahutang)
13
14 perkalian
15 panjang=14
16 lebar=7
17 luas=panjang*lebar
18 print(luas)
19

```

```

20 pembagian
21 kue=15
22 anak=3
23 kueperanak=kue / anak
24 print("setiap anak akan mendapatkan bagian kue sebanyak",kueperanak)
25
26 sisa bagi
27 bilangan1=12
28 bilangan2=4
29 hasil=bilangan1%bilangan2
30 print("sisa bagi dari bilangan",bilangan1,"dan",bilangan2,"adalah",..
    hasil)
31
32 cara mengubah integer ke integer
33 integer merupakan tipe data data bilangan bulat.
34 a=100#variabel/angka yang akan di konversi
35 string=str(a)konversi integer ke string
36 print(string)
37
38 cara mengubah string ke integer
39 string merupakan tipe data yang menampung sebuah teks.
40 a='1212'variabel/angka yang akan di konversi.
41 integer==int(a)konversi string ke integer
42 print(integer)

```

#### 2.25.4 tuliskan dan jelaskan sintak untuk perulangan, jenis-jenisnya contoh kode dan cara pakainya di python

```

1 -statement (pernyataan) di python
2 statement penugasan, if, for dsb.
3 if nilai <=5:
4     print("Nilai merah")
5     print(Tidak lulus")
6 else:
7     print("nilai biru")
8     print("lulus")
9
10 -tanda kutip di python
11 kata='kata'
12 kalimat="ini adalah kalimat"
13 paragraf = """ini adalah paragraf.
14 paragraf terdiri dari beberapa baris."""
15
16 -komentar di python
17 #komentar pertama
18 print("Hello World") # komentar kedua

```

#### 2.25.5 tuliskan dan jelaskan cara pakai sintak untuk memilih kondisi, dan bagaimana contoh sintak kondisi didalam kondisi

contoh penggunaan kondisi if di python :

```

1 umur = 19
2 if umur > 17:
3     print "sudah beranjak dewasa"

```



```

4 hasil eksekusi :
5 $ python if.py
6 sudah beranjak dewasa
7 contoh penggunaan kondisi else di python:
8 umur = 19
9 if umur > 17:
10     print "sudah beranjak dewasa"
11 else:
12     print "masih dibawah umur"
13 hasil eksekusi :
14 $ python if-else.py
15 sudah beranjak dewasa

```

### 2.25.6 tuliskan apa saja jenis error yang sering ditemui di python dalam mengerjakan sintak diatas dan bagaimana cara mengatasinya

```

1 if true:
2     print ("Answer")
3     print ("True")
4 else:
5     print("Answer")
6     print("False")
7 kode diatas error, karena print ("false") dan print ("answer")
8     seharusnya pada blok yang sama seper
9
10 contoh berikut :
11 if true:
12     print ("Answer")
13     print ("True")
14
15 else:
16     print ("Answer")
17     print ("False")
18 ti

```

### 2.25.7 tuliskan dan jelaskan cara memakai Try Except

```

1 cara menangani error pembagian nol =:
2 x 0
3 try:
4     x = 1/0
5 except Exception,
6 :
7     print e
8 print x+1
9 jika kode diatas dieksekusi, maka akan muncul error seperti berikut
10 ini :
11 $ python demo-1.py
12 integer division or modulo by zero
13 1
14 cara menangani error pembacaann file :
15 orang = {"nama": "bibb", "kota": "jepara", "umur": "20"}

```

```
16
17 try:
18     contact = open("contact.txt", 'r')
19     print orang["pekerjaan"]
20 except IOError, e:
21     print "terjadi erorr IO: ",e
22 except keyrror, e:
23     print "terjadi kesalahan pada akses list/dict/tuple:",e
24 print orang
25 jika kode diatas dieksekusi, maka akan muncul output seperti berikut:
26 $ python demo-2.py
27
28 terjadi error IO: [errno 2] No such file or directory: '/home/contact
29     .txt
30 {'nama': 'syuaib', 'umur',: '19', 'kota': 'jepara'}
```



## BAB 3

---

# FUNGSI DAN KELAS

---

### 3.1 D. Irga B. Naufal Fakhri

#### 3.1.1 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 3.1** 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 3.2** 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_1174066
3
4 input( fungsi_1174066.hello("Dirga"))

```

**Listing 3.3** 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 3.4** Contoh import package atau library.

## 4. Cara memanggil library dari instansiasi

Cara memanggilnya:

- Pertama kita import filenya
- kemudian buat variabelnya 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 3.5** 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 3.6** 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 3.7** 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 1174066.fungsi_1174066 import Tambah
3
4 x = 3
5 y = 7
6 print(Tambah(x,y))

```

**Listing 3.8** 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 1174066.fungsi_1174066 import Tambah
3
4 x = 3

```

```

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

```

**Listing 3.9** Contoh import package atau library didalam folder.

### 3.1.2 Keterampilan Pemograman

#### 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="U1")[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("Halo, "+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()

```

### 3.1.3 Ketrampilan Penanganan Error

- 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 variabelnya 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()

```

## 3.2 Nurul Izza Hamka | 1174062 | Teori

1. Fungsi pada python menggunakan kata kunci def. setelah menulis kata def, kita tulis lagi nama fungsi kemudian diikuti dengan parameter yang diberi tanda

kurung dan diakhirnya diberi tanda titi dua (:). Setelah itu kita menulis lagi fungsi yang akan di panggil untuk di jalankan. Inputan fungsi adalah untuk memanggil fungsi dengan menuliskan nama fungsi. Kembalian fungsi adalah keluaran fungsi dan untuk kembali ke baris selanjutnya untuk memanggil fungsi tadi.

```
1 def Izzah() :
2     print ("Apa Kabar Izzah")
3 Izzah()
```

2. Paket adalah sebuah file yang berisi kode program python yang bisa digunakan berulang ketika sebuah paket itu dipanggil.

```
1 #from kelas3lib import *
2 #print(penulisan(int(input("Masukan NPM kamu : "))))
```

3. Apa itu kelas, apa itu objek, apa itu atribut, apa itu method :

- Kelas adalah sebuah objek yang di dalamnya terdapat sebuah metode atau seperangkat atribut.
- Objek adalah struktur data yang di definisikan dalam kelas, objek ini memiliki atribut dan juga aksi /behaviour.
- Atribut adalah data dari variable kelas dan juga method.
- Method adalah sebuah kode yang di gunakan untuk melakukan perintah.

```
1 #class Employee:
2 #     'Common base class for all employees'
3 #     empCount = 0
4
5 #     def __init__(self, name, salary):
6 #         self.name = name
7 #         self.salary = salary
8 #         Employee.empCount += 1
9
10 #     def displayCount(self):
11 #         print ("Total Employee %d" % Employee.empCount)
12
13 #     def displayEmployee(self):
14 #         print ("Name : ", self.name, ", Salary : ", self.salary)
15
16
17 #This would create first object of Employee class"
18 #emp1 = Employee("Izza", 2000)
19 #This would create second object of Employee class"
20 #emp2 = Employee("Manni", 5000)
21 #emp1.displayEmployee()
22 #emp2.displayEmployee()
23 #print ("Total Employee %d" % Employee.empCount)
```

#### 4. Pemanggilan library kelas dari instansiasi

- Pertama, lakukan import file.
- Membuat variable untuk menyimpan data.
- Lakukan pemanggilan data dan class.
- Ketikkan print untuk melihat hasilnya.

```

1 #import belajar
2 #a = 10
3 #b = 5
4
5 #c = belajar.penambahan(a,b)
6 #print(c)

```

#### 5. Penggunaan paket from kalkulator import untuk penambahan berfungsi untuk memanggil file yang di masukkan dan juga fungsinya.

```

1 #import belajar
2 #a = 10
3 #b = 5
4
5 #c = belajar.penambahan(a,b)
6 #print(c)

```

#### 6. Pemakaian paket fungsi adalah sebuah kumpulan fungsi-fungsi.

#### 7. Pemakaian paket kelas apabila file library ada di dalam folder.

```

1 # Employee.empCount += 1

```

#### 8. No1

```

1 #No 1
2 def penulisan(npm):
3     npm = list(str(npm))
4
5     angka1 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
##### ", "4": " ###   ## ", "5": " ##### ", "6": " ##### ", "7": "
##### ", "8": " ##### ", "9": " ##### "}
6     angka2 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
##### ", "4": " ###   ## ", "5": " ##### ", "6": " ##### ", "7": "
##### ", "8": " ##### ", "9": " ##### "}
7     angka3 = {"0": " ##   ## ", "1": " ### ", "2": "   ## ", "3": "
##### ", "4": " ###   ## ", "5": " ##### ", "6": "   ##   ", "7": "
   ##   ", "8": " ##### ", "9": " ##### "}
8     angka4 = {"0": " ##   ## ", "1": " ### ", "2": "   ## ", "3": "
##### ", "4": " ###   ## ", "5": " ##### ", "6": "   ##   ", "7": "
   ##   ", "8": " ##### ", "9": " ##### "}

```

```

9     angka5 = {"0": " ##      ## ", "1": " ### ", "2": " ##### ", "3": "
    ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
    ##### ", "8": " ##### ", "9": " ##### "}
10    angka6 = {"0": " ##      ## ", "1": " ### ", "2": " ##### ", "3": "
    ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
    ##### ", "8": " ##### ", "9": " ##### "}
11    angka7 = {"0": " ##      ## ", "1": " ### ", "2": " ##### ", "3": "
    ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
    ##### ", "8": " ##### ", "9": " ##### "}
12    angka8 = {"0": " ##      ## ", "1": " ### ", "2": " ##### ", "3": "
    ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
    ##### ", "8": " ##### ", "9": " ##### "}
13    angka9 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
    ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
    ##### ", "8": " ##### ", "9": " ##### "}
14    angka10 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
    ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
    ##### ", "8": " ##### ", "9": " ##### "}

15
16    hasil1 = []
17    hasil2 = []
18    hasil3 = []
19    hasil4 = []
20    hasil5 = []
21    hasil6 = []
22    hasil7 = []
23    hasil8 = []
24    hasil9 = []
25    hasil10 = []
26
27    for x in npm:
28        hasil1.append(angka1[x])
29        hasil2.append(angka2[x])
30        hasil3.append(angka3[x])
31        hasil4.append(angka4[x])
32        hasil5.append(angka5[x])
33        hasil6.append(angka6[x])
34        hasil7.append(angka7[x])
35        hasil8.append(angka8[x])
36        hasil9.append(angka9[x])
37        hasil10.append(angka10[x])
38
39    print(*hasil1, sep=' ')
40    print(*hasil2, sep=' ')
41    print(*hasil3, sep=' ')
42    print(*hasil4, sep=' ')
43    print(*hasil5, sep=' ')
44    print(*hasil6, sep=' ')
45    print(*hasil7, sep=' ')
46    print(*hasil8, sep=' ')
47    print(*hasil9, sep=' ')
48    print(*hasil10, sep=' ')

```

## 9. no2

```

2 def perulangan(npm):
3     hitung = 0
4     while(hitung < 62):
5         print("Halo, "+str(npm)+" apa kabar?")
6         hitung = hitung +1
7
8 perulangan(int(input("Masukan NPM : ")))

```

## 10. no3

```

1 #No 3
2 def perulangan_3_digit(npm):
3     hitung = 0
4     npm = str(npm)
5     bil = npm[4:7]
6
7     while(hitung < 8):
8         print("Halo, "+bil+" apa kabar?")
9         hitung = hitung +1
10
11 perulangan_3_digit(int(input("Masukan NPM : ")))

```

## 11. no4

```

1 #No 4
2 def perulangan_3_digit_terakhir(npm):
3     npm = str(npm)
4     bil = npm[-3]
5     print("Halo, "+bil+" apa kabar?")
6
7 perulangan_3_digit_terakhir(int(input("Masukan NPM : ")))

```

## item no5

```

1 #No 5
2 def down(npm):
3     for i in npm:
4         print (i)
5
6 down(input("Masukan NPM : "))

```

## 12. no6

```

1 #No 6
2 def penjumlahan(npm):
3     jumlah = 0
4     for i in npm:
5         jumlah += int(i)
6     print(str(jumlah)+" Adalah hasil penjumlahan")
7
8 penjumlahan(input("Masukan NPM : "))

```

## 13. no7

```

1 #No 7
2 def perkalian(npm):
3     jumlah = 0
4     for i in npm:
5         jumlah += int(i)
6     print(str(jumlah)+" Adalah hasil perkalian")
7
8 perkalian(input("Masukan NPM : "))

```

#### 14. no8

```

1 #No 8
2 def genap():
3     npm = [1,1,7,4,0,2,7]
4     for i in npm:
5         if (i % 2) == 0:
6             print("Bilangan Genapnya : "+str(i))
7 genap()

```

#### 15. no9

```

1 #No 9
2 def ganjil():
3     npm = [1,1,7,4,0,2,7]
4     for i in npm:
5         if (i%2)==1:
6             print("Bilangan Ganjilnya : "+str(i))
7 ganjil()

```

#### 16. no10

```

1 #No 10
2 def prima(npm):
3     npm = str(npm)
4     bil = npm[2]
5     num = int(bil)
6     if num > 1:
7         for i in range(2,num):
8             if (num%i)==0:
9                 print("Bukan Bilangan Prima")
10                break
11            else:
12                print("Bilangan Primanya : "+str(num))
13    else:
14        print("Tidak Ada Bilangan Prima")
15    prima(int(input("Masukan NPM : ")))

```

### 3.3 Fanny Shafira Damayanti | 1174069

#### 3.3.1 Teori

1. fungsi adalah sebuah program untuk melakukan tugas tertentu secara berulang. Fungsi di dalam Python di tandai dengan def, yang artinya definition. Inputan

fungsi berarti memanggil fungsi yang ditulis oleh user dan dikembalikannya dalam bentuk string.

```
1 def nyapa():
2     print ("anyeong Fanny")
3
4 nyapa()
```

2. paket adalah kumpulan fungsi yang siap untuk di pakai. Paket digunakan untuk memudahkan programmer agar tidak menuliskan kembali kode-kodenya.

```
1 #from fungsi_fanny import *
2 #print(penulisan(int(input("Masukan NPM kamu :"))))
```

3. objek adalah instansi dari kelas. Kelas adalah cetakan biru yang berisikan variable dan method. Method adalah fungsi dari suatu objek. Atribut adalah nilai dari suatu objek.

```
1 #class Employee:
2 #     'Common base class for all employees'
3 #     empCount = 0
4
5 #     def __init__(self, name, salary):
6 #         self.name = name
7 #         self.salary = salary
8 #         Employee.empCount += 1
9
10 #     def displayCount(self):
11 #         print ("Total Employee %d" % Employee.empCount)
12
13 #     def displayEmployee(self):
14 #         print ("Name : ", self.name, ", Salary : ", self.salary)
15
16
17 #This would create first object of Employee class"
18 #emp1 = Employee("Fanny", 2000)
19 #This would create second object of Employee class"
20 #emp2 = Employee("Shafira", 5000)
21 #emp1.displayEmployee()
22 #emp2.displayEmployee()
23 #print ("Total Employee %d" % Employee.empCount)
```

4. cara pemanggilan library yaitu :

- Import file yang akan di panggil
- Buat variable nya
- Panggil nama class dan methodnya
- Print untuk menampilkan outputnya

```
1 #import latihan
2 #a = 30
3 #b = 10
```



5. import berfungsi untuk memanggil fungsi dari kelas lain.

```
1 # 'Common base class for all employees'
2 # empCount = 0
```

6. pemakaian paket fungsi apabila ke library ada di dalam folder

```
1 from fungsi_fanny import penulisan
```

7. pemakaian paket kelas apabila ke library ada di dalam folder

```
1 #from 3lib_1174069 import penulisan
```

### 3.3.2 keterampilan pemrograman

1. No 1

```
1 def penulisan(npm):
2     npm = list(str(npm))
3
4     angka1 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
5     angka2 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
6     angka3 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##   ## ", "7": "##### ", "8": "##### ", "9": "##   ## "}
7     angka4 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##   ## ", "7": "##### ", "8": "##### ", "9": "##   ## "}
8     angka5 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##   ## ", "7": "##### ", "8": "##### ", "9": "##   ## "}
9     angka6 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
10    angka7 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##   ## ", "7": "##### ", "8": "##### ", "9": "##### "}
11    angka8 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
12    angka9 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##   ## ", "7": "##### ", "8": "##### ", "9": "##### "}
13    angka10 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
14
15    hasil1 = []
16    hasil2 = []
17    hasil3 = []
18    hasil4 = []
```

```

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

```

## 2. N0 2

```

1 def perulangan(npm):
2     hitung = 0
3     while(hitung < 69):
4         print("Halo, "+str(npm)+" anyeong?")
5         hitung = hitung +1

```

## 3. N0 3

```

1 def perulangan(npm):
2     hitung = 0
3     npm = str(npm)
4     bil = npm[4:7]
5
6     while(hitung < 9):
7         print("Halo, "+bil+" wattsup?")
8         hitung = hitung +1
9
10    perulangan(int(input("Masukan NPM nyaa : ")))

```

## 4. N0 4

```
1 def perulangan2(npm):
2     npm = str(npm)
3     bil = npm[-3]
4     print("Halo, "+bil+" wattsup?")
5
6 perulangan2(int(input("Masukan NPM nyaa : ")))
```

## 5. N0 5

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

## 6. N0 6

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

## 7. N0 7

```
1 def perkalian(npm):
2     jumlah = 0
3     for i in npm:
4         jumlah *= int(i)
5     print(str(jumlah)+" hasil perkaliannya adalah ")
6
7 perkalian(input("Masukan NPM nyaa : "))
```

## 8. N0 8

```
1 def genap():
2     npm = [1,1,7,4,0,6,9]
3     for i in npm:
4         if (i % 2) == 0:
5             print("Bilangan Genapnya : "+str(i))
6 genap()
```

## 9. N0 9

```
1 def ganjil():
2     npm = [1,1,7,4,0,6,9]
3     for i in npm:
4         if (i%2)==1:
5             print("Bilangan Ganjilnya : "+str(i))
6 ganjil()
```

## 10. N0 10

```
1 def prima(npm):
2     npm = str(npm)
3     bil = npm[2]
4     num = int(bil)
```

```

5     if num > 1:
6         for i in range(2,num):
7             if (num%i)==0:
8                 print("Bukan Bilangan Prima")
9                 break
10            else:
11                print("Bilangan Primanya :"+str(num))
12    else:
13        print("Tidak Ada Bilangan Prima")
14    prima(int(input("Masukan NPM nyaa : ")))

```

#### 11. N0 11

```

1 from fungsi_fanny import penulisan

```

#### 12. N0 12

```

1 #from 3lib_1174069 import penulisan

```

### 3.3.3 penanganan Error

```

1 #def penanganan_error(a,b):
2 #     try :
3 #         c = a+b
4 #         print(c)
5 #     except TypeError:
6 #         print("We Are Different")

```

## 3.4 1174054 | Aulyardha Anindita

### 3.4.1 Pemahaman Teori

### 3.4.2 Fungsi

Fungsi adalah bagian dari suatu sub program yang terdiri dari nama fungsi itu sendiri dan variabel yang dapat digunakan ulang dan nama tersebut dapat dipanggil dimana-pun dalam suatu program. Fungsi dalam python menggunakan kata kunci 'def'. Dan setelah 'def' biasanya terdapat nama pengenalan fungsi yang diikuti oleh parameter yang diapit oleh tanda kurung dan diakhiri dengan tanda titik dua (:). Dan baris berikutnya adalah blok fungsi yang akan dijalankan jika fungsi dipanggil.

Contoh fungsi :

```

1 # Contoh fungsi
2 def dita():
3     print("Assalamualaikum , dita!")
4
5 dita()

```

Inputan fungsi adalah memanggil fungsi dengan fungsi yang telah dibuat dari inputan user dan mengembalikannya dalam bentuk string.

Contoh :

```

1 # Contoh inputan fungsi
2 def test(nama):
3     print ("Nama ku adalah : "+str(nama))
4
5 test(input("Masukkan Nama Kamu :"))

```

Kembalian fungsi adalah keluar dari suatu fungsi dan kembali ke baris selanjutnya dimana suatu fungsi dipanggil.

Contoh :

```

1 #Contoh kembalian fungsi
2 def test_return(a,b):
3     r = a + b
4     return r
5
6 a = 15
7 b = 25
8 c = test_return(a,b)
9 print(c)

```

### 3.4.3 Paket

Paket adalah suatu teknik pengumpulan dari beberapa file-file modul. Paket memudahkan programmer dalam mengelompokkan dan mengorganisasikan modul yang telah dibuat. Ringkasnya, kita tidak perlu membuat script untuk beberapa kasus, namun kita bisa mengelompokkannya dalam 1 file tiap kasus dan memanggilnya dalam satu program.

Cara memanggil paket adalah menggunakan kata kunci 'import' untuk mengimport file yang telah dibuat.

Contoh paket :

```

1 # Contoh 2 paket
2 #from fungsi_dita import *
3 #print(penulisan(int(input("Masukkan NPM anda : "))))

```

### 3.4.4 Kelas, Objek, Atribut, dan Method

Kelas adalah suatu entitas atau struktur data yang biasa digunakan yang terdiri dari objek, atribut, dan method didalamnya. Dengan kata lain kelas adalah sebuah cetak biru atau blueprint dari sebuah objek (instans)

Objek adalah suatu entitas yang biasanya memiliki variabel dan method didalamnya dengan kata lain memiliki keadaan (state) dan kelakuan (behavior).

Atribut adalah suatu entitas atau berupa fungsi-fungsi yang dimiliki oleh kelas atau objek. Biasanya atribut berisi variabel-variabel yang telah dideklarasikan.

Method adalah suatu fungsi yang melekat pada sebuah objek atau instan kelas untuk merepresentasikan suatu behavior (kelakuan).

Contoh :

```

1 # Contoh 3 Kelas
2 #class Employee:

```

```

3 # 'Common base class for all employees'
4 # empCount = 0
5
6 # def __init__(self, name, salary):
7 #     self.name = name
8 #     self.salary = salary
9 #     Employee.empCount += 1
10
11 # def displayCount(self):
12 #     print ("Total Employee %d" % Employee.empCount)
13
14 # def displayEmployee(self):
15 #     print ("Name : ", self.name, ", Salary: ", self.salary)
16
17
18 #This would create first object of Employee class"
19 #emp1 = Employee("Zara", 2000)
20 #This would create second object of Employee class"
21 #emp2 = Employee("Manni", 5000)
22 #emp1.displayEmployee()
23 #emp2.displayEmployee()
24 #print ("Total Employee %d" % Employee.empCount)

```

### 3.4.5 Cara Pemanggilan Library Kelas dari Instansiansi

Instansiansi adalah suatu pembuatan instance atau objek dari suatu kelas. Untuk memanggil nama kelas yaitu dengan menggunakan fungsi `init()` pada saat kita mendefinisikannya.

Contoh :

```

1 # Contoh pemanggilan library kelas dari instansiansi
2 #import tugas
3 #a = 25
4 #b = 30
5
6 #c = tugas.penambahan(a,b)
7 #print(c)

```

### 3.4.6 Pemakaian Paket dengan Perintah `from` dan `import`

```

1 # Contoh 2 paket
2 #from fungsi_dita import *
3 #print(penulisan(int(input("Masukkan NPM anda : "))))

```

### 3.4.7 Pemakaian Paket Fungsi

```

1 from kelas3lib_1174054 import penulisan

```

### 3.4.8 Pemakaian Paket Kelas

```

1 #from 3lib_1174054 import penulisan

```

### 3.4.9 Keterampilan Pemrograman

#### 1. Jawaban Soal No.1

```

1 #No 1
2 def penulisan(npm):
3     npm = list(str(npm))
4
5     angka1 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
6     angka2 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
7     angka3 = {"0": "##   ## ", "1": "### ", "2": "   ### ", "3": "##### ", "4": "###   ### ", "5": "###   ", "6": "##### ", "7": "##### ", "8": "   ", "9": "##### "}
8     angka4 = {"0": "##   ## ", "1": "### ", "2": "   ### ", "3": "##### ", "4": "###   ### ", "5": "###   ", "6": "##### ", "7": "##### ", "8": "   ", "9": "##### "}
9     angka5 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "   ", "9": "##### "}
10    angka6 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "   ", "9": "##### "}
11    angka7 = {"0": "##   ## ", "1": "### ", "2": "   ### ", "3": "##### ", "4": "###   ", "5": "   ### ", "6": "##### ", "7": "##### ", "8": "   ", "9": "##### "}
12    angka8 = {"0": "##   ## ", "1": "### ", "2": "   ### ", "3": "##### ", "4": "##### ", "5": "   ### ", "6": "##### ", "7": "##### ", "8": "   ", "9": "##### "}
13    angka9 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "   ", "9": "##### "}
14    angka10 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "   ", "9": "##### "}
15
16    hasil1 = []
17    hasil2 = []
18    hasil3 = []
19    hasil4 = []
20    hasil5 = []
21    hasil6 = []
22    hasil7 = []
23    hasil8 = []
24    hasil9 = []
25    hasil10 = []
26
27    for x in npm:
28        hasil1.append(angka1[x])
29        hasil2.append(angka2[x])
30        hasil3.append(angka3[x])
31        hasil4.append(angka4[x])
32        hasil5.append(angka5[x])
33        hasil6.append(angka6[x])

```

```

34     hasil7.append(angka7[x])
35     hasil8.append(angka8[x])
36     hasil9.append(angka9[x])
37     hasil10.append(angka10[x])
38
39     print(*hasil1, sep=' ')
40     print(*hasil2, sep=' ')
41     print(*hasil3, sep=' ')
42     print(*hasil4, sep=' ')
43     print(*hasil5, sep=' ')
44     print(*hasil6, sep=' ')
45     print(*hasil7, sep=' ')
46     print(*hasil8, sep=' ')
47     print(*hasil9, sep=' ')
48     print(*hasil10, sep=' ')
49
50 penulisan(int(input("Masukkan NPM Anda :")))

```

## 2. Jawaban Soal No.2

```

1 #No 2
2 def perulangan(npm):
3     hitung = 0
4     while(hitung < 54):
5         print("Halo, "+str(npm)+" apa kabar?")
6         hitung = hitung +1
7
8 perulangan(int(input("Masukan NPM Anda : ")))

```

## 3. Jawaban Soal No.3

```

1 #No 3
2 def perulangan_3_digit(npm):
3     hitung = 0
4     npm = str(npm)
5     bil = npm[4:7]
6
7     while(hitung < 9):
8         print("Halo, "+bil+" apa kabar?")
9         hitung = hitung +1
10
11 perulangan_3_digit(int(input("Masukan NPM Anda : ")))

```

## 4. Jawaban Soal No.4

```

1 #No 4
2 def perulangan_3_digit_terakhir(npm):
3     npm = str(npm)
4     bil = npm[-3]
5     print("Halo, "+bil+" apa kabar?")
6
7 perulangan_3_digit_terakhir(int(input("Masukan NPM Anda : ")))

```

## 5. Jawaban Soal No.5



```
1 #No 5
2 def down(npm):
3     for i in npm:
4         print (i)
5
6 down(input("Masukan NPM Anda : "))
```

## 6. Jawaban Soal No.6

```
1 #No 6
2 def penjumlahan(npm):
3     jumlah = 0
4     for i in npm:
5         jumlah += int(i)
6     print(str(jumlah)+" merupakan hasil dari penjumlahan")
7
8 penjumlahan(input("Masukkan NPM Anda : "))
```

## 7. Jawaban Soal No.7

```
1 #No 7
2 def perkalian(npm):
3     jumlah = 0
4     for i in npm:
5         jumlah *= int(i)
6     print(str(jumlah)+" merupakan hasil dari perkalian")
7
8 perkalian(input("Masukan NPM Anda : "))
```

## 8. Jawaban Soal No.8

```
1 #No 8
2 def genap():
3     npm = [1,1,7,4,0,2,7]
4     for i in npm:
5         if (i % 2) == 0:
6             print("Bilangan Genapnya adalah : "+str(i))
7     genap()
```

## 9. Jawaban Soal No.9

```
1 #No 9
2 def ganjil():
3     npm = [1,1,7,4,0,2,7]
4     for i in npm:
5         if (i%2)==1:
6             print("Bilangan Ganjilnya adalah : "+str(i))
7     ganjil()
```

## 10. Jawaban Soal No.10

```
1 #No 10
2 def prima(npm):
3     npm = str(npm)
4     bil = npm[2]
```

```

5     num = int(bil)
6     if num > 1:
7         for i in range(2,num):
8             if (num%i)==0:
9                 print("Bukan Bilangan Prima")
10                break
11            else:
12                print("Bilangan Prima :"+str(num))
13        else:
14            print("Tidak Ada Bilangan Prima")
15    prima(int(input("Masukan NPM Anda : ")))

```

### 11. Jawaban Soal No.11

```

1 from kelas3lib_1174054 import penulisan

```

### 12. Jawaban Soal No.12

```

1 #from 3lib_1174054 import penulisan

```

## 3.4.10 Penanganan Error

```

1 # Contoh inputan fungsi
2 def test(nama):
3     print ("Nama ku adalah : "+str(nama))
4
5 test(input("Masukkan Nama Kamu :"))

```

## 3.5 Dini Permata Putri

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

fungsi / function adalah satu blok kode yang melakukan tugas tertentu atau satu blok kode yang melakukan tugas tertentu atau satu blok instruksi yang eksekusi ketika dipanggil dari bagian lain dalam sebuah program. tujuan pembuatan fungsi adalah : memudahkan dalam pembuatan program.

contoh kodenya :

```

def function_name(parameters):
    """function_docstring"""
    statement(s)
    return [expression]

```

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

paket digunakan untuk mengelompokkan kelas-kelas yang mempunyai kemiripan fungsi (related class). kelas-kelas java yang akan digunakan didalam program, terlebih dahulu harus diimpor beserta dengan nama paket dimana kelas tersebut berada. cara memanggilnya :

```
penampung = namaClass()
penampung.namaMetode()
```

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

- class adalah salah satu cara bagaimana kita membuaat sebuah kode yang mempunyai behaviour tertentu dan lebih mudah dalam mengprganisasi berbagai fungsi dan state-nya. dalam sebuah class kamu dapat menyimpan sebuah state tanpa harus membuat banyak state bila tidak menggunakan class.

contohnya :

Class Product:

```
vendor.message = "Ini adalah rahasia"
name = ""
price = ""
size = ""
unit = ""
```

```
def init_(self, name):
print "ini adalah consuctor"
self.name = name
self.unit = "ml"
self.size = 250
```

```
def get.vendor.message(self):
print self.vendor.message
```

```
def set_price(Self, price):
self.price = price
```

- objek adalah instansi atau perwujudan dari sebuah kelas. bila kelas adalah prototypenya, dan objek adalah barang jadinya.

contohnya :

```
obj=Karyawan("K001", "Dini", "Teknisi")
obj.infoKaryawan()
```

```
#tambah karyawan baru
obj2=Karyawan("K002", "Ayu", "Akunting")
obj2.infoKaryawan()
#tampilkan total karyawan
print "Total Karyawan "
```

- atribut adalah instance spesifik untuk setiap objek, atribut class sama untuk semua contoh – yang dalam hal ini adalah semua dog.

contohnya :

```
class Dog:
#class Attribute
```

```
species = 'mammal'
```

```
#Initializer / Instance Attributes
```

```
def.init(self, name, age):
```

```
self.name = name
```

```
self.age = age
```

jadi, sementara setiap Dog memiliki nama dan umur yang unik, setiap Dog akan menjadi mamalia.

Method digunakan untuk melakukan operasi dengan atribut objek, seperti `init` metodenya, argumen pertama selalu `self`:

contohnya :

```
class dog:
```

```
# instance method def description(self):
```

```
return "() is () years old".format(self.name, self.age)
```

```
# instance method
```

```
def speak(self, sound):
```

```
return "() says {}".format(self.name, sound)
```

```
#instantiate the Dog object
```

```
mikey = Dog("Mikey", 6)
```

```
#call our instance methods
```

```
print(mikey.description())
```

```
print(mikey.speak("Gruff Gruff"))
```

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

library adalah salah satu cara yang paling efisien untuk menghemat waktu ketika membangun aplikasi. untuk memanggil file library digunakan perintah `include()` atau `require()`, selain `include()` dan `require()` ada juga `include.once()` dan `require.once()`, perbedaannya adalah jika suatu `include` ke suatu file dilakukan selama lebih dari 1 kali dalam suatu file, maka akan menghasilkan error karena dianggapnya ada pendklarasian ulang (`redeclare`), tetapi jika menggunakan `include.once()` atau `require.once()` maka kejadian tersebut dihindari.

contohnya :

```
$autoload
```

```
'libraries'] = array('form_validation','database','session'); $thisloadlibrary('nama.li
```

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

paket dengan perintah `from` kalkuator `import` penambahan pertama yaitu tentukan nama fungsi, variabel, dan inputannya apa saja, setiap penulisan harus menggunakan `()` dan `:` dan identasi (jaraknya harus sama)

contoh :

```
def penambahan (a+b):
```

```
r=(a+b)
```

```
return
```

```
a=5
```

```
b=6
```

```
anu=penambahan(a,b)
```

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

```
def tambah(bil1,bil2):
```

```
total = bil1+bil2
```

```
return total
```

```
def kurang(bil1,bil2):
```

```
total = bil1-bil2
```

```
return total
```

```
def kali(bil1,bil2):
```

```
total = bil1*bil2
```

```
return total
```

```
def bagi(bil1,bil2):
```

```
total = bil1/bil2
```

```
return total
```

```
def nilai(n1,n2):
```

```
hasil = n1+n2
```

```
if 80 <= hasil <= 100:
```

```
print("nilai anda adalah A")
```

```
elif 70 <= hasil <= 80:
```

```
print("nilai anda adalah B")
```

```
elif 60 <= hasil <= 70:
```

```
print("nilai anda adalah C")
```

```
elif 50 <= hasil <= 60:
```

```
print("nilai anda adalah D, Silahkan Mengulang kembali")
```

```
else: print("anda GAGAL")
```

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

mendefinisikan sebuah class dengan menggunakan kata kunci class diikuti oleh nama class tersebut.

```
class ClassName: """class docstring"""
```

```
class.body
```

class memiliki docstring atau string dokumentasi yang bersifat opsional artinya bisa atau tidak. Docstring bisa diakses menggunakan format

```
ClassName.doc
```

### 3.6 ainul filiani

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

fungsi / function adalah satu blok kode yang melakukan tugas tertentu atau satu blok kode yang melakukan tugas tertentu atau satu blok instruksi yang eksekusi ketika dipanggil dari bagian lain dalam sebuah program. tujuan pembuatan fungsi adalah : memudahkan dalam pembuatan program.

contoh kodenya :

```
def function_name(parameters):
    """function_docstring"""
    statement(s)
    return [expression]
```

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

paket digunakan untuk mengelompokkan kelas-kelas yang mempunyai kemiripan fungsi (related class). kelas-kelas java yang akan digunakan didalam program, terlebih dahulu harus diimpor beserta dengan nama paket dimana kelas tersebut berada. cara memanggilnya :

```
penampung = namaClass()
penampung.namaMetode()
```

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

- class adalah salah satu cara bagaimana kita membuat sebuah kode yang mempunyai behaviour tertentu dan lebih mudah dalam mengorganisasi berbagai fungsi dan state-nya. dalam sebuah class kamu dapat menyimpan sebuah state tanpa harus membuat banyak state bila tidak menggunakan class.

contohnya :

Class Product:

```
vendor.message = "Ini adalah rahasia"
name = ""
price = ""
size = ""
unit = ""
```

```
def init_(self, name):
    print "ini adalah constructor"
    self.name = name
    self.unit = "ml"
    self.size = 250
```

```
def get.vendor.message(self):
print self.vendor.message
```

```
def set_price(Self, price):
self.price = price
```

- objek adalah instansi atau perwujudan dari sebuah kelas. bila kelas adalah prototipenya, dan objek adalah barang jadinya.

contohnya :

```
obj=Karyawan("K001", "Dini", "Teknisi")
obj.infoKaryawan()
```

```
#tambah karyawan baru
obj2=Karyawan("K002", "Ayu", "Akunting")
obj2.infoKaryawan()
#tampilkan total karyawan
print "Total Karyawan "
```

- atribut adalah instance spesifik untuk setiap objek, atribut class sama untuk semua contoh – yang dalam hal ini adalah semua dog.

contohnya :

```
class Dog:
#class Attribute
species = 'mammal'
#Initializer / Instance Attributes
def.init(self, name, age):
self.name = name
self.age = age
```

jadi, sementara setiap Dog memiliki nama dan umur yang unik, setiap Dog akan menjadi mamalia.

Method digunakan untuk melakukan operasi dengan atribut objek, seperti init metodenya, argumen pertama selalu self:

contohnya :

```
class dog:
# instance method def description(self):
return "() is () years old".format(self.name, self.age)
# instance method
def speak(self, sound):
return "() says ()".format(self.name, sound)
#instantiate the Dog object
mikey = Dog("Mikey", 6)
#call our instance methods
print(mikey.description())
print(mikey.speak("Gruff Gruff"))
```

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

library adalah salah satu cara yang paling efisien untuk menghemat waktu ketika membangun aplikasi. Untuk memanggil file library digunakan perintah `include()` atau `require()`, selain `include()` dan `require()` ada juga `include.once()` dan `require.once()`, perbedaannya adalah jika suatu `include` ke suatu file dilakukan selama lebih dari 1 kali dalam suatu file, maka akan menghasilkan error karena dianggapnya ada pendefinisian ulang (`redeclare`), tetapi jika menggunakan `include.once()` atau `require.once()` maka kejadian tersebut dihindari.

contohnya :

```
$autoload
```

```
'libraries'] = array('form_validation','database','session'); $this->load_library('nama.li
```

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

paket dengan perintah `from` kalkulator import penambahan pertama yaitu tentukan nama fungsi, variabel, dan inputannya apa saja, setiap penulisan harus menggunakan `()` dan `:` dan indentasi (jaraknya harus sama)

contoh :

```
def penambahan(a+b):
```

```
    r=(a+b)
```

```
    return
```

```
    a=5
```

```
    b=6
```

```
    anu=penambahan(a,b)
```

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

```
def tambah(bil1,bil2):
```

```
    total = bil1+bil2
```

```
    return total
```

```
    def kurang(bil1,bil2):
```

```
        total = bil1-bil2
```

```
        return total
```

```
    def kali(bil1,bil2):
```

```
        total = bil1*bil2
```

```
        return total
```

```
    def bagi(bil1,bil2):
```

```
        total = bil1/bil2
```

```
        return total
```

```
    def nilai(n1,n2):
```

```
        hasil = n1+n2
```



```

if 80 <= hasil <= 100:
    print("nilai anda adalah A")
elif 70 <= hasil <= 80:
    print("nilai anda adalah B")
elif 60 <= hasil <= 70:
    print("nilai anda adalah C")
elif 50 <= hasil <= 60:
    print("nilai anda adalah D, Silahkan Mengulang kembali")
else: print("anda GAGAL")

```

7. Jelaskan dengan contoh kodenya, pemakaian paket kelas apabila file library ada di dalam folder mendefinisikan sebuah class dengan menggunakan kata kunci class diikuti oleh nama class tersebut.

```
class ClassName: """class docstring"""
```

```
class.body
```

class memiliki docstring atau string dokumentasi yang bersifat opsional artinya bisa atau tidak. Docstring bisa diakses menggunakan format

```
ClassName.doc
```

## 3.7 Chandra Kirana Poetra

### 3.7.1 Pemahaman Teori

1. Apa itu fungsi, inputan fungsi dan kembalian fungsi dengan contoh kode program lainnya. Fungsi merupakan suatu fitur yang dapat digunakan untuk membuat suatu program lebih mudah dimengerti dan juga agar program tersusun rapih serta mengurangi redudansi kode yang sama, misalkan kita ingin membuat suatu fungsinya penjumlahan maka, kita membuat fungsi penjumlahan lalu ketika kita ingin menghitung penjumlahan, maka yang perlu kita lakukan tinggal memanggil fungsinya itu sendiri agar fungsi itu lah yang akan melakukan eksekusi perhitungan. fungsi pada python biasanya dimulai dengan mengetikan kata kunci def yang selanjutnya diikuti dengan nama fungsinya itu sendiri contohnya seperti dibawah ini :

```
1 def masukannamafungsidisini(inputannyadisini):
```

Inputan fungsi sendiri merupakan suatu input/masukan data yang kita berikan ke program

```
1 outputnya = masukannamafungsidisini("Kembalian Fungsi")
```

Pengembalian fungsi atau return merupakan suatu fitur yang digunakan untuk mengembalikan nilai yang telah di proses oleh fungsi, penulisan return sendiri yaitu return diikuti dengan variabel mana yang nantinya ingin nilainya dikembalikan

```
1 return inputannyadisini
```

2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya. Library atau paket sendiri merupakan suatu kumpulan dari fungsi fungsi yang ditulis oleh berbagai macam developer yang biasanya digunakan untuk menghemat waktu, dikarenakan misalkan untuk fungsi koneksi, biasanya orang sudah menyediakan librarynya, oleh karena itu dikarenakan sudah ada yang membuat, maka akan lebih cepat jika kita menggunakan library yang dibuat oleh orang lain dibandingkan dengan kita harus membuat lagi dari awal.

```
1 #soal no 2
2 import math
3 print("Nilai I adalah : ", math.pi)
```

3. Jelaskan Apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing. Kelas merupakan blueprint atau template dari suatu objek yang nantinya akan di munculkan menjadi instance objek objek merupakan sebuah instance dari suatu kelas, kelas merupakan rancangan yang mendeskripsikan sesuatu hal didunia nyata secara umum misalkan kita punya class mobil, maka kita bisa membuat objek mobil sedan, mobil sport, mobil offroad dan lain lain seperti itulah contoh class dan objek, atribut adalah sesuatu hal yang melekat pada suatu objek atau kelas sedangkan method sendiri merupakan suatu fungsi yang ada pada suatu kelas yang cara penggunaanya dengan dipanggil

```
1 #soal no 3
2 class Mahasiswa:
3     jumlahMahasiswa = 0
4
5     def __init__(self, npm,nama):
6         self.npm = npm
7         self.nama = nama
8         Mahasiswa.jumlahMahasiswa +=1
9
10    def tampilprofile(self):
11        print("npm : ", self.npm)
12        print("NIK : ", self.nama)
13        print()
14
15    mahasiswa1 = Mahasiswa("1174079","Chandra Kirana Poetra")
16    mahasiswa2 = Mahasiswa("1174069","Bakti qilan")
17
18    mahasiswa1.tampilprofile()
19    mahasiswa2.tampilprofile()
20
21    print("total mahasiswa adalah", Mahasiswa.jumlahMahasiswa)
```

4. Jelaskan cara pemanggilan library kelas dari instansiasi dan pemakaianya dengan contoh program lainnya. cara pemanggilannya yaitu dengan import terdahulu filenya, lalu buat variabel untuk menampung data dan setelah itu panggil nama class dan panggil method yang ada di class itu.

```

1 #soal no 4
2
3 from file.Mahasiswa import Mahasiswa
4
5 mahasiswa1 = Mahasiswa("1174079","Chandra Kirana Poetra")
6 mahasiswa2 = Mahasiswa("1174069","Bakti qilan")
7 mahasiswa1.tampilprofile()
8 mahasiswa2.tampilprofile()
9 print("total mahasiswa adalah", Mahasiswa.jumlahMahasiswa)

```

5. Jelaskan dengan contoh pemakaian paket dengan perintah from kalkulator import Penambahan disertai dengan contoh kode lainnya. contohnya sebagai berikut :

```

1 #soal no 5
2 from file import Kalkulator
3
4 hasil = Kalkulator.penambahan(7,6)
5 print(hasil)

```

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila file library ada di dalam folder. pemakaian paket sendiri adalah kumpulan fungsi fungsi contoh dari kodenya sebagai berikut :

```

1 #soal no 6
2 from file import Kalkulator
3
4 a = 120
5 b = 60
6
7 hasil1 = Kalkulator.penambahan(a,b)
8 hasil2 = Kalkulator.pengurangan(a,b)
9 hasil3 = Kalkulator.perkalian(a,b)
10 hasil4 = Kalkulator.pembagian(a,b)
11
12 print(hasil1)
13 print(hasil2)
14 print(hasil3)
15 print(hasil4)

```

7. Jelaskan dengan contoh kodenya, pemakaian paket kelas apabila file library ada di dalam folder. pemakaian paket sendiri adalah kumpulan fungsi fungsi contoh dari kodenya sebagai berikut :

```

1 #soal no 7
2
3 from file.Mahasiswa import Mahasiswa
4
5 mahasiswa1 = Mahasiswa("1174079","Chandra Kirana Poetra")
6 mahasiswa2 = Mahasiswa("1174069","Bakti qilan")
7 mahasiswa1.tampilprofile()
8 mahasiswa2.tampilprofile()
9 print("total mahasiswa adalah", Mahasiswa.jumlahMahasiswa)
10
11 print(117407%9)

```

### 3.7.2 Praktek

#### 1. Jawaban soal no 1

```

1 #soal No.1
2 def AnswerNo1():
3
4     npm = input("Masukan NPM Anda:")
5     npm = list(str(npm))
6
7     number1 = {"0": "++++++ ", "1": " ++", "2": "++++++ ", "3":
8         "++++++ ", "4": "+++++", "5": "++++++", "6": "++++++",
9         "7": "++++++", "8": "+++++", "9": "+++++"}
10    number2 = {"0": "+++ ++", "1": "+++ ", "2": "++ ++", "3": "
11        +++ ++", "4": " ++ ++", "5": "++ ", "6": "+++ ",
12        "7": "++++++", "8": "++ ++", "9": "++ ++"}
13    number3 = {"0": "+++ ++", "1": "+++", "2": "++ ++", "3":
14        "+++ ", "4": " ++ ++", "5": "++ ", "6": "+++ ",
15        "7": "++ ++", "8": "+++++", "9": "++ ++"}
16    number4 = {"0": "+++ ++", "1": "+++", "2": "++ ++", "3":
17        "+++", "4": "++++++", "5": "++++++", "6": "++++++",
18        "7": "+++", "8": "++++++", "9": "++++++"}
19    number5 = {"0": "+++ ++", "1": "+++", "2": "+++", "3":
20        "+++ ++", "4": " ++", "5": "++", "6": "+++ ++",
21        "7": "+++", "8": "++ ++", "9": "++ ++"}
22    number6 = {"0": "++++++ ", "1": "+++", "2": "++++++ ", "3":
23        "++++++ ", "4": " ++", "5": "++++++", "6": "++++++ ",
24        "7": "+++ ", "8": "+++++", "9": "++++++"}
25
26    Result1 = []
27    Result2 = []
28    Result3 = []
29    Result4 = []
30    Result5 = []
31    Result6 = []
32
33    for x in npm:
34        Result1.append(number1[x])
35        Result2.append(number2[x])
36        Result3.append(number3[x])
37        Result4.append(number4[x])
38        Result5.append(number5[x])
39        Result6.append(number6[x])
40
41    print(*Result1, sep=' ')
42    print(*Result2, sep=' ')
43    print(*Result3, sep=' ')
44    print(*Result4, sep=' ')
45    print(*Result5, sep=' ')
46    print(*Result6, sep=' ')
47
48    AnswerNo1()

```

#### 2. Jawaban soal no 2

```

1 #Soal no.2
2 def perulangan(npm):

```

```

3     hitung = 0
4     while (hitung < 79):
5         print("Halo, " + str(npm) + " Apa Kabar?")
6         hitung = hitung + 1
7     perulangan(int(input("Masukkan NPM: ")))

```

### 3. Jawaban soal no 3

```

1 #Soal no.3
2 def perulangan3digitakhir(npm):
3     hitung = 0
4     npm = str(npm)
5     x = npm[4:7]
6     while (hitung < 16):
7         print("Halo, " + x + " Apa Kabar?")
8         hitung = hitung + 1
9     perulangan3digitakhir(int(input("Masukkan NPM: ")))

```

### 4. Jawaban soal no 4

```

1 #Soal no.4
2 def digit3daribelakang(npm):
3     npm = str(npm)
4     x = npm[-3]
5     print("Halo, " + x + " Apa Kabar?")
6     digit3daribelakang(int(input("Masukkan NPM: ")))

```

### 5. Jawaban soal no 5

```

1 #Soal no5
2 def npmkebawah(npm):
3     for i in npm:
4         print (i)
5
6     npmkebawah(input("Masukkan NPM: "))

```

### 6. Jawaban soal no 6

```

1 #Soal no.6
2 def penjumlahannpm(npm):
3     jumlahkan = 0
4     for i in npm:
5         jumlahkan += int(i)
6     print ("Hasil Penjumlahan NPM adalah : " + str(jumlahkan))
7     penjumlahannpm(input("Masukkan NPM: "))

```

### 7. Jawaban soal no 7

```

1 #Soal No.7
2 def perkaliannpm(npm):
3     kalikan = 0
4     for i in npm:
5         kalikan *= int(i)
6     print ("Hasil Perkalian NPM adalah : " + str(kalikan))
7     perkaliannpm(input("Masukkan NPM: "))

```

## 8. Jawaban soal no 8

```

1 #Soal No.8
2 def printdigitgenap(npm):
3     npm = list(map(int, npm))
4     for n in npm:
5         if(n % 2 == 0):
6             if(n != 0):
7                 print(n, end = "")
8     printdigitgenap(input("Masukkan NPM Anda bray: "))

```

## 9. Jawaban soal no 9

```

1 #Soal No.9
2 def printdigitganjil(npm):
3     npm = list(map(int, npm))
4     for n in npm:
5         if(n % 2 != 0):
6             print(n, end = "")
7     printdigitganjil(input("Masukkan NPM Anda bray: "))

```

## 10. Jawaban soal no 10

```

1 #Soal No.10
2 def printdigitprima(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     printdigitprima(input("Masukkan NPM: "))

```

## 11. Jawaban soal no 11

```

1
2 contohimport = __import__('3lib')
3 npm = "1174079"
4
5 result1 = contohimport.AnswerNo1(npm)
6 result2 = contohimport.AnswerNo2(npm)
7 result3 = contohimport.AnswerNo3(npm)
8 result4 = contohimport.AnswerNo4(npm)
9 result5 = contohimport.AnswerNo5(npm)
10 result6 = contohimport.AnswerNo6(npm)
11 result7 = contohimport.AnswerNo7(npm)
12 result8 = contohimport.AnswerNo8(npm)
13 result9 = contohimport.AnswerNo9(npm)

```

```

14 result10 = contohimport.AnswerNo10(npm)
15 print()

```

## 12. Jawaban soal no 12

```

1 from kelas3lib import kelas3lib
2
3 npm = "1174079"
4
5 kel3lib = kelas3lib(npm)
6
7 kel3lib.AnswerNo1()
8 kel3lib.AnswerNo2()
9 kel3lib.AnswerNo3()
10 kel3lib.AnswerNo4()
11 kel3lib.AnswerNo5()
12 kel3lib.AnswerNo6()
13 kel3lib.AnswerNo7()
14 kel3lib.AnswerNo8()
15 kel3lib.AnswerNo9()
16 kel3lib.AnswerNo10()

```

### 3.7.3 Praktek

1. peringatan error dan juga penjelasan
2. syntax error terjadi ketika kode yang dijalankan terdapat kesalahan penulisan, penempatan dan lain lain dengan solusi mencari di line yang terdapat masalah
3. zero division error eror ini terjadi ketika eksekusi program yang memiliki rumus matematika menghasilkan 0 dari eksekusi bagi yang memiliki hasil 0, jangan pernah membagi dengan 0
4. name error name error adalah error yang terjadi ketika kesalahan penamaan terdeteksi, solusinya jangan salah ketik  
contoh fungsi yang menggunakan try dan except

```

1 def cobainaye(masukinama):
2     try:
3         print("UY " + str(masukinama))
4     except:
5         print("ada yang salah bray")
6 cobainaye(input("Nama Mu nak"))

```

=====

## 3.8 Sekar Jasmine

### 3.8.1 Fungsi

Pada contoh dibawah , sebuah fungsi dengan nama perkalian(), memiliki dua buah argumen yaitu a dan b. Isi dari fungsi tersebut adalah melakukan perhitungan perkalian

yang diambil dari nilai a dan b, yang di simpan ke dalam variabel c. Nilai dari c lah yang akan dikembalikan oleh fungsi dari hasil pemanggilan fungsi melalui statemen perkalian(5,10)

```

1 Contoh:\\
2 def perkalian(a,b):
3     c = a*b
4     return c
5     #Program Utama
6 print( perkalian(5,10))
7
8 >>> def nama():
9     gelar = 'Mr'
10    aksi = (lambda x: gelar + ' ' + x)
11    return aksi
12
13 >>> act = nama()
14 >>> act('Namjoon')
15 'Sir Namjoon'

```

**3.8.1.1 Scope Variabel** cakupan variabel merupakan suatu keadaan dimana pen-deklarasian sebuah variabel di tentukan , Dalam scope variabel dikenal dua istilah yaitu local dan global. Contoh penggunaan scope variabel() : x = 12 y = 3 print "Sebelum memanggil fungsi, x bernilai", x print "Sebelum memanggil fungsi, y bernilai", y swap(x,y) print "Setelah memanggil fungsi, x bernilai", x print "Setelah memanggil fungsi, y bernilai", y

**3.8.1.2 Fungsi Rekursif** untuk menyederhanakan penulisan program dan meng-gantikan bentuk iterasi. Dengan rekursi, program akan lebih mudah dilihat.

```

1 # Fungsi Rekursif faktorial
2 def faktorial(nilai):
3     if nilai <= 1:
4         return 1
5     else:
6         return nilai * faktorial(nilai - 1)
7 #Program utama
8 for i in range(11):
9     print "%2d != %d" % (i, faktorial(i))

```

**3.8.1.3 Melewatkan Argumen dengan Kata Kunci** Jika fungsi perkalian kita panggil dengan memberi pernyataan perkalian(10,8), maka nilai 10 akan disalin ke variabel x dan nilai 8 ke variabel y.

```

1 def perkalian(a, b):
2     "Mengalikan dua bilangan"
3     z = x * y
4     print "Nilai a =",a
5     print "Nilai b =",b
6     print "a* b =",c

```



```

7 # program utama mulai di sini
8     perkalian(5,3)
9     print perkalian(b=4,a=2)
10 Hasilnya:
11 Nilai a = 5
12 Nilai b = 3
13 a*b = 15
14 Nilai a = 2

```

Jadi nilai default hanya boleh diberikan kepada deretan akhir parameter. Setelah pemberian nilai default, semua parameter di belakangnya juga harus diberi nilai default. Satu catatan, nilai awal argumen akan dievaluasi pada saat dideklarasikan. Perhatikan contoh berikut :

```

1 usernm="admin"
2 passwd="aa"
3 def login(username=usernm , password=passwd):
4     print "Your username ",username
5     print "Your password ",password
6     print
7     usernm="tamu"
8     passwd="cc"
9 login()
10 Untuk memanggil fungsi dengan deklarasi seperti ini , kita harus
    menyebutkan daftar argumen beserta kata-kuncinya .
11 Contoh () :
12     def cetak1():
13 print 'Hello World'
14     def cetak2(n):
15 print n
16     cetak1()
17 hallo world
18     cetak2(123)
19 123
20     cetak2('apa kabar?')
21 apa kabar
22     def cetak3(x,y,z):
23 print x,y,z
24     def cetak4(x,y,z=4):
25 print x,y,z
26     cetak3(1,2,3)
27 1 2 3
28     cetak4(1,2)
29 1 2 4
30     cetak4(1,2,3)
31 1 2 3
32
33 \class Ngitung:
34     def __init

```

### 3.8.2 Kelas

Class adalah salah satu cara bagaimana kita membuat sebuah kode yang mempunyai behaviour tertentu dan lebih mudah dalam mengorganisasi berbagai fungsi dan state-nya. Dalam sebuah class kamu dapat menyimpan sebuah state tanpa harus membuat

banyak state bila tidak menggunakan class.

Contoh :

```

1 class Product:
2     __vendor_message = "Ini adalah rahasia"
3     name = ""
4     price = ""
5     size = ""
6     unit = ""
7
8     def __init__(self, name):
9         print "Ini adalah constructor"
10        self.name = name
11        self.unit = "ml"
12        self.size = 350
13
14        def get_vendor_message(self):
15            print self.__vendor_message
16
17        def set_price(self, price):
18            self.price = price
19
20 p = Product("Banana Milk")
21 p.set_price(5500)
22
23 print "%s dengan ukuran %s %s harganya Rp. %d" % (p.name, p.size, p.
24         unit, p.price)
25 # print p.__vendor_message
26
27 p.get_vendor_message()
28
29 p1 = Product("UltraMilk")
30 p1.set_price(3000)
31
32 print "%s dengan ukuran %s %s harganya Rp. %d" % (p.name, p.size, p.
33         unit, p.price)
34
35 print p == p
36 print p1 == p1
37 print p == p1

```

### 3.8.3 Pemahaman Teori

1. void(fungsi tanpa nilai balik) Fungsi yang void sering disebut juga prosedur. Disebut void karena fungsi tersebut tidak mengembalikan suatu nilai keluaran yang didapat dari hasil proses tersebut. Ciri-ciri dari jenis fungsi Void, yaitu : 1. tidak adanya keyword return 2. tidak adanya tipe data di dalam deklarasi fungsi 3. menggunakan keyword void 4. tidak memiliki nilai kembalian fungsi 5. Keyword void juga digunakan jika suatu function tidak mengandung suatu parameter apapun. Contohnya:

```

1 void menampilkan_jumlah(int a, int b){
2     in jumlah;

```

```

3     jumlah = a + b;
4     cout << jumlah;
5 }
6

```

2. Non void (fungsi dengan nilai balik) Fungsi non-void disebut juga function. disebut non-void karena mengembalikan nilai kembalian yang berasal dari keluaran hasil proses function tersebut. Ciri-ciri dari jenis fungsi Non-Void, yaitu: 1. Ada keyword return 2. ada tipe data yang mengawali fungsi 3. tidak ada keyword void 4. memiliki nilai keyword 5. Non-void: int jumlah (int a, int b)

3. Prototype Function Sebuah program C++ dapat terdiri dari banyak fungsi. Salah satu fungsi tersebut harus bernama main(). Jika fungsi yang lain dituliskan setelah fungsi main(), sebelum fungsi main ditambahkan. Contohnya:

```

1 #include <stdio.h>
2 //prototype function
3 int hitung(int angka, int bilangan);
4 int tulis(char);
5 int tampil(int angka[], char huruf);
6 // fungsi main
7 int main() {
8     int array[3] = {1, 2, 3};
9     char huruf = "D";
10    // memanggil fungsi
11    hitung(2, 3);
12    tulis("A");
13    tampil(array, huruf);
14 }

```

4. Fungsi Rekursif Fungsi yang memanggil dirinya sendiri. Artinya, fungsi tersebut dipanggil di dalam tubuh fungsi itu sendiri. Parameter yang dilewatkan berubah sebanyak fungsi itu dipanggil.

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

```

1 <?php if ( ! defined('BASEPATH'))
2     exit('No direct script access allowed');
3 class Blog extends CI_Controller {
4     function __construct()
5     {
6         parent::__construct();
7     }
8     function index()
9     {
10        echo "Hallo.. saya min yoongi adalah contoh dari boyband BTS
11           yang mendunia";
12    }
13 }
14 }

```

2. Jelaskan Apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing. gambaran umum tentang sebuah

benda. Di dalam pemrograman nantinya, contoh class seperti: koneksi database dan profile user. penulisan class diawali dengan keyword class, kemudian diikuti dengan nama dari class. Aturan penulisan nama class sama seperti aturan penulisan variabel Contoh :

```

1 <?php
2     class laptop {
3         // isi dari class laptop...
4     }
5 ?>

```

3. Jelaskan cara pemanggilan library kelas dari instansiasi dan penggunaannya dengan contoh program lainnya.

1. Membuat sebuah objek atau sebuah instance pada sebuah kelas disebut instansiasi atau instantiation. Contoh:

```

1 String str = new String("Hello");
2 String str2 = "OOP Yes";
3 Komputer a = new Komputer();
4 Komputer b = new Komputer();

```

2. Atribut suatu class harus didefinisikan sebagai instance variable.

Contoh:

```

1 public class Time {
2     private int hour;
3     private int minute;
4     private int second;
5     //penulisan kode selanjutnya
6 }

```

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

Paket dengan perintah from kalkulator import import penambahan pertama , yaitu tentukan nama fungsi , variabel dan inputnya. setiap penulisan harus menggunakan () dan : dan identasi. Contoh :

```

1 def penambahan (a+b):\
2     r=(a+b)\
3     return \
4     a=5\
5     b=6\
6     anu=penambahan(a,b)

```

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

```

1 // Meletakkan kelas ke paket
2 package bangun.datar;
3
4 // Mendefinisikan kelas Segi3ABC
5 public class Segi3ABC {
6
7     // Metoda hitungKeliling
8     // Untuk mencari keliling segi tiga
9     public static double hitungKeliling(double sisiAB, double
        sisiBC, double sisiCA) {
10
11         double keliling;
12         keliling = sisiAB + sisiBC + sisiCA;
13         return keliling;
14     }
15
16     // Metoda hitungLuas
17     // Untuk mencari luas segi tiga
18     public static double hitungLuas(double sisiAB) {
19
20         // Deklarasi variabel
21         double luas;
22
23         // Mencari tinggi segi tiga
24         double tinggi = Math.sqrt(Math.pow(sisiAB, 2) - Math.pow
            ((0.5 * sisiAB), 2));
25
26         // Mencari luas segi tiga
27         luas = sisiAB * tinggi;
28         return luas;
29     }
30 }

```

### 3.8.4 Keterampilan Pemrograman

```

1 def perulangan(npm):
2     hitung = 0
3     while(hitung < 75):
4         print("Halo, "+str(npm)+" apa kabar?")
5         hitung = hitung +1
6
7 perulangan(int(input("Masukan NPM : ")))

```

```

1 def perulangan_3_digit(npm):
2     hitung = 0
3     npm = str(npm)
4     bil = npm[4:7]
5
6     while(hitung < 9):
7         print("Halo, "+bil+" apa kabar?")
8         hitung = hitung +1
9
10 perulangan_3_digit(int(input("Masukan NPM : ")))

```

```
1 def perulangan_3_digit_terakhir(npm):
2     npm = str(npm)
3     bil = npm[-3]
4     print("Halo, "+bil+" apa kabar?")
5
6 perulangan_3_digit_terakhir(int(input("Masukan NPM : ")))
```

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

```
1 def penjumlahan(npm):
2     jumlah = 0
3     for i in npm:
4         jumlah += int(i)
5     print(str(jumlah)+" Adalah hasil penjumlahan")
6
7 penjumlahan(input("Masukan NPM : "))
```

```
1 def perkalian(npm):
2     jumlah = 0
3     for i in npm:
4         jumlah *= int(i)
5     print(str(jumlah)+" Adalah hasil perkalian")
6
7 perkalian(input("Masukan NPM : "))
```

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

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

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

```

11         print("Bilangan Primanya :"+str(num))
12     else :
13         print("Tidak Ada Bilangan Prima")
14 prima(int(input("Masukan NPM : ")))

```

## 3.9 Bakti Qilan Mufid

### 3.9.1 Teori

1. apa itu fungsi, inputan fungsi, dan kembalian fungsi dan contohnya

Fungsi adalah bagian dari program yang dapat digunakan ulang, yaitu dengan cara memberinya nama dan kemudian nama itu dapat dipanggil dimanapun dalam program. fungsi dalam pyhton didefinisikan dengan kata kunci **def**. Setelah **def** ada nama pengenalan fungsi diikuti dengan parameter yang diapit oleh tanda kurung dan diakhiri dengan tanda titik dua (:). Baris berikutnya berupa blok fungsi atau perintah yang akan dijalankan jika fungsi tersebut dipanggil. untuk pengisian variabel bisa diisikan lebih dari satu variabel, dengan menggunakan tanda pemisah yaitu koma (.). contoh dari fungsi sederhana itu bisa dilihat dibawah dimana hasil akhir dari variabel c adalah 25

```

1 def HasilBagi(a,b):
2     h = a / b
3     return h
4
5 a = 50
6 b = 2
7 c = int(HasilBagi(a,b))
8 #menggunakan int() karena jika tidak, maka hasilnya akan menjadi
   format float.

```

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

Paket atau package ialah sebuah file, contoh nya file bernama *117408.py* yang didalamnya berisi semua fungsi, seperti penambahan, pengurangan, pembagian dan perkalian. sehingga file tersebut kita namakan sebagai paket/ package/ library. contoh nya seperti pada kode program dibawah.

```

1 def penambahan(c,d):
2     r = c + d
3     return r
4 def pengurangan(c,d):
5     r = c - d
6     return r
7 def perkalian(c,d):
8     r = c * d
9     return r
10 def pembagian(c,d):
11     r = c / d
12     return r

```

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

kelas, ialah sebuah prototipe atau kerangka yang digunakan oleh pengguna untuk objek yang mendefinisikan seperangkat atribut yang menjadi ciri objek kelas.

- atribut, ialah data atau apapun yang menempel pada objek.
- objek, ialah contoh unik dari struktur data yang didefinisikan oleh kelasnya.
- method, ialah perilaku, ataupun yang dikerjakan oleh objek.
- contoh kode programnya

```

1
2 class berhitung:
3     def __init__(self, x, y):
4         self.x = x
5         self.y = y
6     def Penambahan(self):
7         z = self.x + self.y
8         return z
9     def Pengurangan(self):
10        z = self.x - self.y
11        return z
12    def Perkalian(self):
13        z = self.x * self.y
14        return z
15    def Pembagian(self):
16        z = self.x / self.y
17        return z

```

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

Untuk memanggil library atau paket, kita cukup dengan menggunakan keyword import, seperti pada kode program dibawah. Di sini kita mencoba memanggil library pada kelas 1174083.py dengan menggunakan nama test.

```

1 import test_1174083 as test
2
3 m = 10
4 n = 5
5
6 coba = test.test_1174083(m,n)
7
8 hasil_1 = coba.Penambahan()
9 hasil_2 = coba.Pengurangan()
10 hasil_3 = coba.Perkalian()
11 hasil_4 = coba.Pembagian()

```

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

Penggunaan paket from *namafile* import, itu berfungsi untuk memanggil file dan fungsinya, seperti pada kode program dibawah



```

1 from contoh_fungsi import
2 print(int(input("input NPM : ")))

```

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

Contoh pemakaian paket seperti pada kode program dibawah

```

1 import coba
2
3 p = 20
4 q = 10
5
6 hasilnya = coba.penambahan(p,q)
7 print(hasilnya)

```

### 3.9.2 Ketrampilan Pemrograman

#### 1. Jawaban

```

1 print(1174083%3)
2 # hasil mod3 = 0
3 def printNPM(npm):
4
5     npm = list(str(npm))
6
7     angka1 = {"0": "*****", "1": "****", "2": "*****", "3": "*****",
8               "4": "*****", "5": "*****", "6": "*****", "7": "*****", "8": "*****", "9": "*****"}
9     angka2 = {"0": "****", "1": "****", "2": "****", "3": "****", "4": "****", "5": "****", "6": "****", "7": "****", "8": "****", "9": "****"}
10    angka3 = {"0": "****", "1": "****", "2": "****", "3": "****", "4": "****", "5": "****", "6": "****", "7": "****", "8": "****", "9": "****"}
11    angka4 = {"0": "****", "1": "****", "2": "****", "3": "****", "4": "****", "5": "****", "6": "****", "7": "****", "8": "****", "9": "****"}
12    angka5 = {"0": "****", "1": "****", "2": "****", "3": "****", "4": "****", "5": "****", "6": "****", "7": "****", "8": "****", "9": "****"}
13    angka6 = {"0": "*****", "1": "****", "2": "*****", "3": "*****", "4": "*****", "5": "*****", "6": "*****", "7": "*****", "8": "*****", "9": "*****"}
14
15    hasil1 = []
16    hasil2 = []
17    hasil3 = []
18    hasil4 = []
19    hasil5 = []
20    hasil6 = []

```

```

21     for x in npm:
22         hasil1.append(angka1[x])
23         hasil2.append(angka2[x])
24         hasil3.append(angka3[x])
25         hasil4.append(angka4[x])
26         hasil5.append(angka5[x])
27         hasil6.append(angka6[x])
28
29     print(*hasil1, sep=' ')
30     print(*hasil2, sep=' ')
31     print(*hasil3, sep=' ')
32     print(*hasil4, sep=' ')
33     print(*hasil5, sep=' ')
34     print(*hasil6, sep=' ')
35
36 printNPM(input("Masukan NPM anda: "))

```

## 2. Jawaban

```

1 def NPMperulangan(npm):
2     i = 1
3     digit = int(npm[5:7])
4     while(i <= digit):
5         print("Halo, "+str(npm)+" apa kabar?")
6         i += 1
7
8 NPMperulangan (input("Masukan NPM : "))

```

## 3. Jawaban

```

1 def SumNPM_3digit(npm):
2     ulang = 1
3     sampai = list(map(int, npm[4:7]))
4     sampai = sum(sampai)
5     while(ulang <= sampai):
6         print("Halo, "+str(npm[-3:])+" apa kabar?")
7         ulang += 1
8
9 SumNPM_3digit(input("Masukan NPM: "))

```

## 4. Jawaban

```

1 def NPMdigitter3(npm):
2     print("Output: Halo, "+str(npm[-3])+" apa kabar?")
3
4 NPMdigitter3(input("Masukan NPM: "))

```

## 5. Jawaban

```

1 def noNPMbarisSatu(npm):
2     npm = list(map(int, npm))
3     for x in npm:
4         print(x)
5
6 noNPMbarisSatu(input("Masukan NPM : "))

```

## 6. Jawaban

```
1 def PenjumlahanNPM(npm):
2     npm = list(map(int, npm))
3     semua = 0
4     for a in npm:
5         semua += a
6
7     print(semua)
8
9 PenjumlahanNPM(input("Masukan NPM : "))
```

## 7. Jawaban

```
1 def PerkalianNPM(npm):
2     npm = list(map(int, npm))
3     semua = 0
4     for a in npm:
5         semua *= a
6
7     print(semua)
8
9 PerkalianNPM(input("Masukan NPM : "))
```

## 8. Jawaban

```
1 def NPMgenapOnly(npm):
2     npm = list(map(int, npm))
3     for x in npm:
4         if(x % 2 == 0):
5             if(x != 0):
6                 print(x, end = "")
7
8 NPMgenapOnly(input("Masukan NPM : "))
```

## 9. Jawaban

```
1 def NPMganjilOnly(npm):
2     npm = list(map(int, npm))
3     for x in npm:
4         if(x % 2 != 0):
5             print(x, end = "")
6
7 NPMganjilOnly(input("Masukan NPM : "))
```

## 10. Jawaban

```
1 def NPMprimaOnly(npm):
2     npm = list(map(int, npm))
3     prima = []
4     for a in npm:
5         bilPrima = True
6         if a == 0 or a == 1:
7             bilPrima = False
8         for b in range(2, a):
9             if a % b == 0:
```

```

10         bilPrima = False
11     if bilPrima:
12         prima.append(a)
13
14     for c in prima:
15         print(c, end = "")
16
17 NPMprimaOnly(input("Masukan NPM : "))

```

## 11. Jawaban

```

1 lib = __import__('3lib')
2
3 npm = str(input("Masukan NPM: "))
4
5 hasil1 = lib.printNPM(npm)
6 hasil2 = lib.NPMperulangan(npm)
7 hasil3 = lib.SumNPM_3digit(npm)
8 hasil4 = lib.NPMdigit3ter(npm)
9 hasil5 = lib.noNPMbarisSatu(npm)
10 hasil6 = lib.PenjumlahanNPM(npm)
11 hasil7 = lib.PerkalianNPM(npm)
12 hasil8 = lib.NPMgenapOnly(npm)
13 print()
14 hasil9 = lib.NPMganjilOnly(npm)
15 print()
16 hasil10 = lib.NPMprimaOnly(npm)
17 print()

```

## 12. Jawaban

```

1 from kelas3lib import kelas3lib
2
3 npm = str(input("Masukan NPM: "))
4
5 kel3lib = kelas3lib(npm)
6
7 kel3lib.printNPM()
8 kel3lib.NPMperulangan()
9 kel3lib.SumNPM_3digit()
10 kel3lib.NPMdigit3ter()
11 kel3lib.noNPMbarisSatu()
12 kel3lib.PenjumlahanNPM()
13 kel3lib.PerkalianNPM()
14 kel3lib.NPMgenapOnly()
15 print()
16 kel3lib.NPMganjilOnly()
17 print()
18 kel3lib.NPMprimaOnly()

```

### 3.9.3 Ketrampilan Penanganan Error

1. Peringatan error yang ditemukan dan penjelasannya serta buat sebuah fungsi try except untuk menanggulangi error.

Peringatan error di praktek ketiga ini, yaitu:

- **Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- **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 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 terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkoversi varibelnya sesuai dengan tipe data yang akan digunakan.

Contoh fungsi yang menggunakan try except

```

1 def coba(inputan):
2     try:
3         print("Hallo, "+str(inputan))
4     except:
5         print("Terjadi error")
6
7 coba(input("Inputan : "))

```

## 3.10 Advent Nopele Olansi Damiahan Sihite 1174089

### 3.10.0.1 Pemahaman Teori

#### 1. Apa itu fungsi, inputan

fungsi dan kembalian fungsi dengan contoh kode program lainnya. Fungsi adalah bagian dari program yang dapat digunakan ulang. Berikut merupakan contoh fungsi dan cara pemanggilannya

```

1 def uji():
2     print("Tugas Web Service")
3
4 uji()

```

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

```

1 def uji_param(nama):
2     print("Nama saya : "+str(nama))
3
4 uji_param(input("Masukan Nama Kamu : "))

```

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

```

1 def uji_return(a,b):
2     r = a + b
3     return r
4
5 a = 10
6 b = 50
7 c = uji_return(a,b)
8 print(c)

```

2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya. Untuk memudahkan dalam pemanggilan fungsi yang di butuhkan, agar dapat dipanggil berulang. Cara pemanggilannya

```

1 #from fungsi_felix import *
2 #print(penulisan(int(input("Masukan NPM kamu : "))))

```

3. Jelaskan Apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing. kelas merupakan sebuah blueprint yang merepresentasikan objek. objek adalah hasil cetakan dari sebuah kelas. method adalah suatu upaya yang digunakan oleh object.

```

1 #class Employee:
2 #     'Common base class for all employees'
3 #     empCount = 0
4
5 #     def __init__(self, name, salary):
6 #         self.name = name
7 #         self.salary = salary
8 #         Employee.empCount += 1
9
10 #     def displayCount(self):
11 #         print ("Total Employee %d" % Employee.empCount)
12
13 #     def displayEmployee(self):
14 #         print ("Name : ", self.name, ", Salary: ", self.salary)
15
16
17 #This would create first object of Employee class"
18 #emp1 = Employee("Zara", 2000)
19 #This would create second object of Employee class"
20 #emp2 = Employee("Manni", 5000)
21 #emp1.displayEmployee()
22 #emp2.displayEmployee()
23 #print ("Total Employee %d" % Employee.empCount)

```

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

- pertama import terlebih dahulu filenya.
- kemudian buat variabel untuk menampung datanya

- setelah itu panggil nama classnya dan panggil methodnya
- Gunakan perintah print untuk menampilkan hasilnya

```

1 #import belajar
2 #a = 100
3 #b = 50
4
5 #c = belajar.penambahan(a,b)
6 #print(c)

```

5. Jelaskan dengan contoh pemakaian paket dengan perintah from kalkulator import Penambahan disertai dengan contoh kode lainnya. Penggunaan paket from namafile import, itu berfungsi untuk memanggil file dan fungsinya

```

1 #from fungsi_felix import *
2 #print(penulisan(int(input("Masukan NPM kamu : "))))

```

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila le library ada di dalam folder. Pemakaian paket adalah perkumpulan fungsi-fungsi. contoh kodenya adalah sebagai berikut :
7. Jelaskan dengan contoh kodenya, pemakaian paket kelas apabila le library ada di dalam folder.

### 3.10.0.2 Ketramplan Pemrograman

1. Buatlah fungsi dengan inputan variabel NPM, dan melakukan print luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk NPM mod 3=0, tanda pagar untuk NPM mod 3 =1, tanda plus untuk NPM mod3=2.

```

1
2 #No 1
3 def speakup(npm):
4     npm = list(str(npm))
5
6     angka1 = {"0": "***** ", "1": " *** ", "2": "***** ", "3": "
7     ***** ", "4": " *** ", "5": "***** ", "6": "***** ", "7": "
8     ***** ", "8": " ***** ", "9": "***** "}
9     angka2 = {"0": "***** ", "1": " *** ", "2": "***** ", "3": "
10    ***** ", "4": " *** ", "5": "***** ", "6": "***** ", "7": "
11    ***** ", "8": " ***** ", "9": "***** "}
12    angka3 = {"0": " *** ", "1": " *** ", "2": " ***** ", "3": "
13    ***** ", "4": " *** ", "5": "***** ", "6": " ***** ", "7": "
14    ***** ", "8": " ***** ", "9": " ***** "}
15    angka4 = {"0": " *** ", "1": " *** ", "2": " ***** ", "3": "
16    ***** ", "4": " *** ", "5": "***** ", "6": " ***** ", "7": "
17    ***** ", "8": " ***** ", "9": " ***** "}
18    angka5 = {"0": " *** ", "1": " *** ", "2": " ***** ", "3": "
19    ***** ", "4": " ***** ", "5": "***** ", "6": "***** ", "7": "
20    ***** ", "8": " ***** ", "9": "***** "}

```

```

11  angka6 = {"0": " ***      *** ", "1": " *** ", "2": " ***** ", "3": "
      *** ", "4": "      *** ", "5": " ***** ", "6": " ***** ", "7": "
      *** ", "8": " ***** ", "9": " ***** "}
12  angka7 = {"0": " ***      *** ", "1": " *** ", "2": " *** ", "3": "
      *** ", "4": "      *** ", "5": " ***** ", "6": " ***** ", "7": "
      *** ", "8": " ***** ", "9": " ***** "}
13  angka8 = {"0": " ***      *** ", "1": " *** ", "2": " *** ", "3": "
      *** ", "4": "      *** ", "5": " ***** ", "6": " ***** ", "7": "
      *** ", "8": " ***** ", "9": " ***** "}
14  angka9 = {"0": " ***** ", "1": " *** ", "2": " ***** ", "3": "
      *** ", "4": "      *** ", "5": " ***** ", "6": " ***** ", "7": "
      *** ", "8": " ***** ", "9": " ***** "}
15  angka10 = {"0": " ***** ", "1": " *** ", "2": " ***** ", "3": "
      *** ", "4": "      *** ", "5": " ***** ", "6": " ***** ", "7": "
      *** ", "8": " ***** ", "9": " ***** "}
16
17  hasil1 = []
18  hasil2 = []
19  hasil3 = []
20  hasil4 = []
21  hasil5 = []
22  hasil6 = []
23  hasil7 = []
24  hasil8 = []
25  hasil9 = []
26  hasil10 = []
27
28  for x in npm:
29      hasil1.append(angka1[x])
30      hasil2.append(angka2[x])
31      hasil3.append(angka3[x])
32      hasil4.append(angka4[x])
33      hasil5.append(angka5[x])
34      hasil6.append(angka6[x])
35      hasil7.append(angka7[x])
36      hasil8.append(angka8[x])
37      hasil9.append(angka9[x])
38      hasil10.append(angka10[x])
39
40  print(*hasil1, sep=' ')
41  print(*hasil2, sep=' ')
42  print(*hasil3, sep=' ')
43  print(*hasil4, sep=' ')
44  print(*hasil5, sep=' ')
45  print(*hasil6, sep=' ')
46  print(*hasil7, sep=' ')
47  print(*hasil8, sep=' ')
48  print(*hasil9, sep=' ')
49  print(*hasil10, sep=' ')
50
51  speakup(int(input("Masukan NPM : 117")))

```

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



```

1 def looping(npm):
2     hitung = 0
3     while(hitung < 89):
4         print("Haiiiii , "+str(npm)+" apa kabar?")
5         hitung = hitung +1
6
7 looping(int(input("Masukan NPM : ")))

```

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

```

1 #No 3
2 def looping_3_digit(npm):
3     hitung = 0
4     npm = str(npm)
5     bil = npm[4:7]
6
7     while(hitung < 17):
8         print("Halo , "+bil+" apa kabar?")
9         hitung = hitung +1
10
11 looping_3_digit(int(input("Masukan NPM : ")))

```

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

```

1 #No 4
2 def looping_3_digit_terakhir(npm):
3     npm = str(npm)
4     bil = npm[-3]
5     print("Haiii , "+bil+" apa kabar?")
6
7 looping_3_digit_terakhir(int(input("Masukan NPM : ")))

```

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

```

1 #No 5
2 def down(npm):
3     for i in npm:
4         print (i)

```

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

```

1
2 #No 6
3 def addition(npm):
4     jumlah = 0
5     for i in npm:
6         jumlah += int(i)
7     print(str(jumlah)+" Adalah hasil penjumlahan")

```

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

```

1
2 #No 7
3 def multiplication(npm):
4     jumlah = 0
5     for i in npm:
6         jumlah *= int(i)
7     print(str(jumlah)+" Adalah hasil perkalian")

```

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

```

1
2 #No 8
3 def even():
4     npm = [1,1,7,4,0,8,9]
5     for i in npm:
6         if (i % 2) == 0:
7             print("Bilangan Genapnya : "+str(i))

```

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

```

1 even()
2
3 #No 9
4 def odd():
5     npm = [1,1,7,4,0,8,9]
6     for i in npm:
7         if (i%2)==1:

```

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

```

1
2 odd()
3
4 #No 10
5 def prime(npm):
6     npm = str(npm)
7     bil = npm[2]
8     num = int(bil)
9     if num > 1:
10        for i in range(2,num):
11            if (num%i)==0:
12                print("Bukan Bilangan Prima")
13                break
14        else:
15            print("Bilangan Primanya :"+str(num))

```

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

```
1 from fungsi_advent import learn
```

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

```
1 from 3lib_advent import learn
```

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

```
1 #def penanganan_error(a,b):
2 #     try :
3 #         c = a+b
4 #         print(c)
5 #     except TypeError:
6 #         print("kita beda")
```

## 3.11 Arrizal Furqona Gifary

### 3.11.0.1 Pemahaman Teori

1. Apa itu fungsi, inputan

fungsi dan kembalian fungsi dengan contoh kode program lainnya, Fungsi adalah bagian dari program yang dapat digunakan ulang. Berikut merupakan contoh fungsi dan cara pemanggilannya

```
1 def uji():
2     print("Tugas Web Service")
3
4     uji()
```

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

```
1 def uji_param(nama):
2     print("Nama saya :"+str(nama))
3
4     uji_param(input("Masukan Nama Kamu : "))
```

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

```

1 def uji_return(a,b):
2     r = a + b
3     return r
4
5 a = 12
6 b = 70
7 c = uji_return(a,b)
8 print(c)

```

2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya. Untuk memudahkan dalam pemanggilan fungsi yang di butuhkan, agar dapat dipanggil berulang. Cara pemanggilannya

```

1 #from fungsi_izal import *
2 #print(penulisan(int(input("Masukan NPM kamu : "))))

```

3. Jelaskan Apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing. kelas merupakan sebuah blueprint yang merepresentasikan objek. objek adalah hasil cetakan dari sebuah kelas. method adalah suatu upaya yang digunakan oleh object.

```

1 #class Employee:
2 #     'Common base class for all employees'
3 #     empCount = 0
4
5 #     def __init__(self, name, salary):
6 #         self.name = name
7 #         self.salary = salary
8 #         Employee.empCount += 1
9
10 #     def displayCount(self):
11 #         print ("Total Employee %d" % Employee.empCount)
12
13 #     def displayEmployee(self):
14 #         print ("Name : ", self.name, ", Salary: ", self.salary)
15
16
17 #This would create first object of Employee class"
18 #emp1 = Employee("Zara", 2000)
19 #This would create second object of Employee class"
20 #emp2 = Employee("Manni", 5000)
21 #emp1.displayEmployee()
22 #emp2.displayEmployee()
23 #print ("Total Employee %d" % Employee.empCount)

```

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

- pertama import terlebih dahulu filenya.
- kemudian buat variabel untuk menampung datanya

- setelah itu panggil nama classnya dan panggil methodnya
- Gunakan perintah print untuk menampilkan hasilnya

```

1 #import belajar
2 #a = 100
3 #b = 50
4
5 #c = belajar.penambahan(a,b)
6 #print(c)

```

5. Jelaskan dengan contoh pemakaian paket dengan perintah from kalkulator import Penambahan disertai dengan contoh kode lainnya. Penggunaan paket from namafile import, itu berfungsi untuk memanggil file dan fungsinya

```

1 #from fungsi_izal import *
2 #print(penulisan(int(input("Masukan NPM kamu :"))))

```

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila le library ada di dalam folder. Pemakaian paket adalah perkumpulan fungsi-fungsi. contoh kodenya adalah sebagai berikut :

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

```

1 #Ketrampilan Pemrograman
2 #No.1
3 def jawabanNo1():
4
5     npm = input("Masukan NPM :")
6     npm = list(str(npm))
7
8     angka1 = {"0": "##### ", "1": " ##", "2": "##### ", "3": "##### ",
9               "4": "##### ", "5": "##### ", "6": "##### ",
10              "7": "##### ", "8": "##### ", "9": "##### "}
11     angka2 = {"0": "### ##", "1": "####", "2": "## ## ", "3": "### ## ",
12               "4": "## ## ", "5": "## ## ", "6": "### ## ",
13               "7": "##### ", "8": "## ## ", "9": "## ## "}
14     angka3 = {"0": "### ##", "1": "### ", "2": "### ", "3": "### ",
15               "4": "## ## ", "5": "## ", "6": "### ",
16               "7": "### ", "8": "##### ", "9": "## ## "}
17     angka4 = {"0": "### ##", "1": "### ", "2": "### ", "3": "### ",
18               "4": "##### ", "5": "##### ", "6": "##### ",
19               "7": "### ", "8": "##### ", "9": "##### "}
20     angka5 = {"0": "### ##", "1": "### ", "2": "### ", "3": "### ",
21               "4": "### ## ", "5": "### ## ", "6": "### ## ",
22               "7": "### ", "8": "### ## ", "9": "### ## "}
23     angka6 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ",
24               "4": "### ", "5": "##### ", "6": "##### ",
25               "7": "### ", "8": "##### ", "9": "##### "}
26
27     hasil1 = []
28     hasil2 = []
29     hasil3 = []
30     hasil4 = []

```

```

19     hasil5 = []
20     hasil6 = []
21
22     for x in npm:
23         hasil1.append(angka1[x])
24         hasil2.append(angka2[x])
25         hasil3.append(angka3[x])
26         hasil4.append(angka4[x])
27         hasil5.append(angka5[x])
28         hasil6.append(angka6[x])
29
30     print(*hasil1, sep=' ')
31     print(*hasil2, sep=' ')
32     print(*hasil3, sep=' ')
33     print(*hasil4, sep=' ')
34     print(*hasil5, sep=' ')
35     print(*hasil6, sep=' ')
36
37 jawabanNo1()

```

### 3.11.0.2 Ketrampilan Pemrograman

1. Buatlah fungsi dengan inputan variabel NPM, dan melakukan print luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk NPM mod 3=0, tanda pagar untuk NPM mod 3 =1, tanda plus untuk NPM mod3=2.

```

1 #Ketrampilan Pemrograman
2 #No.1
3 def jawabanNo1():
4
5     npm = input("Masukan NPM :")
6     npm = list(str(npm))
7
8     angka1 = {"0": "##### ", "1": " ##", "2": "##### ", "3": "##### ",
9               "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ",
10              "8": "##### ", "9": "##### "}
11     angka2 = {"0": "### ##", "1": "####", "2": "## ##", "3": "### ##",
12               "4": "## ##", "5": "## ##", "6": "### ##", "7": "##### ",
13               "8": "## ##", "9": "## ##"}
14     angka3 = {"0": "### ##", "1": "###", "2": "### ", "3": "### ",
15               "4": "## ##", "5": "## ", "6": "### ", "7": "### ",
16               "8": "##### ", "9": "## ##"}
17     angka4 = {"0": "### ##", "1": "###", "2": "### ", "3": "### ",
18               "4": "##### ", "5": "##### ", "6": "##### ", "7": "### ",
19               "8": "##### ", "9": "##### "}
20     angka5 = {"0": "### ##", "1": "###", "2": "### ", "3": "### ",
21               "4": "## ", "5": "## ", "6": "### ##", "7": "### ",
22               "8": "## ##", "9": "##"}
23     angka6 = {"0": "##### ", "1": "###", "2": "##### ", "3": "##### ",
24               "4": "## ", "5": "##### ", "6": "##### ", "7": "### ",
25               "8": "##### ", "9": "##### "}
26
27     hasil1 = []
28     hasil2 = []

```

```

17     hasil3 = []
18     hasil4 = []
19     hasil5 = []
20     hasil6 = []
21
22     for x in npm:
23         hasil1.append(angka1[x])
24         hasil2.append(angka2[x])
25         hasil3.append(angka3[x])
26         hasil4.append(angka4[x])
27         hasil5.append(angka5[x])
28         hasil6.append(angka6[x])
29
30     print(*hasil1, sep=' ')
31     print(*hasil2, sep=' ')
32     print(*hasil3, sep=' ')
33     print(*hasil4, sep=' ')
34     print(*hasil5, sep=' ')
35     print(*hasil6, sep=' ')
36
37     jawabanNo1()

```

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

```

1 #no.2
2 def ulang(npm):
3     hitung = 0
4     while(hitung < 70):
5         print("Halo, " + str(npm) + " Apa Kabar?")
6         hitung = hitung + 1
7     ulang(int(input("Masukkan NPM: ")))

```

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

```

1 #no.3
2 def ulang3digitakhir(npm):
3     hitung = 0
4     npm = str(npm)
5     x = npm[4:7]
6     while(hitung < 7):
7         print("Halo, " + x + " Apa Kabar?")
8         hitung = hitung + 1
9     ulang3digitakhir(int(input("Masukkan NPM: ")))

```

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

```

1 #no.4
2 def digit3daribelakang(npm):

```

```

3     npm = str(npm)
4     x = npm[-3]
5     print("Halo, " + x + " Apa Kabar?")
6 digit3daribelakang(int(input("Masukkan NPM: ")))

```

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

```

1 #no5
2 def kebawah(npm):
3     for i in npm:
4         print(i)
5
6 kebawah(input("Masukkan NPM: "))

```

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

```

1 #no.6
2 def tambah(npm):
3     jumlah = 0
4     for i in npm:
5         jumlah += int(i)
6     print("Hasil Penjumlahan NPM adalah : " + str(jumlah))
7 tambah(input("Masukkan NPM: "))

```

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

```

1 #No.7
2 def kali(npm):
3     kalikan = 0
4     for i in npm:
5         kalikan *= int(i)
6     print("Hasil Perkalian NPM adalah : " + str(kalikan))
7 kali(input("Masukkan NPM: "))

```

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

```

1 #NO.8
2 def digitgenap(npm):
3     npm = list(map(int, npm))
4     for n in npm:
5         if(n % 2 == 0):
6             if(n != 0):
7                 print(n, end = "")
8 digitgenap(input("Masukkan NPM: "))

```

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



```

1 #No.9
2 def digitganjil(npm):
3     npm = list(map(int, npm))
4     for n in npm:
5         if (n % 2 != 0):
6             print(n, end = "")
7 digitganjil(input("Masukkan NPM: "))

```

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

```

1 #No.10
2 def digitprima(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 digitprima(input("Masukkan NPM: "))

```

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

```

1 from fungsi_izal import penulisan

```

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

```

1 from fungsi_izal import penulisan
2 from 3lib import penulisan
3 print(penulisan(int(input("Masukan NPM :"))))

```

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

```

1 #def penanganan_error(a,b):
2 #     try :
3 #         c = a+b
4 #         print(c)
5 #     except TypeError:
6 #         print("We Are Different")

```

## 3.12 Muhammad Reza Syachrani / 1174084

### 3.12.1 Pemahaman Teori

1. Fungsi adalah satu blok program yang terdiri dari nama fungsi, input variabel dan variabel kembalian. Fungsi tersebut dapat digunakan ulang. Hal ini bisa dicapai dengan memberi nama pada blok statemen, kemudian nama ini dapat dipanggil di manapun dalam program.

Nama fungsi dalam Python didefinisikan menggunakan kata kunci *def*. Setelah *def* ada nama pengenalan fungsi diikuti dengan parameter yang merupakan input variable yang dapat bernilai lebih dari satu menggunakan pemisah tanda koma yang diapit oleh tanda kurung dan diakhir dengan tanda titik dua *:*. Variabel kembalian pasti satu.

Contoh Kode Pemrograman :

```
1 def hello():
2     print("Hello world")
3
4 hello()
```

**Listing 3.10** Fungsi sederhana

2. Paket atau library adalah sekumpulan kelas dan fungsi yang dibuat untuk membantu pengembang aplikasi untuk dapat membangun aplikasi dengan lebih cepat dan lebih efisien. Cara pemanggilan yaitu menggunakan perintah *import* pada file tempat digunakan library tersebut.

Contoh Kode Pemrograman :

```
1 def persegi(p,l):
2     luas = p * l #p = panjang, l = lebar
3     return luas
4
5 def lingkaran(r):
6     luas = 3.14 * (r**2) #r = jari-jari
7     return luas
8
9 def segitiga(a,t):
10    luas = (a * t)/2 #a = alas, t = tinggi
11    return luas
```

**Listing 3.11** Library atau paket luas

```
1 import luas
2
3 p = 10
4 l = 5
5 r = 15
6 a = 12
7 t = 5
8
9 jumlah1=luas.persegi(p,l)
```

```

10 jumlah2=luas.lingkaran(r)
11 jumlah3=luas.segitiga(a,t)

```

**Listing 3.12** Cara penggunaan Library luas

3. Kelas merupakan suatu blueprint atau cetakan untuk menciptakan suatu instant dari objek. ciri dari kelas itu diawali dengan kata kunci *class* dan diikuti nama kelas tersebut.

Objek adalah Semua hal yang ada dalam dunia nyata baik konkrit maupun abstrak.

Atribut merupakan nilai data yang terdapat pada suatu objek yang berasal dari kelas.

Method merupakan apa saja yang dapat dilakukan / dialami oleh suatu objek  
Contoh Kode Program :

```

1 class Menghitung:
2     def __init__(self,p,l,r,a,t):
3         self.p = p
4         self.l = l
5         self.r = r
6         self.a = a
7         self.t = t
8
9     def persegi(self):
10        luas = self.p * self.l #p = panjang, l = lebar
11        return luas
12
13    def lingkaran(self):
14        luas = 3.14 * (self.r**2) #r = jari-jari
15        return luas
16
17    def segitiga(self):
18        luas = (self.a * self.t)/2 #a = alas, t = tinggi
19        return luas

```

**Listing 3.13** Kelas Library Menghitung

4. Untuk pemanggilan kelas library menggunakan kata kunci *import* diikuti nama kelas library tersebut. sedangkan cara pemakain untuk membuat objek dari sebuah kelas, kita bisa memanggil nama kelas dengan argumen sesuai dengan fungsi *init* pada saat kita mendefinisikannya. contohnya menggunakan kelas library menghitung :

```

1 import menghitung
2
3 p = 10
4 l = 5
5 r = 15
6 a = 12
7 t = 5
8 hitung = menghitung.Menghitung(p,l,r,s,a,t)
9

```

```

10 jumlah1=hitung.persegi()
11 jumlah2=hitung.lingkaran()
12 jumlah3=hitung.segitiga()

```

**Listing 3.14** Cara penggunaan kelas Library Menghitung

5. Penggunaan paket from nama file import nama kelas library berfungsi untuk memanggil file dan fungsi yang terdapat pada kelas library. contoh penggunaan pada kode program lain :

```

1 from menghitung import Menghitung
2
3 p = 10
4 l = 5
5 r = 15
6 a = 12
7 t = 5
8
9 hitung = Menghitung(p,l,r,a,t)
10
11 jumlah1=hitung.persegi()
12 jumlah2=hitung.lingkaran()
13 jumlah3=hitung.segitiga()

```

**Listing 3.15** Contoh kode lain pemakaian paket fungsi from import

6. Contoh kodenya pemakaian paket fungsi apabila file library ada di dalam folder.

```

1 from folder import kalkulator
2
3 a=100
4 b=50
5
6 hasil1=kalkulator.Penambahan(a,b)
7 hasil2=kalkulator.Pengurangan(a,b)
8 hasil3=kalkulator.Perkalian(a,b)
9 hasil4=kalkulator.Pembagian(a,b)
10
11 print(hasil1)
12 print(hasil2)
13 print(hasil3)
14 print(hasil4)

```

**Listing 3.16** Contoh kode pemakaian paket fungsi dimana file library ada di dalam folder

7. Contoh kode pemakaian paket kelas apabila file library ada di dalam folder. Berikut ini adalah pemakaian paket kelas apabila file library ada di dalam folder.

```

1 from folder import ngitung
2
3 a=100
4 b=50
5
6 hitung = ngitung.Ngitung ( a , b )
7
8 hasil1=hitung.Penambahan ( )

```

```

9  hasil2=hitung.Pengurangan( )
10 hasil3=hitung.Perkalian( )
11 hasil4=hitung.Pembagian( )
12
13 print(hasil1)
14 print(hasil2)
15 print(hasil3)
16 print(hasil4)

```

**Listing 3.17** Contoh kode pemakaian paket kelas dimana file library ada di dalam folder

### 3.12.2 Ketrampilan Pemrograman

#### 1. Jawaban no. 1

```

1  def NPM(npm):
2      npm = list(str(npm))
3
4      angka1 = {"0": "***** ", "1": " **", "2": " ***** ", "3": "
***** ", "4": " ****", "5": " *****", "6": " ***** ", "
7": " *****", "8": " *****", "9": " *****"}
5      angka2 = {"0": "*** **", "1": "****", "2": "*** **", "3": "
** **", "4": " ****", "5": " **", "6": " *** ", "
7": " *** ", "8": " ****", "9": " *** **"}
6      angka3 = {"0": "**** **", "1": " ****", "2": " ****", "3": "
**** ", "4": " ****", "5": " *****", "6": " *****", "
7": " **** ", "8": " ****", "9": " **** **"}
7      angka4 = {"0": "**** **", "1": " ****", "2": " **** ", "3": "
**** ", "4": " *****", "5": " ****", "6": " **** **", "
7": " **** ", "8": " **** **", "9": " *****"}
8      angka5 = {"0": "**** **", "1": " ****", "2": " **** ", "3": "
** ****", "4": " ****", "5": " ****", "6": " **** **", "
7": " **** ", "8": " ****", "9": " **** **"}
9      angka6 = {"0": "***** ", "1": " ****", "2": " *****", "3": "
***** ", "4": " ****", "5": " *****", "6": " *****", "
7": " **** ", "8": " *****", "9": " *****"}
10
11  hasil1 = []
12  hasil2 = []
13  hasil3 = []
14  hasil4 = []
15  hasil5 = []
16  hasil6 = []
17
18  for x in npm:
19      hasil1.append(angka1[x])
20      hasil2.append(angka2[x])
21      hasil3.append(angka3[x])
22      hasil4.append(angka4[x])
23      hasil5.append(angka5[x])
24      hasil6.append(angka6[x])
25
26  print(*hasil1, sep=' ')
27  print(*hasil2, sep=' ')
28  print(*hasil3, sep=' ')
29  print(*hasil4, sep=' ')

```

```

30     print(*hasil5 , sep=' ')
31     print(*hasil6 , sep=' ')
32
33 NPM(input("Masukan NPM anda: "))

```

## 2. Jawaban no. 2

```

1 def NPMDuaDijit(npm):
2     ulang = 1
3     sampai = int(npm[5:7])
4     while(ulang <= sampai):
5         print("Halo, "+str(npm)+" apa kabar?")
6         ulang += 1
7
8 NPMDuaDijit(input("Masukan NPM anda: "))

```

## 3. Jawaban no. 3

```

1 def NPMTigaDijit(npm):
2     ulang = 1
3     sampai = list(map(int , npm[4:7]))
4     sampai = sum(sampai)
5     while(ulang <= sampai):
6         print("Halo, "+str(npm[-3:])+ " apa kabar?")
7         ulang += 1
8
9 NPMTigaDijit(input("Masukan NPM anda: "))

```

## 4. Jawaban no. 4

```

1 def NPMDigitKetiga(npm):
2     print("Output:")
3     print("Halo, "+str(npm[-3])+ " apa kabar?")
4
5 NPMDigitKetiga(input("Masukan NPM anda: "))

```

## 5. Jawaban no. 5

```

1 def NPMSatuPersatu(npm):
2     npm = list(map(int , npm))
3     for n in npm:
4         print(n)
5
6 NPMSatuPersatu(input("Masukan NPM anda: "))

```

## 6. Jawaban no. 6

```

1 def penjumlahanNPM(npm):
2     npm = list(map(int , npm))
3     hasil = 0
4     for x in npm:
5         hasil += x
6     print(hasil)
7
8 penjumlahanNPM(input("Masukan NPM anda: "))

```

## 7. Jawaban no. 7

```
1 def perkalianNPM(npm):
2     npm = list(map(int, npm))
3     hasil = 0
4     for x in npm:
5         hasil *= x
6     print(hasil)
7
8 perkalianNPM(input("Masukan NPM anda: "))
```

## 8. Jawaban no. 8

```
1 def genap(npm):
2     npm = list(map(int, npm))
3     for n in npm:
4         if (n % 2 == 0):
5             if (n != 0):
6                 print(n, end = "")
7     genap(input("Masukan NPM anda: "))
```

## 9. Jawaban no. 9

```
1 def ganjil(npm):
2     npm = list(map(int, npm))
3     for n in npm:
4         if (n % 2 != 0):
5             print(n, end = "")
6     ganjil(input("Masukan NPM anda: "))
```

## 10. Jawaban no. 10

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

## 11. Jawaban no. 11

```
1 lib = __import__('3lib')
2
3 npm = "1174084"
4
5 lib.NPM(npm)
```

```

6 lib . NPMDuaDijit (npm)
7 lib . NPMTigaDijit (npm)
8 lib . NPMDigitKetiga (npm)
9 lib . NPMSatuPersatu (npm)
10 lib . penjumlahanNPM (npm)
11 lib . perkalianNPM (npm)
12 lib . genap (npm)
13 lib . ganjil (npm)
14 lib . prima (npm)

```

## 12. Jawaban no. 12

```

1 from kelas3lib import kelas3lib
2
3 npm = "1174084"
4
5 k3lib = kelas3lib (npm)
6
7 k3lib .NPM()
8 k3lib .NPMDuaDijit ()
9 k3lib .NPMTigaDijit ()
10 k3lib .NPMDigitKetiga ()
11 k3lib .NPMSatuPersatu ()
12 k3lib .penjumlahanNPM ()
13 k3lib .perkalianNPM ()
14 k3lib .genap ()
15 k3lib .ganjil ()
16 k3lib .prima ()

```

### 3.12.3 Ketrampilan Penanganan Error

1. Peringatan error yang ditemukan dan penjelasannya serta buat sebuah fungsi try except untuk menanggulangi error. Peringatan error di praktek ketiga ini, yaitu:
  - **Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
  - **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 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 terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.



Contoh fungsi yang menggunakan try except :

```

1 def siapa(nama):
2     try:
3         print("Hallo, "+str(nama))
4     except:
5         print("Terjadi error")
6
7 siapa(input("Masukan nama anda: "))

```

### 3.13 Difa Al Fansha

#### 3.13.1 Jawaban Teori

1. Fungsi adalah satu blok program yang terdiri dari nama fungsi, input variabel dan variabel kembalian.
2. Paket atau library adalah kumpulan dari fungsi-fungsi.  
Contoh Nomor 1,2 dan 3

```

1 # Nomor 1 & 2
2 def Penambahan(a,b):
3     z = a + b
4     return z
5 def Pengurangan(a,b):
6     z = a - b
7     return z
8 def Perkalian(a,b):
9     z = a * b
10    return z
11 def Pembagian(a,b):
12    z = a / b
13    return z

```

3. Pengertian Kelas, Objek, Atribut, dan Method.

- Kelas : Cetak biru dari objek.
- Objek : Segala sesuatu dapat dipanggil objek.
- Atribut : Segala sesuatu yang melekat pada objek.
- Method : Perilaku dari objek, biasa disebut function.

4. Cara memanggil Library

- mengimport library.
- File harus dalam satu folder.

5. pemakaian paket dengan perintah from kalkulator import Penambahan
6. pemakaian fungsi paket function
7. pemakaian fungsi paket function  
Contoh Nomor 4,5,6 dan 7

```

1 # Nomor 3, 4, 5, 6 dan 7 TEORI
2 class Npm :
3     from kalkulator import Penambahan, Pengurangan, Perkalian,
4     Pembagian
5     a = 20
6     b = 10
7     hasil1 = Penambahan(a,b)
8     hasil2 = Pengurangan(a,b)
9     hasil3 = Perkalian(a,b)
10    hasil4 = Pembagian(a,b)
11    print(hasil1)
12    print(hasil2)
13    print(hasil3)
14    print(hasil4)

```

### 3.13.2 Jawaban Praktek

#### 1. Inputan variabel NPM dengan tanda pagar

```

1 # Nomor 1 Praktek
2 print("Nomor 1 Praktek")
3
4 def NPM(npm):
5     npm = list(str(npm))
6
7     angka1 = {"0": "##### ", "1": " ##", "2": "##### ", "3": "##### ",
8     "4": "      ##", "5": "##### ", "6": "##### ", "7": "##### ",
9     "8": "##### ", "9": "##### "}
10    angka2 = {"0": "### ##", "1": "####", "2": "## ##", "3": "## ##",
11    "4": "##### ", "5": "## ", "6": "### ", "7": "## ",
12    "8": "### ##", "9": "## ##"}
13    angka3 = {"0": "### ##", "1": "## ", "2": "## ", "3": "##### ",
14    "4": "## ## ", "5": "##### ", "6": "##### ", "7": "## ",
15    "8": "## ## ", "9": "## ##"}
16    angka4 = {"0": "### ##", "1": "## ", "2": "## ", "3": "##### ",
17    "4": "##### ", "5": "## ", "6": "### ##", "7": "## ",
18    "8": "## ## ", "9": "##### "}
19    angka5 = {"0": "### ##", "1": "## ", "2": "## ", "3": "## ",
20    "4": "## ", "5": "## ##", "6": "### ##", "7": "## ",
21    "8": "## ## ", "9": "##"}
22    angka6 = {"0": "##### ", "1": "## ", "2": "##### ", "3": "##### ",
23    "4": "      ##", "5": "##### ", "6": "##### ", "7": "## ",
24    "8": "##### ", "9": "##### "}
25
26    hasil1 = []
27    hasil2 = []
28    hasil3 = []
29    hasil4 = []
30    hasil5 = []
31    hasil6 = []
32
33    for x in npm:
34        hasil1.append(angka1[x])
35        hasil2.append(angka2[x])
36        hasil3.append(angka3[x])
37        hasil4.append(angka4[x])
38        hasil5.append(angka5[x])
39        hasil6.append(angka6[x])

```

```

25     hasil4.append(angka4[x])
26     hasil5.append(angka5[x])
27     hasil6.append(angka6[x])
28
29     print(*hasil1, sep=' ')
30     print(*hasil2, sep=' ')
31     print(*hasil3, sep=' ')
32     print(*hasil4, sep=' ')
33     print(*hasil5, sep=' ')
34     print(*hasil6, sep=' ')
35
36 NPM(input("Masukan NPM anda: "))

```

## 2. Perulangan NPM sampai 87 kali

```

1 # Nomor 2 Praktek
2 print("Nomor 2 Praktek")
3
4 def NPM(npm):
5     i = 1
6     x = int(npm[5:7])
7     while(i <= x):
8         print("Halo, "+str(npm)+" apa kabar?")
9         i += 1
10
11 NPM(input("Masukan NPM anda: "))

```

## 3. Perulangan berupa tiga karakter belakang dari NPM sebanyak penjumlahan tiga dijit

```

1 # Nomor 3 Praktek
2 print("Nomor 3 Praktek")
3
4 def NPM(npm):
5     i = 1
6     x = list(map(int, npm[4:7]))
7     x = sum(x)
8     while(i <= x):
9         print("Halo, "+str(npm[-3:])+ " apa kabar?")
10        i += 1
11
12 NPM(input("Masukan NPM anda: "))

```

## 4. fungsi hello word dengan input variabel string

```

1 # Nomor 4 Praktek
2 print("Nomor 4 Praktek")
3
4 def NPM(npm):
5     print("Halo, "+str(npm[-3])+ " apa kabar?")
6
7 NPM(input("Input: "))

```

## 5. Mengurutkan NPM

```

1 # Nomor 5 Praktek
2 print("Nomor 5 Praktek")
3
4 def NPM(npm):
5     npm = list(map(int, npm))
6     for n in npm:
7         print(n)
8
9 NPM(input("Masukan NPM anda: "))

```

6. Fungsi dengan inputan variabel NPM, didalamnya melakukan penjumlahan dari seluruh digit NPM tersebut

```

1 # Nomor 6 Praktek
2 print("Nomor 6 Praktek")
3
4 def NPM(npm):
5     npm = list(map(int, npm))
6     hasil = 0
7     for x in npm:
8         hasil += x
9
10    print(hasil)
11
12 NPM(input("Masukan NPM anda: "))

```

7. fungsi dengan inputan variabel NPM, didalamnya melakukan melakukan perkalian dari seluruh digit NPM

```

1 # Nomor 6 Praktek
2 print("Nomor 6 Praktek")
3
4 def NPM(npm):
5     npm = list(map(int, npm))
6     hasil = 0
7     for x in npm:
8         hasil += x
9
10    print(hasil)
11
12 NPM(input("Masukan NPM anda: "))
13
14 # Nomor 7 Praktek
15 print("Nomor 7 Praktek")
16
17 def NPM(npm):
18     npm = list(map(int, npm))
19     hasil = 0
20     for x in npm:
21         hasil *= x
22
23    print(hasil)
24
25 NPM(input("Masukan NPM anda: "))

```

8. fungsi dengan inputan variabel NPM, Lakukan print NPM anda tapi hanya dijit genap saja

```

1 # Nomor 8 Praktek
2 print("Nomor 8 Praktek")
3
4 def NPM(npm):
5     npm = list(map(int, npm))
6     for n in npm:
7         if(n % 2 == 0):
8             if(n != 0):
9                 print(n, end = "")
10
11 NPM(input("Masukan NPM anda: "))
12 print("")

```

9. fungsi dengan inputan variabel NPM, Lakukan print NPM anda tapi hanya dijit ganjil saja

```

1 #Nomor 9 Praktek
2 print("Nomor 9 Praktek")
3
4 def NPM(npm):
5     npm = list(map(int, npm))
6     for n in npm:
7         if(n % 2 != 0):
8             print(n, end = "")
9
10 NPM(input("Masukan NPM anda : "))
11 print("")

```

10. fungsi dengan inputan variabel NPM, Lakukan print NPM anda tapi hanya dijit yang termasuk bilangan prima

```

1 # Nomor 10 Praktek
2 print("Nomor 10 Praktek")
3
4 def NPM(npm):
5     npm = list(map(int, npm))
6     prima = []
7     for n in npm:
8         isPrime = True
9         if n == 0 or n == 1:
10             isPrime = False
11         for x in range(2, n):
12             if n % x == 0:
13                 isPrime = False
14         if isPrime:
15             prima.append(n)
16
17     for p in prima:
18         print(p, end = "")
19
20 NPM(input("Masukan NPM anda: "))

```

## 3.14 Alfadian Owen

### 3.14.1 Teori

1. Apa itu fungsi, inputan fungsi dan kembalian fungsi dengan contoh kode program lainnya! fungsi adalah sebuah construct untuk menyusun program. Fungsi digunakan untuk memanfaatkan kode lebih dari satu tempat dalam suatu program.

```

1 def fungsi(a,b):
2     c=a+b
3     return c
4
5 a="nama saya "
6 b="owen"
7
8 jadi=fungsi(a,b)

```

2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya! paket adalah kumpulan fungsi-fungsi di dalam sebuah file. cara pemanggilannya yaitu dengan cara "import"

```

1 def penambahan(x,y):
2     hasil=x+y
3     return hasil
4 def pengurangan(x,y):
5     hasil=x-y
6     return hasil
7 def pembagian(x,y):
8     hasil=x/y
9     return hasil
10 def perkalian(x,y):
11     hasil=x*y
12     return hasil

```

```

1 import p_1174091
2
3 x=100
4 y=1000
5
6 perkalian=p_1174091.perkalian(x,y)

```

3. Jelaskan Apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing! kelas adalah blueprint untuk menciptakan instant dari objek, objek adalah instance dari class secara umum merepresentasikan sebuah objek. attribute adalah nilai yang dimiliki objek. Terdapat penambahan, pengurangan perkalian, perkalian, pembagian, modulus

```

1 class aritmatika:
2     def __init__(self,a,b):
3         self.a=a
4         self.b=b
5     def tambah(self):

```

```

6         c=self.a+self.b
7         return c
8     def kurang(self):
9         c=self.a-self.b
10        return c
11    def kali(self):
12        c=self.a*self.b
13        return c
14    def bagi(self):
15        c=self.a/self.b
16        return c

```

4. Jelaskan cara pemanggilan library kelas dari instansiasi dan pemakaiannya dengan contoh program lainnya! pertama buat file kelas, setelah itu buat file yang isinya untuk memanggil method” yang ada di kelas

```

1 class aritmatika:
2     def __init__(self,a,b):
3         self.a=a
4         self.b=b
5     def tambah(self):
6         c=self.a+self.b
7         return c
8     def kurang(self):
9         c=self.a-self.b
10        return c
11    def kali(self):
12        c=self.a*self.b
13        return c
14    def bagi(self):
15        c=self.a/self.b
16        return c

```

```

1 import c_1174091 as kelas
2
3 a=25
4 b=5
5
6 hitung=kelas.aritmatika(a,b)
7
8 hasil1 = hitung.tambah()
9 hasil2 = hitung.kurang()

```

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

```

1 from kalkulator_1174091 import tambah
2
3 hasil=tambah(a,b)

```

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila file library ada di dalam folder! pertama buat file kelas, setelah itu import fungsi nya kedalam file lain

```

1 def tambah(a,b):
2     c=a+b
3     return c
4 def kurang(a,b):
5     c=a-b
6     return c
7 def kali(a,b):
8     c=a*b
9     return c
10 def bagi(a,b):
11     c=a/b
12     return c

```

```

1 from kalkulator_1174091 import tambah
2
3 hasil=tambah(a,b)

```

7. Jelaskan dengan contoh kodenya, pemakaian paket kelas apabila file library ada di dalam folder! buat paket kelas terlebih dahulu, setelah itu import file kelas kedalam file lain.

```

1 class aritmatika:
2     def __init__(self,a,b):
3         self.a=a
4         self.b=b
5     def tambah(self):
6         c=self.a+self.b
7         return c
8     def kurang(self):
9         c=self.a-self.b
10        return c
11    def kali(self):
12        c=self.a*self.b
13        return c
14    def bagi(self):
15        c=self.a/self.b
16        return c

```

```

1 import c_1174091 as kelas
2
3 a=25
4 b=5
5
6 hitung=kelas.aritmatika(a,b)
7
8 hasil1 = hitung.tambah()
9 hasil2 = hitung.kurang()

```

### 3.14.2 Praktek

1. Jawaban soal no 1

```

1     print("hallo " + npm + " apa kabar")
2 loop("1174091")

```



```
3 #yabg atas jawaban no 2
4 #bawah jawaban no 3
5 npm=1174091
6 def loop(npm):
7     for x in range(15):
8         print("hallo " + str(npm)[4:7] + " apa kabar")
9
10 loop(npm)
11
12 #bawah jawaban no 4
```

## 2. Jawaban soal no 2

```
1 def loop(npm = "1174091"):
2     for x in range(87):
3         print("hallo " + npm + " apa kabar")
4 loop("1174091")
5 #yabg atas jawaban no 2
```

## 3. Jawaban soal no 3

```
1 npm=1174091
2 def loop(npm):
3     for x in range(15):
4         print("hallo " + str(npm)[4:7] + " apa kabar")
5
6 loop(npm)
```

## 4. Jawaban soal no 4

```
1 def helloworld(npm):
2     npm="1174091"
3     print("halo " + str(npm)[4] + " apa kabar")
4 helloworld(npm)
```

## 5. Jawaban soal no 5

```
1 def test(npm):
2     npm = ['1', '1', '7', '4', '0', '9', '1']
3     for x in npm:
4         print(x)
5 test(npm)
```

## 6. Jawaban soal no 6

```
1 def jmlnpm(npm):
2     npm = [1,1,7,4,0,9,1]
3     hasil = 0
4     for x in npm:
5         hasil += x
6
7     print(hasil)
8 jmlnpm(npm)
```

## 7. Jawaban soal no 7

```

1 def kalinpm(npm):
2     npm = [1,1,7,4,0,9,1]
3     hasil = 0
4     for x in npm:
5         hasil *= x
6
7     print(hasil)
8 kalinpm(npm)

```

#### 8. Jawaban soal no 8

```

1 def npmgenap(npm):
2     npm = [1,1,7,4,0,9,1]
3     for num in npm:
4         if (num % 2 == 0):
5             print(num, end=" ")
6 npmgenap(npm)

```

#### 9. Jawaban soal no 9

```

1 def helloworld(npm):
2     npm="1174091"
3     print("halo " + str(npm)[4] + " apa kabar")
4 helloworld(npm)

```

#### 10. Jawaban soal no 10

```

1 def helloworld(npm):
2     npm="1174091"
3     print("halo " + str(npm)[4] + " apa kabar")
4 helloworld(npm)

```

#### 11. Jawaban soal no 11

```

1 def helloworld(npm):
2     npm="1174091"
3     print("halo " + str(npm)[4] + " apa kabar")
4 helloworld(npm)

```

### 3.15 Muhammad Abdul Gani Wijaya

#### 3.15.0.1 Teori

##### 1. Apa itu fungsi, inputan

fungsi dan kembalian fungsi dengan contoh kode program lainnya. Fungsi adalah bagian dari program yang dapat digunakan ulang. Berikut merupakan contoh fungsi dan cara pemanggilannya

```

1 def uji():
2     print("Tugas Web Service")
3
4 uji()

```

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

```
1 def uji_param(nama):
2     print("Nama saya :"+str(nama))
3
4 uji_param(input("Masukan Nama Kamu : "))
```

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

```
1 def uji_return(a,b):
2     r = a + b
3     return r
4
5 a = 10
6 b = 50
7 c = uji_return(a,b)
8 print(c)
```

2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya. Untuk memudahkan dalam pemanggilan fungsi yang dibutuhkan, agar dapat dipanggil berulang. Cara pemanggilannya

```
1 #from fungsi_felix import *
2 #print(penulisan(int(input("Masukan NPM kamu : ")))))
```

3. Jelaskan Apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing. kelas merupakan sebuah blueprint yang merepresentasikan objek. objek adalah hasil cetakan dadri sebuah kelas. method adalah suatu upaya yang digunakan oleh object.

```
1 #class Employee:
2 #     'Common base class for all employees'
3 #     empCount = 0
4
5 #     def __init__(self, name, salary):
6 #         self.name = name
7 #         self.salary = salary
8 #         Employee.empCount += 1
9
10 #     def displayCount(self):
11 #         print ("Total Employee %d" % Employee.empCount)
12
13 #     def displayEmployee(self):
14 #         print ("Name : ", self.name, ", Salary: ", self.salary)
15
16
17 #This would create first object of Employee class"
18 #emp1 = Employee("Zara", 2000)
19 #This would create second object of Employee class"
20 #emp2 = Employee("Manni", 5000)
21 #emp1.displayEmployee()
22 #emp2.displayEmployee()
23 #print ("Total Employee %d" % Employee.empCount)
```

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

- pertama import terlebih dahulu filenya.
- kemudian buat variabel untuk menampung datanya
- setelah itu panggil nama classnya dan panggil methodnya
- Gunakan perintah print untuk menampilkan hasilnya

```

1 import belajar
2 #a = 100
3 #b = 50
4
5 #c = belajar.penambahan(a,b)
6 #print(c)

```

5. Jelaskan dengan contoh pemakaian paket dengan perintah from kalkulator import Penambahan disertai dengan contoh kode lainnya. Penggunaan paket from namafile import, itu berfungsi untuk memanggil file dan fungsinya

```

1 #from fungsi_felix import *
2 #print(penulisan(int(input("Masukan NPM kamu : "))))

```

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila le library ada di dalam folder. Pemakaian paket adalah perkumpulan fungsi-fungsi. contoh kodenya adalah sebagai berikut :
7. Jelaskan dengan contoh kodenya, pemakaian paket kelas apabila le library ada di dalam folder.

### 3.15.0.2 Ketampilan Pemrograman

1. Buatlah fungsi dengan inputan variabel NPM, dan melakukan print luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk NPM mod 3=0, tanda pagar untuk NPM mod 3 =1, tanda plus untuk NPM mod3=2.

```

1
2 #No 1
3 def speakup(npm):
4     npm = list(str(npm))
5
6     angka1 = {"0": "***** ", "1": "*** ", "2": "***** ", "3": "***** ", "4": "***** ", "5": "***** ", "6": "***** ", "7": "***** ", "8": "***** ", "9": "***** "}
7     angka2 = {"0": "***** ", "1": "*** ", "2": "***** ", "3": "***** ", "4": "***** ", "5": "***** ", "6": "***** ", "7": "***** ", "8": "***** ", "9": "***** "}
8     angka3 = {"0": "***** ", "1": "*** ", "2": "***** ", "3": "***** ", "4": "***** ", "5": "***** ", "6": "***** ", "7": "***** ", "8": "***** ", "9": "***** "}

```



```
51 speakup(int(input("Masukan NPM : 117")))
```

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

```
1 def looping(npm):
2     hitung = 0
3     while(hitung < 71):
4         print("Haiiii , "+str(npm)+" apa kabar?")
5         hitung = hitung +1
6
7 looping(int(input("Masukan NPM : ")))
```

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

```
1 #No 3
2 def looping_3_digit(npm):
3     hitung = 0
4     npm = str(npm)
5     bil = npm[4:7]
6
7     while(hitung < 8):
8         print("Halo , "+bil+" apa kabar?")
9         hitung = hitung +1
10
11 looping_3_digit(int(input("Masukan NPM : ")))
```

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

```
1 #No 4
2 def looping_3_digit_terakhir(npm):
3     npm = str(npm)
4     bil = npm[-3]
5     print("Haiii , "+bil+" apa kabar?")
6
7 looping_3_digit_terakhir(int(input("Masukan NPM : ")))
```

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

```
1 #No 5
2 def down(npm):
3     for i in npm:
4         print (i)
```

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

```

1
2 #No 6
3 def addition(npm):
4     jumlah = 0
5     for i in npm:
6         jumlah += int(i)
7     print(str(jumlah)+" Adalah hasil penjumlahan")

```

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

```

1
2 #No 7
3 def multiplication(npm):
4     jumlah = 0
5     for i in npm:
6         jumlah *= int(i)
7     print(str(jumlah)+" Adalah hasil perkalian")

```

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

```

1
2 #No 8
3 def even():
4     npm = [1,1,7,4,0,7,1]
5     for i in npm:
6         if (i % 2) == 0:
7             print("Bilangan Genapnya : "+str(i))

```

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

```

1 even()
2
3 #No 9
4 def odd():
5     npm = [1,1,7,4,0,7,1]
6     for i in npm:
7         if (i%2)==1:

```

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

```

1
2 odd()
3
4 #No 10
5 def prime(npm):
6     npm = str(npm)
7     bil = npm[2]
8     num = int(bil)

```

```

9     if num > 1:
10         for i in range(2,num):
11             if (num%i)==0:
12                 print("Bukan Bilangan Prima")
13                 break
14             else:
15                 print("Bilangan Primanya :"+str(num))

```

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

```

1 from fungsi_gani import learn

```

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

```

1 from 3lib_gani import learn

```

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

```

1 #def penanganan_error(a,b):
2 #     try :
3 #         c = a+b
4 #         print(c)
5 #     except TypeError:
6 #         print("kita beda")

```

## 3.16 Ilham Muhammad Ariq D4TI2C 1174087

### 3.16.1 Pemahaman Teori

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

Fungsi adalah satu blok program yang terdiri dari nama fungsi, input variabel dan variabel kembalian. Berikut merupakan contoh fungsi :

```

1 def cek(a,b):
2     c = a + b
3     return c
4

```



```

5 a = 1
6 b = 2
7 hasil = cek(a,b)
8 print(hasil)

```

- **cek** merupakan nama fungsi
  - **a dan b** merupakan inputan fungsi yang mana adalah inputan yang akan diproses program
  - **c** merupakan hasil dari kembalian fungsi tersebut
  - **hasil** merupakan output program
2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya.

Paket merupakan kumpulan dari beberapa fungsi untuk mempermudah pemanggilan fungsi cara pemanggilannya :

```

1 import paket
2
3 a = 1
4 b = 2
5
6 hasil =paket.penambahan(a,b)
7 print(hasil)

```

- **import** merupakan cara pemanggilan paket tersebut
3. Jelaskan apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing.

```

1 class Mahasiswa:
2     def __init__(self,nama):
3         self.nama = nama
4     def biodata(self):
5         print("Nama saya", self.nama)
6
7 NamaMhs = Mahasiswa("Ariq")
8 NamaMhs.biodata()

```

- **Mahasiswa** = kelas adalah sebuah blueprint yang mepresentasikan objek.
  - **self.nama** = atribut adalah variabel yang menyimpan data.
  - **NamaMhs** = objek adalah perwujudan dari sebuah kelas.
  - **Biodata** = method adalah suatu tindakan yang digunakan oleh object.
4. Jelaskan cara pemanggikan library kelas dari instansiasi dan pemakaiannya dengan contoh program lainnya.

```

1 import kelas3lib
2
3 NamaMhs = kelas3lib.Mahasiswa("Ariq")
4 NamaMhs.biodata()

```

- pertama import terlebih dahulu filenya.
- kemudian buat variabel untuk nenampung nilai data nya
- setelah itu panggil nama class dan methodnya

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

Berikut cara pemakaiannya,

```
1 from kalkulator import Penambahan
2 a = 2
3 b = 3
4 hasil = Penambahan(a, b)
5 print(hasil)
```

- import paketnya
- panggil fungsi Penambahan

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila le librar yada di dalam folder.

Pemanggilan fungsi-fungsi pada suatu file, berikut kodenya :

```
1 from namafolder.namafile import namafungsi
2
3 a=10
4 b=5
5
6 hasil=namafungsi.Penambahan(a,b)
7 print(hasil)
```

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

Pemanggilan nama kelas pada suatu file, berikut kodenya :

```
1 from namafolder.namafile import namapaket
2
3 NamaMhs = namapaket.Mahasiswa("Ariq")
4 NamaMhs.biodata()
```

### 3.16.2 Ketrampilan Pemrograman

1. Buatlah fungsi dengan inputan variabel NPM, dan melakukan print luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk NPM mod 3=0, tanda pagar untuk NPM mod 3 =1, tanda plus untuk NPM mod3=2.

```

1 def cetaknpm (npm):
2
3     npm = list(str(npm))
4
5     angka1 = {"0": "##### ", "1": "  ##", "2": "#####", "3": "#####",
6               "4": "      ##", "5": "#####", "6": "#####", "7": "#####",
7               "8": "#####", "9": "#####"}
8     angka2 = {"0": "###  ##", "1": "####", "2": "      ##", "3": "      ##",
9               "4": "#####", "5": "##      ", "6": "###      ", "7": "      ##",
10              "8": "###  ##", "9": "###  ##"}
11     angka3 = {"0": "###  ##", "1": "  ##", "2": "      ##", "3": "#####",
12               "4": "  ##  ##", "5": "#####", "6": "#####", "7": "      ##",
13               "8": "#####", "9": "###  ##"}
14     angka4 = {"0": "###  ##", "1": "  ##", "2": "      ##", "3": "#####",
15               "4": "#####", "5": "#####", "6": "###  ##", "7": "      ##",
16               "8": "#####", "9": "#####"}
17     angka5 = {"0": "###  ##", "1": "  ##", "2": "      ##", "3": "      ##",
18               "4": "      ##", "5": "      ##", "6": "      ##", "7": "      ##",
19               "8": "      ##", "9": "      ##"}
20     angka6 = {"0": "##### ", "1": "  ##", "2": "#####", "3": "#####",
21               "4": "      ##", "5": "#####", "6": "#####", "7": "      ##",
22               "8": "#####", "9": "#####"}
23
24     hasil1 = []
25     hasil2 = []
26     hasil3 = []
27     hasil4 = []
28     hasil5 = []
29     hasil6 = []
30
31     for x in npm:
32         hasil1.append(angka1[x])
33         hasil2.append(angka2[x])
34         hasil3.append(angka3[x])
35         hasil4.append(angka4[x])
36         hasil5.append(angka5[x])
37         hasil6.append(angka6[x])
38
39     print(*hasil1, sep=' ')
40     print(*hasil2, sep=' ')
41     print(*hasil3, sep=' ')
42     print(*hasil4, sep=' ')
43     print(*hasil5, sep=' ')
44     print(*hasil6, sep=' ')
45
46 cetaknpm(input("Masukan NPM anda: "))

```

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

```

1 def cetaknpm2digit(npm):
2     i = 1
3     while(i <= 87):
4         print("Hallo , "+npm[4:8]+" apa kabar?")
5         i += 1

```

```

6 print("Masukan Npm: ")
7 npm = input()
8 print(cetaknpm2dijit(npm))

```

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

```

1 def cetaknpm2dijit(npm):
2     i = 1
3     sampai = list(map(int, npm[4:8]))
4     sampai = sum(sampai)
5     while(i <= sampai):
6         print("Halo, "+str(npm)+" apa kabar?")
7         i += 1
8     print("Masukan Npm: ")
9     npm = input()
10    print(cetaknpm2dijit(npm))

```

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

```

1 def cetaknpmdijitke3(npm):
2     print("Halo, "+str(npm[-3])+" apa kabar?")
3     print("Masukan Npm: ")
4     npm = input()
5     print(cetaknpmdijitke3(npm))

```

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

```

1 def cetaknpmsatupersatu(npm):
2     npm = list(map(int, npm))
3     for n in npm:
4         print(n)
5     print("Masukan Npm: ")
6     npm = input()
7     print(cetaknpmsatupersatu(npm))

```

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

```

1 def cetakpenjumlahannpm(npm):
2     npm = list(map(int, npm))
3     hasil = 0
4     for x in npm:
5         hasil += x
6
7     print(hasil)
8     print("Masukan Npm: ")
9     npm = input()
10    print(cetakpenjumlahannpm(npm))

```

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

```

1 def cetakperkaliannpm(npm):
2     npm = list(map(int, npm))
3     hasil = 0
4     for x in npm:
5         hasil *= x
6
7     print(hasil)
8 print("Masukan Npm: ")
9 npm = input()
10 print(cetakperkaliannpm(npm))

```

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

```

1 def cetaknpmgenap(npm):
2     npm = list(map(int, npm))
3     for n in npm:
4         if(n % 2 == 0):
5             if(n != 0):
6                 print(n, end = "")
7 print("Masukan Npm: ")
8 npm = input()
9 print(cetaknpmgenap(npm))

```

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

```

1 def cetaknpmganjil(npm):
2     npm = list(map(int, npm))
3     for n in npm:
4         if(n % 2 != 0):
5             print(n, end = "")
6 print("Masukan Npm: ")
7 npm = input()
8 print(cetaknpmganjil(npm))

```

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

```

1 def cetaknpmprima(npm):
2     npm = list(map(int, npm))
3     prima = []
4     for n in npm:
5         isPrime = True
6         if n == 0 or n == 1:
7             isPrime = False
8         for x in range(2, n):
9             if n % x == 0:
10                isPrime = False

```

```

11         if isPrime:
12             prima.append(n)
13
14         for p in prima:
15             print(p, end = "")
16     print("Masukan Npm: ")
17     npm = input()
18     print(cetaknpmprima(npm))

```

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

```

1  #Teori
2  #No 2
3  import paket
4
5  a = 1
6  b = 2
7
8  hasil =paket.penambahan(a,b)
9  print(hasil)
10
11
12 #No 4
13 import kelas3lib
14
15 NamaMhs = kelas3lib.Mahasiswa("Ariq")
16 NamaMhs.biadata()
17
18 #No 5
19 from kalkulator import Penambahan
20 a = 2
21 b = 3
22 hasil = Penambahan(a, b)
23 print(hasil)
24
25 #No 6
26 from namafolder.namafile import namafungsi
27
28 a=10
29 b=5
30
31 hasil=namafungsi.Penambahan(a,b)
32 print(hasil)
33
34 #No 7
35 from namafolder.namafile import namapaket
36
37 NamaMhs = namapaket.Mahasiswa("Ariq")
38 NamaMhs.biadata()

```

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

```

1 class Mahasiswa:
2     def __init__(self, nama):
3         self.nama = nama
4     def biodata(self):
5         print("Nama saya", self.nama)
6
7 NamaMhs = Mahasiswa("Ariq")
8 NamaMhs.biodata()

```

### 3.16.3 Keterampilan Penanganan Error

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

```

1 def menyapa(nama):
2     try:
3         print("Hallo, "+str(nama))
4     except:
5         print("Terjadi error")
6
7 menyapa(input("Masukan nama anda: "))

```

## 3.17 Alvan Alvanzah/1174077

### 3.17.1 Pemahaman Teori

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

- Fungsi adalah blok program untuk melakukan tugas-tugas tertentu yang dilakukan berulang dan dapat digunakan berulang kali dari tempat lain di dalam program. namaFungsi dari fungsi yang dibuat.

```
1 def namaFungsi(inputanFungsi):
```

- Inputan fungsi adalah inputan yang berasal dari luar fungsi yang akan di proses di dalam fungsi itu sendiri. inputanFungsi dari inputan fungsi yang diterima dari luar fungsi namaFungsi.

```
1 def namaFungsi(inputanFungsi):
```

- Kembalian fungsi adalah untuk mengembalikan suatu nilai ekspresi dari proses yang dilakukan fungsi. return inputanFungsi merupakan kembalian dari fungsi namaFungsi.

```
1 print(z)
```

2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya. Untuk memudahkan dalam pemanggilan fungsi yang di butuhkan, agar dapat dipanggil berulang. Cara pemanggilannya

```

1 import fungsiAlvan
2 print("Nilai pi adalah: ", fungsiAlvan.pi)

```

### 3. Pengertian kelas, objek, atribut, method, dan contoh kode programnya.

#### ▪ Kelas

Kelas adalah cetak biru atau prototipe dari objek dimana kita mendefinisikan atribut dari suatu objek. Contoh penggunaan kelas di python.

```

1 class Mahasiswa:
2     jumlahMahasiswa = 0
3
4     def __init__(self, npm, nama):
5         self.npm = npm
6         self.nama = nama
7         Mahasiswa.jumlahMahasiswa +=1
8
9     def tampilkanProfil(self):
10        print("NPM :", self.npm)
11        print("Nama :", self.nama)
12        print()
13
14 mahasiswa1 = Mahasiswa("1174077", "Alvan Alvanzah")
15 mahasiswa2 = Mahasiswa("1174067", "Kaka Kamaludin")
16
17 mahasiswa1.tampilkanProfil()
18 mahasiswa2.tampilkanProfil()
19
20 print("Total mahasiswa adalah ", Mahasiswa.jumlahMahasiswa)

```

#### ▪ Objek

Objek adalah instansi atau perwujudan dari sebuah kelas. Contoh penggunaan objek di python.

```

1 mahasiswa1 = Mahasiswa("1174077", "Alvan Alvanzah")
2 mahasiswa2 = Mahasiswa("1174067", "Kaka Kamaludin")

```

#### ▪ Atribut

Atribut adalah variabel yang menyimpan data yang berhubungan dengan kelas dan objeknya. Contoh penggunaan atribut di python.

```

1     jumlahMahasiswa = 0

```

#### ▪ Method

Metode adalah fungsi yang didefinisikan di dalam suatu kelas. Contoh penggunaan method di python.

```

1     def tampilkanProfil(self):
2         print("NPM :", self.npm)
3         print("Nama :", self.nama)
4         print()

```



#### 4. Cara pemanggilan library kelas, dan contoh kode programnya.

Berikut ini adalah cara pemanggilan library kelas dari instansi dan pemakaiannya. Library kelasnya adalah Mahasiswa dari file Mahasiswa.py. Lalu dipanggil dengan import. Kemudian instansi dengan mhs1 dan mhs1, dengan 2 parameter. Contoh pemanggilan library kelas dari instansi dan pemakaiannya.

```
1 from Mahasiswa import Mahasiswa
2
3 mhs1 = Mahasiswa("1174077", "Alvan Alvanzah")
4 mhs2 = Mahasiswa("1174067", "Kaka Kamaludin")
5
6 mhs1.tampilkanProfil()
7 mhs2.tampilkanProfil()
8
9 print("Total mahasiswa adalah ", Mahasiswa.jumlahMahasiswa)
```

#### 5. Penjelasan pemakaian paket disertai dengan contoh kode programnya.

Berikut ini adalah contoh pemakaian paket dengan perintah from kalkulator import Penambahan. Setelah mengimport paketnya, lalu panggil fungsi penambahannya. Contoh pemakaian paket dengan perintah from kalkulator import Penambahan.

```
1 from kalkulator import Penambahan
2
3 hasil = Penambahan(10, 5)
4 print(hasil)
```

#### 6. Contoh kode pemakaian paket fungsi apabila file library ada di dalam folder. Berikut ini adalah pemakaian paket fungsi apabila file library ada di dalam folder. Contoh kode pemakaian paket fungsi dimana file library ada di dalam folder.

```
1 from folder import kalkulator
2
3 a=100
4 b=50
5
6 hasil1=kalkulator.Penambahan(a,b)
7 hasil2=kalkulator.Pengurangan(a,b)
8 hasil3=kalkulator.Perkalian(a,b)
9 hasil4=kalkulator.Pembagian(a,b)
10
11 print(hasil1)
12 print(hasil2)
13 print(hasil3)
14 print(hasil4)
```

#### 7. Contoh kode pemakaian paket kelas apabila file library ada di dalam folder. Berikut ini adalah pemakaian paket kelas apabila file library ada di dalam folder. Contoh kode pemakaian paket kelas dimana file library ada di dalam folder.

```

1 from folder.Mahasiswa import Mahasiswa
2
3 mhs1 = Mahasiswa("1174077", "Alvan Alvanzah")
4 mhs2 = Mahasiswa("1174067", "Kaka Kamaludin")
5
6 mhs1.tampilkanProfil()
7 mhs2.tampilkanProfil()
8
9 print("Total mahasiswa adalah ", Mahasiswa.jumlahMahasiswa)

```

### 3.17.2 Ketrampilan Pemrograman

#### 1. Jawaban soal No. 1

```

1 def printNPM(npm):
2
3     npm = list(str(npm))
4
5     angka1 = {"0": "***** ", "1": " **", "2": "***** ", "3": "
6 ***** ", "4": "    **", "5": "***** ", "6": "***** ", "
7 ": "*****", "8": "***** ", "9": "*****"}
8 angka2 = {"0": "*** **", "1": "***", "2": " ** **", "3": "
9 ** **", "4": "*****", "5": " **", "6": "*** ", "
10 ": " **", "8": " ** **", "9": " ** **"}
11 angka3 = {"0": "*** **", "1": " **", "2": " **", "3": "
12 *** ", "4": " ** **", "5": "***** ", "6": "***** ", "
13 ": " **", "8": " ** **", "9": " ** **"}
14 angka4 = {"0": "*** **", "1": " **", "2": " **", "3": "
15 *** ", "4": "*****", "5": " **", "6": "*** **", "
16 ": " **", "8": " ** **", "9": "*****"}
17 angka5 = {"0": "*** **", "1": " **", "2": " **", "3": "
18 ** **", "4": " **", "5": " ** **", "6": "*** **", "
19 ": " **", "8": " ** **", "9": " **"}
20 angka6 = {"0": "***** ", "1": " **", "2": "*****", "3": "
21 ***** ", "4": " **", "5": "***** ", "6": "***** ", "
22 ": " **", "8": "***** ", "9": "*****"}
23
24 hasil1 = []
25 hasil2 = []
26 hasil3 = []
27 hasil4 = []
28 hasil5 = []
29 hasil6 = []
30
31 for x in npm:
32     hasil1.append(angka1[x])
33     hasil2.append(angka2[x])
34     hasil3.append(angka3[x])
35     hasil4.append(angka4[x])
36     hasil5.append(angka5[x])
37     hasil6.append(angka6[x])
38
39 print(*hasil1, sep=' ')
40 print(*hasil2, sep=' ')
41 print(*hasil3, sep=' ')

```

```

30     print(*hasil4, sep=' ')
31     print(*hasil5, sep=' ')
32     print(*hasil6, sep=' ')
33
34 printNPM(input("Masukan NPM anda: "))

```

## 2. Jawaban soal No. 2

```

1 def printNPMDuaDijit(npm):
2     ulang = 1
3     sampai = int(npm[5:7])
4     while(ulang <= sampai):
5         print("Halo, "+str(npm)+" apa kabar?")
6         ulang += 1
7
8 printNPMDuaDijit(input("Masukan NPM anda: "))

```

## 3. Jawaban soal No. 3

```

1 def printNPMTigaDijit(npm):
2     ulang = 1
3     sampai = list(map(int, npm[4:7]))
4     sampai = sum(sampai)
5     while(ulang <= sampai):
6         print("Halo, "+str(npm[-3:])+ " apa kabar?")
7         ulang += 1
8
9 printNPMTigaDijit(input("Masukan NPM anda: "))

```

## 4. Jawaban soal No. 4

```

1 def printNPMDigitKetiga(npm):
2     print("Output:")
3     print("Halo, "+str(npm[-3])+ " apa kabar?")
4
5 printNPMDigitKetiga(input("Input: "))

```

## 5. Jawaban soal No. 5

```

1 def printNPMSatuPersatu(npm):
2     npm = list(map(int, npm))
3     for n in npm:
4         print(n)
5
6 printNPMSatuPersatu(input("Masukan NPM anda: "))

```

## 6. Jawaban soal No. 6

```

1 def printNPMPenjumlahan(npm):
2     npm = list(map(int, npm))
3     hasil = 0
4     for x in npm:
5         hasil += x
6
7     print(hasil)
8
9 printNPMPenjumlahan(input("Masukan NPM anda: "))

```

## 7. Jawaban soal No. 7

```

1 def printNPMPerkalian(npm):
2     npm = list(map(int, npm))
3     hasil = 0
4     for x in npm:
5         hasil *= x
6
7     print(hasil)
8
9 printNPMPerkalian(input("Masukan NPM anda: "))

```

## 8. Jawaban soal No. 8

```

1 def printNPMDijitGenap(npm):
2     npm = list(map(int, npm))
3     for n in npm:
4         if (n % 2 == 0):
5             if (n != 0):
6                 print(n, end = "")
7
8 printNPMDijitGenap(input("Masukan NPM anda: "))

```

## 9. Jawaban soal No. 9

```

1 def printNPMDijitGanjil(npm):
2     npm = list(map(int, npm))
3     for n in npm:
4         if (n % 2 != 0):
5             print(n, end = "")
6
7 printNPMDijitGanjil(input("Masukan NPM anda: "))

```

## 10. Jawaban soal No. 10

```

1 def printNPMDijitPrima(npm):
2     npm = list(map(int, npm))
3     prima = []
4     for n in npm:
5         isPrime = True
6         if n == 0 or n == 1:
7             isPrime = False
8         for x in range(2, n):
9             if n % x == 0:
10                isPrime = False
11         if isPrime:
12             prima.append(n)
13
14     for p in prima:
15         print(p, end = "")
16
17 printNPMDijitPrima(input("Masukan NPM anda: "))

```

## 11. Jawaban soal No. 11

```

1 lib = __import__('3lib')
2
3 npm = "1174006"
4
5 lib.printNPM(npm)
6 lib.printNPMDuaDijit(npm)
7 lib.printNPMTigaDijit(npm)
8 lib.printNPMDigitKetiga(npm)
9 lib.printNPMSatuPersatu(npm)
10 lib.printNPMPenjumlahan(npm)
11 lib.printNPMPerkalian(npm)
12 lib.printNPMDijitGenap(npm)
13 lib.printNPMDijitGanjil(npm)
14 lib.printNPMDijitPrima(npm)

```

## 12. Jawaban soal No. 12

```

1 from kelas3lib import kelas3lib
2
3 npm = "1174006"
4
5 k3lib = kelas3lib(npm)
6
7 k3lib.printNPM()
8 k3lib.printNPMDuaDijit()
9 k3lib.printNPMTigaDijit()
10 k3lib.printNPMDigitKetiga()
11 k3lib.printNPMSatuPersatu()
12 k3lib.printNPMPenjumlahan()
13 k3lib.printNPMPerkalian()
14 k3lib.printNPMDijitGenap()
15 k3lib.printNPMDijitGanjil()
16 k3lib.printNPMDijitPrima()

```

### 3.17.3 Ketrampilan Penanganan Error

1. Peringatan error yang ditemukan dan penjelasannya serta buat sebuah fungsi try except untuk menanggulangi error.

Peringatan error di praktek ketiga ini, yaitu:

- **Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- **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 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 terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

Contoh fungsi yang menggunakan try except

```
1 def sapa(nama):
2     try:
3         print("Hallo, "+str(nama))
4     except:
5         print("Terjadi error")
6
7 sapa(input("Masukan nama anda: "))
```

## 3.18 Engelbertus Adiputra Mau Leto/1174078

### 3.18.0.1 Pemahaman Teori

#### 1. Apa itu fungsi, inputan

fungsi dan kembalian fungsi dengan contoh kode program lainnya. Fungsi adalah bagian dari program yang dapat digunakan ulang. Berikut merupakan contoh fungsi dan cara pemanggilannya

```
1 def uji():
2     print("Tugas Web Service")
3
4 uji()
```

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

```
1 def uji_param(nama):
2     print("Nama saya :"+str(nama))
3
4 uji_param(input("Masukan Nama Kamu : "))
```

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

```
1 def uji_return(a,b):
2     r = a + b
3     return r
4
5 a = 10
6 b = 50
7 c = uji_return(a,b)
8 print(c)
```

- #### 2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya. Untuk memudahkan dalam pemanggilan fungsi yang dibutuhkan, agar dapat dipanggil berulang. Cara pemanggilannya

```

1 #from fungsi_felix import *
2 #print(penulisan(int(input("Masukan NPM kamu : "))))

```

3. Jelaskan Apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing. kelas merupakan sebuah blueprint yang merepresentasikan objek. objek adalah hasil cetakan dari sebuah kelas. method adalah suatu upaya yang digunakan oleh object.

```

1 #class Employee:
2 #   'Common base class for all employees'
3 #   empCount = 0
4
5 #   def __init__(self, name, salary):
6 #       self.name = name
7 #       self.salary = salary
8 #       Employee.empCount += 1
9
10 #   def displayCount(self):
11 #       print ("Total Employee %d" % Employee.empCount)
12
13 #   def displayEmployee(self):
14 #       print ("Name : ", self.name, ", Salary: ", self.salary)
15
16
17 #This would create first object of Employee class"
18 #emp1 = Employee("Zara", 2000)
19 #This would create second object of Employee class"
20 #emp2 = Employee("Manni", 5000)
21 #emp1.displayEmployee()
22 #emp2.displayEmployee()
23 #print ("Total Employee %d" % Employee.empCount)

```

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

- pertama import terlebih dahulu filenya.
- kemudian buat variabel untuk menampung datanya
- setelah itu panggil nama classnya dan panggil methodnya
- Gunakan perintah print untuk menampilkan hasilnya

```

1 #import belajar
2 #a = 100
3 #b = 50
4
5 #c = belajar.penambahan(a,b)
6 #print(c)

```

5. Jelaskan dengan contoh pemakaian paket dengan perintah from kalkulator import Penambahan disertai dengan contoh kode lainnya. Penggunaan paket from namafile import, itu berfungsi untuk memanggil file dan fungsinya

```

1 #from fungsi_felix import *
2 #print(penulisan(int(input("Masukan NPM kamu : "))))

```

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila  
le library ada di dalam folder. Pemakaian paket adalah perkumpulan fungsi-fungsi. contoh kodenya adalah sebagai berikut :
7. Jelaskan dengan contoh kodenya, pemakaian paket kelas apabila  
le library ada di dalam folder.

### 3.18.0.2 Ketrampilan Pemrograman

1. Buatlah fungsi dengan inputan variabel NPM, dan melakukan print luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk NPM mod 3=0, tanda pagar untuk NPM mod 3 =1, tanda plus untuk NPM mod3=2.

```

1
2 #No 1
3 def speakup(npm):
4     npm = list(str(npm))
5
6     angka1 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
7     ##### ", "4": " ### ", "5": " ##### ", "6": " ##### ", "7": "
8     ##### ", "8": " ##### ", "9": " ##### "}
9     angka2 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
10    ##### ", "4": " ### ", "5": " ##### ", "6": " ##### ", "7": "
11    ##### ", "8": " ##### ", "9": " ##### "}
12    angka3 = {"0": " ### ", "1": " ### ", "2": " ### ", "3": "
13    ##### ", "4": " ### ", "5": " ##### ", "6": " ### ", "7": "
14    ##### ", "8": " ##### ", "9": " ##### "}
15    angka4 = {"0": " ### ", "1": " ### ", "2": " ### ", "3": "
16    ##### ", "4": " ### ", "5": " ##### ", "6": " ### ", "7": "
17    ##### ", "8": " ##### ", "9": " ##### "}
18    angka5 = {"0": " ### ", "1": " ### ", "2": " ##### ", "3": "
19    ##### ", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
20    ##### ", "8": " ##### ", "9": " ##### "}
21    angka6 = {"0": " ### ", "1": " ### ", "2": " ##### ", "3": "
22    ##### ", "4": " ### ", "5": " ##### ", "6": " ##### ", "7": "
23    ##### ", "8": " ##### ", "9": " ##### "}
24    angka7 = {"0": " ### ", "1": " ### ", "2": " ### ", "3": "
25    ##### ", "4": " ### ", "5": " ##### ", "6": " ### ", "7": "
26    ##### ", "8": " ##### ", "9": " ##### "}
27    angka8 = {"0": " ### ", "1": " ##### ", "2": " ### ", "3": "
28    ##### ", "4": " ### ", "5": " ##### ", "6": " ### ", "7": "
29    ##### ", "8": " ##### ", "9": " ##### "}
30    angka9 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
31    ##### ", "4": " ### ", "5": " ##### ", "6": " ##### ", "7": "
32    ##### ", "8": " ##### ", "9": " ##### "}
33    angka10 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
34    ##### ", "4": " ### ", "5": " ##### ", "6": " ##### ", "7": "
35    ##### ", "8": " ##### ", "9": " ##### "}

```



```

16
17     hasil1 = []
18     hasil2 = []
19     hasil3 = []
20     hasil4 = []
21     hasil5 = []
22     hasil6 = []
23     hasil7 = []
24     hasil8 = []
25     hasil9 = []
26     hasil10 = []
27
28     for x in npm:
29         hasil1.append(angka1[x])
30         hasil2.append(angka2[x])
31         hasil3.append(angka3[x])
32         hasil4.append(angka4[x])
33         hasil5.append(angka5[x])
34         hasil6.append(angka6[x])
35         hasil7.append(angka7[x])
36         hasil8.append(angka8[x])
37         hasil9.append(angka9[x])
38         hasil10.append(angka10[x])
39
40     print(*hasil1, sep=' ')
41     print(*hasil2, sep=' ')
42     print(*hasil3, sep=' ')
43     print(*hasil4, sep=' ')
44     print(*hasil5, sep=' ')
45     print(*hasil6, sep=' ')
46     print(*hasil7, sep=' ')
47     print(*hasil8, sep=' ')
48     print(*hasil9, sep=' ')
49     print(*hasil10, sep=' ')
50
51 speakup(int(input("Masukan NPM : 117")))

```

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

```

1 def looping(npm):
2     hitung = 0
3     while(hitung < 78):
4         print("Haiiiii, "+str(npm)+" apa kabar?")
5         hitung = hitung +1
6
7 looping(int(input("Masukan NPM : ")))

```

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

```

1 #No 3
2 def looping_3_digit(npm):

```

```

3     hitung = 0
4     npm = str(npm)
5     bil = npm[4:7]
6
7     while(hitung < 15):
8         print("Halo, "+bil+" apa kabar?")
9         hitung = hitung +1
10
11 looping_3_digit(int(input("Masukan NPM : ")))

```

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

```

1 #No 4
2 def looping_3_digit_terakhir(npm):
3     npm = str(npm)
4     bil = npm[-3]
5     print("Haiii, "+bil+" apa kabar?")
6
7 looping_3_digit_terakhir(int(input("Masukan NPM : ")))

```

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

```

1 #No 5
2 def down(npm):
3     for i in npm:
4         print(i)

```

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

```

1
2 #No 6
3 def addition(npm):
4     jumlah = 0
5     for i in npm:
6         jumlah += int(i)
7     print(str(jumlah)+" Adalah hasil penjumlahan")

```

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

```

1
2 #No 7
3 def multiplication(npm):
4     jumlah = 0
5     for i in npm:
6         jumlah *= int(i)
7     print(str(jumlah)+" Adalah hasil perkalian")

```

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

```

1
2 #No 8
3 def even():
4     npm = [1,1,7,4,0,7,8]
5     for i in npm:
6         if (i % 2) == 0:
7             print("Bilangan Genapnya : "+str(i))

```

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

```

1 even()
2
3 #No 9
4 def odd():
5     npm = [1,1,7,4,0,7,8]
6     for i in npm:
7         if (i%2)==1:

```

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

```

1
2 odd()
3
4 #No 10
5 def prime(npm):
6     npm = str(npm)
7     bil = npm[2]
8     num = int(bil)
9     if num > 1:
10        for i in range(2,num):
11            if (num%i)==0:
12                print("Bukan Bilangan Prima")
13                break
14            else:
15                print("Bilangan Primanya :"+str(num))

```

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

```

1 from fungsi_engel import learn

```

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

```
1 from 3lib_engel import learn
```

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

```
1 #def penanganan_error(a,b):
2 #     try :
3 #         c = a+b
4 #         print(c)
5 #     except TypeError:
6 #         print("kita beda")
```

## 3.19 Tia Nur Candida

### 3.19.0.1 Pemahaman Teori

#### 1. Apa itu fungsi, inputan

fungsi dan kembalian fungsi dengan contoh kode program lainnya. Fungsi merupakan blok kode atau program yang strukturnya terdiri dari nama fungsi, input variable, dan variable kembalian. Fungsi juga merupakan bagian dari program yang dapat dipakai ulang, dapat membaca parameter. Penulisan fungsi diawali oleh def dan setelahnya titik dua.

```
1 #No. 1
2 def anu():
3     print("Halo Sayang")
4
5 anu()
```

Fungsi input digunakan untuk menerima baris input dari user dan mengembalikannya dalam string.

```
1
2 def cek_inputan (nama):
3     print("Nama saya :"+str(nama))
4
5 cek_inputan (input("Masukan Nama Kamu : "))
```

Fungsi kembalian atau yang sering disebut dengan return merupakan fungsi yang mengembalikan nilai.

```
1 def cek_return(a,b):
2     r = a + b
3     return r
4
5 a = 10
6 b = 50
7 c = cek_return(a,b)
8 print(c)
```

2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya. Untuk memudahkan dalam pemanggilan fungsi yang di butuhkan, agar dapat dipanggil berulang. Cara pemanggilannya

```
1 #from 1174086 import *
2 #print(penulisan(int(input("Masukan NPM kamu : "))))
```

3. Jelaskan Apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing. Class merupakan cetakan atau blueprint dari objek, class disebut juga konstruktor objek. Objek merupakan sesuatu yang unik dari struktur data yang didefinisikan oleh kelas dan method. Atribut merupakan data atau segala sesuatu yang ada pada class dan method yang diakses melalui notasi titik. Method adalah fungsi khusus yang didefinisikan dalam kelas.

```
1
2 #class Employee:
3 #   'Common base class for all employees'
4 #   empCount = 0
5
6 #   def __init__(self, name, salary):
7 #       self.name = name
8 #       self.salary = salary
9 #       Employee.empCount += 1
10
11 #   def displayCount(self):
12 #       print ("Total Employee %d" % Employee.empCount)
13
14 #   def displayEmployee(self):
15 #       print ("Name : ", self.name, ", Salary: ", self.salary)
16
17
18 #This would create first object of Employee class"
19 #emp1 = Employee("Zara", 2000)
20 #This would create second object of Employee class"
21 #emp2 = Employee("Manni", 5000)
22 #emp1.displayEmployee()
23 #emp2.displayEmployee()
24 #print ("Total Employee %d" % Employee.empCount)
```

4. Jelaskan cara pemanggilan library kelas dari instansiasi dan pemakaianannya dengan contoh program lainnya. Cara Pemanggilanya

- pertama import file terlebih dahulu.
- kemudian membuat variabel untuk menampung data,
- setelah itu panggil nama class dan panggil method,
- Untuk menampilkan hasilnya, gunakan perintah print.

```
1 #import belajar
2 #a = 100
3 #b = 50
```

```

4
5 #c = belajar.penambahan(a,b)
6 #print(c)

```

5. Jelaskan dengan contoh pemakaian paket dengan perintah from kalkulator import Penambahan disertai dengan contoh kode lainnya. Penggunaan paket from namafile import, itu berfungsi untuk memanggil file dan fungsinya

```

1 #c = belajar.penambahan(a,b)
2 #print(c)

```

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila le library ada di dalam folder. Pemakaian paket adalah perkumpulan fungsi-fungsi. contoh kodenya adalah sebagai berikut :
7. Jelaskan dengan contoh kodenya, pemakaian paket kelas apabila le library ada di dalam folder.

```

1 #Chapter 3

```

### 3.19.0.2 Ketrampilan Pemrograman

1. Buatlah fungsi dengan inputan variabel NPM, dan melakukan print luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk NPM mod 3=0, tanda pagar untuk NPM mod 3 =1, tanda plus untuk NPM mod3=2.

```

1 #No 1
2 def penulisan(npm):
3     npm = list(str(npm))
4
5     angka1 = {"0": " ##### ", "1": " ### ", "2": " ### ", "3": "
##### ", "4": " ###  ## ", "5": " ##### ", "6": " ##### ", "7": "
##### ", "8": " ##### ", "9": " #### "}
6     angka2 = {"0": " ##### ", "1": " ### ", "2": " ### ", "3": "
##### ", "4": " ###  ## ", "5": " ##### ", "6": " ##### ", "7": "
##### ", "8": " ##### ", "9": " #### "}
7     angka3 = {"0": " ##  ## ", "1": " ### ", "2": " ### ", "3": "
##### ", "4": " ###  ## ", "5": " ##### ", "6": " ###  ", "7": "
      ", "8": " ##  ## ", "9": " #### "}
8     angka4 = {"0": " ##  ## ", "1": " ### ", "2": " ### ", "3": "
##### ", "4": " ###  ## ", "5": " ##### ", "6": " ###  ", "7": "
      ", "8": " ##  ## ", "9": " #### "}
9     angka5 = {"0": " ##  ## ", "1": " ### ", "2": " ### ", "3": "
##### ", "4": " ##### ", "5": " ##### ", "6": " ###  ", "7": "
      ", "8": " ##### ", "9": " #### "}
10    angka6 = {"0": " ##  ## ", "1": " ### ", "2": " ### ", "3": "
##### ", "4": "      ", "5": " ##### ", "6": " ##### ", "7": "
      ", "8": "      ", "9": " ##### "}
11    angka7 = {"0": " ##  ## ", "1": " ### ", "2": " ### ", "3": "
##### ", "4": "      ", "5": " ##### ", "6": " ###  ## ", "7": "
      ", "8": " ##  ## ", "9": " #### "}

```

```

12  angka8 = {"0": " ##      ## ", "1": " ### ", "2": " ### ", "3": "
      ##### ", "4": "      ### ", "5": " ##### ", "6": " ###  ## ", "7": "
      ###      ", "8": " ##  ## ", "9": " ##### "}
13  angka9 = {"0": " ##### ", "1": " ### ", "2": " ### ", "3": "
      ##### ", "4": "      ### ", "5": " ##### ", "6": " ###  ## ", "7": "
      ###      ", "8": " ##### ", "9": " ##### "}
14  angka10 = {"0": " ##### ", "1": " ### ", "2": " ### ", "3": "
      ##### ", "4": "      ### ", "5": " ##### ", "6": " ##### ", "7": "
      ###      ", "8": " ##### ", "9": " ##### "}

15
16  hasil1 = []
17  hasil2 = []
18  hasil3 = []
19  hasil4 = []
20  hasil5 = []
21  hasil6 = []
22  hasil7 = []
23  hasil8 = []
24  hasil9 = []
25  hasil10 = []
26
27  for x in npm:
28      hasil1.append(angka1[x])
29      hasil2.append(angka2[x])
30      hasil3.append(angka3[x])
31      hasil4.append(angka4[x])
32      hasil5.append(angka5[x])
33      hasil6.append(angka6[x])
34      hasil7.append(angka7[x])
35      hasil8.append(angka8[x])
36      hasil9.append(angka9[x])
37      hasil10.append(angka10[x])
38
39  print(*hasil1, sep=' ')
40  print(*hasil2, sep=' ')
41  print(*hasil3, sep=' ')
42  print(*hasil4, sep=' ')
43  print(*hasil5, sep=' ')
44  print(*hasil6, sep=' ')
45  print(*hasil7, sep=' ')
46  print(*hasil8, sep=' ')
47  print(*hasil9, sep=' ')
48  print(*hasil10, sep=' ')
49
50  penulisan(int(input("Masukan NPM :")))

```

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

```

1  #No 2
2  def perulangan(npm):
3      hitung = 0
4      while(hitung < 86):
5          print("Halo, "+str(npm)+" apa kabar?")
6          hitung = hitung +1

```

```

7
8 perulangan(int(input("Masukan NPM : ")))

```

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

```

1 #No 3
2 def perulangan_3_digit(npm):
3     hitung = 0
4     npm = str(npm)
5     bil = npm[4:7]
6
7     while(hitung < 14):
8         print("Halo, "+bil+" apa kabar?")
9         hitung = hitung +1
10
11 perulangan_3_digit(int(input("Masukan NPM : ")))

```

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

```

1 #No 4
2 def perulangan_3_digit_terakhir(npm):
3     npm = str(npm)
4     bil = npm[-3]
5     print("Halo, "+bil+" apa kabar?")
6
7 perulangan_3_digit_terakhir(int(input("Masukan NPM : ")))

```

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

```

1 #No 5
2 def down(npm):
3     for i in npm:
4         print (i)
5
6 down(input("Masukan NPM : "))

```

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

```

1 #No 6
2 def penjumlahan(npm):
3     jumlah = 0
4     for i in npm:
5         jumlah += int(i)
6     print(str(jumlah)+" Adalah hasil penjumlahan")
7
8 penjumlahan(input("Masukan NPM : "))

```



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

```

1 #No 7
2 def perkalian(npm):
3     jumlah = 0
4     for i in npm:
5         jumlah *= int(i)
6         print(str(jumlah)+" Adalah hasil perkalian")
7
8     perkalian(input("Masukan NPM : "))

```

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

```

1 #No 8
2 def genap():
3     npm = [1,1,7,4,0,8,6]
4     for i in npm:
5         if (i % 2) == 0:
6             print("Bilangan Genapnya : "+str(i))
7     genap()

```

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

```

1 #No 9
2 def ganjil():
3     npm = [1,1,7,4,0,8,6]
4     for i in npm:
5         if (i%2)==1:
6             print("Bilangan Ganjilnya : "+str(i))
7     ganjil()

```

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

```

1 #No 10
2 def prima(npm):
3     npm = str(npm)
4     bil = npm[2]
5     num = int(bil)
6     if num > 1:
7         for i in range(2,num):
8             if (num%i)==0:
9                 print("Bukan Bilangan Prima")
10                break
11            else:
12                print("Bilangan Primanya :"+str(num))
13        else:
14            print("Tidak Ada Bilangan Prima")
15    prima(int(input("Masukan NPM : ")))

```

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

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

## 3.20 Mochamad Arifqi Ramadhan

### 3.20.0.1 Pemahaman Teori

#### 1. Apa itu fungsi, inputan

fungsi dan kembalian fungsi dengan contoh kode program lainnya. Fungsi adalah kode atau instruksi yang dibuat untuk melakukan tugas tertentu yang dapat digunakan ulang. inputan fungsi adalah data yang dimasukkan untuk mengeksekusi atau menjalankan fungsi kembalian fungsi adalah hasil dari fungsi yang akan dikembalikan. Berikut contoh fungsi dan cara pemanggilannya

```
1 #no 1
2 def perkalian(x, y):
3     z = x * y
4     return z
5 x = 2
6 y = 3
7 hasil = perkalian(x, y)
```

#### 2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya. Paket atau library adalah kumpulan fungsi untuk memudahkan pemrogram membuat program dan tidak perlu mengakses langsung sistem komputer untuk memprogram. Cara pemanggilannya

```
1 print("***      ***      ***      ***      ***      ***      ***")
2 print("      ***      ***      ***      ***      ***      ***      ***")
3 print("      ***      ***      ***      ***      ***      ***      ***")
4 #No.2
5 npm = input("Masukan NPM :")
```

#### 3. Jelaskan Apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing. Kelas adalah cetak biru dari objek. Objek adalah perwujudan dari sebuah kelas yang memiliki atribut dan method. Atribut adalah segala sesuatu yang menempel atau yang ada pada objek. Metode adalah fungsi yang didefinisikan di dalam suatu kelas.

```

1 while(hitung < 74):
2     print("Halo, " + str(npm) + " Apa Kabar?")
3     hitung = hitung +1
4 #No.3
5 npm = input("Masukan NPM :")
6 hitung = 0
7 while(hitung < 3):
8     print("Halo, " + str(npm[4:7]) + " Apa Kabar?")
9     hitung = hitung +1
10 #No.4
11 npm = input("Masukan NPM :")
12 print("Halo, " + str(npm[-3]) + " Apa Kabar?")
13 #No.5
14 a = 1
15 b = 1
16 c = 7

```

4. Jelaskan cara pemanggilan library kelas dari instansiasi dan pemakaiannya dengan contoh program lainnya. Berikut ini adalah cara pemanggilan library kelas dari instansi dan pemakaiannya. Nama dari Library kelasnya adalah pakan dari file pakan.py

```

1 e = 0
2 f = 7
3 g = 4
4 h = a+b+c+d+e+f+g
5 i = a*b*c*d*e*f*g
6 print(a,b,c,d,e,f,g)
7 #No.6

```

5. Jelaskan dengan contoh pemakaian paket dengan perintah from kalkulator import Penambahan disertai dengan contoh kode lainnya. Berikut ini adalah pemakaian paket fungsi apabila file library ada di dalam folder.

```

1 #No.7
2 print(i)
3 #No.8
4 print(a)

```

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila file library ada di dalam folder. Pemakaian paket adalah perkumpulan fungsi-fungsi. contoh kodenya adalah sebagai berikut :

```

1 print(c)
2 print(d)
3 print(e)
4 print(f)
5 print(g)
6 #No.9
7 print(d,g)
8 #No.10
9 print(a,b,c,f)
10 #No.11
11 print(c,g)

```

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

1 lastline

### 3.20.0.2 Keterampilan Pemrograman

1. Buatlah fungsi dengan inputan variabel NPM, dan melakukan print luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk  $\text{NPM} \bmod 3 = 0$ , tanda pagar untuk  $\text{NPM} \bmod 3 = 1$ , tanda plus untuk  $\text{NPM} \bmod 3 = 2$ .
2. Buatlah fungsi dengan inputan variabel berupa NPM. kemudian dengan menggunakan perulangan mengeluarkan print output sebanyak dua digit belakang NPM.
3. Buatlah fungsi dengan dengan input variabel string bernama NPM dan beri luaran output dengan perulangan berupa tiga karakter belakang dari NPM sebanyak penjumlahan tiga digit tersebut.
4. Buatlah fungsi hello word dengan input variabel string bernama NPM dan beri luaran output berupa digit ketiga dari belakang dari variabel NPM menggunakan akses langsung manipulasi string pada baris ketiga dari variabel NPM.
5. buat fungsi program dengan input variabel NPM dan melakukan print nomor npm satu persatu kebawah.
6. Buatlah fungsi dengan inputan variabel NPM, didalamnya melakukan penjumlahan dari seluruh digit NPM tersebut, wajib menggunakan perulangan dan atau kondisi.
7. Buatlah fungsi dengan inputan variabel NPM, didalamnya melakukan melakukan perkalian dari seluruh digit NPM tersebut, wajib menggunakan perulangan dan atau kondisi.
8. Buatlah fungsi dengan inputan variabel NPM, Lakukan print NPM anda tapi hanya digit genap saja. wajib menggunakan perulangan dan atau kondisi.
9. Buatlah fungsi dengan inputan variabel NPM, Lakukan print NPM anda tapi hanya digit ganjil saja. wajib menggunakan perulangan dan atau kondisi.

10. Buatlah fungsi dengan inputan variabel NPM, Lakukan print NPM anda tapi hanya dijit yang termasuk bilangan prima saja. wajib menggunakan perulangan dan atau kondisi.
11. Buatlah satu library yang berisi fungsi-fungsi dari nomor diatas dengan nama le 3lib.py dan berikan contoh cara pemanggilannya pada le main.py.

```

1 from 1174086 import penulisan
2 from 3lib import penulisan
3
4 print(penulisan(int(input("Masukan NPM :"))))

```

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

**3.20.0.3 Keterampilan Penanganan Error** Error yang di dapat dari mengerjakan tugas ini adalah type error, cara menaggulaginya dengan cara mengecek kembali codingannya dan merubah kode yang erorrnya, lalu jalankan lagi programnya. berikut contoh Penggunaan fungsi try dan exception

## 3.21 Chapter 3 | Kaka Kamaludin

### 3.21.1 Teori

1. pada Python penulisan fungsi diawali dengan "def" lalu nama fungsi yang akan dibuat, dan biasanya terdapat juga parameter yang berisikan variable didalamnya. contoh :

```

1 #fungsi
2 def fungsi():
3     print("string")
4 fungsi() #memanggil fungsi
5
6 #inputan fungsi
7 def luasSegitiga(alas , tinggi):
8     luas = (alas * tinggi)/2
9     print ("Luas segitiga : %f" % luas) # %f float
10 luasSegitiga(8, 12) #memanggil fungsi luasSegitiga
11
12 #kembalian fungsi

```

```

13 def luasSegitiga(alas , tinggi):
14     luas = (alas * tinggi)/2
15     return luas
16 print ("Luas segitiga : %f" % luasSegitiga(8, 12))

```

2. paket atau library merupakan file yang berisi kode yang dipanggil menggunakan perintah "import". contoh :

```

1 ###2###
2 import tes.moduleKaka
3 print (tes.moduleKaka.fullName())

```

3. kelas bisa diartikan sebagai cetakan, object merupakan hasil dari cetakan tersebut, dan atribut merupakan ciri khas yang melekat pada object contoh : class yang bernama Identisa yang berisi nama.

```

1 #file identitas.py
2 class Identitas:
3     def __init__(self , saya):
4         self.saya = saya
5     def nama(self):
6         r = self.saya
7         return r
8
9 #file nain.py
10 from identitas import Identitas
11 saya = "Kaka"
12 siapa = Identitas(saya)
13 hasil=siapa.nama()

```

4. pemanggilan library kelas, contoh : - buat file class python dan bernama - buat file main.py untuk memanggil class yang sudah diberinama tadi - udah

```

1 #file identitas.py
2 class Identitas:
3     def __init__(self , saya):
4         self.saya = saya
5     def nama(self):
6         r = self.saya
7         return r
8
9 #file nain.py
10 from identitas import Identitas
11 saya = "Kaka"
12 siapa = Identitas(saya)
13 hasil=siapa.nama()

```

5. from identitas merupakan pemanggilan nama file, import Identitas merupakan nama class yang dipanggil. contoh :

```

1 ###2###
2 import tes.moduleKaka
3 print (tes.moduleKaka.fullName())

```

6. dengan menggunakan perintah `import namafolder.namafile` dan penggunaan `(.)` contoh :

```

1 lastlinelastline
2 ###2###
3 import tes.moduleKaka
4 print(tes.moduleKaka.fullName())
5
6 ###3##
7 #file identitas.py
8 class Identitas:
9     def __init__(self, saya):
10         self.saya = saya
11     def nama(self):
12         r = self.saya
13         return r
14
15 #file nain.py
16 from identitas import Identitas
17 saya = "Kaka"
18 siapa = Identitas(saya)
19 hasil1=siapa.nama()
20
21 ###4###

```

7. `from identitas` merupakan pemanggilan nama file, `import Identitas` merupakan nama class yang dipanggil. contoh :

```

1 #file identitas.py
2 class Identitas:
3     def __init__(self, saya):
4         self.saya = saya
5     def nama(self):
6         r = self.saya
7         return r
8
9 #file nain.py
10 from identitas import Identitas
11 saya = "Kaka"
12 siapa = Identitas(saya)
13 hasil1=siapa.nama()

```

### 3.21.2 Praktek

1. jawaban :

```

1 #Jawaban No. 1
2 def variableNPM(npm):
3     npm = list(str(npm))
4
5     row1 = {"0": "##### ", "1": "   ##", "2": "##### ", "3": "##### ", "4": "   ##", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
6     row2 = {"0": "###   ##", "1": "#####", "2": "###   ##", "3": "###   ##", "4": "#####", "5": "###   ", "6": "###   ", "7": "###   ", "8": "###   ##", "9": "###   ##"}

```

```

7 row3 = {"0": "###  ##", "1": "  ##", "2": "    ## ", "3": "
    ### ", "4": "  ##  ##", "5": "##### ", "6": "##### ", "7":
: "  ## ", "8": "  ##  ## ", "9": "##    ##"}
8 row4 = {"0": "###  ##", "1": "  ##", "2": "    ## ", "3": "
    ### ", "4": "##### ", "5": "    ##", "6": "###  ## ", "7":
: "    ## ", "8": "  ##  ## ", "9": "    #####"}
9 row5 = {"0": "###  ##", "1": "  ##", "2": "    ## ", "3": "##
    ## ", "4": "    ## ", "5": "##  ## ", "6": "###  ## ", "7":
: "  ## ", "8": "##### ", "9": "    ##"}
10 row6 = {"0": "    ## ", "1": "  ##", "2": "##### ", "3": "
##### ", "4": "    ## ", "5": "    ## ", "6": "    ## ", "
7": "  ## ", "8": "    ## ", "9": "    ## "}
11
12 scan1 = []
13 scan2 = []
14 scan3 = []
15 scan4 = []
16 scan5 = []
17 scan6 = []
18
19 for x in npm:
20     scan1.append(row1[x])
21     scan2.append(row2[x])
22     scan3.append(row3[x])
23     scan4.append(row4[x])
24     scan5.append(row5[x])
25     scan6.append(row6[x])
26
27     print(*scan1, sep=' ')
28     print(*scan2, sep=' ')
29     print(*scan3, sep=' ')
30     print(*scan4, sep=' ')
31     print(*scan5, sep=' ')
32     print(*scan6, sep=' ')
33
34 variableNPM(input("NPM : "))

```

## 2. jawaban :

```

1 #Jawaban No. 2
2 def variableNPM(npm):
3     i = 1
4     dua = int(npm[5:7])
5     while(i <= dua):
6         print("Halo, "+str(npm)+" apa kabar?")
7         i += 1
8
9 variableNPM(input("NPM : "))

```

## 3. jawaban :

```

1 #Jawaban No. 3
2 def variableNPM(npm):
3     i = 1
4     tiga = list(map(int, npm[4:7]))
5     sampai = sum(tiga)

```



```

6     while(i <= sampai):
7         print("Halo, "+str(npm[-3:])+ " apa kabar?")
8         i += 1
9
10 variableNPM(input("NPM : "))

```

#### 4. jawaban :

```

1 def variableNPM(npm):
2     i = 1
3     tiga = list(map(int, npm[4:7]))
4     sampai = sum(tiga)
5     while(i <= sampai):
6         print("Halo, "+str(npm[-3:])+ " apa kabar?")
7         i += 1
8
9 variableNPM(input("NPM : "))
10
11 #Jawaban No. 4
12 def variableNPM(npm):
13     print("Halo, "+str(npm[-3:])+ " apa kabar?")
14
15 variableNPM(input("NPM : "))

```

#### 5. jawaban :

```

1 #Jawaban No. 5
2 def variableNPM(npm):
3     npm = list(map(int, npm))
4     for x in npm:
5         print(x)
6
7 variableNPM(input("NPM : "))

```

#### 6. jawaban :

```

1 #Jawaban No. 6
2 def variableNPM(npm):
3     npm = list(map(int, npm))
4     hasil = 0
5     for x in npm:
6         hasil += x
7
8     print(hasil)
9
10 variableNPM(input("NPM : "))

```

#### 7. jawaban :

```

1 #Jawaban No. 7
2 def variableNPM(npm):
3     npm = list(map(int, npm))
4     hasil = 0
5     for x in npm:
6         hasil *= x

```

```

7
8     print(hasil)
9
10 variableNPM(input("NPM : "))

```

8. jawaban :

```

1 #Jawaban No. 8
2 def variableNPM(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 variableNPM(input("NPM : "))

```

9. jawaban :

```

1 #Jawaban No. 9
2 def variableNPM(npm):
3     npm = list(map(int, npm))
4     for x in npm:
5         if(x % 2 != 0):
6             print(x, end = "")
7
8 variableNPM(input("NPM : "))

```

10. jawaban :

```

1 #Jawaban No. 10
2 def variableNPM(npm):
3     npm = list(map(int, npm))
4     prima = []
5     for n in npm:
6         isPrime = True
7         if n == 0 or n == 1:
8             isPrime = False
9         for x in range(2, n):
10            if n % x == 0:
11                isPrime = False
12        if isPrime:
13            prima.append(n)
14
15    for p in prima:
16        print(p, end = "")
17
18 variableNPM(input("NPM : "))

```

11. jawaban :

```

1 from 3lib import variableNPM
2 variableNPM(input("NPM : "))

```

12. jawaban :

```

1 from kelas3lib import kelas3lib
2 npm = "1174067"
3 kelas = kelas3lib(npm)
4
5 kelas.variableNPM()

```

### 3.21.3 Praktek

#### 1. Syntax Errors

penulisan kode tidak rapih, rapihkan kembali dengan mengecek tab setiap baris.

```

1 from kelas3lib import kelas3lib
2 npm = "1174067"
3 kelas = kelas3lib(npm)
4
5 kelas.variableNPM()

```

## 3.22 Fernando Lorencius Sihite

### 3.22.0.1 Pemahaman Teori

#### 1. Apa itu fungsi, inputan

fungsi dan kembalian fungsi dengan contoh kode program lainnya. Fungsi yaitu bagian dari program yang dapat digunakan ulang.fungsi, blok kode terorganisir dan dapat digunakan kembali yang digunakan untuk melakukan sebuah tindakan/action Setelah def ada nama pengenalan fungsi diikuti dengan parameter yang diapit oleh tanda kurung dan diakhir dengan tanda titik dua : .yang merupakan sebuah program kecil untuk memproses sebagian dari pekerjaan program utama. Input variabel bisa lebih dari satu dengan pemisah tanda koma. variabel kembalian pasti satu, bebas apakah itu jenis *string*, *integer*, *list* atau *dictionary*

```

1 def Tugas() :
2     print("Kerjakan Tugas")
3
4 Tugas()

```

Fungsi dapat membaca parameter, parameter adalah nilai yang disediakan kepada fungsi, dimana nilai ini akan menentukan output yang akan dihasilkan fungsi. Parameter dikirim dalam tanda kurung saat pemanggilan fungsi. Statemen return digunakan untuk keluar dari fungsi. return digunakan untuk mengembalikan nilai semula.

```

1 def Tugas(inputanTugas) :
2     return inputanTugas

```

#### 2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya. dengan adanya paket dapat memudahkan dalam proses pemanggilan, agar dapat dipanggil berulang. Cara pemanggilannya

```

1 import math
2 print("Nilai: ", math.pi)

```

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

- Kelas adalah sebuah blueprint untuk mempresentasikan objek. selain itu dalam sebuah kelas yang didalamnya terdapat beberapa metode yang memang merupakan isi dari sebuah class ini. Class dan metode ini biasa di kenal dengan sebutan OOP atau object oriented programing.

```

1 class Jurusan:
2     jumlahMahasiswa = 0
3
4     def __init__(self, prodi, nama):
5         self.prodi = prodi
6         self.nama = nama
7         Jurusan.jumlahMahasiswa +=1
8
9     def tampilkanProfil(self):
10        print("Prodi :", self.prodi)
11        print("Nama :", self.nama)
12        print()
13
14 mahasiswa1 = Jurusan("TI", "FernandoLS")
15 mahasiswa2 = Jurusan("TI", "bebas")
16
17 mahasiswa1.tampilkanProfil()
18 mahasiswa2.tampilkanProfil()
19
20 print("Jumlah Mahasiswa ", Jurusan.jumlahMahasiswa)

```

**Listing 3.18** Contoh penggunaan kelas di python.

- Objek adalah instansiasi atau perwujudan dari sebuah kelas. Bila kelas adalah prototipenya, dan objek adalah barang jadinya.

```

1 mahasiswa1 = Jurusan("TI", "FernandoLS")
2 mahasiswa2 = Jurusan("TI", "bebas")

```

**Listing 3.19** Contoh penggunaan Objek di python.

- Atribut adalah data anggota (variabel kelas dan variabel contoh) dan metode, diakses melalui notasi titik.

```

1     jumlahMahasiswa = 0

```

**Listing 3.20** Contoh penggunaan Atribut di python.

- Metode adalah fungsi yang didefinisikan di dalam suatu kelas.

```

1     def tampilkanProfil(self):
2         print("Prodi :", self.prodi)
3         print("Nama :", self.nama)
4         print()

```

**Listing 3.21** Contoh penggunaan Metode di python.

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

- pertama import terlebih dahulu filenya.
- kemudian buat variabel untuk menampung datanya
- setelah itu panggil nama classnya dan panggil methodnya
- Gunakan perintah print untuk menampilkan hasilnya

```

1 #4
2 from Jurusan import Jurusan
3
4 mhs1 = Jurusan("TI", "Fernandols")
5 mhs2 = Jurusan("TI", "bebas")
6
7 mhs1.tampilkanProfil()
8 mhs2.tampilkanProfil()
9
10 print("Total mahasiswa ", Jurusan.jumlahMahasiswa)

```

5. Jelaskan dengan contoh pemakaian paket dengan perintah from kalkulator import Penambahan disertai dengan contoh kode lainnya. Penggunaan paket from namafile import, bergungsi untuk memanggil fungsi yang terdapat di file yang di panggil dengan menggunakan import kita hanya memanggil nama filenya saja tanpa harus menulis atau mengetik kembali hasil yang di buat.

```

1 from kalkulator import Penambahan
2
3 hasil = Penambahan(20, 10)
4 print(hasil)

```

6. Jelaskan dengan contoh kodenya, pemakaian paket fungsi apabila le library ada di dalam folder. Pemakaian paket gabungan atau kumpulan fungsi-fungsi dalam satu hasil yang dibuat. contoh kodenya adalah sebagai berikut :

```

1
2 #6
3 from folder import kalkulator

```

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

```

1 from folder.Jurusan import Jurusan
2
3 mhs1 = Jurusan("TI", "fernandols")
4 mhs2 = Jurusan("TI", "bebas")
5
6 mhs1.tampilkanProfil()
7 mhs2.tampilkanProfil()
8
9 print("Total mahasiswa ", Jurusan.jumlahMahasiswa)

```

### 3.22.0.2 Ketrampilan Pemrograman

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

```

1 def printNPM(npm):
2
3     npm = list(str(npm))
4
5     angka1 = {"0": "+++++ ", "1": " ++", "2": "+++++ ", "3": "
6     ++++++ ", "4": "    ++", "5": "++++++", "6": "+++++ ", "
7     7": "++++++", "8": "+++++ ", "9": "+++++ "}
8     angka2 = {"0": "+++ ++", "1": "++++", "2": "++    ++", "3": "
9     ++ ++", "4": "    +++++", "5": "++    ", "6": "+++    ",
10    "7": "    ++ ", "8": "+++ ++", "9": "++    ++"}
11    angka3 = {"0": "+++ ++", "1": " ++", "2": "    ++ ", "3": "
12    +++++ ", "4": "    ++ ++", "5": "++++++ ", "6": "++++++ ", "
13    7": "    ++ ", "8": "    ++ ++", "9": "++    ++"}
14    angka4 = {"0": "+++ ++", "1": " ++", "2": "    ++ ", "3": "
15    +++++ ", "4": "++++++", "5": "    ++", "6": "+++ ++", "
16    7": "    ++ ", "8": "    ++ ++", "9": "++++++"}
17    angka5 = {"0": "+++ ++", "1": " ++", "2": "    ++ ", "3": "
18    ++ ++", "4": "    ++", "5": "++ ++", "6": "+++ ++", "
19    7": "    ++ ", "8": "+++ ++", "9": "    ++"}
20    angka6 = {"0": "+++++ ", "1": " ++", "2": "++++++", "3": "
21    ++++++ ", "4": "    ++", "5": "+++++ ", "6": "+++++ ", "
22    7": "    ++ ", "8": "+++++ ", "9": "+++++ "}
23
24    hasil1 = []
25    hasil2 = []
26    hasil3 = []
27    hasil4 = []
28    hasil5 = []
29    hasil6 = []
30
31    for x in npm:
32        hasil1.append(angka1[x])
33        hasil2.append(angka2[x])
34        hasil3.append(angka3[x])
35        hasil4.append(angka4[x])
36        hasil5.append(angka5[x])
37        hasil6.append(angka6[x])
38
39    print(*hasil1, sep=' ')
40    print(*hasil2, sep=' ')
41    print(*hasil3, sep=' ')
42    print(*hasil4, sep=' ')
43    print(*hasil5, sep=' ')
44    print(*hasil6, sep=' ')
45
46    printNPM(input("Masukan NPM anda: "))

```

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

```

1 def printNPM(npm):
2     ulang = 1
3     sampai = int(npm[5:7])
4     while(ulang <= sampai):
5         print("Halo, "+str(npm)+" apa kabar?")
6         ulang += 1
7
8 printNPM(input("Harap MASUKAN NPM: "))

```

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

```

1 def printNPMTigaNomor(npm):
2     ulang = 1
3     sampai = list(map(int, npm[4:7]))
4     sampai = sum(sampai)
5     while(ulang <= sampai):
6         print("Halo, "+str(npm[-3:])+" apa kabar?")
7         ulang += 1
8
9 printNPMTigaNomor(input("Masukan NPM anda: "))

```

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

```

1 def printNPMAngkaUruttiga(npm):
2     print("Output:")
3     print("Halo, "+str(npm[-3])+" apa kabar?")
4
5 printNPMAngkaUruttiga(input("Input: "))

```

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

```

1 def printNPMBerbaris(npm):
2     npm = list(map(int, npm))
3     for n in npm:
4         print(n)
5
6 printNPMBerbaris(input("Masukan NPM anda: "))

```

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

```

1 def printPenjumlahanNPM(npm):
2     npm = list(map(int, npm))

```

```

3     hasil = 0
4     for x in npm:
5         hasil += x
6
7     print(hasil)
8
9 printPenjumlahanNPM(input("Masukan NPM anda: "))

```

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

```

1 def printPerkalianNPM(npm):
2     npm = list(map(int, npm))
3     hasil = 0
4     for x in npm:
5         hasil *= x
6
7     print(hasil)
8
9 printPerkalianNPM(input("Masukan NPM anda: "))

```

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

```

1 def printAngkaGenapNPM(npm):
2     npm = list(map(int, npm))
3     for n in npm:
4         if(n % 2 == 0):
5             if(n != 0):
6                 print(n, end = "")
7
8 printAngkaGenapNPM(input("Masukan NPM anda: "))

```

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

```

1 def printAngkaGanjilNPM(npm):
2     npm = list(map(int, npm))
3     for n in npm:
4         if(n % 2 != 0):
5             print(n, end = "")
6
7 printAngkaGanjilNPM(input("Masukan NPM anda: "))

```

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

```

1 def printAngkaPrimaNPM(npm):
2     npm = list(map(int, npm))
3     prima = []
4     for n in npm:
5         isPrime = True

```



```

6         if n == 0 or n == 1:
7             isPrime = False
8         for x in range(2, n):
9             if n % x == 0:
10                isPrime = False
11        if isPrime:
12            prima.append(n)
13
14    for p in prima:
15        print(p, end = " ")
16
17 printAngkaPrimaNPM(input("Masukan NPM anda: "))

```

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

```

1 lib = __import__('3lib')
2
3 npm = "1174072"
4
5 lib.printNPM(npm)
6 lib.printNPMDuaDijit(npm)
7 lib.printNPMTigaNomor(npm)
8 lib.printNPMAngkaUrutTiga(npm)
9 lib.printNPMBerbaris(npm)
10 lib.printPenjumlahanNPM(npm)
11 lib.printPerkalianNPM(npm)
12 lib.printAngkaGenapNPM(npm)
13 lib.printAngkaGenjilNPM(npm)
14 lib.printAngkaPrimaNPM(npm)
15 print()

```

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

```

1 from kelas3lib import kelas3lib
2
3 npm = "1174072"
4
5 k3lib = kelas3lib(npm)
6
7 k3lib.printNPM()
8 k3lib.printNPMDuaDijit()
9 k3lib.printNPMTigaNomor()
10 k3lib.printNPMAngkaUrutTiga()
11 k3lib.printNPMSatuPersatu()
12 k3lib.printPenjumlahanNPM()
13 k3lib.printPerkalianNPM()

```

```

14 k3lib.printAngkaGenapNPM()
15 k3lib.printAngkaGanjilNPM()
16 k3lib.printAngkaPrimaNPM()

```

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

```

1 def panggil(nama):
2     try:
3         print("Hallo , "+str(nama))
4     except:
5         print("Terjadi kesalahan")
6
7 panggil(input("Masukan nama anda: "))

```

## 3.23 Handi Hermawan

### 3.23.1 Pemahaman Teori

#### 1. Fungsi

- Fungsi adalah blok program untuk melakukan tugas-tugas tertentu yang dilakukan berulang dan dapat digunakan berulang kali dari tempat lain di dalam program.
- Inputan fungsi adalah inputan yang berasal dari luar fungsi yang akan di proses di dalam fungsi itu sendiri.
- Kembalian fungsi adalah untuk mengembalikan suatu nilai ekspresi dari proses yang dilakukan fungsi.

#### 2. Pengertian paket dan cara pemanggilannya serta contoh kode programnya.

Paket atau library adalah file yang berisi kode program python yang bisa digunakan berulang dimana paket itu dipanggil.

Cara pemanggilan paket atau library yaitu dengan meng-import paket atau library yang akan digunakan. Lalu panggil dengan cara mendefinisikan nama-paket.namafunksinya.

#### 3. Kelas

- Kelas Kelas adalah cetak biru atau prototipe dari objek dimana kita mendefinisikan atribut dari suatu objek. Contoh penggunaan kelas di python.
- Objek Objek adalah instansi atau perwujudan dari sebuah kelas.
- Atribut Atribut adalah variabel yang menyimpan data yang berhubungan dengan kelas dan objeknya.

- Method Metode adalah fungsi yang didefinisikan di dalam suatu kelas.

#### 4. Cara pemanggilan library kelas, dan contoh kode programnya.

Berikut ini adalah cara pemanggilan library kelas dari instansi dan pemakaiannya. Library kelasnya adalah Mahasiswa dari file Mahasiswa.py. Lalu dipanggil dengan import. Kemudian instansi dengan mhs1 dan mhs1, dengan 2 parameter.

#### 5. Penjelasan pemakaian paket disertai dengan contoh kode programnya.

Berikut ini adalah contoh pemakaian paket dengan perintah from kalkulator import Penambahan. Setelah mengimport paketnya, lalu panggil fungsi penambahannya.

#### 6. Contoh kode pemakaian paket fungsi apabila file library ada di dalam folder. Berikut ini adalah pemakaian paket fungsi apabila file library ada di dalam folder.

#### 7. Contoh kode pemakaian paket kelas apabila file library ada di dalam folder. Berikut ini adalah pemakaian paket kelas apabila file library ada di dalam folder.

### 3.23.2 Ketrampilan Pemrograman

#### 1. Jawaban soal No. 1

```

1 #Jawaban No. 1
2 def printNPM(npm):
3
4     npm = list(str(npm))
5
6     angka1 = {"0": "##### ", "1": " ##", "2": " ##### ", "3": "
##### ", "4": "    ##", "5": "##### ", "6": " ##### ", "
7": "##### ", "8": " ##### ", "9": " #####"}
7     angka2 = {"0": "###  ##", "1": "####", "2": "##  ##", "3": "
##  ##", "4": " #####", "5": "##  ", "6": "###  ",
"7": "    ## ", "8": "###  ##", "9": "##  ##"}
8     angka3 = {"0": "###  ##", "1": "  ##", "2": "    ## ", "3": "
#### ", "4": "  ##  ##", "5": "##### ", "6": "##### ", "
7": "    ## ", "8": "  ##  ##", "9": "##  ##"}
9     angka4 = {"0": "###  ##", "1": "  ##", "2": "    ## ", "3": "
#### ", "4": "##### ", "5": "    ## ", "6": "###  ##", "
7": "  ##  ", "8": "  ##  ##", "9": "  #####"}
10    angka5 = {"0": "###  ##", "1": "  ##", "2": "    ## ", "3": "
##  ##", "4": "    ##", "5": "##  ##", "6": "###  ##",
"7": "  ##  ", "8": "###  ##", "9": "    ##"}
11    angka6 = {"0": "##### ", "1": "  ##", "2": "##### ", "3": "
##### ", "4": "    ##", "5": " ##### ", "6": " ##### ", "
7": "  ##  ", "8": " ##### ", "9": " ##### "}
12
13    hasil1 = []
14    hasil2 = []
15    hasil3 = []
16    hasil4 = []
17    hasil5 = []

```

```

18     hasil6 = []
19
20     for x in npm:
21         hasil1.append(angka1[x])
22         hasil2.append(angka2[x])
23         hasil3.append(angka3[x])
24         hasil4.append(angka4[x])
25         hasil5.append(angka5[x])
26         hasil6.append(angka6[x])
27
28     print(*hasil1, sep=' ')
29     print(*hasil2, sep=' ')
30     print(*hasil3, sep=' ')
31     print(*hasil4, sep=' ')
32     print(*hasil5, sep=' ')
33     print(*hasil6, sep=' ')
34
35 printNPM(input("Masukan NPM anda: "))

```

**Listing 3.22** Jawaban soal No. 1 Ketrampilan Pemrograman.

## 2. Jawaban soal No. 2

```

1 #Jawaban No. 2
2 def printNPMDuaDijit(npm):
3     ulang = 1
4     sampai = int(npm[5:7])
5     while(ulang <= sampai):
6         print("Halo, "+str(npm)+" apa kabar?")
7         ulang += 1
8
9 printNPMDuaDijit(input("Masukan NPM anda: "))

```

**Listing 3.23** Jawaban soal No. 2 Ketrampilan Pemrograman.

## 3. Jawaban soal No. 3

```

1 #Jawaban No. 3
2 def printNPMTigaDijit(npm):
3     ulang = 1
4     sampai = list(map(int, npm[4:7]))
5     sampai = sum(sampai)
6     while(ulang <= sampai):
7         print("Halo, "+str(npm[-3:])+ " apa kabar?")
8         ulang += 1
9
10 printNPMTigaDijit(input("Masukan NPM anda: "))

```

**Listing 3.24** Jawaban soal No. 3 Ketrampilan Pemrograman.

## 4. Jawaban soal No. 4

```

1 #Jawaban No. 4
2 def printNPMDigitKetiga(npm):
3     print("Output:")
4     print("Halo, "+str(npm[-3])+ " apa kabar?")

```

```

5
6 printNPMDigitKetiga(input("Input: "))

```

**Listing 3.25** Jawaban soal No. 4 Ketrampilan Pemrograman.

#### 5. Jawaban soal No. 5

```

1 #Jawaban No. 5
2 def printNPMSatuPersatu(npm):
3     npm = list(map(int, npm))
4     for n in npm:
5         print(n)

```

**Listing 3.26** Jawaban soal No. 5 Ketrampilan Pemrograman.

#### 6. Jawaban soal No. 6

```

1 #Jawaban No. 6
2 def printNMPPenjumlahan(npm):
3     npm = list(map(int, npm))
4     hasil = 0
5     for x in npm:
6         hasil += x
7
8     print(hasil)
9
10 printNMPPenjumlahan(input("Masukan NPM anda: "))

```

**Listing 3.27** Jawaban soal No. 6 Ketrampilan Pemrograman.

#### 7. Jawaban soal No. 7

```

1 #Jawaban No. 7
2 def printNMPPerkalian(npm):
3     npm = list(map(int, npm))
4     hasil = 0
5     for x in npm:
6         hasil *= x
7
8     print(hasil)
9
10 printNMPPerkalian(input("Masukan NPM anda: "))

```

**Listing 3.28** Jawaban soal No. 7 Ketrampilan Pemrograman.

#### 8. Jawaban soal No. 8

```

1 #Jawaban No. 8
2 def printNPMDigitGenap(npm):
3     npm = list(map(int, npm))
4     for n in npm:
5         if(n % 2 == 0):
6             if(n != 0):
7                 print(n, end = "")
8
9 printNPMDigitGenap(input("Masukan NPM anda: "))

```

**Listing 3.29** Jawaban soal No. 8 Ketrampilan Pemrograman.

## 9. Jawaban soal No. 9

```

1 #Jawaban No. 9
2 def printNPMDijitGanjil(npm):
3     npm = list(map(int, npm))
4     for n in npm:
5         if (n % 2 != 0):
6             print(n, end = "")
7
8 printNPMDijitGanjil(input("Masukan NPM anda: "))

```

**Listing 3.30** Jawaban soal No. 9 Ketrampilan Pemrograman.

## 10. Jawaban soal No. 10

```

1 #Jawaban No. 10
2 def printNPMDijitPrima(npm):
3     npm = list(map(int, npm))
4     prima = []
5     for n in npm:
6         isPrime = True
7         if n == 0 or n == 1:
8             isPrime = False
9         for x in range(2, n):
10             if n % x == 0:
11                 isPrime = False
12         if isPrime:
13             prima.append(n)
14
15     for p in prima:
16         print(p, end = "")
17
18 printNPMDijitPrima(input("Masukan NPM anda: "))

```

**Listing 3.31** Jawaban soal No. 10 Ketrampilan Pemrograman.

## 11. Jawaban soal No. 11

```

1 def namaFungsi(inputanFungsi):
2     return inputanFungsi
3
4 output = namaFungsi("Kembalian Fungsi")
5 print(output)
6
7 #Jawaban No. 2
8 import math
9 print("Nilai pi adalah: ", math.pi)
10
11 #Jawaban No. 3
12 class Mahasiswa:
13     jumlahMahasiswa = 0
14
15     def __init__(self, npm, nama):
16         self.npm = npm
17         self.nama = nama
18         Mahasiswa.jumlahMahasiswa += 1
19

```

```

20     def tampilkanProfil(self):
21         print("NPM :", self.npm)
22         print("Nama :", self.nama)
23         print()
24
25     mahasiswa1 = Mahasiswa("1174080", "Handi Hermawan")
26     mahasiswa2 = Mahasiswa("1174076", "Difa Al")
27
28     mahasiswa1.tampilkanProfil()
29     mahasiswa2.tampilkanProfil()
30
31     print("Total mahasiswa adalah ", Mahasiswa.jumlahMahasiswa)

```

**Listing 3.32** Jawaban soal No. 11 Ketrampilan Pemrograman.

## 12. Jawaban soal No. 12

```

1     def namaFungsi(inputanFungsi):
2         return inputanFungsi
3
4     output = namaFungsi("Kembalian Fungsi")
5     print(output)
6
7     #Jawaban No. 2
8     import math
9     print("Nilai pi adalah: ", math.pi)
10
11    #Jawaban No. 3
12    class Mahasiswa:
13        jumlahMahasiswa = 0
14
15        def __init__(self, npm, nama):
16            self.npm = npm
17            self.nama = nama
18            Mahasiswa.jumlahMahasiswa +=1
19
20        def tampilkanProfil(self):
21            print("NPM :", self.npm)
22            print("Nama :", self.nama)
23            print()
24
25        mahasiswa1 = Mahasiswa("1174080", "Handi Hermawan")
26        mahasiswa2 = Mahasiswa("1174076", "Difa Al")
27
28        mahasiswa1.tampilkanProfil()
29        mahasiswa2.tampilkanProfil()
30
31        print("Total mahasiswa adalah ", Mahasiswa.jumlahMahasiswa)

```

**Listing 3.33** Jawaban soal No. 12 Ketrampilan Pemrograman.

### 3.23.3 Ketrampilan Penanganan Error

1. Peringatan error yang ditemukan dan penjelasannya serta buat sebuah fungsi try except untuk menanggulangi error.

Peringatan error di praktek ketiga ini, yaitu:

- **Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- **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 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 terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.





## BAB 4

---

# PENGELOLAAN FILE CSV

---

### 4.1 D. Irga B. Naufal Fakhri

#### 4.1.1 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
2 11714066,D. Irga B.Naufal Fakhri,D4 Teknik Informatika 2C
```

##### 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
2 11714066,D. Irga B.Naufal Fakhri,D4 Teknik Informatika 2C
```

### 4. Jelaskan sejarah library csv

Module csv mengimplementasikan kelas untuk membaca dan menulis data kedalam format CSV. Hal ini memungkinkan programmer untuk "tuliskan 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():
2     with open('1174066.csv') as csv_file:
3         csv_reader = csv.reader(csv_file, delimiter=',')
4         line_count = 0
5         for row in csv_reader:
6             if line_count == 0:
```

```

7         print(f'Column names are {", ".join(row)}')
8         line_count += 1
9     else:
10        print(f'\tNomor NPM: {row[0]} Nama: {row[1]}
        Kelas: {row[2]}')
11        line_count += 1
12        print(f'Processed {line_count} lines.')
13 read()

```

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

```

#### ■ 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():
2     with open('1174066.csv', mode='r') as csv_file:
3         csv_reader = csv.DictReader(csv_file)
4         line_count = 0
5         for row in csv_reader:
6             if line_count == 0:
7                 print(f'Column names are {", ".join(row)}')
8                 line_count += 1
9                 print(f'\tNomor NPM: {row["npm"]} Nama: {row["nama"]}
        Kelas: {row["kelas"]} ')
10                line_count += 1
11                print(f'Processed {line_count} lines.')
12 readdict()

```

## 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():
2     df = pandas.read_csv('1174066.csv')
3     print(df)
4 readpanda()
```

### ▪ `to_csv`

Berfungsi untuk mengedit data didalam csv dan menulisnya kedalam file csv

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

# Index

---

disruptif, xxxi  
modern, xxxi