

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 Tunnisia

Diana Asri Wijayanti

Desain sampul dan Tata letak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasisih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

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

*'Jika Kamu tidak dapat
menahan lelahnya
belajar, Maka kamu harus
sanggup menahan
perihnya Kebodohan.'*

Imam Syafi'i

CONTRIBUTORS

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

CONTENTS IN BRIEF

1 SEJARAH DAN KARAKTERISTIK PYTHON	1
2 Judul Bagian Kedua	87
3 Judul Bagian Ketiga	179

DAFTAR ISI

Daftar Gambar	xv
Daftar Tabel	xxi
Foreword	xxxv
Kata Pengantar	xxxvii
Acknowledgments	xxxix
Acronyms	xli
Glossary	xliii
List of Symbols	xlvi
Introduction	xlvii
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
1 SEJARAH DAN KARAKTERISTIK PYTHON	1
1.1 Dwi Yulianingsih	1
1.1.1 Sejarah Phyton	1
1.1.2 Instalasi Anaconda	2
1.1.3 Menggunakan Spyder	3

1.2	Dwi Septiani Tsaniyah	3
1.2.1	Sejarah Python	3
1.2.2	Instalasi Anaconda	4
1.2.3	Penggunaan Spyder	4
1.3	Kadek Diva Krishna Murti	4
1.3.1	Sejarah Python	4
1.3.2	Instalasi Anaconda	5
1.3.3	Penggunaan Spyder	10
1.4	Harun Ar-Rasyid	16
1.4.1	Sejarah	16
1.4.2	Instalasi Anaconda	16
1.4.3	Menggunakan Spyder	19
1.5	Rahmatul Ridha / 1144124	20
1.5.1	Sejarah Python	20
1.5.2	Instalasi Anaconda	20
1.6	Dwi Yulianingsih	23
1.6.1	Sejarah Phyton	23
1.6.2	Instalasi Anaconda	25
1.6.3	Menggunakan Spyder	26
1.7	Dwi Septiani Tsaniyah	27
1.7.1	Sejarah Python	27
1.7.2	Instalasi Anaconda	27
1.7.3	Penggunaan Spyder	27
1.8	Kadek Diva Krishna Murti	28
1.8.1	Sejarah Python	28
1.8.2	Instalasi Anaconda	28
1.8.3	Penggunaan Spyder	33
1.9	Harun Ar-Rasyid	39
1.9.1	Sejarah	39
1.9.2	Instalasi Anaconda	39
1.9.3	Menggunakan Spyder	42
1.10	Felix Lase	43
1.10.1	sejarah	43
1.10.2	Instalasi Anaconda	43
1.10.3	Menggunakan Spyder	45
1.11	Oniwaldus Bere Mali	46
1.11.1	Sejarah Python	46
1.11.2	Instalasi Anaconda	46

1.11.3	Menggunakan Spyder	49
1.12	DezhaAidiMartha	50
1.12.1	Sejarah	50
1.12.2	InstalasiAnaconda	50
1.12.3	Spyder	52
1.13	Choirul Anam	52
1.13.1	Sejarah Phyton	52
1.13.2	Instalasi Anaconda	53
1.13.3	Menggunakan Spyder	55
1.14	Nico Ekklesia Sembiring	55
1.14.1	sejarah pyton	55
1.14.2	Menginstal Anaconda	56
1.14.3	Menggunakan Spyder	58
1.15	Damara Benedicta	60
1.15.1	Sejarah	60
1.15.2	Instalasi Anaconda	60
1.15.3	(63
1.16	Damara Benedicta	63
1.16.1	Sejarah	63
1.16.2	Instalasi Anaconda	63
1.16.3	(65
1.17	Arjun Yuda Firwanda	67
1.17.1	Sejarah Python	67
1.17.2	Instalasi Anaconda	67
1.17.3	Spyder Python	69
1.18	muhammad dzihan al-banna	71
1.18.1	Instalasi Anaconda	71
1.18.2	Menggunakan Spyder	74
1.19	Muhammad Fahmi	74
1.19.1	Sejarah Pyton	74
1.19.2	Instalasi Anaconda	74
1.19.3	Penggunaan Spider	76
1.20	Evietania Charis Sujadi	76
1.21	Python	76
1.21.1	Background	76
1.21.2	Problems	77
1.21.3	Objective and Contribution	77
1.21.4	Scoop and Environtment	77

1.22	Habib Abdul Rasyid	77
1.22.1	Sejarah Python	77
1.22.2	Instalasi Anaconda	78
1.22.3	Spyder	79
1.23	Muhammad Tomy Nur Maulidy	80
1.23.1	Sejarah Python	80
1.23.2	Instalasi Anaconda	80
1.23.3	(83
1.24	Doli Jonviter NT Simbolon / 1154016	83
1.24.1	Sejarah Python	83
1.24.2	Instalasi Aconda	83
1.24.3	Menggunakan Spyder	85

2 Judul Bagian Kedua 87

2.1	Harun Ar - Rasyid	87
2.1.1	Teori	87
2.1.2	Ketrampilan Pemrograman	90
2.1.3	Ketrampilan Penanganan Error	92
2.2	Dwi Yulianingsih	92
2.2.1	Teori	92
2.2.2	praktek	95
2.2.3	Keterampilan dan penanganan eror	97
2.3	Kadek Diva Krishna Murti	97
2.3.1	Teori	97
2.3.2	Ketrampilan Pemrograman	103
2.3.3	Ketrampilan Penanganan Error	105
2.4	Dezha Aidil Martha	105
2.4.1	Teori	105
2.4.2	praktek	107
2.4.3	Keterampilan dan penanganan eror	109
2.5	Evietania Charis Sujadi	109
2.5.1	Teori	109
2.5.2	praktek	111
2.5.3	Keterampilan dan penanganan eror	113
2.6	Perintah Navigasi	113
2.7	Damara Benedikta	113
2.7.1	Teori	113
2.7.2	praktek	116

2.7.3	Keterangan dan Penanganan eror	118
2.8	Felix Setiawan Lase	118
2.8.1	Teori	118
2.8.2	Ketrampilan Pemrograman	120
2.8.3	Ketrampilan Penanganan Error	122
2.9	Perintah Navigasi	122
2.10	Muhammad Dzihan Al-Banna	122
2.10.1	Pemrograman Dasar	122
2.10.2	Varialbel	122
2.10.3	input dan output	123
2.10.4	operator aritmatika	124
2.10.5	Perulangan	124
2.10.6	If Else	125
2.10.7	Penanganan Error	125
2.10.8	Try Exception	125
2.10.9	Ketrampilan Penanganan Error	126
2.11	Nico Ekklesia Sembiring	127
2.11.1	Tugas Teori	127
2.11.2	Tugas Ketrampilan Pemrograman	131
2.11.3	Ketrampilan Penanganan Error	132
2.12	Muhammad Tomy Nur Maulidy	133
2.12.1	Teori	133
2.12.2	Ketrampilan Pemrograman	136
2.12.3	Error	138
2.13	Choirul Anam	138
2.13.1	Teori	138
2.13.2	Ketrampilan Pemrograman	141
2.13.3	Ketrampilan Penanganan Error	143
2.14	Oniwaldus Bere Mali	144
2.14.1	Teori	144
2.14.2	Ketrampilan Pemrograman	146
2.14.3	Ketrampilan Penanganan Error	149
2.15	Habib Abdul Rasyid	149
2.15.1	Teori	149
2.15.2	Ketrampilan Pemrograman	152
2.15.3	Ketrampilan Penanganan Error	154
2.16	Arjun Yuda Firwanda	154
2.16.1	Teori	154

2.16.2	Keterampilan Pemrograman	158
2.16.3	Keterampilan Penanganan Error	159
2.17	Muh. Rifky Prananda	160
2.17.1	Teori	160
2.17.2	Ketrampilan Pemrograman	162
2.17.3	Ketrampilan Penanganan Error	164
2.18	Muhammad Fahmi	164
2.18.1	Teori	164
2.18.2	Keterampilan Pemrograman	171
2.18.3	Keterampilan Penanganan Error	173
2.19	Rahmatul Ridha	173
2.19.1	Teori	173
2.19.2	praktek	176
2.19.3	Keterangan dan Penanganan eror	178
3	Judul Bagian Ketiga	179
3.1	Harun Ar - Rasyid	179
3.2	Evietania	185
3.3	Kadek Diva Krishna Murti	191
3.3.1	Pemahaman Teori	191
3.3.2	Ketrampilan Pemrograman	195
3.3.3	Ketrampilan Penanganan Error	199

DAFTAR GAMBAR

1.1	gambar1	2
1.2	gambar2	2
1.3	gambar3	2
1.4	gambar4	3
1.5	gambar5	3
1.6	gambar6	3
1.7	Tampilan Awal	17
1.8	License Agreement	17
1.9	Pemilihan User	17
1.10	Pemilihan Direktori Penyimpanan	18
1.11	Pemilihan Opsi	18
1.12	Proses Instal	18
1.13	Proses Instal Selesai	18

1.14	Penawaran Instal MS VSC	19
1.15	Instalasi Selesai	19
1.16	Ini adalah tampilan spyder	19
1.17	Download Anaconda	20
1.18	Proses 1	21
1.19	proses 2	21
1.20	Proses 3	22
1.21	Proses 4	22
1.22	Proses 5	23
1.23	Proses 6	23
1.24	Proses 7	24
1.25	Proses 8	24
1.26	gambar1	25
1.27	gambar2	25
1.28	gambar3	25
1.29	gambar4	26
1.30	gambar5	26
1.31	gambar6	26
1.32	Tampilan Awal	40
1.33	License Agreement	40
1.34	Pemilihan User	40
1.35	Pemilihan Direktori Penyimpanan	41
1.36	Pemilihan Opsi	41
1.37	Proses Instal	41
1.38	Proses Instal Selesai	41
1.39	Penawaran Instal MS VSC	42
1.40	Instalasi Selesai	42
1.41	Ini adalah tampilan spyder	42

1.42	Tampilan Awal	43
1.43	License	44
1.44	Proses	44
1.45	Proses	44
1.46	Proses	44
1.47	Proses	45
1.48	Proses	45
1.49	Proses	45
1.50	Proses	46
1.51	Klik Next	47
1.52	Klik pada I Agree	47
1.53	Ceklis pada Add Anaconda to my PATH environment varable dan Register Anaconda as my default Python 3.7 .selanjutnya klik Install	48
1.54	Proses Instali	48
1.55	klik next	49
1.56	Klik install microsoft vscode	49
1.57	Instalasi Selesai	49
1.58	gambar1	53
1.59	gambar2	54
1.60	gambar3	54
1.61	gambar4	54
1.62	gambar5	55
1.63	gambar6	55
1.64	File Anaconda	56
1.65	Tampilan Awal	57
1.66	Persetujuan Lisensi	57
1.67	Tipe Instalasi	57

1.68	Pilih Lokasi	57
1.69	Advanced Options	58
1.70	Instal Selesai	58
1.71	Instal VSCode	58
1.72	Instalasi Selesai	59
1.73	Kode	59
1.74	Hasil Kode	59
1.75	Tampilan Awal	60
1.76	License Agreement	60
1.77	Pemilihan User	61
1.78	Pemilihan Direktori Penyimpanan	61
1.79	Pemilihan Opsi	61
1.80	Proses Instal	62
1.81	Proses Instal Selesai	62
1.82	Penawaran Instal MS VSC	62
1.83	Penawaran Instal MS VSC	62
1.84	Instalasi Selesai	63
1.85	Tampilan Awal	64
1.86	License Agreement	64
1.87	Pemilihan User	64
1.88	Pemilihan Direktori Penyimpanan	64
1.89	Pemilihan Opsi	65
1.90	Proses Instal	65
1.91	Proses Instal Selesai	65
1.92	Penawaran Instal MS VSC	66
1.93	Penawaran Instal MS VSC	66
1.94	Instalasi Selesai	66
1.95	File Instalasi Anaconda	67

1.96	Tampilan Awal Instalasi Anconda	68
1.97	Tampilan Licence Anaconda	68
1.98	Memilih Type Instalasi	68
1.99	Penyimpanan Instalasi Anaconda	68
1.100	Opsi Anaconda	69
1.101	Proses Installing	69
1.102	Installing Completed	69
1.103	Hak Akses Anaconda	70
1.104	Instalasi Selesai	70
1.105	Spyder Python	70
1.106	Instal file	71
1.107	Lisensi	72
1.108	Pemilihan User	72
1.109	Pemilihan Direktori Penyimpanan	72
1.110	Pemilihan Opsi	72
1.111	Proses Instalasi	73
1.112	Instal selesai	73
1.113	Penawaran Instal MS VSC	73
1.114	Instalasi Selesai	73
1.115	instalasi	78
1.116	instalasi	78
1.117	instalasi	78
1.118	instalasi	79
1.119	instalasi	79
1.120	instalasi	79
1.121	instalasi	80
1.122	Tampilan Awal	80
1.123	License Agreement	81

1.124	Pemilihan User	81
1.125	Pemilihan Direktori Penyimpanan	81
1.126	Pemilihan Opsi	82
1.127	Proses Instal	82
1.128	Proses Instal Selesai	82
1.129	Penawaran Instal MS VSC	82
1.130	Penawaran Instal MS VSC	83
1.131	Instalasi Selesai	83
1.132	Tampilan Awal	84
1.133	License	84
1.134	Proses	84
1.135	Proses	85
1.136	Proses	85
1.137	Proses	85
1.138	Proses	86
1.139	Proses	86
1.140	Proses	86

DAFTAR TABEL

Listings

src/1174027.py	87
src/1174027.py	87
src/1174027.py	88
src/1174027.py	90
src/1174027.py	90
src/1174027.py	91
src/1174027.py	92
src/1174027.py	92
src/1174027.py	92

src/1174027.py	92
src/err2.py	92
src/teori.py	93
src/teori.py	93
src/teori.py	93
src/teori.py	94
src/teori.py	94
src/teori.py	95
src/1174009.py	95
src/1174009.py	96
src/1174009.py	97
src/1174009.py	97
src/1174009.py	97
src/eror.py	97
2.1 Contoh kode penggunaan Boolean.	97
2.2 Contoh kode penggunaan String.	97
2.3 Contoh kode penggunaan Integer.	97
2.4 Contoh kode penggunaan Float.	98
2.5 Contoh kode penggunaan Hexadecimal.	98
2.6 Contoh kode penggunaan Complex.	98
2.7 Contoh kode penggunaan List.	98
2.8 Contoh kode penggunaan Tuple.	98
2.9 Contoh kode penggunaan Set.	98
2.10 Contoh kode penggunaan Dictionary.	98
2.11 Contoh kode input dan output.	99
2.12 Contoh kode operasi pertambahan.	99
2.13 Contoh kode operasi pengurangan.	99
2.14 Contoh kode operasi perkalian.	99
2.15 Contoh kode operasi pembagian.	99
2.16 Contoh kode operasi modulus.	100
2.17 Contoh kode operasi perpangkatan.	100
2.18 Contoh kode operasi pembulatan hasil bagi.	100

2.19 Contoh kode konversi string ke integer.	100
2.20 Contoh kode konversi integer ke string.	100
2.21 Contoh kode penggunaan while loop.	100
2.22 Contoh kode penggunaan for loop.	101
2.23 Contoh kode penggunaan nested loop.	101
2.24 Contoh kode penggunaan if.	101
2.25 Contoh kode penggunaan if else.	101
2.26 Contoh kode penggunaan elif.	101
2.27 Contoh kode penggunaan kondisi di dalam kondisi.	102
2.28 Contoh kode penggunaan try except.	102
src/1174006.py	103
src/1174006.py	104
src/1174006.py	105
src/1174025_teori.py	105
src/1174025_teori.py	106
src/1174025_teori.py	107
src/1174025_teori.py	107
src/1174025_teori.py	107
src/1174025_praktek.py	107
src/1174025_praktek.py	108

src/1174025_praktek.py	109
src/erro2.py	109
src/1174051_teori.py	109
src/1174051_teori.py	109
src/1174051_teori.py	110
src/1174051_teori.py	111
src/1174051_teori.py	111
src/1174051_praktek.py	111
src/1174051_praktek.py	111
src/1174051_praktek.py	112
src/1174051_praktek.py	113
src/1174051_praktek.py	113
src/eror.py	113
src/teori1.py	113
src/teori1.py	114
src/teori1.py	114
src/teori1.py	115
src/teori1.py	116
src/1174012.py	117

src/1174012.py	117
src/error1.py	118
src/1174026.py	119
src/1174026.py	120
src/1174026.py	120
src/1174026.py	120
src/1174026.py	121
src/1174026.py	122
src/felix.py	122
src/dzihan.py	123
src/dzihan.py	123
src/dzihan.py	124
src/dzihan.py	125
src/dzihan.py	125
src/dzihan.py	125
src/tugasweb2.py	126
src/1174096.py	127

src/1174096.py	127
src/1174096.py	128
src/1174096.py	128
src/1174096.py	128
src/1174096.py	129
src/1174096.py	130
src/1174096.py	130
src/1174096.py	131
src/1174096.py	132
src/2err.py	132
src/1174031.py	133
src/1174031.py	134
src/1174031.py	134
src/1174031.py	134
src/1174031.py	135
src/1174031.py	136
src/1174031.py	136
src/1174031.py	136

src/1174031.py	137
src/1174031.py	138
src/err26.py	138
src/1174004.py	139
src/1174004.py	140
src/1174004.py	141
src/1174004.py	141
src/1174004.py	142
src/1174004.py	143
src/1174027.py	143
src/1174004.py	143
src/1174005.py	144
src/1174005.py	145
src/1174005.py	145
src/1174005.py	146
src/1174005.py	147
src/1174005.py	147

src/1174008.py	155
src/1174008.py	156
src/1174008.py	156
src/1174008.py	156
src/1174008.py	157
src/1174008.py	158
src/1174008.py	159
src/rrr2e.py	160
src/1174017.py	162
src/1174017.py	162
src/1174017.py	163
src/1174017.py	164
src/err2.py	164
2.29 Variabel	165

2.30 Tipe Data	165
2.31 Input Output	167
2.32 Operator Dasar Aritmatika	167
2.33 Operator Dasar Aritmatika	168
2.34 Operator Dasar Aritmatika	169
2.35 Operator Dasar Aritmatika	170
src/1174021.py	171
src/1174021.py	172
src/2err_1174021.py	173
src/1144124_Teori.py	173
src/1144124_Teori.py	174
src/1144124_Teori.py	175
src/1144124_Teori.py	175
src/1144124_Teori.py	176
src/1144124_Praktek.py	176
src/1144124_Praktek.py	176
src/1144124_Praktek.py	177
src/1144124_Praktek.py	178
src/1144124_Praktek.py	178
src/error.py	178
src/1174027.py	179
src/1174027.py	179
src/1174027.py	180

src/1174027.py	180
src/1174027.py	180
src/1174027.py	181
src/1174027.py	183
src/1174027.py	184
src/main.py	185
src/main.py	185
src/1174027.py	185
src/1174051_praktek.py	185
src/1174051_praktek.py	186
src/1174051_praktek.py	187
src/1174051_praktek.py	189
src/1174051_praktek.py	189
src/1174051_praktek.py	189
src/1174051_praktek.py	190
src/1174051_praktek.py	190
src/1174051_praktek.py	190
src/1174051_praktek.py	191
src/mainn.py	191
src/mainn.py	191

src/1174051_praktek.py	191
3.1 namaFungsi merupakan nama dari fungsi yang dibuat.	192
3.2 inputanFungsi merupakan nama dari inputan fungsi yang diterima dari luar fungsi namaFungsi.	192
3.3 return inputanFungsi merupakan kembalian fungsi dari fungsi namaFungsi.	192
3.4 Contoh penggunaan fungsi di Python.	192
3.5 Contoh penggunaan paket atau library.	192
3.6 Contoh penggunaan kelas di python.	193
3.7 Contoh penggunaan objek di python.	193
3.8 Contoh penggunaan atribut di python.	193
3.9 Contoh penggunaan method di python.	193
3.10 Contoh pemanggilan library kelas dari instansi dan pemakaianya. .	193
3.11 Contoh pemakaian paket dengan perintah from kalkulator import Penambahan.	194
3.12 Contoh kode pemakaian paket fungsi dimana file library ada di dalam folder.	194
3.13 Contoh kode pemakaian paket kelas dimana file library ada di dalam folder.	194
3.14 Jawaban soal No. 1 Ketrampilan Pemrograman.	195
3.15 Jawaban soal No. 2 Ketrampilan Pemrograman.	196
3.16 Jawaban soal No. 3 Ketrampilan Pemrograman.	196
3.17 Jawaban soal No. 4 Ketrampilan Pemrograman.	196
3.18 Jawaban soal No. 5 Ketrampilan Pemrograman.	196
3.19 Jawaban soal No. 6 Ketrampilan Pemrograman.	197
3.20 Jawaban soal No. 7 Ketrampilan Pemrograman.	197
3.21 Jawaban soal No. 8 Ketrampilan Pemrograman.	197
3.22 Jawaban soal No. 9 Ketrampilan Pemrograman.	197
3.23 Jawaban soal No. 10 Ketrampilan Pemrograman.	198
3.24 Jawaban soal No. 11 Ketrampilan Pemrograman.	198
3.25 Jawaban soal No. 12 Ketrampilan Pemrograman.	198
3.26 Contoh fungsi yang menggunakan try except .	199

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

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[?].

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

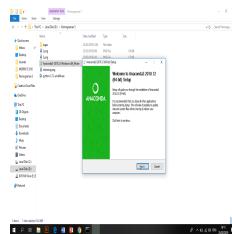
BAB 1

SEJARAH DAN KARAKTERISTIK PYTHON

1.1 Dwi Yulianingsih

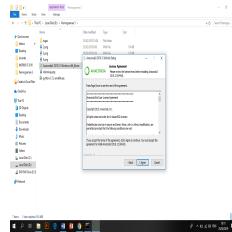
1.1.1 Sejarah Phyton

Phyton adalah sebuah bahasa pemrograman dengan perancangan yang berfokus pada tingkat keterbacaan kode, menggabungkan kapabilitas, kemampuan dan sintaks kode yang sangat jelas. Phyton juga dilengkapi dengan fungsi pustaka atau library standar yang besar dan didukung oleh komunitas yang besar. Phyton dibuat oleh seseorang keturunan belanda yaitu Guido Van Rossum, awalnya pembuatan phyton ini digunakan untuk pembuatan bahasa tingkat tinggi pada sebuah sistem operasi. Phyton telah digunakan oleh perusahaan-perusahaan untuk membuat perangkat lunak komersil. Pemrograman bahasa python merupakan pemrogram gratis atau freeware, sehingga bisa dikembangkan, dan tidak memiliki batasan dalam peng-copy-an dan didistribusikan. Terdapat beberapa layanan yang diberikan dalam phyton lengkap dengan source kodennya, debugger dan profiler, antarmuka, fungsi sistem, GUI, dan database-nya. Python dapat digunakan untuk berbagai Sistem Operasi, yang diantaranya Unix (linux), PCs (DOS, Windows, OS/2), Machintosh dan sebagainya.

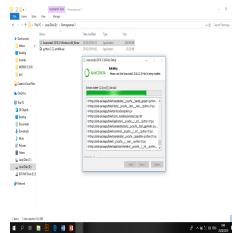
**Gambar 1.1** gambar1

1.1.2 Instalasi Anaconda

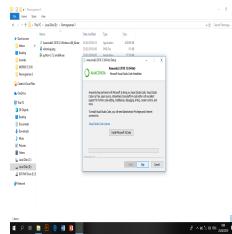
1. Kita harus menyiapkan instalasi anaconda, kita dapat mendownload nya melalui internet.
2. Kemudian kita bisa mengklik installer yang telah kita miliki dan tunggu.
3. Lalu pada tampilan seperti gambar di bawah klik next.

**Gambar 1.2** gambar2

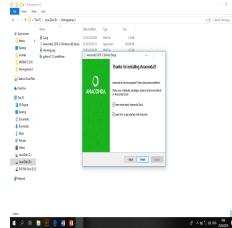
4. Setelah itu setujui lisensi yang ada dengan mengklik I Agree

**Gambar 1.3** gambar3

5. Tunggu instalasi selesai, lalu klik skip
6. setelah itu klik finish, dan selesai yeay



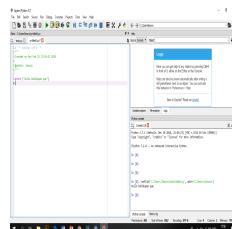
Gambar 1.4 gambar4



Gambar 1.5 gambar5

1.1.3 Menggunakan Spyder

berikut adalah contoh dalam menggunakan spyder



Gambar 1.6 gambar6

1.2 Dwi Septiani Tsaniyah

1.2.1 Sejarah Python

Python dikembangkan oleh Guido van Rossum sebagai bahasa pemrograman ABC pada tahun 1990 di Stichting Mathematisch Centrum (CWI) di Amsterdam. Versi terbaru yang dirilis oleh CWI adalah 1.2.

Pada 1995, Guido pindah ke CNRI di Virginia, AS, dan terus mengembangkan Python. Versi terakhir yang dirilis 1.6. Pada tahun 2000, insinyur Guido dan Python

menjadi perusahaan komersial untuk BeOpen.com dan menciptakan BeOpen Python-Labs. Python 2.0 dirilis oleh BeOpen. Setelah menghapus Python 2.0, beberapa anggota Guido dan PythonLabs pindah ke DigitalCreations.

Saat ini, pengembangan Python sedang dilanjutkan oleh sekelompok programmer yang dikoordinir oleh Guido dan Yayasan Perangkat Lunak Python. Python Software Foundation, versi 2.1, memiliki hak cipta Python, dan Python adalah organisasi nirlaba yang memblokir kepemilikan perusahaan komersial. Saat ini, distribusi Python telah mencapai versi 2.7.14 dan versi 3.6.3

Program televisi Guido Monty Python Flying Circus telah dinamai nama Python oleh Guido sebagai bahasa ciptaannya. Oleh karena itu, sering kali ungkapan khas suatu acara sering muncul dalam korespondensi antara pengguna Python.

1.2.2 Instalasi Anaconda

1. Pastikan Bahwa Python telah terinstal dilaptop anda.
2. Jika anda belum punya anaconda
3. Kemudian buka installer yang telah di download barusan
4. Klik next
5. Kemudian Klik I Agree
6. Kemudian pilih akan di instal untuk siapa, kemudian pilih next
7. Kemudian tentukan dicrotory nya
8. Kemudian Centang yang register Anaconda as default Python, Kemudian Pilih Next
9. Tunggu Proses Instalasi hingga selesai
10. Instalasi telah selesai

1.2.3 Penggunaan Spyder

kodingan sederhana Hello Word

1.3 Kadek Diva Krishna Murti

1.3.1 Sejarah Python

Python merupakan salah satu bahasa pemrograman tingkat tinggi yang menggunakan metode pemrosesan *interpreted*, dimana kode program akan diproses baris per baris secara langsung dari kode program.

Bahasa pemrograman Python dirilis pertama kali oleh Guido van Rossum di Sciticting Mathematisch Centrum (CWI) Belanda pada tahun 1991. Bahasa python terinspirasi dari bahasa pemrograman ABC. Nama python tidak berasal dari namaular yang kita kenal. Guido merupakan penggemar grup komedi Inggris bernama Monty Python. Kemudian, ia menamakan Bahasa pemrograman ciptaannya dengan nama Python.

Pada tahun 1994, Python 1.0 dirilis, yang diikuti dengan Python 2.0 pada tahun 2000. Python 3.0 keluar pada tahun 2008. Sampai saat ini Python masih dikembangkan oleh *Python Software Foundation*. Bahasa Python mendukung hampir semua sistem operasi, bahkan untuk sistem operasi Linux, hampir semua distrionya sudah menyertakan Python di dalamnya [?].

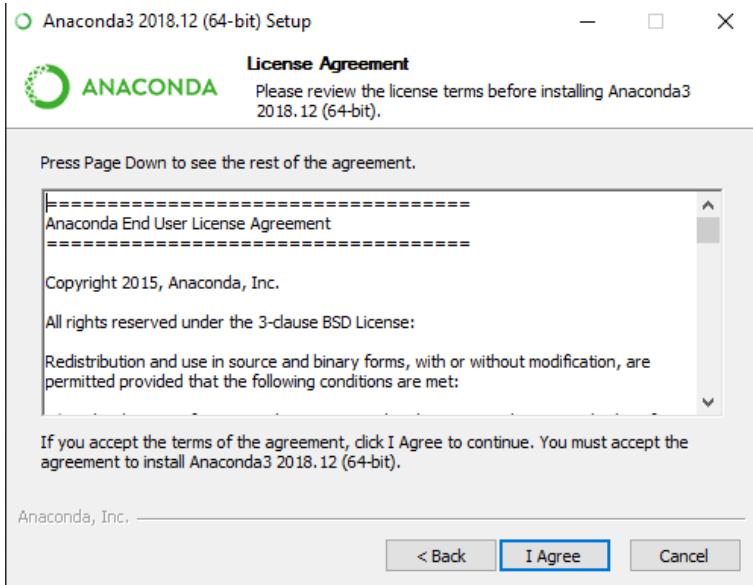
1.3.2 Instalasi Anaconda

Berikut ini merupakan langkah-langkah cara instalasi Anaconda di windows:

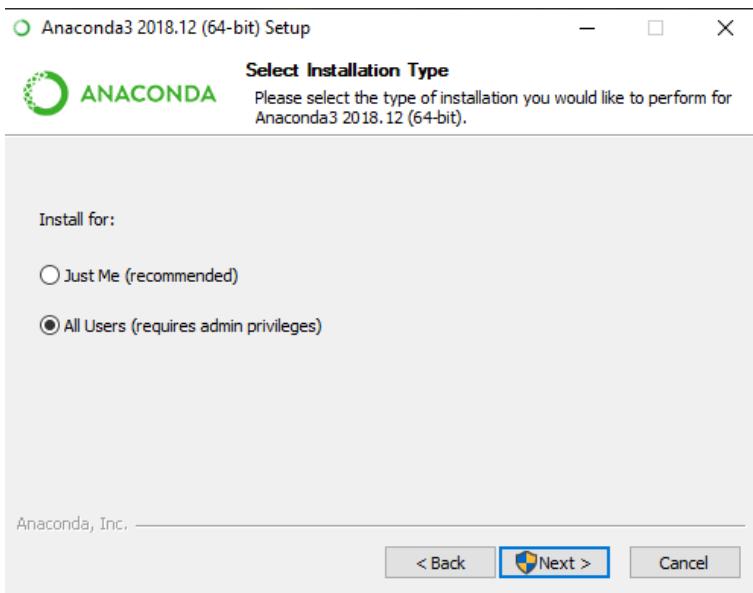
1. Pastikan kalian telah menginstall Python sebelumnya.
2. Klik dua kali pada installer Anaconda. Installer anaconda bisa anda dapatkan di <https://www.anaconda.com/distribution/>
3. Setelah itu akan muncul window installernya. Kemudian klik "Next" untuk memulai instalasi.



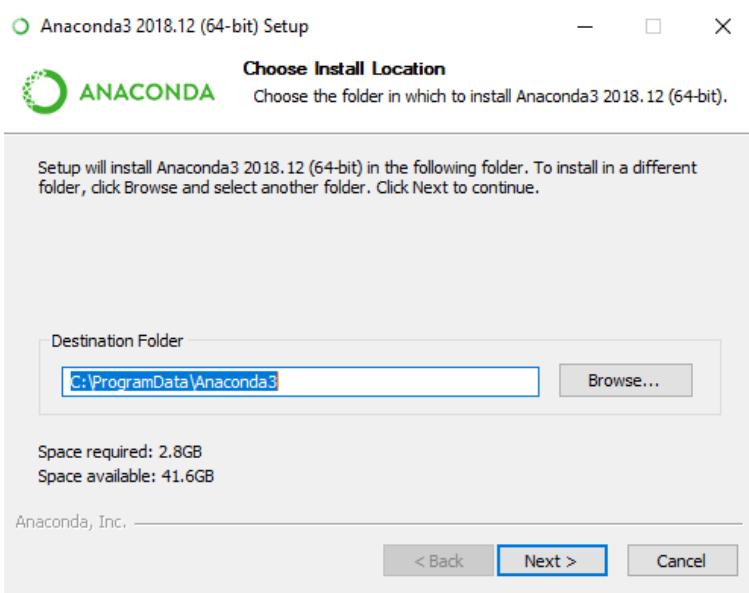
4. Baca Lisensi Agreement Anacondanya. Lalu klik "I Agree" jika kalian menerimanya dan untuk melanjutkannya instalasinya.



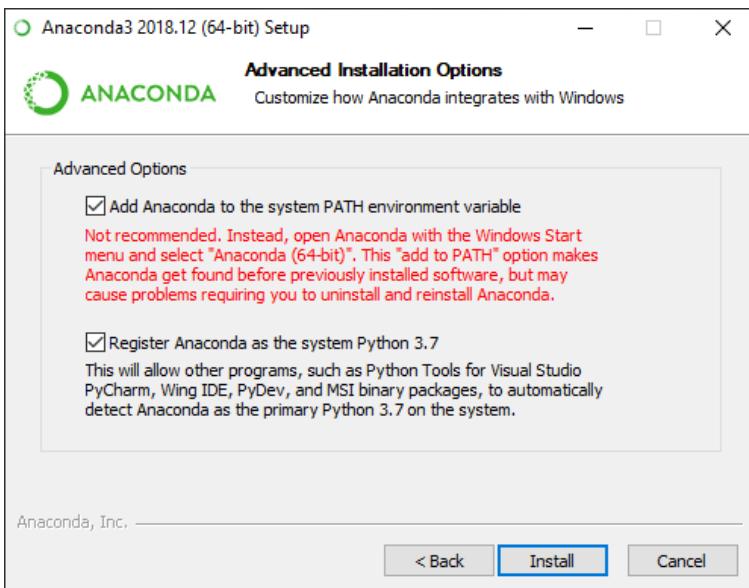
5. Selanjutnya diberi pilihan untuk menginstallnya, apakah hanya untuk kalian atau untuk semua pengguna. Disini saya memilih "All Users", lalu klik "Next".



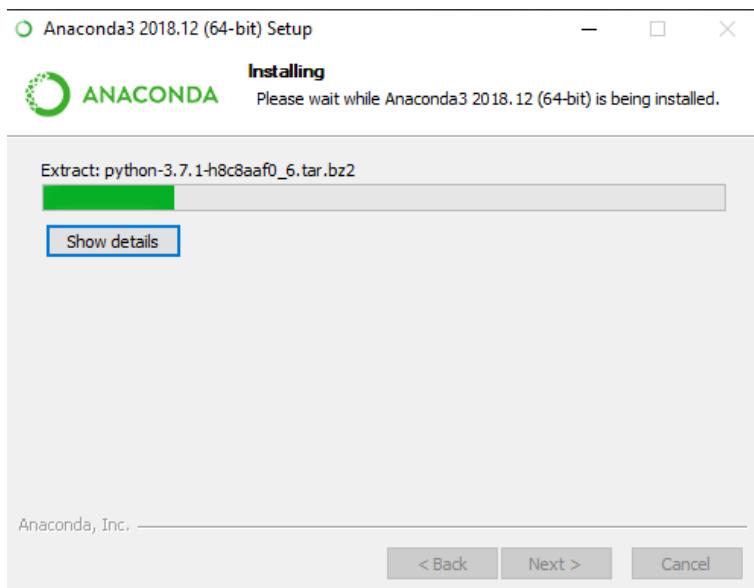
6. Kemudian pilih tujuan instalasinya. Disini saya biarkan default folder instalasinya. Setelah itu, klik "Next".



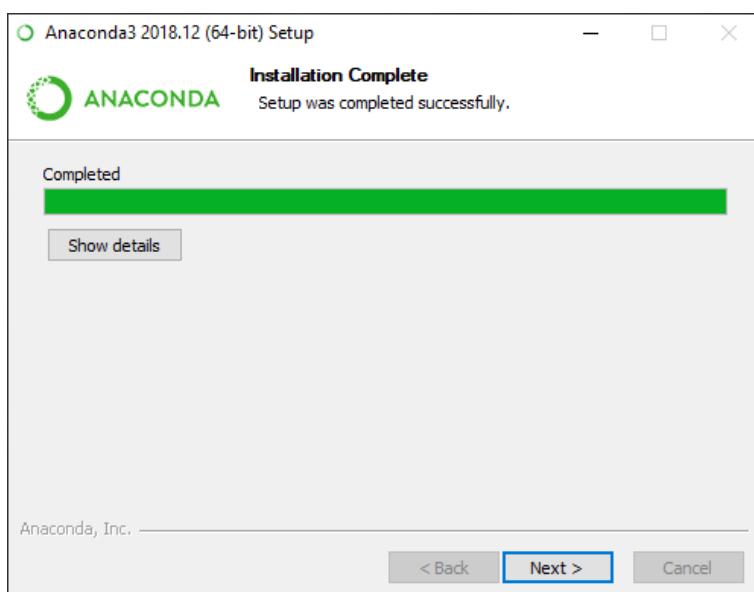
- Setelah itu, kalian diberi beberapa opsi tambahan. Opsi pertama yaitu, "Add Anaconda to my PATH environment variable". Opsi ini akan menambahkan Anaconda ke PATH sistem environment variable. Opsi kedua yaitu, "Register Anaconda as my default Python 3.7". Opsi ini akan mendaftarkan Anaconda sebagai system Python 3.7. Saya centang kedua opsi tersebut, lalu klik "Install".



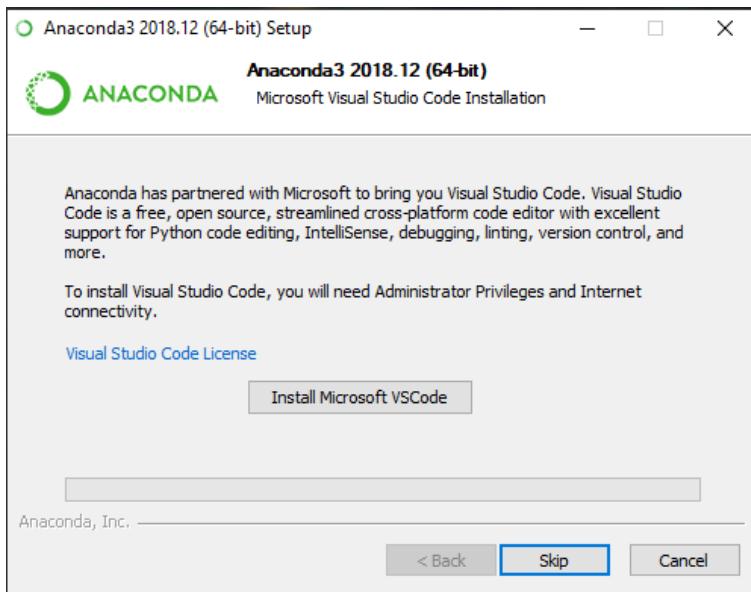
8. Tunggu hingga proses instalasi selesai.



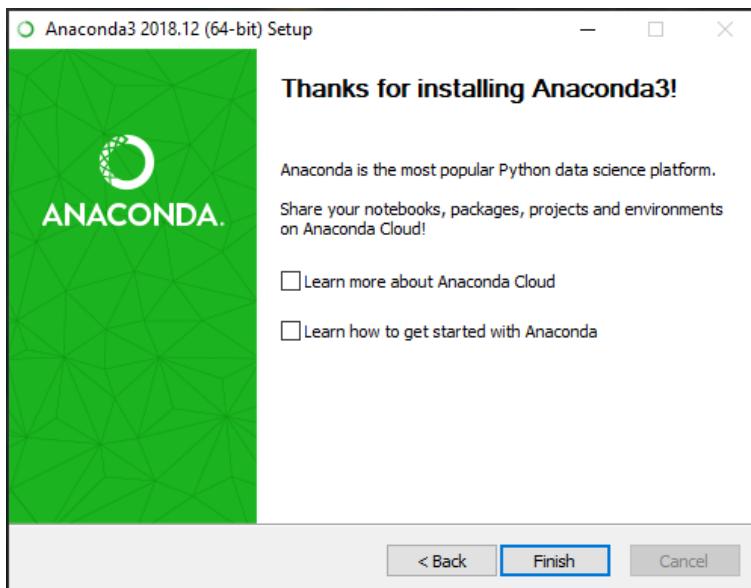
9. Kemudian klik "Next" untuk melanjutkan.



10. Selanjutnya kalian diberi pilihan untuk menginstall Microsoft VSCode. Saya klik "Skip" untuk melanjutkan.



11. Kemudian klik "Finish" untuk selesai.



12. Untuk mengecek apakah Anaconda telah terinstall yaitu dengan cara membuka Command Prompt. Lalu ketikan "conda -V" dan tekan enter, kode itu akan mengecek versi Anaconda yang terinstall.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.316]
(c) 2018 Microsoft Corporation. All rights reserved.

c:\Users\vanerz>conda -V
conda 4.5.12

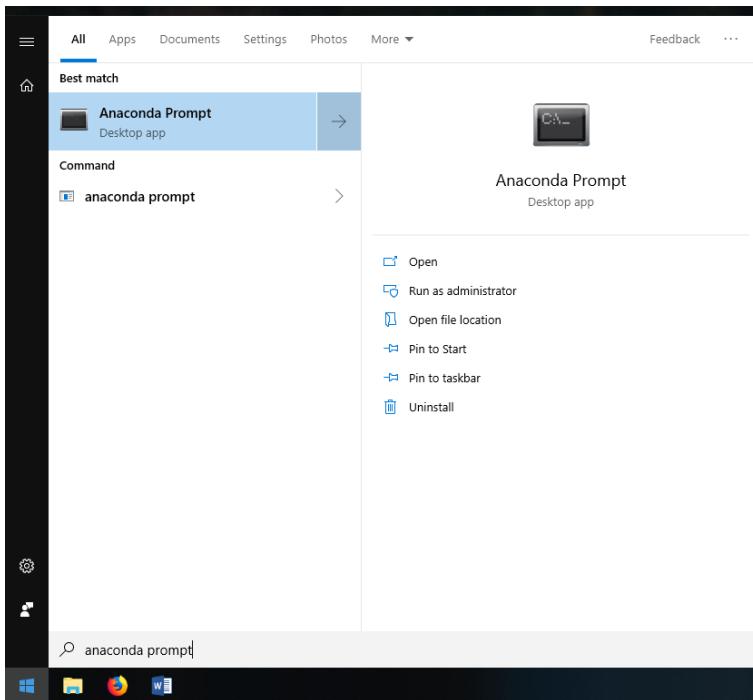
c:\Users\vanerz>
```

1.3.3 Penggunaan Spyder

Terdapat 2 cara menjalankan Spyder. Yang pertama dengan Anaconda Prompt dan yang kedua dengan Anaconda Navigation. Berikut ini merupakan langkah-langkah cara menjalankan Spyder di windows:

- Anaconda Prompt

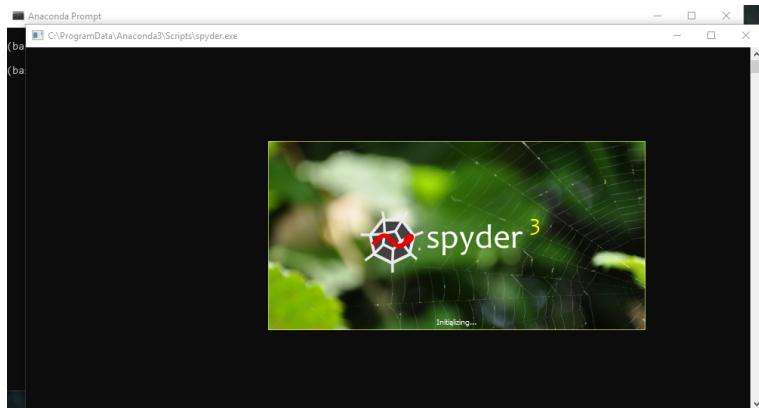
1. Pertama klik start, lalu cari "Anaconda Prompt".



2. Selanjutnya akan muncul sebuah prompt. Kemudian ketikan "start spyder" dan tekan enter.

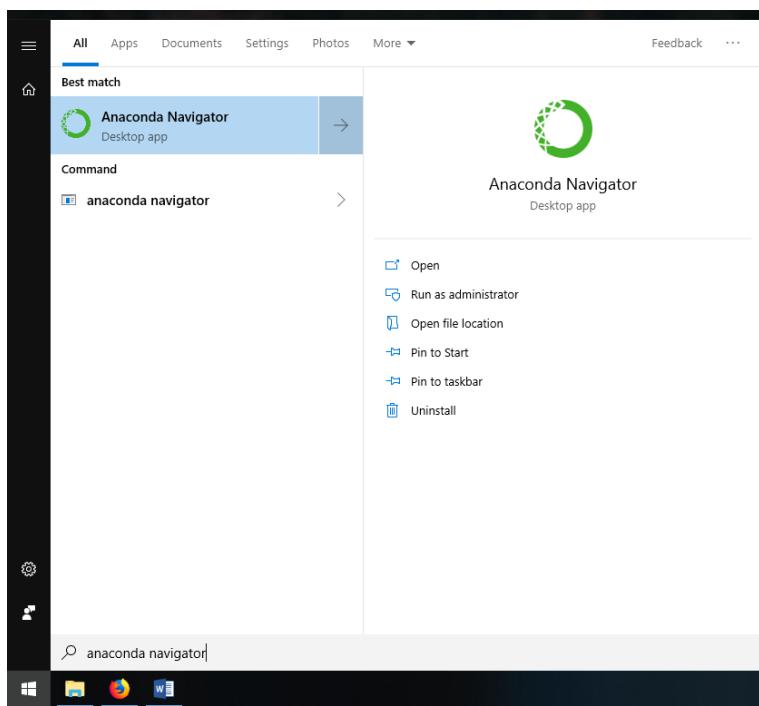
A screenshot of the Anaconda Prompt window. The title bar says "Anaconda Prompt". The command line shows "(base) C:\Users\vanerz>start spyder". The window is mostly black, indicating it's a terminal or command-line interface.

3. Lalu tunggu sampai selesai.

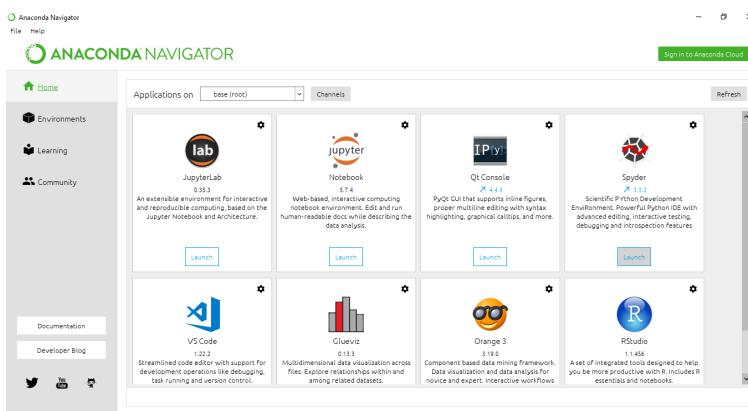


- Anaconda Navigation

1. Pertama klik start, lalu cari "Anaconda Navigation".



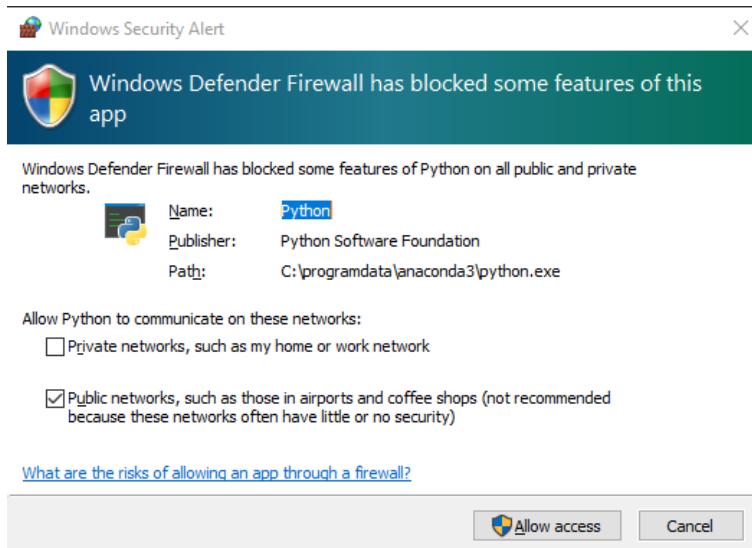
2. Selanjutnya akan muncul sebuah window. Kemudian klik "Launch" pada menu Spyder.



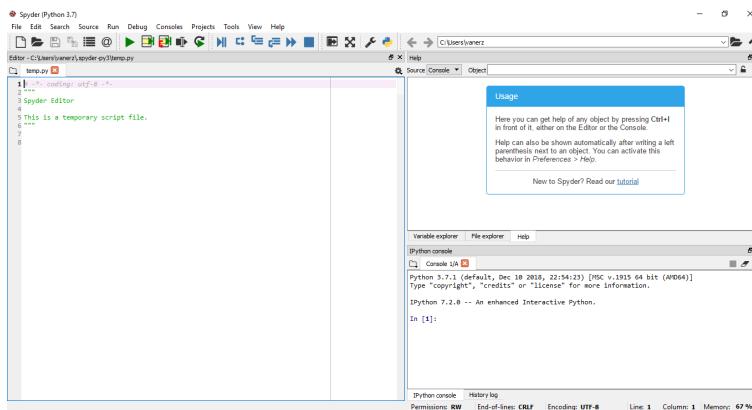
3. Lalu tunggu sampai selesai.



Apabila muncul window ini ketika pertama kali menjalankan Spyder, pilih Allow Access.

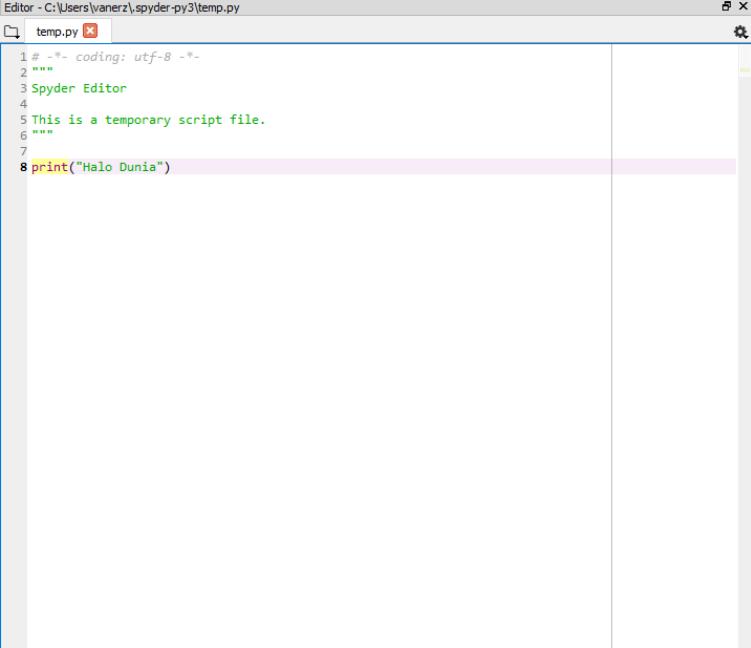


Berikut ini merupakan gambar dari Spyder



Berikut cara menggunakan Spyder:

1. Silahkan ketikan script Python anda di sini.

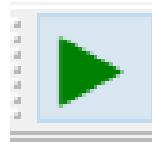


Editor - C:\Users\vanerz\spyder-py3\temp.py

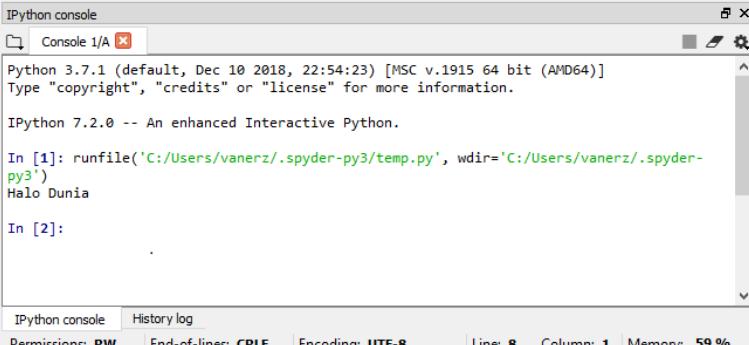
temp.py

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8 print("Halo Dunia")
```

- Setelah mengetik script Python, kemudian klik tombol play atau tekan tombol F5 untuk mengeksekusi script Python yang telah diketik tadi.



- Hasil dari eksekusi akan muncul disini.



IPython console

Console 1/A

```
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.

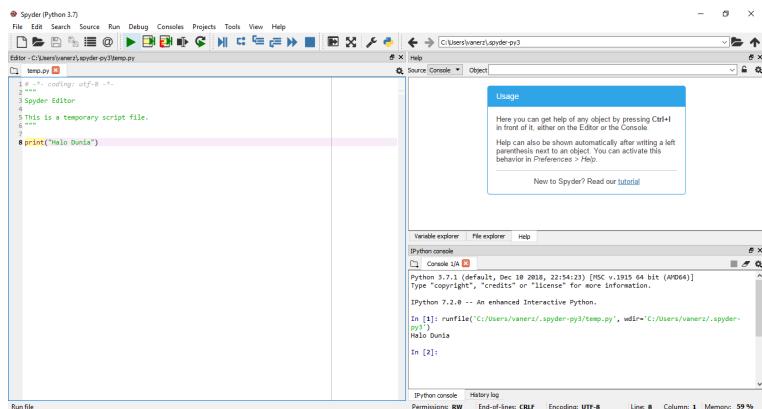
In [1]: runfile('C:/Users/vanerz/.spyder-py3/temp.py', wdir='C:/Users/vanerz/.spyder-py3')
Halo Dunia

In [2]:
```

IPython console History log

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 8 Column: 1 Memory: 59 %

4. Berikut tampilan penuhnya.



1.4 Harun Ar-Rasyid

1.4.1 Sejarah

Python diciptakan oleh Guido van Rossum pertama kali di Scitchting Mathematisch Centrum (CWI) di Belanda pada awal tahun 1990-an. Bahasa python terinspirasi dari bahasa pemrograman ABC. Sampai sekarang, Guido masih menjadi penulis utama untuk python, meskipun bersifat open source sehingga ribuan orang juga berkontribusi dalam mengembangkannya.

Pada 1995, Guido pindah ke CNRI di Virginia Amerika sambil terus mengembangkan Python. Versi terakhir yang dirilis adalah 1.6. Pada tahun 2000, pengembang inti Guido dan Python pindah ke BeOpen.com yang merupakan perusahaan komersial dan membentuk BeOpen PythonLabs. Python 2.0 dirilis oleh BeOpen. Setelah menghapus Python 2.0, Guido dan beberapa anggota tim PythonLabs pindah ke DigitalCreations.

Saat ini pengembangan Python terus dilakukan oleh sekelompok programmer yang dikoordinir oleh Guido dan Python Software Foundation. Python Software Foundation adalah organisasi nirlaba 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 telah mencapai versi 2.7.14 dan versi 3.6.3

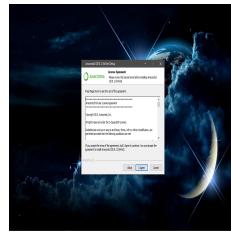
Nama Python dipilih oleh Guido sebagai nama bahasa ciptaananya karena kecintaan Guido pada acara televisi Flying Circus Monty Python. Oleh karena itu sering ekspresi khas acara sering muncul dalam korespondensi antara pengguna Python.

1.4.2 Instalasi Anaconda

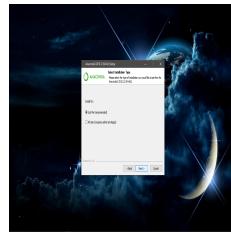
- Pastikan Bahwa Python telah terinstal dilaptop anda.



Gambar 1.7 Tampilan Awal

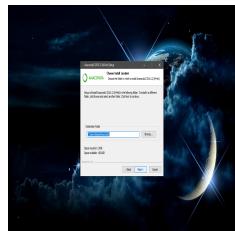


Gambar 1.8 License Agreement

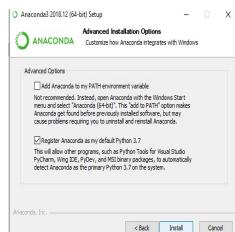


Gambar 1.9 Pemilihan User

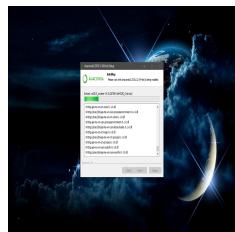
2. Kemudian Download Anaconda pada websitenya langsung.
3. Kemudian buka installer yang telah di download barusan
4. Klik next
5. Kemudian Klik I Agree
6. Kemudian pilih akan di instal untuk siapa, kemudian pilih next
7. Kemudian tentukan dicitrectory nya
8. Kemudian Centang yang register Anaconda as default Python, Kemudian Pilih Next
9. Tunggu Proses Instalasi hingga selesai



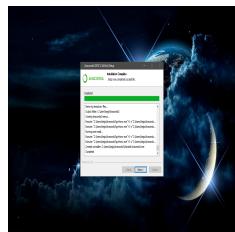
Gambar 1.10 Pemilihan Direktori Penyimpanan



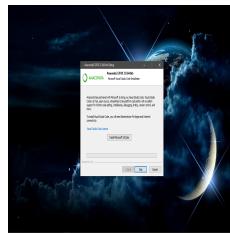
Gambar 1.11 Pemilihan Opsi



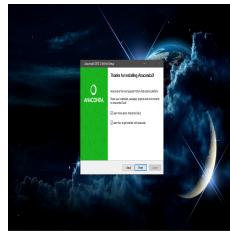
Gambar 1.12 Proses Instal



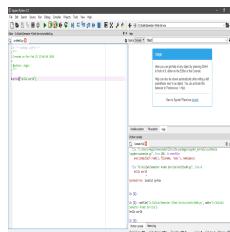
Gambar 1.13 Proses Instal Selesai



Gambar 1.14 Penawaran Instal MS VSC



Gambar 1.15 Instalasi Selesai



Gambar 1.16 Ini adalah tampilan spyder

10. Klik next
11. kemudian jika kalian belum instal MS VSC di sarankan menginstalnya dlu, jika sudah klik skip
12. Instalasi anaconda telah selesai

1.4.3 Menggunakan Spyder

Setelah selesai melakukan instalasi anaconda, maka ada beberapa tool yang digunakan seperti spyder

Gambar diatas menjelaskan tentang tampilan spider dan mengexekusi program halo world.

1.5 Rahmatul Ridha / 1144124

1.5.1 Sejarah Python

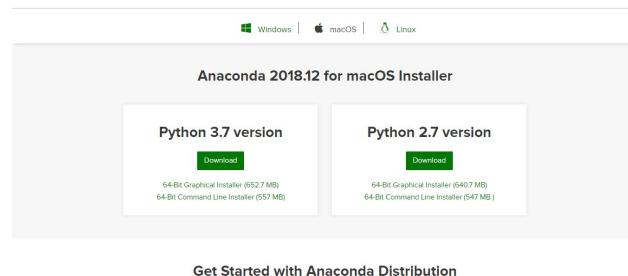
Bahasa pemrograman Python adalah bahasa yang dibuat oleh seorang keturunan Belanda yaitu Guido van Rossum. Sampai saat ini Python masih dikembangkan oleh *Python Software Foundation*. Awalnya, pembuatan bahasa pemrograman ini adalah untuk membuat skrip bahasa tingkat tinggi pada sebuah sistem operasi yang terdistribusi Amoeba. Python telah digunakan oleh beberapa pengembang dan bahkan digunakan oleh beberapa perusahaan untuk pembuatan perangkat lunak komersial. Pemrograman bahasa python ini adalah pemrogram gratis atau *freeware*, sehingga dapat dikembangkan, dan tidak ada batasan dalam penyalinannya dan mendistribusikan.

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.

1.5.2 Instalasi Anaconda

- Instalasi Anaconda Berikut adalah langkah-langkah cara menginstal Anaconda di Windows:

 1. Download installer anaconda terbaru, seperti pada gambar 1.17. Kalian dapat memilih versi 2 atau 3, dengan versi Anaconda berapa.

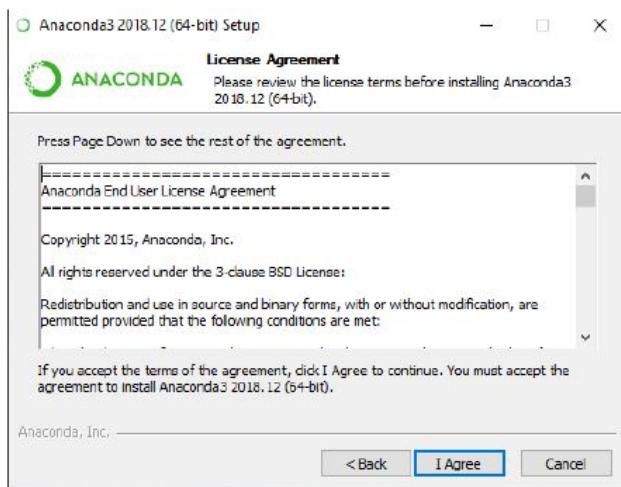


Gambar 1.17 Download Anaconda

2. Setelah selesai mendownload, klik 2 kali pada installer Anaconda.
3. Kemudian akan tampil seperti gambar 1.18, lalu klik next.
4. Setelah itu read lisensi dan klik I Agree.
5. Selanjutnya ada pilihan untuk menginstallnya, yaitu just me atau all users. Lalu klik next.

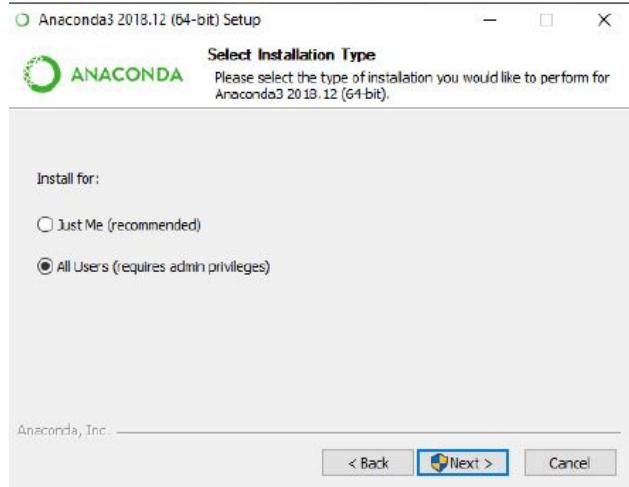


Gambar 1.18 Proses 1

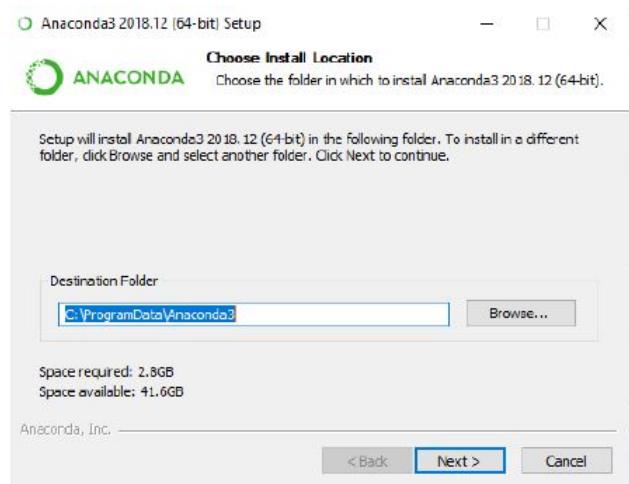


Gambar 1.19 proses 2

6. Kemudian pilih okasi yang diinginkan, lalu klik next.
7. Pilih add anaconda to PATH atau tidak. Disini kalian memilih apakah akan mendaftarkan Anaconda sebagai default Python 3.7. Kacuali kalian berencana menginstal dan menjalankan beberapa versi Anaconda, atau beberapa versi Python, biarkan default dan biarkan kotaknya dicentang. Kemudian klik tombol Install. Jika kalian ingin melihat packages Anaconda yang sedang dipasang, klik Show Details.
8. Jika instalasi selesai, kemudian klik next.



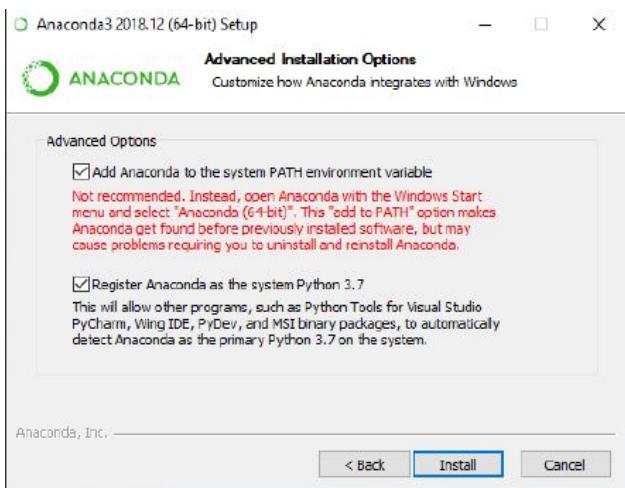
Gambar 1.20 Proses 3



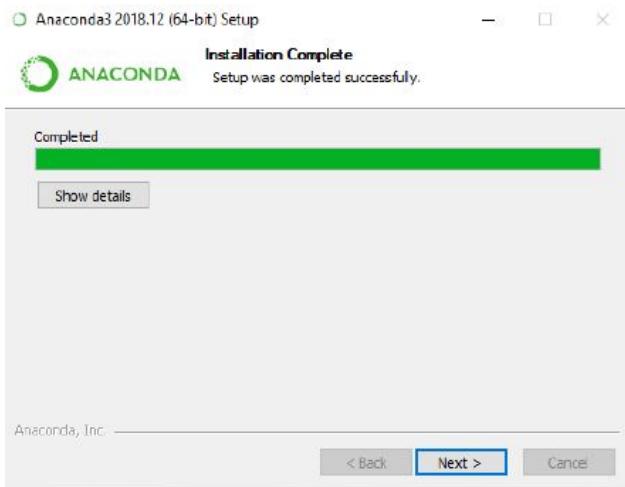
Gambar 1.21 Proses 4

9. Jika packages telah selesai diinstall, maka akan ada perintah untuk menginstall VS Code, lalu klik tombol Install Microsoft VS Code.

10. Setelah instalasi selesai, maka akan terlihat kotak dialog Thanks for Installing Anaconda3. Lalu klik Finish.



Gambar 1.22 Proses 5

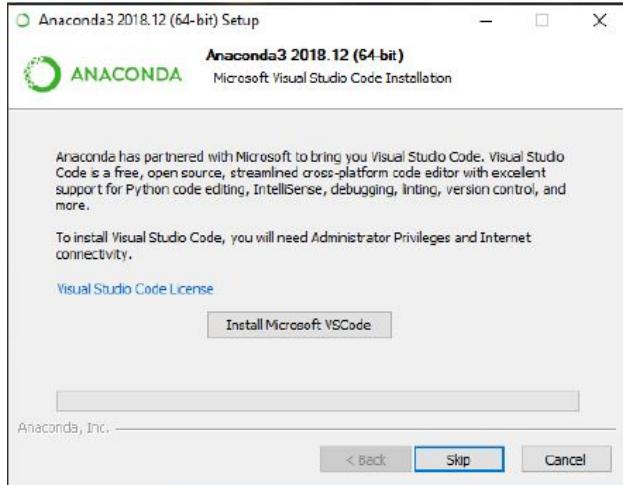


Gambar 1.23 Proses 6

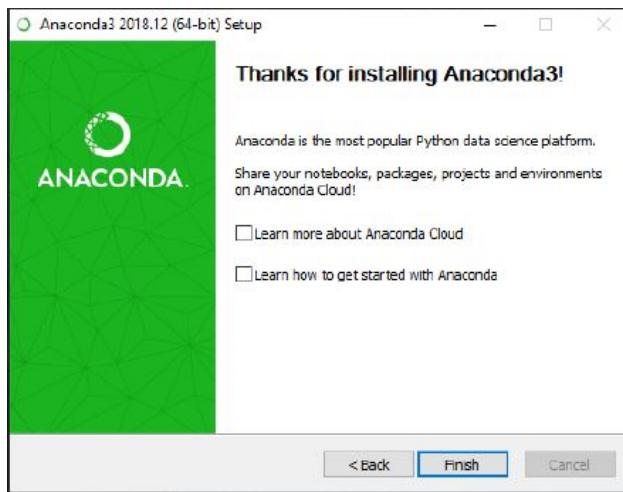
1.6 Dwi Julianingsih

1.6.1 Sejarah Phyton

Phyton adalah sebuah bahasa pemrograman dengan perancangan yang berfokus pada tingkat keterbacaan kode, menggabungkan kapabilitas, kemampuan dan sintaks kode yang sangat jelas. Phyton juga dilengkapi dengan fungsi pustaka atau library standar yang besar dan didukung oleh komunitas yang besar. Phyton dibuat oleh seseorang



Gambar 1.24 Proses 7



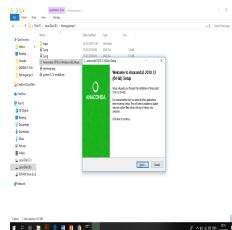
Gambar 1.25 Proses 8

rang keturunan belanda yaitu Guido Van Rossum, awalnya pembuatan phyton ini digunakan untuk pembuatan bahasa tingkat tinggi pada sebuah sistem operasi. Phyton telah digunakan oleh perusahaan-perusahaan untuk membuat perangkat lunak komersil. Pemrograman bahasa python merupakan pemrogram gratis atau freeware, sehingga bisa dikembangkan, dan tidak memiliki batasan dalam peng-copy-an dan didistribusikan. Terdapat beberapa layanan yang diberikan dalam phyton lengkap dengan source kodennya, debugger dan profiler, antarmuka, fungsi sistem, GUI, dan

database-nya. Python dapat digunakan untuk berbagai Sistem Operasi, yang diantaranya Unix (linux), PCs (DOS, Windows, OS/2), Machintosh dan sebagainya.

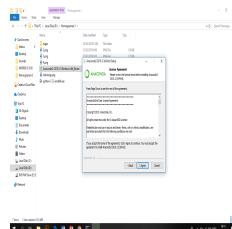
1.6.2 Instalasi Anaconda

1. Kita harus menyiapkan instalasi anaconda, kita dapat mendownload nya melalui internet.
2. Kemudian kita bisa mengklik installer yang telah kita miliki dan tunggu.



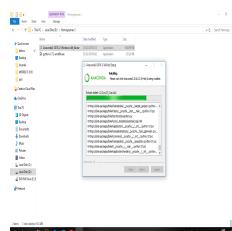
Gambar 1.26 gambar1

3. Lalu pada tampilan seperti gambar di bawah klik next.

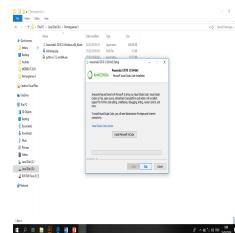


Gambar 1.27 gambar2

4. Setelah itu setujui lisensi yang ada dengan mengklik I Agree



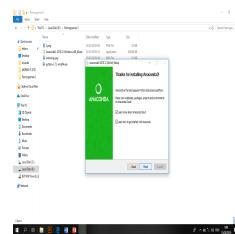
Gambar 1.28 gambar3



Gambar 1.29 gambar4

5. Tunggu instalasi selesai, lalu klik skip

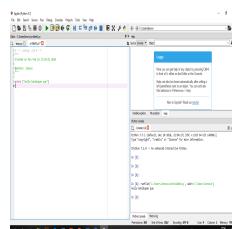
6. setelah itu klik finish, dan selesai yeay



Gambar 1.30 gambar5

1.6.3 Menggunakan Spyder

berikut adalah contoh dalam menggunakan spyder



Gambar 1.31 gambar6

1.7 Dwi Septiani Tsaniyah

1.7.1 Sejarah Python

Python dikembangkan oleh Guido van Rossum sebagai bahasa pemrograman ABC pada tahun 1990 di Stichting Mathematisch Centrum (CWI) di Amsterdam. Versi terbaru yang dirilis oleh CWI adalah 1.2.

Pada 1995, Guido pindah ke CNRI di Virginia, AS, dan terus mengembangkan Python. Versi terakhir yang dirilis 1.6. Pada tahun 2000, insinyur Guido dan Python menjadi perusahaan komersial untuk BeOpen.com dan menciptakan BeOpen Python-Labs. Python 2.0 dirilis oleh BeOpen. Setelah menghapus Python 2.0, beberapa anggota Guido dan PythonLabs pindah ke DigitalCreations.

Saat ini, pengembangan Python sedang dilanjutkan oleh sekelompok programmer yang dikoordinir oleh Guido dan Yayasan Perangkat Lunak Python. Python Software Foundation, versi 2.1, memiliki hak cipta Python, dan Python adalah organisasi nirlaba yang memblokir kepemilikan perusahaan komersial. Saat ini, distribusi Python telah mencapai versi 2.7.14 dan versi 3.6.3

Program televisi Guido Monty Python Flying Circus telah dinamai nama Python oleh Guido sebagai bahasa ciptaannya. Oleh karena itu, sering kali ungkapan khas suatu acara sering muncul dalam korespondensi antara pengguna Python.

1.7.2 Instalasi Anaconda

1. Pastikan Bahwa Python telah terinstal dilaptop anda.
2. Jika anda belum punya anaconda
3. Kemudian buka installer yang telah di download barusan
4. Klik next
5. Kemudian Klik I Agree
6. Kemudian pilih akan di instal untuk siapa, kemudian pilih next
7. Kemudian tentukan directory nya
8. Kemudian Centang yang register Anaconda as default Python, Kemudian Pilih Next
9. Tunggu Proses Instalasi hingga selesai
10. Instalasi telah selesai

1.7.3 Penggunaan Spyder

kodingan sederhana Hello Word

1.8 Kadek Diva Krishna Murti

1.8.1 Sejarah Python

Python merupakan salah satu bahasa pemrograman tingkat tinggi yang menggunakan metode pemrosesan *interpreted*, dimana kode program akan diproses baris per baris secara langsung dari kode program.

Bahasa pemrograman Python dirilis pertama kali oleh Guido van Rossum di Scitthing Mathematisch Centrum (CWI) Belanda pada tahun 1991. Bahasa python terinspirasi dari bahasa pemrograman ABC. Nama python tidak berasal dari nama ular yang kita kenal. Guido merupakan penggemar grup komedi Inggris bernama Monty Python. Kemudian, ia menamakan Bahasa pemrograman ciptaannya dengan nama Python.

Pada tahun 1994, Python 1.0 dirilis, yang diikuti dengan Python 2.0 pada tahun 2000. Python 3.0 keluar pada tahun 2008. Sampai saat ini Python masih dikembangkan oleh *Python Software Foundation*. Bahasa Python mendukung hampir semua sistem operasi, bahkan untuk sistem operasi Linux, hampir semua distrionya sudah menyertakan Python di dalamnya [?].

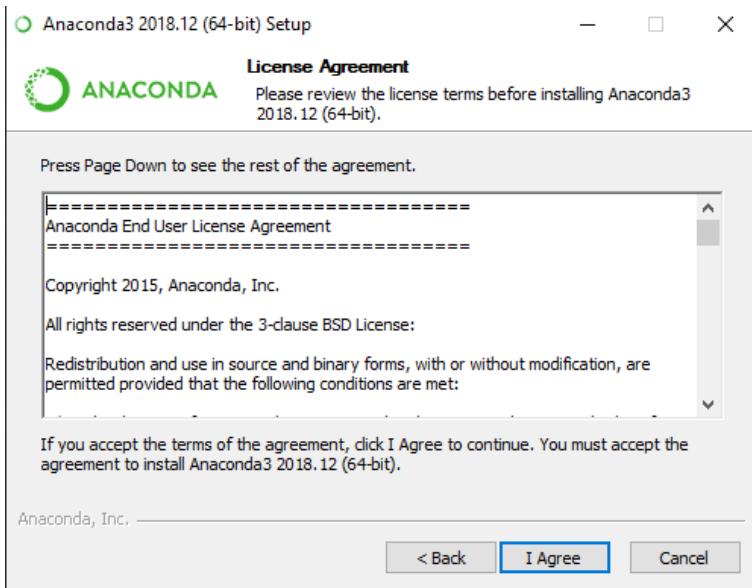
1.8.2 Instalasi Anaconda

Berikut ini merupakan langkah-langkah cara instalasi Anaconda di windows:

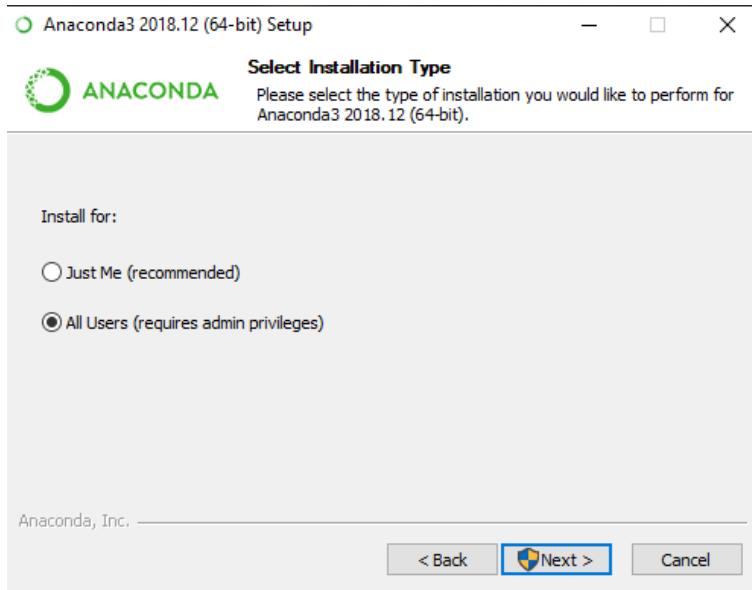
1. Pastikan kalian telah menginstall Python sebelumnya.
2. Klik dua kali pada installer Anaconda. Installer anaconda bisa anda dapatkan di <https://www.anaconda.com/distribution/>
3. Setelah itu akan muncul window installernya. Kemudian klik "Next" untuk memulai instalasi.



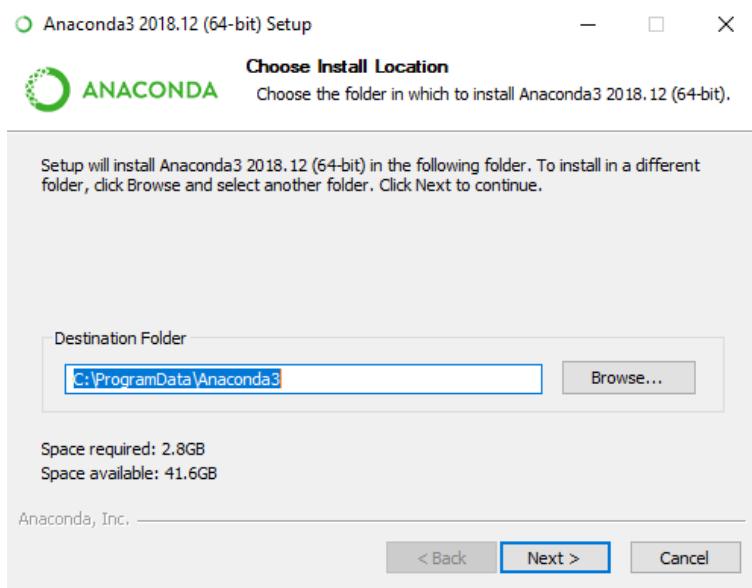
4. Baca Lisensi Agreement Anacondanya. Lalu klik "I Agree" jika kalian menerimanya dan untuk melanjutkannya instalasinya.



5. Selanjutnya diberi pilihan untuk menginstallnya, apakah hanya untuk kalian atau untuk semua pengguna. Disini saya memilih "All Users", lalu klik "Next".

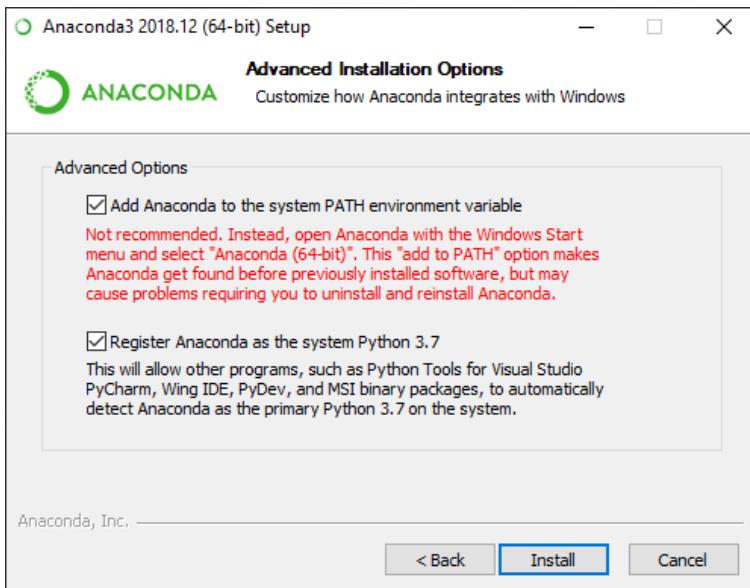


6. Kemudian pilih tujuan instalasinya. Disini saya biarkan default folder instalasinya. Setelah itu, klik "Next".

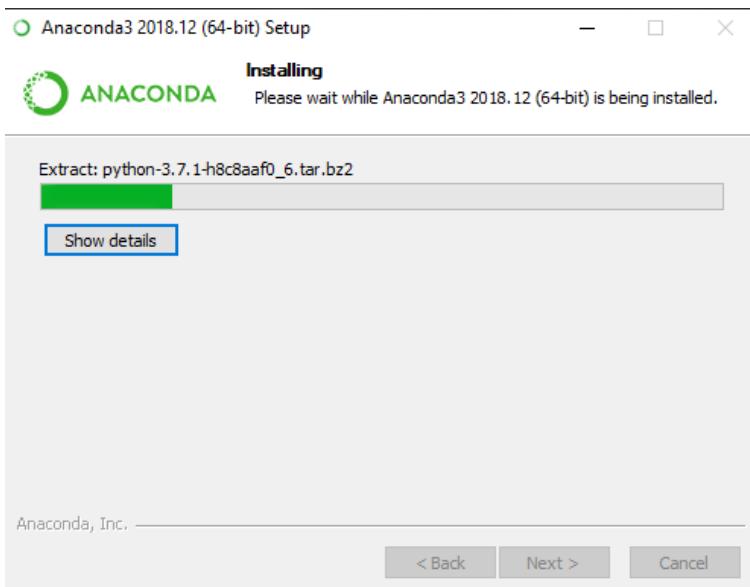


7. Setelah itu, kalian diberi beberapa opsi tambahan. Opsi pertama yaitu, "Add Anaconda to my PATH environment variable". Opsi ini akan menambahkan

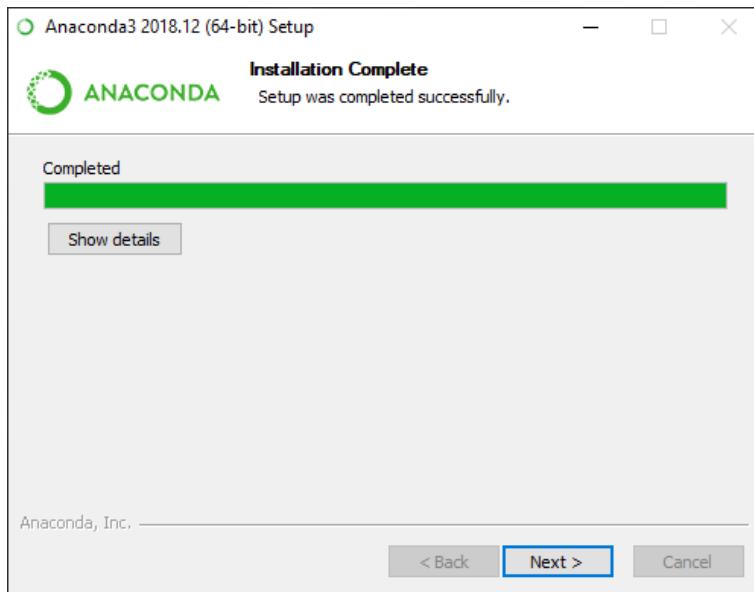
Anaconda ke PATH sistem environment variable. Opsi kedua yaitu, "Register Anaconda as my default Python 3.7". Opsi ini akan mendaftarkan Anaconda sebagai system Python 3.7. Saya centang kedua opsi tersebut, lalu klik "Install".



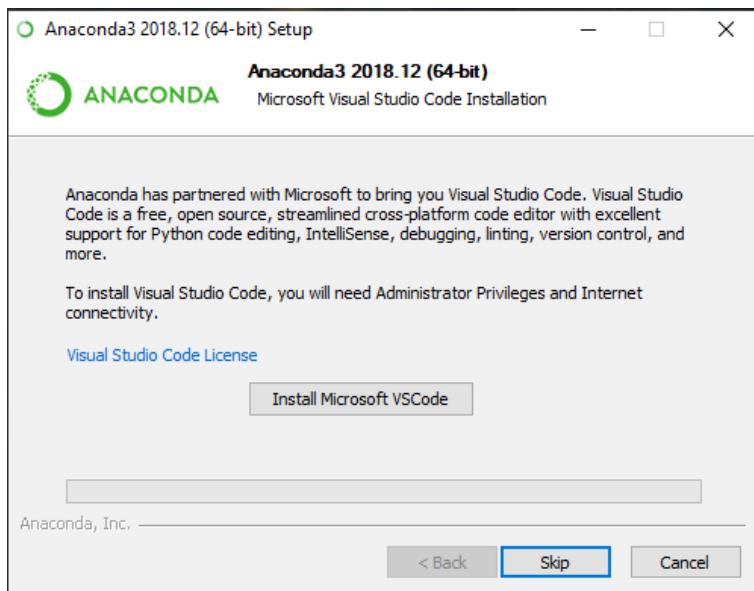
8. Tunggu hingga proses instalasi selesai.



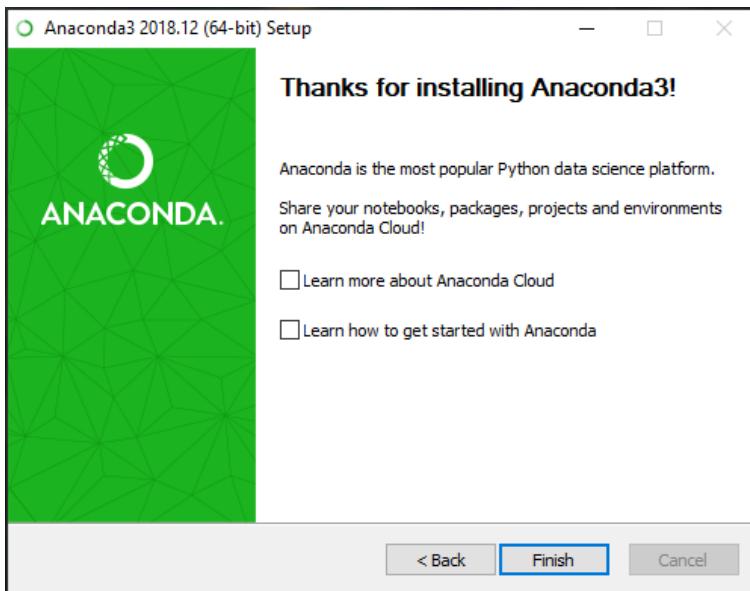
9. Kemudian klik "Next" untuk melanjutkan.



10. Selanjutnya kalian diberi pilihan untuk menginstall Microsoft VSCode. Saya klik "Skip" untuk melanjutkan.



11. Kemudian klik "Finish" untuk selesai.



12. Untuk mengecek apakah Anaconda telah terinstall yaitu dengan cara membuka Command Prompt. Lalu ketikan "conda -V" dan tekan enter, kode itu akan mengecek versi Anaconda yang terinstall.

A screenshot of a Windows Command Prompt window. The title bar says "C:\Windows\system32\cmd.exe". The window shows the following text:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.310]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\vanerz>conda -V
conda 4.5.12

C:\Users\vanerz>
```

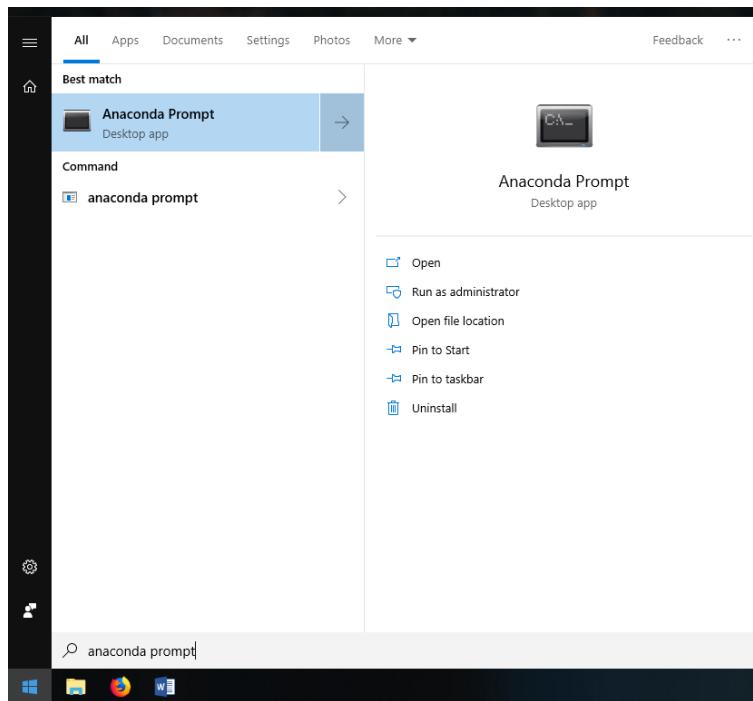
The "Finish" button from the previous screenshot is overlaid on the bottom right of this window.

1.8.3 Penggunaan Spyder

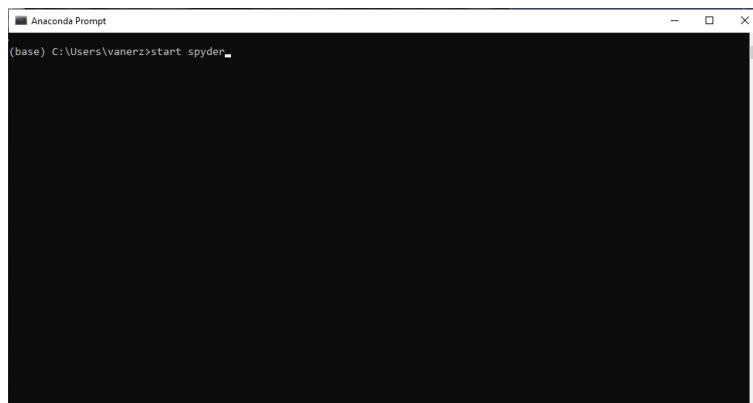
Terdapat 2 cara menjalankan Spyder. Yang pertama dengan Anaconda Prompt dan yang kedua dengan Anaconda Navigation. Berikut ini merupakan langkah-langkah cara menjalankan Spyder di windows:

- Anaconda Prompt

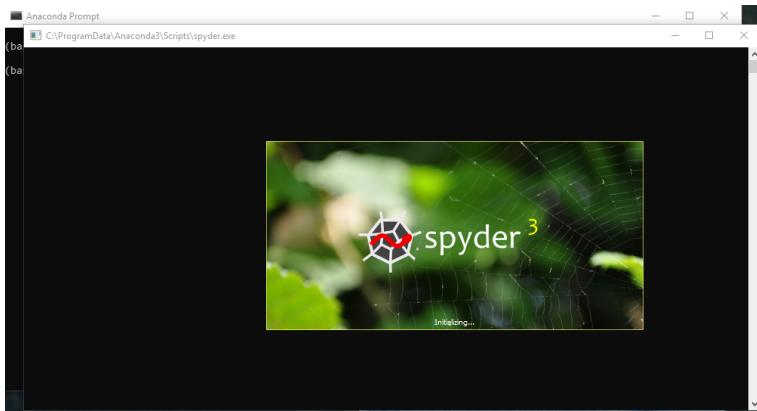
1. Pertama klik start, lalu cari "Anaconda Prompt".



2. Selanjutnya akan muncul sebuah prompt. Kemudian ketikan "start spyder" dan tekan enter.

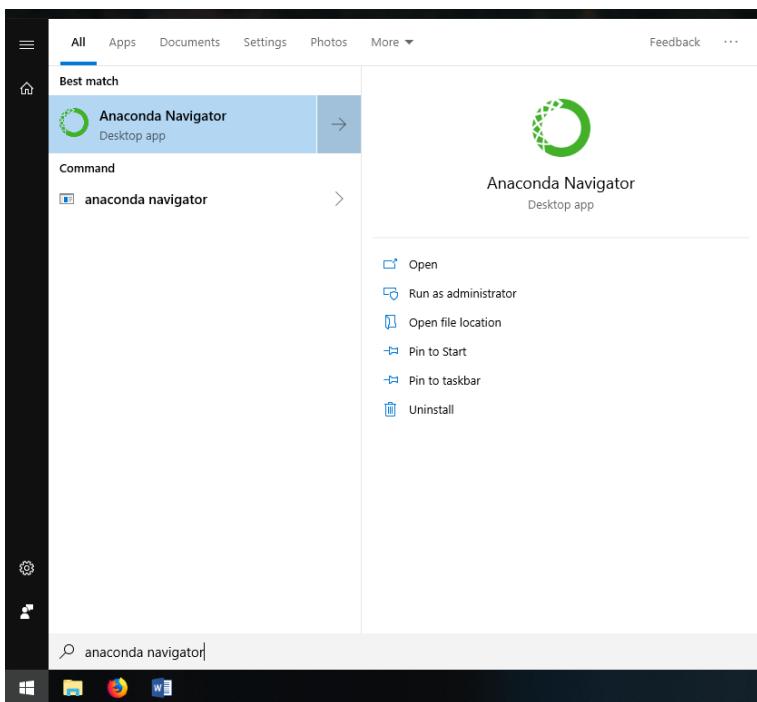


3. Lalu tunggu sampai selesai.

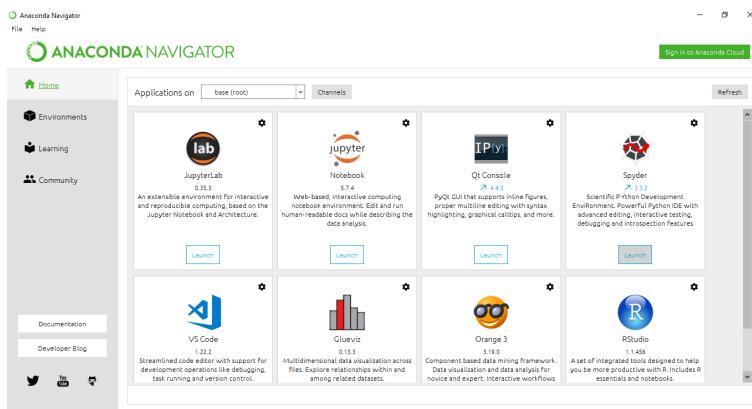


▪ Anaconda Navigation

1. Pertama klik start, lalu cari "Anaconda Navigation".



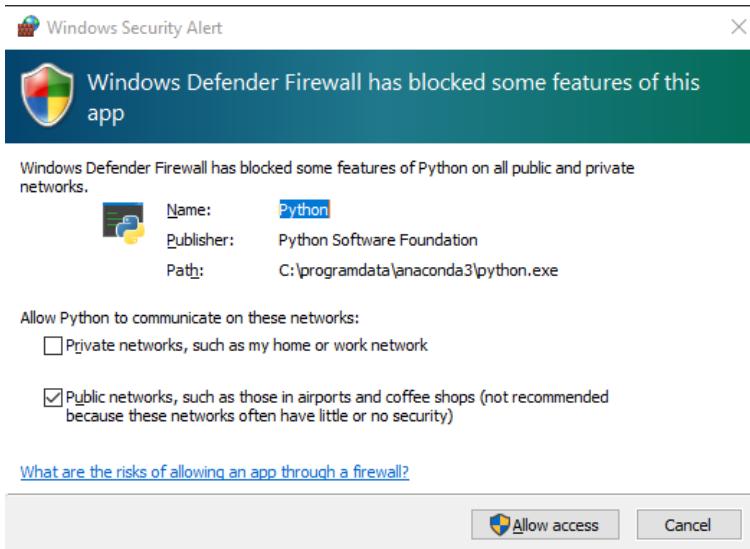
2. Selanjutnya akan muncul sebuah window. Kemudian klik "Launch" pada menu Spyder.



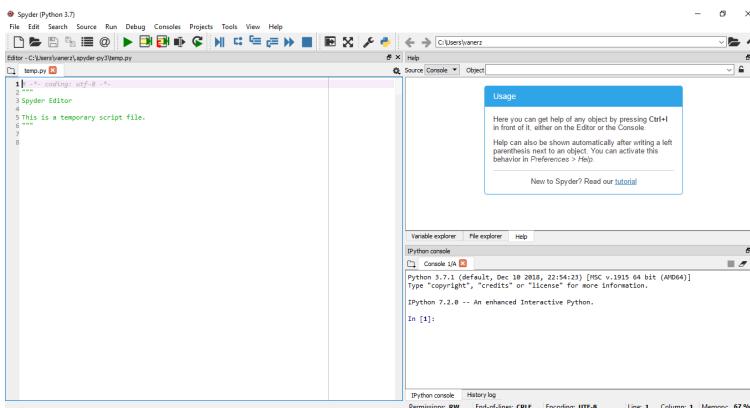
3. Lalu tunggu sampai selesai.



Apabila muncul window ini ketika pertama kali menjalankan Spyder, pilih Allow Access.

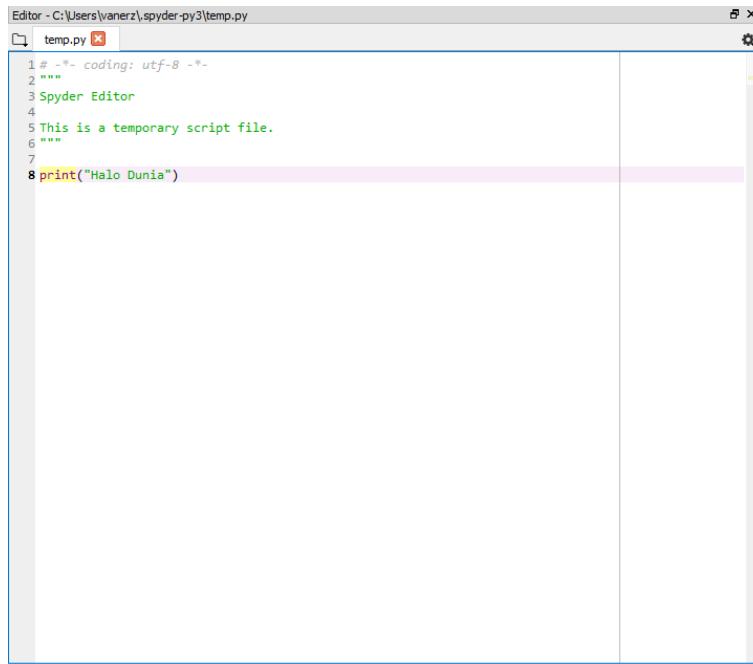


Berikut ini merupakan gambar dari Spyder



Berikut cara menggunakan Spyder:

1. Silahkan ketikan script Python anda di sini.



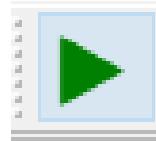
```

Editor - C:\Users\vanerz\spyder-py3\temp.py
temp.py

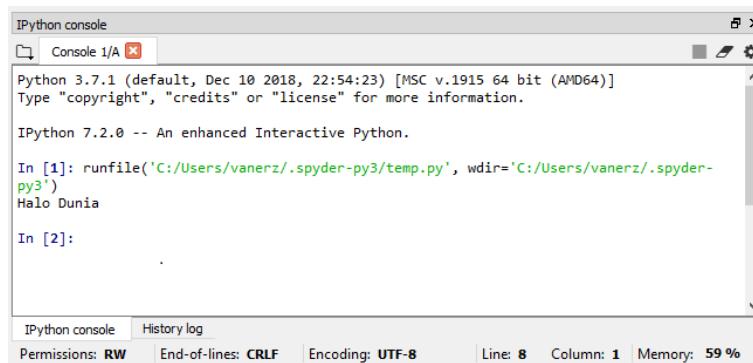
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8 print("Halo Dunia")

```

- Setelah mengetik script Python, kemudian klik tombol play atau tekan tombol F5 untuk mengeksekusi script Python yang telah diketik tadi.



- Hasil dari eksekusi akan muncul disini.



```

IPython console
Console 1/A

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.

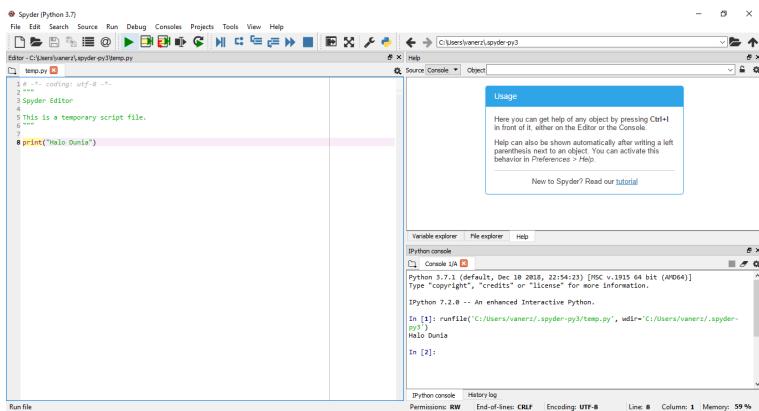
In [1]: runfile('C:/Users/vanerz/.spyder-py3/temp.py', wdir='C:/Users/vanerz/.spyder-py3')
Halo Dunia

In [2]:

```

IPython console History log
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 8 Column: 1 Memory: 59 %

4. Berikut tampilan penuhnya.



1.9 Harun Ar-Rasyid

1.9.1 Sejarah

Python diciptakan oleh Guido van Rossum pertama kali di Scitchting Mathematisch Centrum (CWI) di Belanda pada awal tahun 1990-an. Bahasa python terinspirasi dari bahasa pemrograman ABC. Sampai sekarang, Guido masih menjadi penulis utama untuk python, meskipun bersifat open source sehingga ribuan orang juga berkontribusi dalam mengembangkannya.

Pada 1995, Guido pindah ke CNRI di Virginia Amerika sambil terus mengembangkan Python. Versi terakhir yang dirilis adalah 1.6. Pada tahun 2000, pengembang inti Guido dan Python pindah ke BeOpen.com yang merupakan perusahaan komersial dan membentuk BeOpen PythonLabs. Python 2.0 dirilis oleh BeOpen. Setelah menghapus Python 2.0, Guido dan beberapa anggota tim PythonLabs pindah ke DigitalCreations.

Saat ini pengembangan Python terus dilakukan oleh sekelompok programmer yang dikoordinir oleh Guido dan Python Software Foundation. Python Software Foundation adalah organisasi nirlaba 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 telah mencapai versi 2.7.14 dan versi 3.6.3

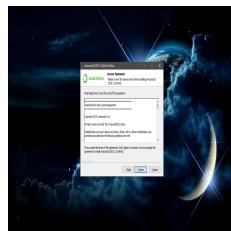
Nama Python dipilih oleh Guido sebagai nama bahasa ciptaananya karena kecintaan Guido pada acara televisi Flying Circus Monty Python. Oleh karena itu sering ekspresi khas acara sering muncul dalam korespondensi antara pengguna Python.

1.9.2 Instalasi Anaconda

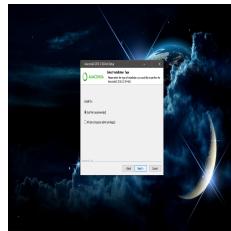
1. Pastikan Bahwa Python telah terinstal dilaptop anda.



Gambar 1.32 Tampilan Awal

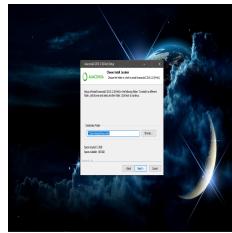


Gambar 1.33 License Agreement

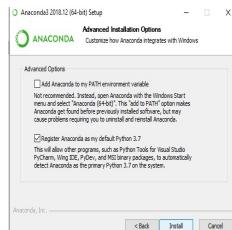


Gambar 1.34 Pemilihan User

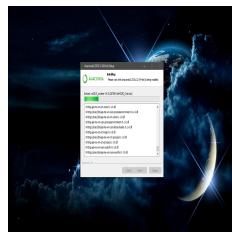
2. Kemudian Download Anaconda pada websitenya langsung.
3. Kemudian buka installer yang telah di download barusan
4. Klik next
5. Kemudian Klik I Agree
6. Kemudian pilih akan di instal untuk siapa, kemudian pilih next
7. Kemudian tentukan dicitrectory nya
8. Kemudian Centang yang register Anaconda as default Python, Kemudian Pilih Next
9. Tunggu Proses Instalasi hingga selesai



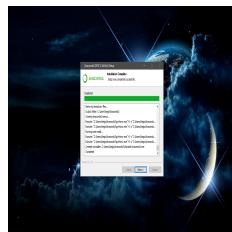
Gambar 1.35 Pemilihan Direktori Penyimpanan



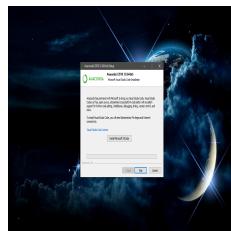
Gambar 1.36 Pemilihan Opsi



Gambar 1.37 Proses Instal



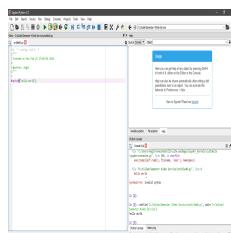
Gambar 1.38 Proses Instal Selesai



Gambar 1.39 Penawaran Instal MS VSC



Gambar 1.40 Instalasi Selesai



Gambar 1.41 Ini adalah tampilan spyder

10. Klik next
11. kemudian jika kalian belum instal MS VSC di sarankan menginstalnya dlu, jika sudah klik skip
12. Instalasi anaconda telah selesai

1.9.3 Menggunakan Spyder

Setelah selesai melakukan instalasi anaconda, maka ada beberapa tool yang digunakan seperti spyder

Gambar diatas menjelaskan tentang tampilan spider dan mengexekusi program halo world.

1.10 Felix Lase

1.10.1 sejarah

Python diciptakan oleh Guido van Rossum untuk pertama kalinya di Scitchting Mathematisch Centrum (CWI) di Belanda pada awal 1990-an. Bahasa Python terinspirasi oleh bahasa pemrograman ABC. Sampai sekarang, Guido masih menjadi penulis utama untuk Python, meskipun open source terbuka untuk ribuan orang yang juga berkontribusi pada pengembangannya.

Pada tahun 1995, Guido terus membuat python di Corporate for National Research Initiative (CNRI) di Virginia America, tempat ia merilis beberapa versi Python.

Pada bulan Mei 2000, Guido dan tim Python pindah ke BeOpen.com dan membentuk tim BeOpen PythonLabs. Pada bulan Oktober tahun yang sama, tim Python pindah ke Digital Creation (sekarang Zope Company). Pada tahun 2001, Organisasi Python dibentuk, Yayasan Perangkat Lunak Python (PSF). PSF adalah organisasi nirlaba yang khusus dibuat untuk semua hal yang berkaitan dengan kekayaan intelektual Python. Perusahaan Zope adalah anggota sponsor PSF.

Semua versi Python yang dirilis adalah open source. Dalam sejarahnya, hampir semua rilis python menggunakan lisensi yang kompatibel dengan GFL. Berikutnya adalah versi minor dari walikota dan python bersama dengan tanggal rilis. Instalasi anaconda

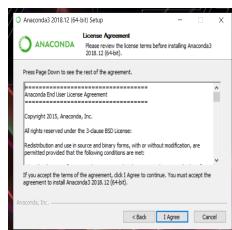
1.10.2 Instalasi Anaconda

1. Terlebih dahulu kita harus mendownload python, sebelum anaconda diinstall
2. Buka installer klik Next

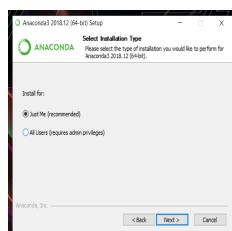


Gambar 1.42 Tampilan Awal

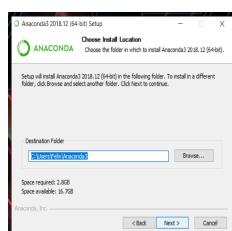
3. Klik I Agree untuk membuka lisensi
4. Pilih untuk siapa aplikasi diinstal bisa just me dan juga bisa all users
5. Pilih lokasi instalasi
6. Pilih register anaconda karna add aconda environment tidak recommended



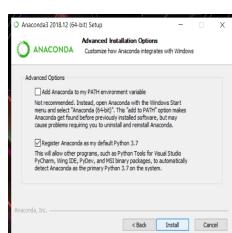
Gambar 1.43 License



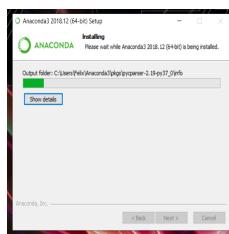
Gambar 1.44 Proses



Gambar 1.45 Proses



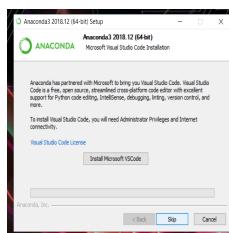
Gambar 1.46 Proses



Gambar 1.47 Proses

7. Tunggu hingga selesai

8. Klik skip



Gambar 1.48 Proses

9. Dan anaconda berhasil di install

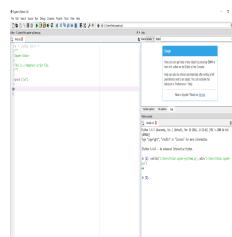


Gambar 1.49 Proses

1.10.3 Menggunakan Spyder

setelah selesai menggunakan instalasi anaconda, maka ada beberapa tool yang digunakan seperti spyder

Gambar tersebut menjelaskan tentang tampilan spyder dan mengeksekusi program aa



Gambar 1.50 Proses

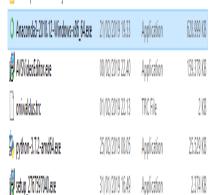
1.11 Oniwaldus Bere Mali

1.11.1 Sejarah Python

Python dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Amsterdam sebagai kelanjutan dari bahasa pemrograman ABC. Tahun 1995, Guido pindah ke CNRI 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 DigitalCreations. 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 2.6.1 dan versi 3.0. Nama Python dipilih oleh Guido sebagai nama bahasa ciptaannya karena kecintaan guido pada acara televisi Monty Python's Flying Circus. Oleh karena itu seringkali ungkapan-ungkapan khas dari acara tersebut seringkali muncul dalam korespondensi antar pengguna Python.

1.11.2 Instalasi Anaconda

1. Pastikan Bawa Python telah terinstal dilaptop anda.
2. Jika anda belum punya anaconda, kalian bisa download
3. Kemudian buka installer yang telah di download barusan
4. Klik next

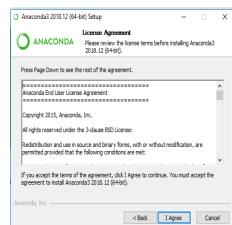


5. klik next

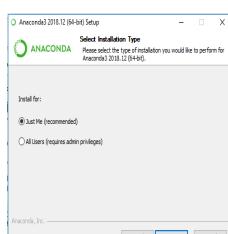


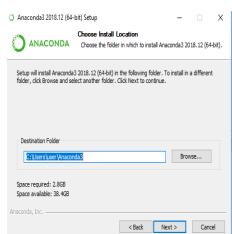
Gambar 1.51 Klik Next

6. Klik pada I Agreee

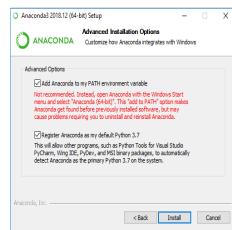


Gambar 1.52 Klik pada I Agreeer



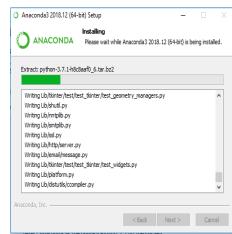


7. Ceklis pada Add Anaconda to my PATH environment variable dan Register Anaconda as my default Python 3.7 . selanjutnya klik Install



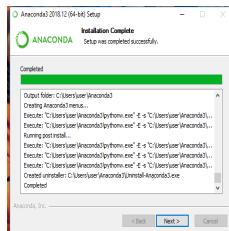
Gambar 1.53 Ceklis pada Add Anaconda to my PATH environment variable dan Register Anaconda as my default Python 3.7 .selanjutnya klik Install

- ## 8. Klik next



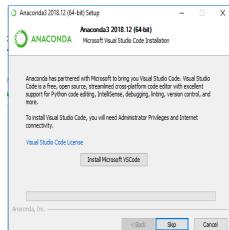
Gambar 1.54 Proses Instalasi

9. klik next



Gambar 1.55 klik next

10. instal MS VSC di sarankan menginstalnya dulu



Gambar 1.56 Klik install microsoft vscode

11. Instalasi anaconda telah selesai



Gambar 1.57 Instalasi Selesai

1.11.3 Menggunakan Spyder

```

temp.py x
1 # -*- coding: utf-8 -*-
2 ***
3 Spyder Editor
4
5 This is a temporary script file.
6 ***
7 print("Hello world")
8

```

```

Python 3.7.1 (default, Dec 10 2018, 23:54:52) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

Python 3.7.0 -- An enhanced Interactive Python.

[1]:
```

```

In [1]: readline('C:\users\user\syahrizq\temp.py', write=C:\users\user\syahrizq\temp.py')
Hello world

[2]:
```

1.12 DezhaAidiMartha

1.12.1 Sejarah

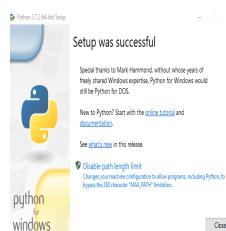
Python merupakan bahasa pemrograman yang diciptakan oleh Guido van Rossum. Hingga sekarang, bahasa pemrograman Python ini masih dikembangkan oleh *Python Software Foundation*. Pada awalnya bahasa pemrograman ini dikembangkan untuk membuat skrip bahasa pemrograman tingkat tinggi pada suatu sistem operasi yang terdistribusi oleh Amoeba. Python ini telah digunakan oleh beberapa pengembang dan bahkan telah digunakan oleh beberapa perusahaan untuk pembuatan perangkat lunak komersial. Dan lebihnya lagi bahasa pemrograman Python ini gratis atau *freeware*, sehingga dapat dikembangkan, dan tidak ada batasan dalam penyalinannya.

Bahasa pemrograman python dikembangkan oleh Guido van Rossum pada akhir tahun 80-an dan awal tahun 90-an di National Research Institute for Mathematics and Computer Science di Belanda. Python berasal dari banyak bahasa latin, termasuk ABC, Modula-3, C, C++, Algol-68, SmallTalk, dan shell Unix dan bahasa script lainnya.

Saat ini pengembangan Python terus dilakukan oleh sekompulan programmer yang di koordinir oleh 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 mencegah Python dimiliki oleh perusahaan komersial. Saat ini distribusi Python sudah mencapai versi 3.7.2.

1.12.2 InstalasiAnaconda

- Pastikan bahwa ython telah terinstall.



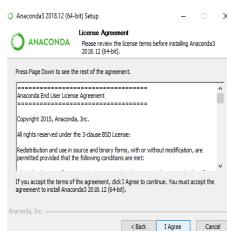
- Jika anda belum punya anaconda,kalian bisa download.

3. Kemudian buka installer yang telah di download.

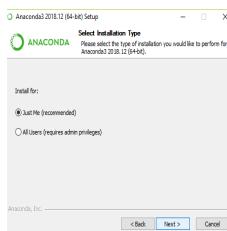
4. Klik next



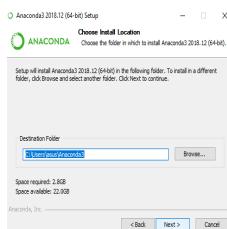
5. Klik I Agree



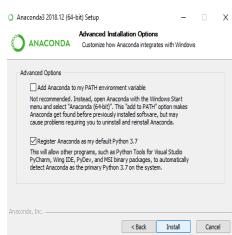
6. Klik next



7. Klik next



8. Klik Install

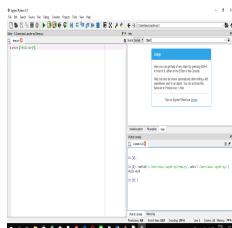


9. Tunggu hingga selesai



1.12.3 Spyder

1. Running Spyder dan lakukan code dibawah ini



1.13 Choirul Anam

1.13.1 Sejarah Phyton

Guido Van Rossum adalah orang yang menciptakan python di Scitching Matematisch Centrum (CWI) Belanda pada wal tahun 1990-an. Pada bahasa python sendiri terinspirasi dari salahsatu pemrograman ABC. Sampai sekarangpun Guido Van Rossum masih menjadi orang yang pertama untuk python, walaupun bersifat open source, dengan itu banyak orang bahkan ribuan orang dapat berkontribusi dalam mengembangkannya.

Pada tahun 1995, Di Virginia Amerika Guido Van Rossum melanjutkan mengembangkan python di Corporation for National Reserch Initiative (CNRI), dia merilis beberapa versi pada python.

Pada Mei tahun 2000, Guido Van Rossum dan tim pythonnya pindah ke BeOpen.com dan membentuk tim baru yaitu BeOpen PythonLabs, Pada bulan Oktober di tahun yang sama, timnya pindah ke Digital Creation dan sekarang menjadi perusahaan Zope.

Dibentuklah organisasi python yaitu Python Software Foundation (PSF). Organisasi ini merupakan organisasi nirlaba yang dibentuk khusus untuk semua hal yang bertentangan dengan hak intelektual python. Perusahaan Zope menjadi anggota sponsor dari organisasi PSF (Python Software Foundation).

Pada nama python sendiri berasal dari nama ular yang kita tau. Guido Van Rossum seorang yang menggemarkan grub komedi inggris yang bernama Monty Python, Lalu ia menamakan ciptaananya dengan nama Python.

Untuk versi python yang pernah dirilis semuanya bersifat open source. Python yang dirilis hamper semuanya menggunakan lisensi GFL-compatible.

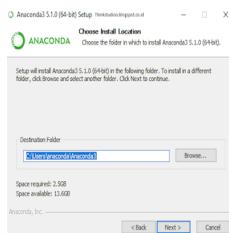
1.13.2 Instalasi Anaconda

- Sebelumnya anda download file installernya di <http://repo.continuum.io/archive> silahkan anda pilih versi python 2 atau python 3 dan versi anaconda berapa, disini menggunakan anaconda versi Anaconda3 5.1.0 64bit.

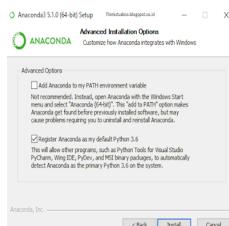
Anaconda installer archive http://repo.continuum.io/archive			
Filename	Size	Last Modified	MD5
anaconda-5.1.0-Linux-x86_64.sh	257,37	2016-01-13 01:01:49	e9f4e5c2a297d989f8a722a
anaconda-5.1.0-Linux-x86.sh	403,39	2016-01-13 01:01:51	a2f05e5d5099f22227bfef07
anaconda-5.1.0-Linux-x86_64.exe	553,39	2016-01-13 01:01:59	93fb54e4cf1f4d64058070
anaconda-5.1.0-Linux-x86.exe	560,39	2016-01-13 01:01:59	63f054e4cf1f4d64058070
anaconda-5.1.0-MacOSX-x86_64.dmg	108,00	2016-01-13 01:01:59	63f054e4cf1f4d64058070
anaconda-5.1.0-MacOSX-x86.dmg	108,00	2016-01-13 01:01:59	63f054e4cf1f4d64058070
anaconda-5.1.0-MacOSX-x86_64.tar.gz	43,49	2016-01-13 01:01:59	63f054e4cf1f4d64058070
anaconda-5.1.0-MacOSX-x86.tar.gz	52,07	2016-01-13 01:01:59	63f054e4cf1f4d64058070
anaconda-5.1.0-Linux-x86_64.tgz	265,75	2016-01-13 01:01:59	45b6895919e7ca74e7a70
anaconda-5.1.0-Linux-x86.sh	48,37	2016-01-13 01:01:59	7156eff78a6d948e4e4d
anaconda-5.1.0-Linux-x86.exe	50,22	2016-01-13 01:01:59	7156eff78a6d948e4e4d
anaconda-5.1.0-MacOSX-x86_64.dmg	108,00	2016-01-13 01:01:59	7156eff78a6d948e4e4d
anaconda-5.1.0-MacOSX-x86.dmg	108,00	2016-01-13 01:01:59	7156eff78a6d948e4e4d
anaconda-5.1.0-MacOSX-x86_64.tar.gz	53,17	2016-01-13 01:01:59	7156eff78a6d948e4e4d
anaconda-5.1.0-MacOSX-x86.tar.gz	65,75	2016-01-13 01:01:59	7156eff78a6d948e4e4d
anaconda-5.1.0-Linux-x86_64.exe	557,37	2016-01-13 01:01:59	63f054e4cf1f4d64058070
anaconda-5.1.0-Linux-x86.sh	403,39	2016-01-13 01:01:59	a2f05e5d5099f22227bfef07

Gambar 1.58 gambar1

- Jika anda sudah memilih versi yang mana silahkan download dan tunggu proses downloadnya selesai, Setelah di download silahkan ke file download tersebut lalu klik dukali pada file installernya kemudian klik next.
- Lalu read lisensi dan klik I Agree, setelah itu akan muncul tampilan seperti berikut dan silahkan klik next.
- Jika anda sudah klik next lalu pilih Ad anaconda to path lalu aka nada tampilan seperti berikut dan biarkan kotak tersebut tercentang anda klik next saja.
- Jika anda sudah klik next lalu klik tombol install, dan jika anda ingin melihat pakages anaconda yang sedang di pasang silahkan anda klik show details.
- Selanjutnya anda klik next, lalu untuk menginstal VS codenya silahkan klik Install Microsoft VS code, setelah proses instalasi selesai lalu anda pilih next, jika anda ingin menginstal anaconda tanpa VS codenya pilih skip.



Gambar 1.59 gambar2



Gambar 1.60 gambar3



Gambar 1.61 gambar4

- Setelah proses instalasinya selesai maka akan muncul tampilan tersebut dengan penanda bahwa anaconda sudah terinstal di laptop atau computer anda lalu anda pilih Finish.

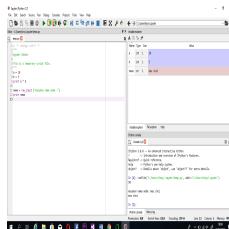


Gambar 1.62 gambar5

1.13.3 Menggunakan Spyder

Pada penulisan script kode-kode python biasanya di tulis di text editor seperti notepad, sublime dan semacamnya lalu untuk menjalankannya di jalankan melalui cmd. Tapi disini ada yang beda dimana ada text editor untuk kode-kode python dan bisa di jalankan saat klik run saja, text editor tersebut yaitu Spyder.

Spyder merupakan text editor dari anaconda dimana di tools ini bisa menjalankan perintah-perintah dengan klik run saja. Di anaconda sendiri mempunyai beberapa text editor untuk python tapi yang akan dijelaskan kali ini menggunakan spyder.



Gambar 1.63 gambar6

1.14 Nico Ekklesia Sembiring

1.14.1 sejarah python

Python merupakan bahasa pemrograman yang dikembangkan oleh Guido van Rossum pada tahun akhir tahun 1980 di CWI, Belanda. Kemudian diimplementasikan pada Desember 1989 sebagai kelanjutan dari bahasa pemrograman ABC dan dapat berinteraksi dengan sistem operasi Amuba. Versi terakhir yang dirilis oleh CWI adalah

1.2. Kemudian pada tahun 1995 Guido pindah ke CNRI sambil terus melanjutkan pengembangan Python. Versi terakhir yang dirilis oleh CNRI adalah versi 1.6 yang dirilis pada Tahun 2000.

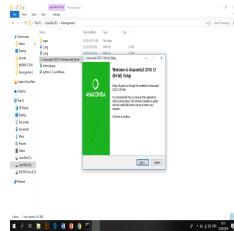
Selanjutnya Guido dan para pengembang inti Python pindah ke BeOpen.com yang merupakan sebuah perusahaan komersial dan mereka membentuk BeOpen Python-Labs. BeOpen kemudian merilis Python versi 2.0. Setelah merilis Python 2.0, Guido dan beberapa anggota tim PythonLabs pindah ke DigitalCreations.

Nama Python ini dipilih oleh Guido yang merupakan pengembangnya sebagai nama Bahasa pemrograman yang dibuatnya karena kecintaan Guido pada salah satu acara televisi Monty Pythons Flying Circus. Oleh karena itu seringkali ungkapan-ungkapan khas dari acara tersebut muncul dalam korespondensi antar pengguna Python

Hingga saat ini pengembangan Python terus dilakukan oleh sekumpulan pemrogram yang dikoordinir Guido dan Python Software Foundation. Python Software Foundation merupakan sebuah organisasi non-profit yang dibentuk sebagai pemegang hak cipta intelektual Python sejak versi 2.1. Hal ini untuk mencegah Python dimiliki oleh perusahaan komersial. Saat ini Python sudah dirilis hingga versi 2.7.13 dan juga versi 3.6.0.

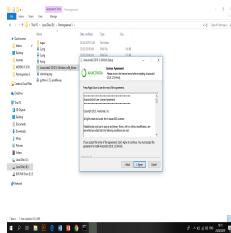
1.14.2 Menginstal Anaconda

1. Pastikan terlebih dahulu bahwa Python telah terinstal dilaptop anda.
2. Untuk melakukan instalasi Anaconda, kita harus punya file setup Anaconda nya terlebih dahulu. Kemudian Pilih file nya

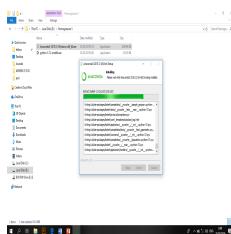


Gambar 1.64 File Anaconda

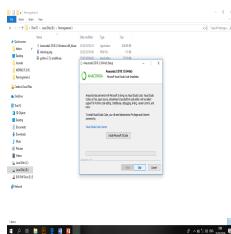
3. Setelah setup terbuka, maka akan muncul informasi seperti berikut. Kemudian klik Next
4. Setelah itu akan muncul persetujuan lisensi. Setelah membaca lisensi tersebut, kita memilih I Agree untuk melanjutkan menginstal Anaconda
5. Setelah menyetui lisensi, kita diharuskan memilih tipe instalasi. Setelah memilih tipe instalasi, maka kita klik Next
6. Selanjutnya kita diharuskan memilih lokasi untuk diinstal. Setelah memilih lokasi, klik Next



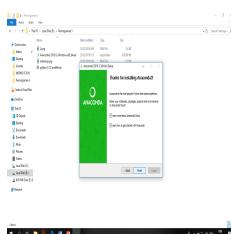
Gambar 1.65 Tampilan Awal



Gambar 1.66 Persetujuan Lisensi

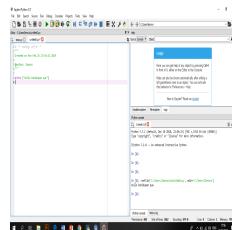


Gambar 1.67 Tipe Instalasi



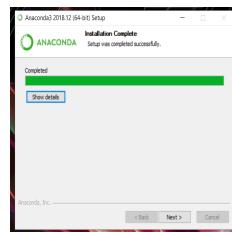
Gambar 1.68 Pilih Lokasi

- Setelah itu centang pada kotak register. Lalu pilih Instal



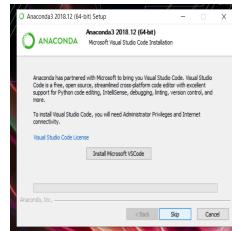
Gambar 1.69 Advanced Options

- Setelah install selesai, maka pilih Next.



Gambar 1.70 Instal Selesai

- Jika tidak akan menginstal VSCode, maka bisa memilih Skip



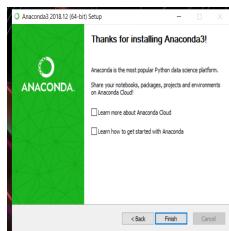
Gambar 1.71 Instal VSCode

- Proses instalasi telah selesai. Kemudian pilih finish.

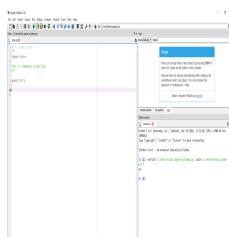
1.14.3 Menggunakan Spyder

Spyder merupakan tool yang tersedia saat anaconda diinstall.

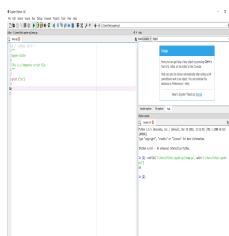
Gambar merupakan tampilan kode sederhana saat menggunakan spyder. berikut adalah hasilnya



Gambar 1.72 Instalasi Selesai



Gambar 1.73 Kode



Gambar 1.74 Hasil Kode

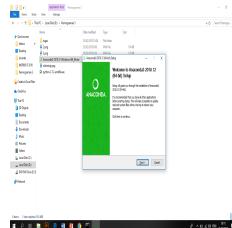
1.15 Damara Benedicta

1.15.1 Sejarah

Python merupakan suatu bahasa pemrograman skrip yang tidak sulit untuk dibaca dan ditulis, dan tidak memiliki fungsi yang terbatas sehingga dapat digunakan untuk menyelesaikan berbagai macam tugas Python awal mula dikembangkan tahun 1990 oleh guido van Rossum di Amsterdam. Nama python berasal dari nama yang dipilih oleh guido dari acara televisi sirkus monty python. Pada 1995 versi yang dirilis oleh CWI adalah versi 1.2. dan versi terakhir pada tahun 2000 adalah versi 1.6. dan python 2.0 mulai dirilis oleh BeOpen. Setelah menghapus Python 2.0, Guido dan anggota tim PythonLabs lainnya pindah ke DigitalCreations. Distribusi Python telah mencapai versi 2.6.1 dan versi 3.0, dan dengan demikian mencegah Python dimiliki oleh perusahaan komersial. Saat ini distribusi Python telah mencapai versi 2.7.14 dan versi 3.6.3

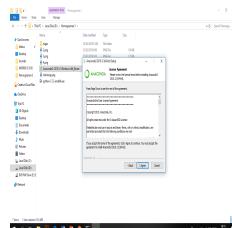
1.15.2 Instalasi Anaconda

1. Pastikan Bawa Python telah terinstal dilaptop anda.
2. Kemudian buka installer yang telah di download barusan
3. Klik next



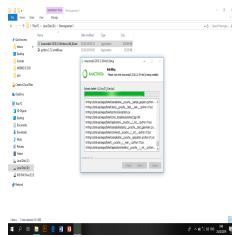
Gambar 1.75 Tampilan Awal

4. Kemudian Klik Next



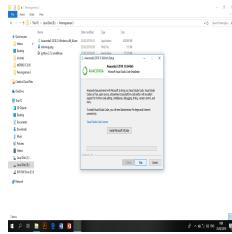
Gambar 1.76 License Agreement

5. Kemudian akan muncul persetujuan lisensi, setelah itu pilih " I Agree"



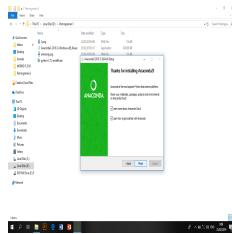
Gambar 1.77 Pemilihan User

6. Kemudian memilih tipe instalasi



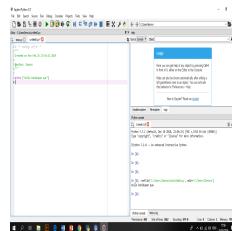
Gambar 1.78 Pemilihan Direktori Penyimpanan

7. Kemudian pilih lokasi untuk instal, setelah itu klik Next

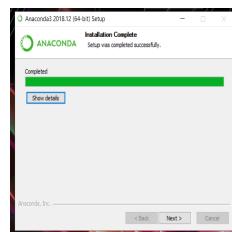


Gambar 1.79 Pemilihan Opsi

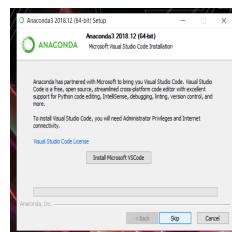
8. Kemudian centang pada kotak register dan pilih instal
9. Kemudian Klik next
10. kemudian jika tidak akan mengistal VSCode, maka pilih skip
11. kemudian Proses intalasi telah selesai pilih finish



Gambar 1.80 Proses Instal



Gambar 1.81 Proses Instal Selesai



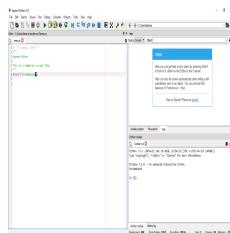
Gambar 1.82 Penawaran Instal MS VSC



Gambar 1.83 Penawaran Instal MS VSC

1.15.3 (

Spyder) Spyder adalah tools yang dipakai untuk python



Gambar 1.84 Instalasi Selesai

1.16 Damara Benedicta

1.16.1 Sejarah

Python merupakan suatu bahasa pemrograman skrip yang tidak sulit untuk dibaca dan ditulis, dan tidak memiliki fungsi yang terbatas sehingga dapat digunakan untuk menyelesaikan berbagai macam tugas Python awal mula dikembangkan tahun 1990 oleh guido van Rossum di Amsterdam. Nama python berasal dari nama yang dipilih oleh guido dari acara televisi sirkus monty python. Pada 1995 versi yang dirilis oleh CWI adalah versi 1.2. dan versi terakhir pada tahun 2000 adalah versi 1.6. dan python 2.0 mulai dirilis oleh BeOpen. Setelah menghapus Python 2.0, Guido dan anggota tim PythonLabs lainnya pindah ke DigitalCreations. Distribusi Python telah mencapai versi 2.6.1 dan versi 3.0, dan dengan demikian mencegah Python dimiliki oleh perusahaan komersial. Saat ini distribusi Python telah mencapai versi 2.7.14 dan versi 3.6.3

1.16.2 Instalasi Anaconda

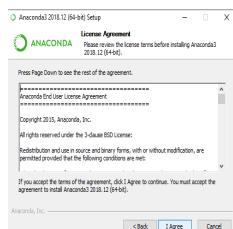
1. Pastikan Bahwa Python telah terinstal dilaptop anda.
2. Jika anda belum punya anaconda, kalian bisa download
3. Kemudian buka installer yang telah di download barusan
4. Klik next
5. Kemudian Klik Next
6. Kemudian akan muncul persetujuan lisensi, setelah itu pilih " I Agree"
7. Kemudian memilih tipe instalasi



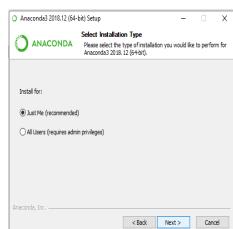
Gambar 1.85 Tampilan Awal



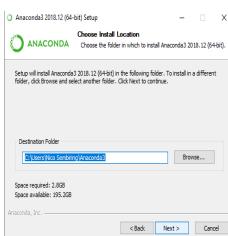
Gambar 1.86 License Agreement



Gambar 1.87 Pemilihan User

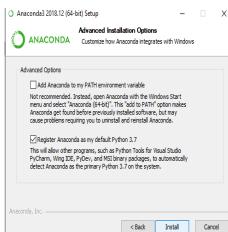


Gambar 1.88 Pemilihan Direktori Penyimpanan



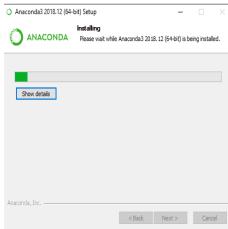
Gambar 1.89 Pemilihan Opsi

8. Kemudian pilih lokasi untuk instal, setelah itu klik Next
9. Kemudian centang pada kotak register dan pilih instal



Gambar 1.90 Proses Instal

10. Kemudian Klik next

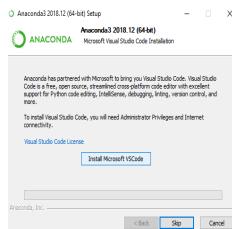


Gambar 1.91 Proses Instal Selesai

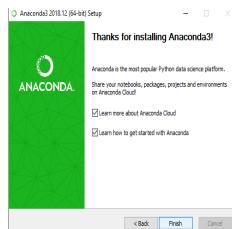
11. kemudian jika tidak akan mengistal VSCode, maka pilih skip
12. kemudian Proses intalasi telah selesai pilih finish

1.16.3 (

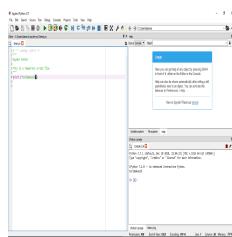
Spyder) Spyder adalah tools yang dipakai untuk python



Gambar 1.92 Penawaran Instal MS VSC



Gambar 1.93 Penawaran Instal MS VSC



Gambar 1.94 Instalasi Selesai

1.17 Arjun Yuda Firwanda

1.17.1 Sejarah Python

Bahasa Python merupakan Bahasa pemrograman interpretatif multiguna dengan memanfaatkan perkembangan Bahasa pemrograman yang lebih fokus meningkatkan keterbacaan sebuah kode. Python digunakan sebagai Bahasa pemrograman yang menggabungkan kapabilitas, kemampuan, sintak kode yang sangat jelas.

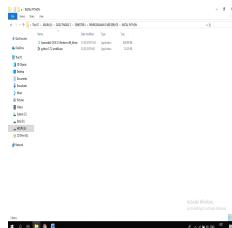
Salah satu fitur yang tersedia pada bahasa python sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis.

Saat ini kode python dapat dijalankan di berbagai platform sistem operasi, beberapa diantaranya Linux/Unix, Windows, Mac OS X, Java Virtual Machine, OS/2, Amiga, Palm.

Perkembangan Bahasa python digunakan untuk keperluan pengembangan perangkat lunak dan dapat berjalan diberbagai platform sistem operasi. Bahasa pemrograman Python awalnya dikembangkan Guido van Rossum pada tahun 1990 di CWI, Amsterdam. Versi terakhir yang dikeluarkan CWI adalah 1.2 yang saat ini banyak digunakan oleh berbagai perusahaan di dunia.

1.17.2 Instalasi Anaconda

1. Langkah pertama harus install Anaconda di Laptop/Pc anda.
2. Jika yang belum download, silahkan download di google.
3. Klik kanan pada unduhan Anaconda, klik kanan, pilih Run As Administrator.
4. Klik next

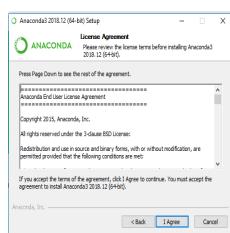


Gambar 1.95 File Instalasi Anaconda

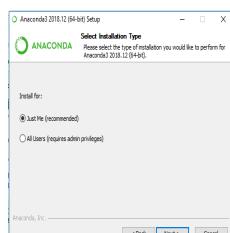
5. Klik Next
6. Klik I Agree
7. Klik Just Me dan Klik Next
8. Klik Next



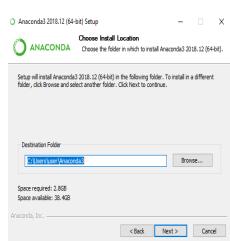
Gambar 1.96 Tampilan Awal Instalasi Anconda



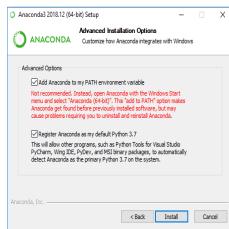
Gambar 1.97 Tampilan Licence Anaconda



Gambar 1.98 Memilih Type Instalasi



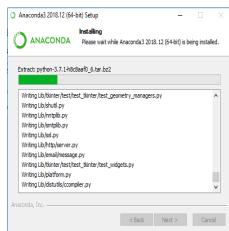
Gambar 1.99 Penyimpanan Instalasi Anaconda



Gambar 1.100 Opsi Anaconda

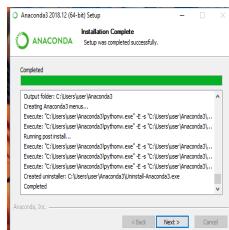
9. Pilih Semua Kemudian Klik Next

10. Menunggu Instalasi



Gambar 1.101 Proses Installing

11. Setelah Completed Klik Next



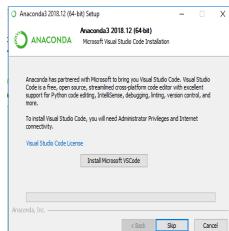
Gambar 1.102 Installing Completed

12. Setelah Completed Klik Instal Microsoft VSCode

13. Klik Finish

1.17.3 Spyder Python

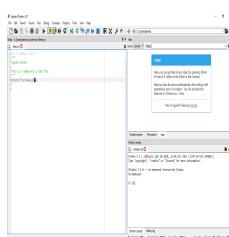
Spyder merupakan sebuah tools atau media yang dipakai dalam bahasa python.



Gambar 1.103 Hak Akses Anaconda



Gambar 1.104 Instalasi Selesai



Gambar 1.105 Spyder Python

1.18 muhammad dzihan al-banna

Python dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Amsterdam, Belanda sebagai pengembangan dari bahasa pemrograman ABC. Pada Tahun 1995, Guido pindah ke CNRI dan melanjutkan pengembangan Python. Saat ini pengembangan Python terus dilakukan oleh sekumpulan developer yang dipimpin oleh 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 dan versi 3.7

Nama Python diambil oleh Guido sebagai nama bahasa ciptaannya karena kecintaan Guido pada acara televisi Monty Pythons Flying Circus. Oleh karena itu seringkali ungkapan-ungkapan khas dari acara tersebut seringkali muncul dalam korespondensi antar pengguna Python.

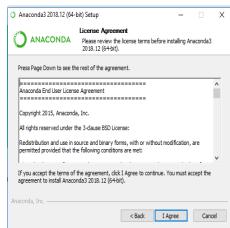
1.18.1 Instalasi Anaconda

1. Instal python terlebih dahulu.
2. setelah itu download anaconda.
3. Kemudian lakukan instalasi dan ikuti langkah-langkahnya.
4. Klik next

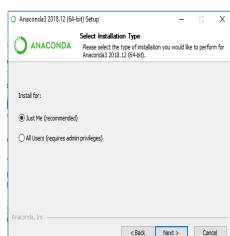


Gambar 1.106 Instal file

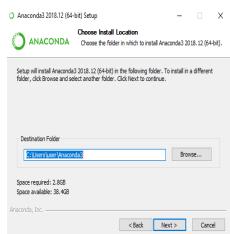
5. Kemudian Klik I Agree
6. Kemudian pilih akan di instal untuk siapa, kemudian pilih next
7. Kemudian tentukan directory nya, secara default akan berada di C
8. Kemudian Centang yang register Anaconda as default Python, Kemudian Pilih Next
9. Tunggu Proses Instalasi hingga selesai
10. Klik next



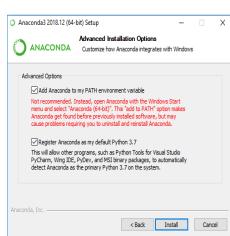
Gambar 1.107 Lisensi



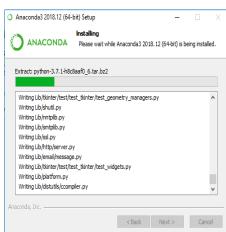
Gambar 1.108 Pemilihan User



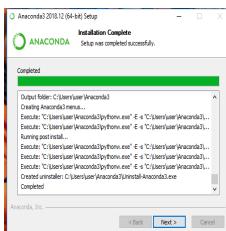
Gambar 1.109 Pemilihan Direktori Penyimpanan



Gambar 1.110 Pemilihan Opsi

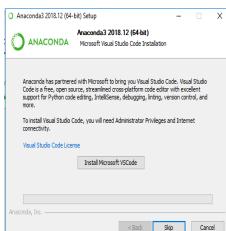


Gambar 1.111 Proses Instalasi



Gambar 1.112 Instal selesai

11. kemudian jika kalian belum instal MS VSC di sarankan menginstalnya dlu, jika sudah klik skip



Gambar 1.113 Penawaran Instal MS VSC

- 12 Instalasi anaconda telah selesai



Gambar 1.114 Instalasi Selesai

1.18.2 Menggunakan Spyder

Merupakan tool bawaan dari anaconda

1.19 Muhammad Fahmi

1.19.1 Sejarah Python

Guido Van Rossum adalah orang yang menciptakan python di Scitching Mathematisch Centrum (CWI) Belanda pada wal tahun 1990-an. Pada nama python sendiri berasal dari nama ular yang kita tau. Guido Van Rossum seorang yang mengge-markan grub komedi inggris yang bernama Monty Python, Lalu ia menamakan ciptaannya dengan nama Python Bahasa python terinspirasi dari bahasa pemrograman ABC. Nama python tidak berasal dari nama ular yang kita kenal. Guido merupakan penggemar grup komedi Inggris bernama Monty Python. Kemudian, ia menamakan Bahasa pemrograman ciptaannya dengan nama Python. Pada tahun 1994, Python 1.0 dirilis, yang diikuti dengan Python 2.0 pada tahun 2000. Python 3.0 keluar pada tahun 2008.

1.19.2 Instalasi Anaconda

1. Pastikan Bahwa Python telah terinstall dilaptop anda.

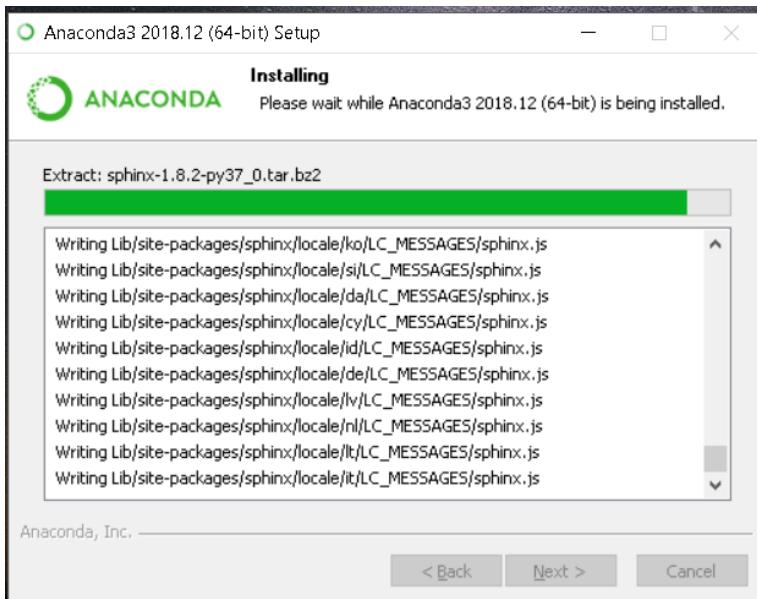
2. Download File Installer Anaconda pada www.anaconda.com

3. Install seperti biasa

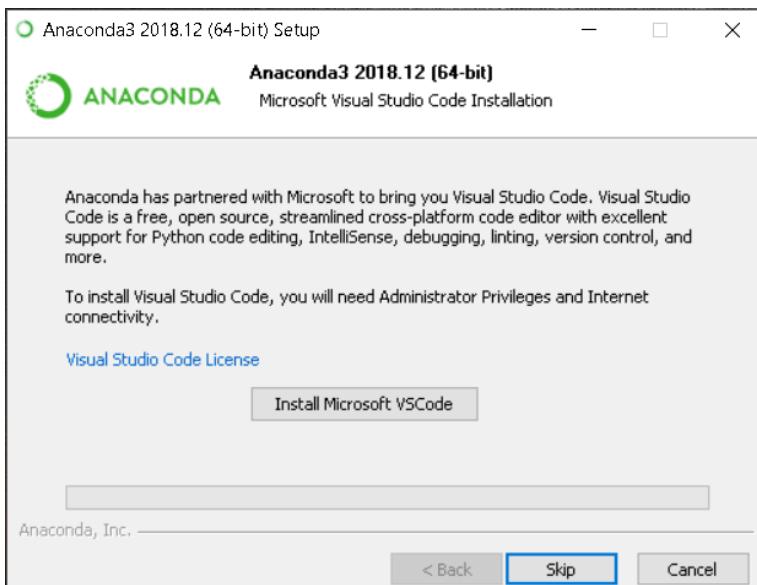
4. Kemudian Klik I Agree

5. Kemudian Centang yang register Anaconda as default Python, Kemudian Pilih Next

6. Tunggu Proses Instalasi hingga selesai



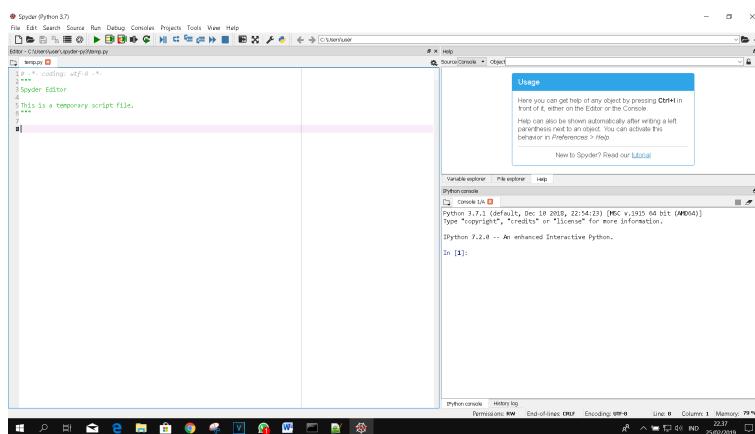
7. Instalasi telah selesai



1.19.3 Penggunaan Spider

Spider merupakan text editor dari anaconda dimana di tools ini bisa menjalankan perintah-perintah dengan klik run saja. Di anaconda sendiri mempunyai beberapa text editor untuk python tapi yang akan dijelaskan kali ini menggunakan spyder.

Spider sendiri sudah terinstall bersama pada saat kita menginstall Anacoda tadi. ini adalah contoh tampilan pada Spider.



1.20 Evietania Charis Sujadi

1.21 Python

1.21.1 Background

Python adalah sebuah bahasa pemrograman dengan level tinggi yang interaktif, dan mendukung berbagai paradigma pemrograman. Python sudah terkenal pada kala-gan programmer sebagai bahasa yang mudah dipahami dan memiliki kompleksitas yang dinamis sehingga dapat dipakai di algoritma maupun platform yang berbagai macam. Python sudah memiliki banyak komunitas pendukung karena penggunanya yang banyak. Selain pada komunitas biasa, Python sudah diimplementasikan pada banyak perusahaan ternama dan dipasang pada aplikasi yang sudah terkenal seperti pada search engine google yang dimiliki oleh perusahaan Google.

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

blok program, sub program tersebut harus diletakkan pada satu atau lebih spasi dari kolom sebuah blok

1.21.2 Problems

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

1.21.3 Objective and Contribution

1.21.3.1 Objective

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

1.21.3.2 Contribution

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

1.21.4 Scoop and Environtment

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

1.22 Habib Abdul Rasyid

1.22.1 Sejarah Python

Python adalah salah satu bahasa pemrograman yang paling portabel, nyaman dan kuat yang tersedia saat ini. Ini juga didistribusikan secara bebas sebagai kode sumber yang dapat dimodifikasi dan didistribusikan kembali dalam produk komersial. Setiap programmer layak menikmati Python, dan Pemrograman Internet dengan Python menunjukkan caranya. Watters, van Rossum (penulis utama Python), dan James C. Ahlstrom mendemonstrasikan metode pemrograman dalam Python, dengan penekanan khusus pada aplikasi terkait internet[?].

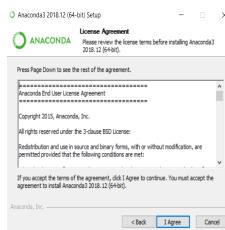
1.22.2 Instalasi Anaconda

1. Pastikan Bahwa Python telah terinstal dilaptop anda.
2. Kemudian Download Anaconda pada websitenya.
3. Buka installer yang telah di download barusan
4. Klik next



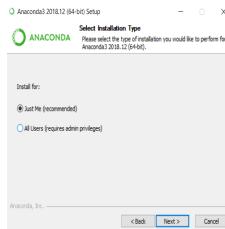
Gambar 1.115 instalasi

5. klik I Agree



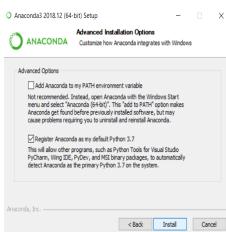
Gambar 1.116 instalasi

6. Klik Next

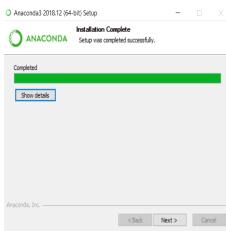


Gambar 1.117 instalasi

7. klik Install

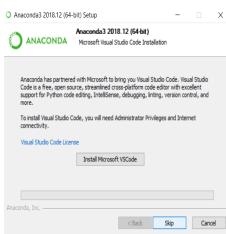


Gambar 1.118 instalasi



Gambar 1.119 instalasi

8. Setelah Completed klik next
9. jika anda sudah mengintall Visual Studio code anda tinggal klik Skip saja.



Gambar 1.120 instalasi

10. Terakhir tinggal finish

1.22.3 Spyder

Setelah install anaconda didalamnya terdapat tools yang bisa di gunakan pada sesi ini menggunakan tool spyder



Gambar 1.121 instalasi

1.23 Muhammad Tomy Nur Maulidy

1.23.1 Sejarah Python

Python merupakan salah satu bahasa pemrograman yang interpretatif multiguna, python memiliki filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python disebut sebagai bahasa yang menggabungkan kemampuan, kapabilitas, dengan sintaksis kode yang jelas. Pemograman python dikembangkan oleh Guido Van Rossum pada tahun 1990 di Stichting Mathematisch Centrum, yang merupakan pengembangan dari Bahasa Pemograman ABC. Pada tahun 1995 Guido pindah ke Virginia amerika sambal terus mengembangkan Bahasa pemograman python. Versi terakhir dari python terakhir pada waktu itu adalah versi 1.6

1.23.2 Instalasi Anaconda

1. Pastikan Bawa Bahasa Pemograman Python 3 telah terinstal.
2. Jika anda belum mempunyai aplikasi anaconda, kalian bisa download sendiri
3. Kemudian buka installer yang telah di download.
4. Klik next

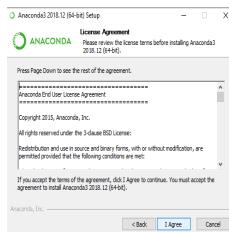


Gambar 1.122 Tampilan Awal

5. Kemudian Klik Next

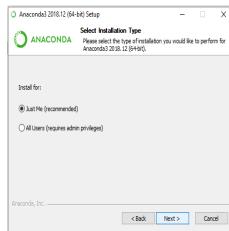


Gambar 1.123 License Agreement



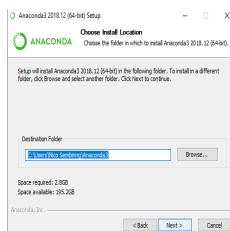
Gambar 1.124 Pemilihan User

6. Kemudian akan muncul persetujuan lisensi, setelah itu anda pilih I Agree
7. Kemudian anda akan disuruh untuk memilih tipe instalasi

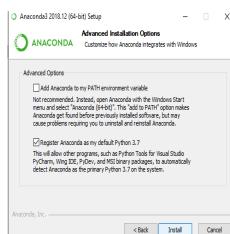


Gambar 1.125 Pemilihan Direktori Penyimpanan

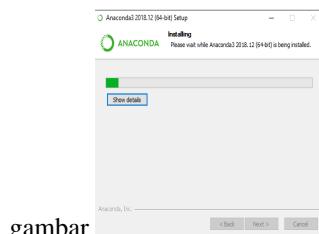
8. Kemudian pilih lokasi untuk penyimpanan instal, setelah itu klik Next
9. Kemudian klik pada kotak register dan pilih instal
10. Kemudian Klik next
11. kemudian jika anda tidak akan menginstal VSCode, maka pilih skip
12. kemudian Proses instalasi telah selesai pilih finish



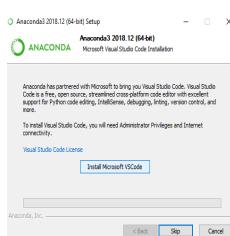
Gambar 1.126 Pemilihan Opsi



Gambar 1.127 Proses Instal



Gambar 1.128 Proses Instal Selesai



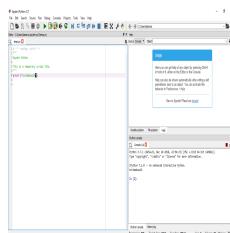
Gambar 1.129 Penawaran Instal MS VSC



Gambar 1.130 Penawaran Instal MS VSC

1.23.3 (

Spyder) Spyder adalah tools yang dipakai untuk python



Gambar 1.131 Instalasi Selesai

1.24 Doli Jonviter NT Simbolon / 1154016

1.24.1 Sejarah Python

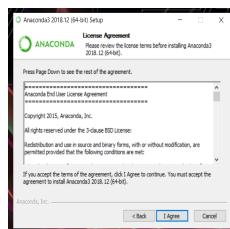
Bahasa Pemograman Python ditemukan oleh Guido van Rossum pertama kali di Scitctting Mathematisch Centrum (CWI) pada awal tahun 1990 an yang terinspirasi oleh bahasa pemograman ABC bahasa pemograman Python bersifat open source sehingga setiap orang dapat mengembangkan bahasa pemograman tersebut

1.24.2 Instalasi Aconda

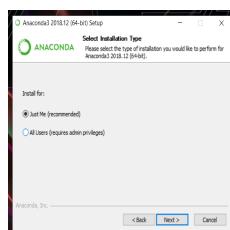
1. Download dan instal bahasa pemograman Python 3
2. Download aplikasi anakonda
3. Install dan update aplikasi anakonda
4. Klik I Agree untuk membuka lisensi
5. Pilih untuk siapa aplikasi tersebut diinstall apakah just me dan juga bisa all users



Gambar 1.132 Tampilan Awal

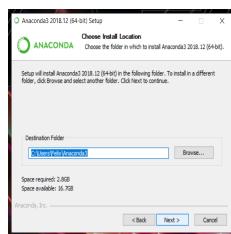


Gambar 1.133 License



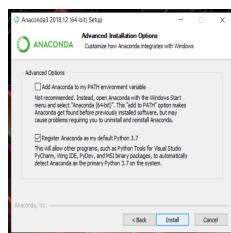
Gambar 1.134 Proses

6. Pilih lokasi instalasi



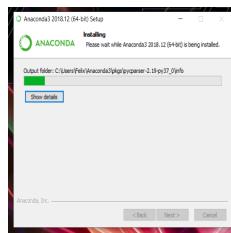
Gambar 1.135 Proses

7. Pilih register anaconda karna add aconda environment tidak remomeded



Gambar 1.136 Proses

8. Tunggu proses instalasi hingga selesai



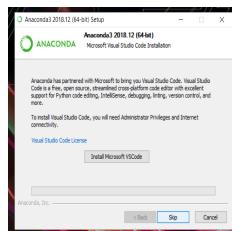
Gambar 1.137 Proses

9. Klik skip

10. Dan anaconda berhasil di install

1.24.3 Menggunakan Spyder

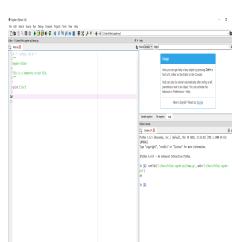
setelah selesai menggunakan instalasi anaconda, maka ada beberapa tool yang digunakan seperti spyder



Gambar 1.138 Proses



Gambar 1.139 Proses



Gambar 1.140 Proses

BAB 2

JUDUL BAGIAN KEDUA

2.1 Harun Ar - Rasyid

2.1.1 Teori

1. sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut di kode Python Variabel merupakan tempat menyimpan data. Dalam phyton kita dapat membuat variable dengan cara sebagai berikut

```
1 #j = "hati"  
2 #k = 1  
3 #print("Tetap" + str(j),k)
```

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

```
1 #1 = input("Masukan NPM :")  
2 #print("NPM Kamu Adalah : "+1)
```

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 perhitungan

```

1 #m = 10
2 #n = 2
3 #o = j + k
4 #p = j * k
5 #q = j - k
6 #r = j / k
7 #s = "10"
8 #print(o,p,q,r)
9 #print(int(s))
10 #print(str(m))

```

4. Tuliskan dan jelaskan sintak untuk perulangan, jenis-jenisnya contoh kode dan cara pakainya di python Untuk Perulangan Pada Python ada For dan While, Untuk Contohnya bisa lihat gambar berikut :

```

1 #t = 12
2 #for u in range(t):
3 #    print("Ini Yang Ke : "+str(u))
4
5 #while(t <= 15):
6 #    print("Yap Ini Betul")
7 #    t = t + 1

```

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 #v = 12
2 #if(v==12):
3 #    print("Dua Belas")

```

Ifelse

```

1 #if(v==13):
2 #    print("Tiga Belas")
3 #else:
4 #    print("Dua Belas")

```

IfNested

```

1 #if(v==11):
2 #    print("Sebelas")
3 #elif(v==12):
4 #    print("Sebelas")
5 #else :
6 #    print("Tiga Belas")

```

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.
- ZeroDivisionError 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 pungasan.
- 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 pengenal 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 #w = 10
2 #x = "10"
3 #try :
4 #     y = w+x
5 #     print(y)
6 #except TypeError :
7 #     print("We Are Different")

```

2.1.2 Ketampilan 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("###  ###  #####  ###  ###  #####  #####")
3 print("###  ###  #####  ###  ###  #####  #####")
4 print("###  ###      ###  ###  ##  ##  ##")
5 print("###  ###      ###  ###  ##  ##  ##")
6 print("###  ###      ###  #####  ##  ##  #####")
7 print("###  ###      ###      ###  ##  ##  #####")
8 print("###  ###      ###      ###  ##  ##  ##")
9 print("###  ###      ###      ###  ##  ##  ##")
10 print("###  ###      ###      ###  #####  #####  ##")
11 print("###  ###      ###      ###  #####  #####  ##")

```

2. Buatlah program hello word dengan input NPM yang disimpan dalam sebuah variabel string bernama NPM dan output sebanyak dua digit belakang NPM.

```

1 #No 2
2 npm = input("Masukan NPM :")
3 hitung = 0
4 while(hitung < 27):
5     print("Halo, " + str(npm) + " Apa kabar?")
6     hitung = hitung +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 dijít tersebut.

```

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

```

5. 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 = 2
8 g = 7
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. Dari soal no 5, Lakukan penjumlahan dari seluruh variabel tersebut.

```

1 #No 6
2 print(h)

```

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

```

1 #No 7
2 print(i)

```

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

```

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

```

1 #No 9
2 print(d,f)

```

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

```

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

```

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

```

1 #No 11
2 print(a,b,c,g)

```

2.1.3 Ketrampilan Penanganan Error

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar  4 11:15:33 2019
4
5 @author: Aegis
6 """
7
8 a = 1
9 b = "2"
10
11 try :
12     c = a + b
13     print(c)
14 except TypeError :
15     print("Perbedaan Tipe Data")

```

2.2 Dwi Yulianingsih

2.2.1 Teori

1. Jenis-jenis variable phyton dan cara pemakaiannya Variabel merupakan tempat menyimpan data. Dalam Phyton terdapat variabel biasa dan variabel array dengan berbagai type data diantaranya adalah variabel dengan type data number, string, dan boolean. Dalam phyton kita dapat membuat variable dengan cara sebagai gambar1 berikut

```

1 #no1
2 #integer
3 x = 1
4 y = 1.1
5 z = 3+5j
6
7 print(type(x))
8 print(type(y))
9 print(type(z))
10
11 #string
12 a = "Hello , World!"
13 print(a[1]) #untuk mencetak huruf terpilih tidak semua
14
15 b = "Hello , World!"
16 print(b[2:5]) #mencetak huruf dari rentang huruf ke 2–5
17
18 a = " Hello , World! "
19 print(a.strip()) #mengembalikan "Hello , World!"
20
21 a = "Hello , World!"
22 print(len(a)) #mencetak berapa jumlah huruf yang ada
23
24 a = "Hello , World!"
25 print(a.lower()) #menjadikan lowercase
26
27 a = "Hello , World!"
28 print(a.upper()) #menjadikan uppercase
29
30 #boolean
31 a = 12
32 b = 10
33 if (a < b):
34     print("Isi variabel a lebih kecil daripada variabel b")
35 elif (a > b):
36     print("Isi variabel a lebih besar daripada variabel b")
37 else:
38     print("Isi variabel a sama dengan variabel b")

```

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

```

1 #no2
2 nama = input("Masukan namamu :")
3 print("Halo , " + str(nama))

```

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

```

1 #no3
2 x = 5
3 y = 3
4 print(x + y) #penambahan
5
6 x = 5

```

```

7 y = 3
8
9 print(x - y) #mengurangan
10
11 x = 5
12 y = 3
13
14 print(x * y) #perkalian
15
16 x = 12
17 y = 3
18
19 print(x / y) #pembagian
20
21 x = 5
22 y = 2
23
24 print(x % y) #modulus

```

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

```

1 #no4
2 #perulangan while
3 i = 1
4 while i < 6:
5     print(i)
6     i += 1
7
8 #perulangan for
9 fruits = ["apple", "banana", "cherry"]
10 for x in fruits:
11     print(x)

```

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

```

1 #no5
2 #If statement
3 a = 33
4 b = 200
5 if b > a:
6     print("b is greater than a")
7
8 #elif statement
9 a = 33
10 b = 33
11 if b > a:
12     print("b is greater than a")
13 elif a == b:
14     print("a and b are equal")
15
16 #else statement

```

```

17 a = 200
18 b = 33
19 if b > a:
20     print("b is greater than a")
21 elif a == b:
22     print("a and b are equal")
23 else:
24     print("a is greater than b")

```

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

```

1 #no7
2 try :
3     print(x)
4 except :
5     print("An exception occurred")

```

2.2.2 praktek

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("Masukan NPM :")
3 hitung = 0
4 while(hitung < 9):
5     print("Halo, " + str(npm) + " Apa kabar?")
6     hitung = hitung +1

```

3. Jawaban soal no 3

```

1 #No 3
2 npm = input("Masukan 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("Masukan 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 = 0
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 #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,e,f)
```

10. Jawaban soal no 10

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

11. Jawaban soal no 11

```
1 #No 11
2 print(a,b,c,g)
```

2.2.3 Keterampilan dan penanganan eror

```
1 a = 1
2 b = "2"
3
4 try :
5     c = a + b
6     print(c)
7 except TypeError :
8     print("type datanya beda tau")
```

2.3 Kadek Diva Krishna Murti

2.3.1 Teori

1. Jenis-jenis variabel dan cara pemakaianya di Python.

- (a) Boolean

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

Listing 2.1 Contoh kode penggunaan Boolean.

- (b) String

```
1 #String
2 var = "Kadek Diva Krishna Murti"
3 print(var)
```

Listing 2.2 Contoh kode penggunaan String.

- (c) Integer

```
1 #Integer
2 var = 1174006
3 print(var)
```

Listing 2.3 Contoh kode penggunaan Integer.

(d) Float

```

1 #Float
2 var = 5.5
3 print(var)

```

Listing 2.4 Contoh kode penggunaan Float.

(e) Hexadecimal

```

1 #Hexadecimal
2 var = 0x5
3 print(var)

```

Listing 2.5 Contoh kode penggunaan Hexadecimal.

(f) Complex

```

1 #Complex
2 var = 5j
3 print(var)

```

Listing 2.6 Contoh kode penggunaan Complex.

(g) List

```

1 #List
2 var = [1,2,3,4,5]
3 print(var)
4 print(var[0])

```

Listing 2.7 Contoh kode penggunaan List.

(h) Tuple

```

1 #Tuple
2 var = (1,2,3,4,5)
3 print(var)
4 print(var[0])

```

Listing 2.8 Contoh kode penggunaan Tuple.

(i) Set

```

1 #Set
2 var = {1,2,3,4,5}
3 print(var)

```

Listing 2.9 Contoh kode penggunaan Set.

(j) Dictionary

```

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

```

Listing 2.10 Contoh kode penggunaan Dictionary.

2. Kode untuk meminta input dan melakukan output di Python.

```

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

```

Listing 2.11 Contoh kode input dan output.

3. Operator dasar aritmatika dan mengubah tipe data di Python.

Operator dasar aritmatika

- (a) Pertambahan Operator ini dipergunakan untuk melakukan operasi pertambahan.

```

1 #Pertambahan
2 angka1 = 2
3 angka2 = 3
4 hasil = angka1+angka2
5 print(hasil)

```

Listing 2.12 Contoh kode operasi pertambahan.

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

```

1 #Pengurangan
2 angka1 = 5
3 angka2 = 3
4 hasil = angka1-angka2
5 print(hasil)

```

Listing 2.13 Contoh kode operasi pengurangan.

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

```

1 #Perkalian
2 angka1 = 2
3 angka2 = 3
4 hasil = angka1*angka2
5 print(hasil)

```

Listing 2.14 Contoh kode operasi perkalian.

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

```

1 #Pembagian
2 angka1 = 6
3 angka2 = 3
4 hasil = angka1/angka2
5 print(hasil)

```

Listing 2.15 Contoh kode operasi pembagian.

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

```

1 #Modulus
2 angka1 = 7
3 angka2 = 3
4 hasil = angka1%angka2
5 print(hasil)

```

Listing 2.16 Contoh kode operasi modulus.

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

```

1 #Perpangkatan
2 angka1 = 6
3 angka2 = 2
4 hasil = angka1**angka2
5 print(hasil)

```

Listing 2.17 Contoh kode operasi perpangkatan.

- (g) Pembulatan Hasil Bagi Operator ini dipergunakan untuk melakukan operasi pembulatan hasil bagi.

```

1 #Pembulatan Pembagian
2 angka1 = 7
3 angka2 = 3
4 hasil = angka1 // angka2
5 print(hasil)

```

Listing 2.18 Contoh kode operasi pembulatan hasil bagi.

Mengubah tipe data

- (a) String ke Integer

```

1 #String ke Integer
2 var_str = '5'
3 var_int = int(var_str)
4 print(var_int)

```

Listing 2.19 Contoh kode konversi string ke integer.

- (b) Integer ke String

```

1 var_int = 5
2 var_str = str(var_int)
3 print(var_str)

```

Listing 2.20 Contoh kode konversi integer ke string.

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

- (a) While Loop While Loop adalah perulangan yang mengeksekusi statement berkali-kali selama kondisi bernilai benar atau 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 perulangan yang mengulangi item dari urutan apapun, seperti list atau string.

```
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 perulangan yang berada di perulangan atau biasa disebut dengan perulangan bersarang.

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

Listing 2.23 Contoh kode penggunaan nested loop.

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

- (a) If Kondisi ini dipergunakan jika penkondisiannya hanya satu.

```
1 #If
2 angka = 5
3 if angka > 0:
4     print(angka, "adalah bilangan positif.")
```

Listing 2.24 Contoh kode penggunaan if.

- (b) If Else Kondisi ini dipergunakan jika pengkondisiannya ada dua.

```
1 #If Else
2 angka = -5
3 if angka > 0:
4     print(angka, "adalah bilangan positif.")
5 else:
6     print(angka, "adalah bilangan negatif.)")
```

Listing 2.25 Contoh kode penggunaan if else.

- (c) Elif Kondisi ini dipergunakan jika pengkondisiannya lebih dari dua.

```
1 #Elif
2 angka = 0
3 if angka > 0:
4     print(angka, "adalah bilangan positif.")
```

```

5 elif angka < 0:
6     print(angka, "adalah bilangan negatif.")
7 else:
8     print(angka, "adalah bilangan nol.")

```

Listing 2.26 Contoh kode penggunaan elif.

- (d) Kondisi di dalam kondisi Kondisi ini dipergunakan jika pengkondisianya memerlukan pengkondisian di dalamnya.

```

1 #Kondisi di dalam kondisi
2 if angka > 0:
3     print(angka, "adalah bilangan positif.")
4     if angka > 50:
5         print(angka, "adalah bilangan lebih dari 50.")
6     else:
7         print(angka, "adalah bilangan kurang dari 50.")
8 elif angka < 0:
9     print(angka, "adalah bilangan negatif.")
10 else:
11     print(angka, "adalah bilangan nol.")

```

Listing 2.27 Contoh kode penggunaan kondisi di dalam kondisi.

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 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 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.

7. Cara pemakaian Try Except di Python. Berikut ini adalah contoh penggunaan try except.

```

1 #Try Except
2 try:
3     print(te)
4 except:
5     print("Terjadi kesalahan penulisan kode")
6 finally:

```

```
7     print("Try except telah selesai")
```

Listing 2.28 Contoh kode penggunaan try except.

2.3.2 Ketampilan Pemrograman

1. Jawaban Soal 1

```
1 #Jawaban No. 1
2
3 print(1174006%3)
4 #mod3 = 1
5
6 print(" ##   ## ##### #####   ###   #####   #####   ##### #####")
7 print("### ## ## ##### #####   ###   ## ## ##### #####")
8 print(" ##   ## ## ## ## ## ##   ## ## ## ## ##")
9 print(" ##   ## ## ##### #####   ##   ## ## ## ## ##### #####")
10 print(" ##   ## ## ## ## ## ##   ## ## ## ## ## ## ## ## ##")
11 print(" ##   ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##")
```

2. Jawaban Soal 2

```
1 #Jawaban No. 2
2
3 npm = input("Input : ")
4 ulang = 1
5 while(ulang <= 6):
6     print("Halo, "+str(npm)+" apa kabar?")
7     ulang += 1
```

3. Jawaban Soal 3

```
1 #Jawaban No. 3
2
3 npm = input("Input : ")
4 ulang = 1
5 while(ulang <= 6):
6     print("Halo, "+str(npm[4:7])+" apa kabar?")
7     ulang += 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 = 0
9 g = 6
10
11 npm = [a,b,c,d,e,f,g]
12
13 for n in npm:
14     print(n, end ='''')

```

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 ='''')

```

10. Jawaban Soal 10

```

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

```

11. Jawaban Soal 11

```

1 #Jawaban No. 11
2
3 print(c)

```

2.3.3 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 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 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.

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")
```

2.4 Dezhia Aidil Martha

2.4.1 Teori

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

```
1 c = 7
2 d = "Dezha Martha"
3
4 print(c)
5 print(d)
```

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

```
1 npm = input()
2 print(npm)
```

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

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

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

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

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

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

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

```

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

```

```

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

```

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

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

```

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

```

2.4.2 praktek

1. Jawaban soal no 1

```

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

```

```

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

```

2. Jawaban soal no 2

```

1 npm = input("Masukkan NPM :")
2 hitung = 0
3 while(hitung <= 25):
4     print("Hallo, " + str(npm) + " Apa kabar?")
5     hitung = hitung + 1

```

3. Jawaban soal no 3

```

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

```

4. Jawaban soal no 4

```

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

```

5. Jawaban soal no 5

```

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

```

6. Jawaban soal no 6

```

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

```

7. Jawaban soal no 7

```

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

```

8. Jawaban soal no 8

```

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

```

9. Jawaban soal no 9

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

10. Jawaban soal no 10

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

11. Jawaban soal no 11

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

2.4.3 Keterampilan dan penanganan eror

```

1 c = 1
2 d = "4"
3
4 try:
5     a = c - d
6     print(a)
7 except:
8     print("Ubah error nya")

```

2.5 Evietania Charis Sujadi

2.5.1 Teori

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

```

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

```

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

```
1 npm = input()
2 print(npm)
```

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

```

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

```

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

```

1 ulang = ["Motor", "Mobil", "Pesawat"]
2 for i in range(ulang):
3     print(i)

```

```

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

```

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

```

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

```

```

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

```

```

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

```

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

```

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

```

2.5.2 praktek

1. Jawaban soal no 1

```

1 print("###    ###    ###    ###    ###    ##    ##    ##    ##")
2 print("###    ###    ###    #####    ##    ##    #####    #####")
3 print("###    ###    ###    ###    ##    ##    ##    #####    #####")
4 print("###    ###    ###    ###    ##    ##    ##    ##    ##")
5 print("###    ###    ###    ###    ##    ##    ##    ##    ##")
6 print("###    ###    ###    ###    #####    #####    #####    #####")
7 print("###    ###    ###    ###    #####    #####    #####    #####")
8
9 #Nomor 2
10 npm = input("Masukkan NPM :")

```

2. Jawaban soal no 2

```

1 hitung = hitung +1
2

```

```

3 #Nomor 3
4 npm = input("Masukkan NPM : ")
5 hitung = 0

```

3. Jawaban soal no 3

```

1 #Nomor 4
2 npm = input("Masukkan NPM : ")
3 print("Hello , " + str(npm[-3]) + " Apa kabar?")
4
5 #Nomor 5

```

4. Jawaban soal no 4

```

1 c = 7
2 d = 4

```

5. Jawaban soal no 5

```

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

```

6. Jawaban soal no 6

```

1 print(c)
2 print(d)

```

7. Jawaban soal no 7

```

1 print(g)

```

8. Jawaban soal no 8

```

1
2 #No 10
3 print(a,b,c,g)
4
5 #No 11

```

9. Jawaban soal no 9

```

1 #j = "hati"

```

10. Jawaban soal no 10

11. Jawaban soal no 11

2.5.3 Keterampilan dan penanganan eror

```
1
2 try :
3     c = a + b
4     print(c)
5 except TypeError :
6     print("type datanya beda tau")
```

2.6 Perintah Navigasi

Perintah navigasi direktori

2.7 Damara Benedikta

2.7.1 Teori

1. jenis-jenis variable phyton dan cara pemakaiannya Variable merupakan tempat untuk menyimpan data, Isi dari variabel itu dapat berubah atau mutable sesuai dengan operasi yang diinginkan. Saat program dieksekusi maka variabellah yang bertugas menyimpan data. Dimana didalam phyton terdapat beberapa variable diantaranya number, boolean,string. Dalam membuat variabel Python caranya adalah sebagai berikut

```
1 #no1
2 #integer
3 x = 1
4 y = 1.2
5 z = 2+5j
6
7 print(type(x))
8 print(type(y))
9 print(type(z))
10
11 #string
12 a = "Hello , World!"
13 print(a[1]) #untuk mencetak huruf terpilih tidak semua
14
15 b = "Hello , World!"
16 print(b[2:5]) #mencetak huruf dari rentang huruf ke 2-5
17
18 a = " Hello , World! "
19 print(a.strip()) #mengembalikan "Hello , World!"
```

```

21 a = "Hello , World!"
22 print(len(a)) #mencetak berapa jumlah huruf yang ada
23
24 a = "Hello , World!"
25 print(a.lower()) #menjadikan lowercase
26
27 a = "Hello , World!"
28 print(a.upper()) #menjadikan uppercase
29
30 #boolean
31 a = 12
32 b = 10
33 if (a < b):
34     print("Isi variabel a lebih kecil daripada variabel b")
35 elif (a > b):
36     print("Isi variabel a lebih besar daripada variabel b")
37 else:
38     print("Isi variabel a sama dengan variabel b")

```

2. operator dasar aritmatika dimana terdapat penjumlahan,pengurangan,pembagian,perkalian nominal

```

1 #no2
2 nama = input("Masukan namamu :")
3 print("Halo , " + str(nama))
4
5 #no3
6 x = 4
7 y = 2
8 print(x + y) #penambahan
9
10 x = 4
11 y = 2
12
13 print(x - y) #mengurangan
14
15 x = 4
16 y = 2
17
18 print(x * y) #perkalian
19
20 x = 5
21 y = 3
22
23 print(x / y) #pembagian
24
25 x = 6
26 y = 2
27
28 print(x % y) #modulus

```

3. Perulangan dalam phyton terdapat perulangan while dan for

```

1 #no4
2 #perulangan while
3 i = 1

```

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

4. Dimana terdapat sintak untuk meilih kondisi didalam kondisi Untuk memilih keputusan menggunakan (kondisi if) dimana digunakan untuk mengantisipasi kondisi yang terjadi saat jalannya suatu program dan menentukan tindakan apa yang akan dilakukan sesuai dengan kondisi.

```
1 #no5  
2 #If statement  
3 a = 33  
4 b = 200  
5 if b > a:  
6     print("b is greater than a")  
7  
8 #elif statement  
9 a = 33  
10 b = 33  
11 if b > a:  
12     print("b is greater than a")  
13 elif a == b:  
14     print("a and b are equal")  
15  
16 #else statement  
17 a = 200  
18 b = 33  
19 if b > a:  
20     print("b is greater than a")  
21 elif a == b:  
22     print("a and b are equal")  
23 else:  
24     print("a is greater than b")
```

5. Jenis-jenis sintak error pada phyton Syntax errors Jika dalam program terdapat kesalahan sintaks maka proses akan berhenti dan menampilkan pesan kesalahan. Runtime errors, disebut begitu karenakesalahan 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.

Type eror merupakan eror yang terjadi saat dilakukan eksekusi pada suatu operasi dengan type object yang tidak sesuai. ZeroDivision eror merupakan eror yang terjadi saat eksekusi program menghasilkan perhitungan matematika dengan angka 0

6. Try except cara memakai try except adalah sebagai berikut

```

1 #no1
2 #integer
3 x = 1
4 y = 1.2
5 z = 2+5j
6
7 print(type(x))
8 print(type(y))
9 print(type(z))

```

2.7.2 praktek

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("###   ###       ###           #####   ###   ###   ###"
      "###")

```

2. Jawaban soal no 2

```

1 #No.2
2 npm = input("Masukan NPM :")
3 hitung = 0
4 while(hitung < 12):
5     print("Hallo, " + str(npm) + " Apa Kabar?")
6     hitung = hitung + 1

```

3. Jawaban soal no 3

```

1 #No.3
2 npm = input("Masukan NPM :")
3 hitung = 0
4 while(hitung < 3):
5     print("Hallo, " + str(npm[4:7]) + " Apa Kabar?")
6     hitung = hitung + 1

```

4. Jawaban soal no 4

```
1 #No.4
2 npm = input("Masukan 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 = 1
8 g = 2
9 h = a+b+c+d+e+f+g
10 i = a*b*c*d*e*f*g
11 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,g)
```

2.7.3 Keterangan dan Penanganan eror

```

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

```

2.8 Felix Setiawan Lase

2.8.1 Teori

1. sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut di kode Python

Variabel merupakan tempat menyimpan data. Dalam phyton kita dapat membuat variable dengan cara sebagai berikut

```

1 j = "hati"
2 k = 1
3 print("Tetap" +str(j),k)

```

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

```

1 l = input("Masukan NPM :")
2 print("NPM Kamu Adalah : "+l)

```

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 perhitungan

```

1 m = 10
2 n = 2
3 o = j + k
4 p = j * k
5 q = j - k
6 r = j / k
7 s = "10"
8 print(o,p,q,r)
9 print(int(s))
10 print(str(m))

```

4. Tuliskan dan jelaskan sintak untuk perulangan, jenis-jenisnya contoh kode dan cara pakainya di python Untuk Perulangan Pada Python ada For dan While, Untuk Contohnya bisa lihat gambar berikut :

```

1 t = 12
2 for u in range(t):
3     print("Ini Yang Ke : "+str(u))
4
5 while(t <= 15):
6     print("Yap Ini Betul")
7     t = t + 1

```

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 v = 12
2 if(v==12):
3     print("Dua Belas")

```

Ifelse

```

1 if(v==13):
2     print("Tiga Belas")
3 else:
4     print("Dua Belas")

```

IfNested

```

1 if(v==11):
2     print("Sebelas")
3 elif(v==12):
4     print("Sebelas")
5 else :
6     print("Tiga Belas")

```

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.
- ZeroDivisionError 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 penu-gasan.
 - **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`.

7. Tuliskan dan jelaskan cara memakai Try Except.

```
1 w = 10
2 x = "10"
3 try:
4     y = w+x
5     print(y)
6 except TypeError:
7     print("We Are Different")
```

2.8.2 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(1174026%3)
3 print("***** *** ***** * * * * * * * * * * * * * * * *")
4 print("***** *** ***** * * * * * * * * * * * * * * * *")
5 print("***** *** * * * * * * * * * * * * * * * * * * * *")
6 print("***** *** * * * * * * * * * * * * * * * * * * * *")
7 print("***** *** * * * * * * * * * * * * * * * * * * * *")
8 print("***** *** * * * * * * * * * * * * * * * * * * * *")
9 print("***** *** * * * * * * * * * * * * * * * * * * * *")
10 print("***** *** * * * * * * * * * * * * * * * * * * * *")
11 print("***** *** * * * * * * * * * * * * * * * * * * * *")
```

2. Buatlah program hello word dengan input NPM yang disimpan dalam sebuah variabel string bernama NPM dan output sebanyak dua digit belakang NPM.

```
1 #No.2  
2 npm = input("Masukan NPM :")
```

```

3 hitung = 0
4 while(hitung < 26):
5     print("Halo, " + str(npm) + " Apa Kabar?")
6     hitung = hitung + 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 npm = input("Masukan NPM :")
3 hitung = 0
4 while(hitung < 8):
5     print("Halo, " + str(npm[4:7]) + " Apa Kabar?")
6     hitung = hitung + 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("Halo, " + str(npm[-3]) + " Apa Kabar?")

```

5. 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 = 2
8 g = 6
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. Dari soal no 5, Lakukan penjumlahan dari seluruh variabel tersebut.

```

1 #No.6
2 print(h)

```

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

```

1 #No.7
2 print(i)

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

2.8.3 Ketrampilan Penanganan Error

```

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

```

2.9 Perintah Navigasi

Perintah navigasi direktori

2.10 Muhammad Dzihan Al-Banna

2.10.1 Pemrograman Dasar

2.10.2 Variabel

Variabel merupakan sebuah ruang kosong yang digunakan untuk menyimpan suatu nilai atau data. Pada saat membuat sebuah variabel berarti sedang melakukan pemesanan pada sebuah ruang kosong di memori. Isi dari variabel itu dapat berubah

atau mutable sesuai dengan operasi yang diinginkan. Saat program dieksekusi maka variabel yang akan bertugas menyimpan data. Variabel dapat menyimpan berbagai macam tipe data. Dalam pemrograman Python, variabel mempunyai sifat dinamis, yang berarti variabel Python tidak perlu ditentukan tipe data tertentu dan variabel pada Python dapat diubah saat program dieksekusi. Cara penulisan python :

1. Karakter pertama harus berupa huruf atau garis bawah
2. Karakter selanjutnya dapat berupa huruf, garis bawah atau angka
3. Karakter pada nama variabel bersifat sensitif (case-sensitif). Artinya penggunaan urut besar dan huruf kecil sangat berpengaruh. Sebagai contoh, variabel Mahasiswa dan mahasiswa adalah variabel yang berbeda.

```

1 #1 variabel
2 x = 5
3 y = "John"
4 print(x)
5 print(y)
6
7 x = 4
8 x = "Sally"
9 print(x)
```

2.10.3 input dan output

Input adalah masukan yang anda berikan kepada sebuah program dan system akan memproses lalu menampilkan ouputnya. Di dalam python sudah terdapat fungsi input dan rawinput. input digunakan untuk masukan bernilai angka sedangkan raw input untuk masukan bernilai teks. Beikut adalah cara untuk meminta input dan outputnya.

```

1 #2 input output
2 npm = input("Masukan NPM :")
3 hitung = 0
4 while(hitung < 95):
5     print("Halo, " + str(npm) + " Apa Kabar?")
6     hitung = hitung +1
7
8 #No.1
9 npm = input("Masukan NPM :")
10 hitung = 0
11 while(hitung < 14):
12     print("Halo, " + str(npm[4:7]) + " Apa Kabar?")
13     hitung = hitung +1
14
15 #No.2
16 npm = input("Masukan NPM :")
17 print("Halo, " + str(npm[-3]) + " Apa Kabar?")
```

2.10.4 operator aritmatika

4. Penjumlahan

```

1 #1 penjumlahan
2 x = 5
3 y = 3

```

5. Pengurangan

```

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

```

6. Perkalian

```

1 print(x * y)
2 #4 pembagian
3 x = 12
4 y = 3

```

7. Mengubah String ke Integer

```

1 #5 perubahan data ke string
2 kamu = 'cantik'
3 kamu2 = 1
4 kamuitu = kamu + str(kamu2)
5 print (kamuitu)

```

8. Mengubah Integer ke String

```

1 #6 string ke int
2 a = 5
3 b = str(a)
4 print(b)

```

2.10.5 Perulangan

Secara umum, perintah pada bahasa pemrograman Python akan dijalankan secara berurutan. Pernyataan pertama dalam sebuah fungsi dijalankan pertama, lalu diikuti oleh yang kedua, dan seterusnya. Tetapi akan ada situasi yang mengharuskan menulis banyak kode, dimana kode tersebut sangat banyak.

```

1 #4 sintak perulangan
2 i = 1
3 while i < 6:
4     print(i)
5     if i == 3:
6         break
7     i += 1

```

2.10.6 If Else

Pada saat pengambilan keputusan (kondisi if) digunakan untuk mengantisipasi kondisi yang akan terjadi saat program dijalankan. Dan akan menentukan tindakan apa yang akan diambil sesuai dengan kondisi program pada saat dieksekusi. Pada python ada beberapa kondisi diantaranya adalah if, else, dan elif. Kondisi if hanya bisa digunakan pada saat kondisi benar saja.

```

1 #5 kondisi
2 a = 33
3 b = 200
4 if b > a:
5     print("b is greater than a")

```

2.10.7 Penanganan Error

Biasanya yang paling mudah dikenali, kesalahan sintaksis terjadi ketika Anda membuat kesalahan ketik. Tidak mengakhiri pernyataan if dengan titik dua adalah contoh kesalahan sintak, seperti salah mengeja kata kunci Python (mis. Menggunakan whille alih-alih sementara). Kesalahan sintak biasanya muncul pada waktu kompilasi dan dilaporkan oleh interpreter. Berikut ini contoh kesalahan sintaks:

```

1 x = int(input('Enter a number: '))
2
3 whille x%2 == 0:
4     print('You have entered an even number.')
5 else:
6     print ('You have entered an odd number.')
7 #penanganan error

```

2.10.8 Try Exception

Python memiliki banyak exceptions bawaan yang memaksa program untuk menghasilkan kesalahan ketika ada sesuatu yang salah di dalamnya. Ketika exceptions ini terjadi, itu menyebabkan proses saat ini berhenti dan meneruskannya ke proses panggilan sampai ditangani. Jika tidak ditangani, program ini akan macet. Misalnya, jika fungsi A memanggil fungsi B yang pada gilirannya memanggil fungsi C dan pengecualian terjadi di fungsi C. Jika tidak ditangani dalam C, pengecualian beralih ke B dan kemudian ke A. Jika tidak pernah ditangani, pesan kesalahan dilemparkan dan program ini terhenti secara tiba-tiba.

```

1 x = int(input('Enter a number: '))
2
3 while x%2 == 0:
4     print('You have entered an even number.')
5 else:
6     print ('You have entered an odd number.')

```

```

7 #7 try except
8 a = 2
9 b = "4"

```

2.10.9 Keterampilan Penanganan Error

```

1 print(1174095%3)
2 print("**** *** **** * ***** * *** * *** * ***** * ****")
3 print("**** *** * ***** * *** * *** * ***** * ****")
4 print("**** *** *** * *** * *** * *** * *** * *** * ***")
5 print("**** *** *** * *** * *** * *** * *** * *** * ***")
6 print("**** *** *** * *** * *** * *** * *** * *** * ***")
7 print("**** *** *** * *** * ***** * *** * *** * *****")
8 print("**** *** *** * *** * *** * *** * *** * ***")
9 print("**** *** *** * *** * *** * *** * *** * ***")
10 print("**** *** *** * *** * ***** * ****")
11 print("**** *** *** * *** * ***** * ****")
12
13 #No.2
14 npm = input("Masukan NPM :")
15 hitung = 0
16 while(hitung < 95):
17     print("Halo, " + str(npm) + " Apa Kabar?")
18     hitung = hitung + 1
19
20 #No.3
21 npm = input("Masukan NPM :")
22 hitung = 0
23 while(hitung < 14):
24     print("Halo, " + str(npm[4:7]) + " Apa Kabar?")
25     hitung = hitung + 1
26
27 #No.4
28 npm = input("Masukan NPM :")
29 print("Halo, " + str(npm[-3]) + " Apa Kabar?")
30
31 #No.5
32 a = 1
33 b = 1
34 c = 7
35 d = 4
36 e = 0
37 f = 9
38 g = 5
39 h = a+b+c+d+e+f+g
40 i = a*b*c*d*e*f*g

```

```

41
42 print(a,b,c,d,e,f,g)
43
44 #No.6
45 print(h)
46
47 #No.7
48 print(i)
49
50 #No.8
51 print(a)
52 print(b)
53 print(c)
54 print(d)
55 print(e)
56 print(f)
57 print(g)
58
59 #No.9
60 print(d)
61
62 #No.10
63 print(a,b,c,f,g)
64
65 #No.11
66 print(c,g)

```

2.11 Nico Ekklesia Sembiring

2.11.1 Tugas Teori

1. Variabel Number. Python memiliki berbagai jenis variabel yang merupakan lokasi untuk memori yang dipersiapkan untuk menyimpan nilai. Jenis-jenis variabel yang terdapat dalam python adalah sebagai berikut :

- Variabel Number. Variabel Number terdiri dari Integer, float, dan complex. Cara pemakaian variabel tersebut adalah dengan cara menuliskan perintah print. dapat dilihat sebagai berikut.

```

1 x = 5
2 print(x, "tipenya adalah ", type(x))
3 x = 2.0
4 print(x, "tipenya adalah ", type(x))
5 x = 1+2j
6 print(x, "tipenya adalah ", type(x))

```

- Variabel String. Variabel String merupakan rangkaian dari karakter yang berada diantara tanda kutip (""). String terdiri dari huruf dan angka yang digabung hingga menjadi bentuk teks. Cara Penggunaan variabel string adalah sebagai berikut.

```

1 kalimat = "Nama saya Nico"
2
3 print(kalimat)      # menampilkan string lengkap

```

```

4 print(kalimat[0]) # menampilkan karakter pertama
5 print(kalimat[-1]) # menampilkan karakter terakhir
6 print(kalimat[4:7]) # menampilkan dari indeks 4 – 6
7 print(kalimat[:4]) # menampilkan dari indeks 0 – 3

```

- Variabel List Variabel list merupakan variabel yang memiliki item yang berurutan. Cara Pemakaian variabel list yaitu dengan memanggil nama list yang diikuti dengan indeks dari item yang dipanggil. Dapat dilihat pada gambar berikut

```

1 a = [5,10,15,20,25,30,35,40]
2
3 # a[2] = 15
4 print("a[2] = ", a[2])
5
6 # a[0:3] = [5, 10, 15]
7 print("a[0:3] = ", a[0:3])
8
9 # a[5:] = [30, 35, 40]
10 print("a[5:] = ", a[5:])

```

- Variabel Tuple Variabel Tupel mirip dengan list, namun bersifat immutable(anggotanya tidak dapat diubah). Cara pemakaiannya yaitu dengan menggunakan tanda kurung dan anggotanya dipisah oleh tanda komma. dapat dilihat pada gambar berikut.

```

1 white = (255,255, 255)
2 red = (255,0,0)
3 print(white)
4 print(red[0])
5 print(red[1])

```

- Variabel Set Variabel Set merupakan variabel yang tidak terurut. biasa digunakan untuk melakukan operasi himpunan matematika. Cara penggunaannya dapat dilihat dari gambar berikut

```

1 my_set = {1,2,3}
2 print(my_set)
3
4 # set dengan menggunakan fungsi set()
5 my_set = set([1,2,3])
6 print(my_set)
7
8 # set data campuran
9 my_set = {1, 2.0, "Python", (3,4,5)}
10 print(my_set)
11
12 # bila kita mengisi duplikasi, set akan menghilangkan salah satu
13 # output: {1,2,3}
14 my_set = {1,2,2,3,3,3}
15 print(my_set)

```

2. Kode meminta Input dari User dan Output ke Layar Contoh kode yang dapat digunakan untuk meminta input dari User

```

1 npm = input("Masukan NPM :")
2 print(npm)

```

3. Operator Dasar Aritmatika

- Penjumlahan(+).

```

1 a = 2
2 b = 3
3 print(a+b)

```

- Pengurangan(-).

```

1 a = 2
2 b = 3
3 print(a-b)

```

- Perkalian(*) .

```

1 a = 2
2 b = 3
3 print(a*b)

```

- Pembagian(/).

```
1 a = 2
```

- Mengubah integer ke string.

```

1 npm = 1174096
2 print("Halo, " + str(npm) + " Apa Kabar?")

```

- Mengubah string ke integer.

```

1 npm = "5"
2 print(int(npm))

```

4. Sintaks Untuk Perulangan

- Pengulangan While

```

1 i = 1
2 while i < 9:
3     print(i)
4     i += 1

```

- Pengulangan For

```

1 kue = ["Bolu", "Bakpao", "Apem"]
2 for x in kue:
3     print(x)

```

5. Cara Memakai sintaks kondisi

```
1 x = 100
2 y = 20
3 if x > y:
4     print("x lebih besar dari y")
5 else:
6     print("x tidak lebih besar dari y")
```

6. Macam Macam error

- SystemError Muncul saat penafsir menemukan masalah internal, namun bila kesalahan ini ditemui, bahasa Python tidak keluar.
- SystemExit Muncul saat bahasa Python berhenti dengan menggunakan fungsi sys.exit (). Jika tidak ditangani dalam kode, menyebabkan penafsir untuk keluar.
- TypeError Muncul saat operasi atau fungsi dicoba yang tidak valid untuk tipe data yang ditentukan.
- ValueError Muncul ketika fungsi bawaan untuk tipe data memiliki jenis argumen yang valid, namun argumen tersebut memiliki nilai yang tidak valid yang ditentukan.
- RuntimeError Muncul saat kesalahan yang dihasilkan tidak termasuk dalam kategori apa pun.
- NotImplementedError Muncul ketika metode abstrak yang perlu diimplementasikan di kelas warisan sebenarnya tidak dilaksanakan.
- NameError Muncul saat pengenal tidak ditemukan di namespace lokal atau global.
- UnboundLocalError Muncul saat mencoba mengakses variabel lokal dalam suatu fungsi atau metode namun tidak ada nilai yang ditugaskan padanya.
- EnvironmentError Muncul dasar untuk semua pengecualian yang terjadi di luar lingkungan Python.
- IOError Muncul saat operasi input / output gagal, seperti pernyataan cetak atau fungsi open () saat mencoba membuka file yang tidak ada.
- OSError Muncul untuk kesalahan terkait sistem operasi.
- SyntaxError Muncul saat ada kesalahan dengan sintaks Python.
- IndentationError Muncul saat indentasi tidak ditentukan dengan benar.

7. Cara Memakai Try dan Except

```
1 x = 17
2 y = "19"
3 try :
4     z = x+y
5     print(z)
6 except TypeError:
7     print("Berbeda tipe datanya")
```

2.11.2 Tugas Keterampilan Pemrograman

- 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 print("###  ###  #####  ###  ###  #####  ######")
2 print("###  ###  #####  ###  ###  #####  ######")
3 print("###  ###      ###  ###  ###  ###  ###  ###  ###")
4 print("###  ###      ###  ###  ###  ###  ###  ###  ###")
5 print("###  ###      ###  #####  ###  ###  #####  ######")
6 print("###  ###      ###          ###  ###  ###  ###  ###")
7 print("###  ###      ###          ###  ###  ###  ###  ###")
8 print("###  ###      ###          ###  #####  ######  ######")
9 print("###  ###      ###          ###  #####  ######  ######")

```

- Buatlah program hello word dengan input NPM yang disimpan dalam sebuah variabel string bernama NPM dan output sebanyak dua digit belakang NPM.

```

1 npm = input("Masukan NPM :")
2 hitung = 0
3 while(hitung < 96):
4     print("Halo, " + str(npm) + " Apa Kabar?")
5     hitung = hitung + 1

```

- 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 npm = input("Masukan NPM :")
2 hitung = 0
3 while(hitung < 15):
4     print("Halo, " + str(npm[4:7]) + " Apa Kabar?")
5     hitung = hitung + 1

```

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

```

- Buat program dengan mengisi variabel alfabet dengan nomor npm satu persatu berurut.

```

1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 9
7 g = 6
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. Dari soal no 5, Lakukan penjumlahan dari seluruh variabel tersebut.

```
1 print(h)
```

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

```
1 print(i)
```

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

```

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

```

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

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

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

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

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

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

2.11.3 Ketampilan Penanganan Error

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar  4 11:39:52 2019
4
5 @author: Nico Sembiring
6 """

```

```

7 a = 2
8 b = "4"
9
10 try :
11     c = a + b
12     print(c)
13 except TypeError :
14     print ("Beda Tipe Data Cuk")

```

2.12 Muhammad Tomy Nur Maulidy

2.12.1 Teori

1. sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut di kode Python Variabel merupakan tempat menyimpan data, sedangkan tipe data adalah jenis data yang terseimpan dalam variable. Variabel ada 2 yaitu, biasa dan array.

```

1 x = "mantan 1"
2 z = 5
3 print("Terap" +str(x),z)

```

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

```

1 l = input("Masukan NPM :")
2 print("NPM Kamu Adalah : "+l)

```

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 perhitungan

```

1 m = 100
2 n = 25
3 o = m + n
4 p = m * n
5 q = m - n
6 r = m / n
7 s = "100"
8 print(o,p,q,r)
9 print(int(s))
10 print(str(m))

```

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

```

1 t = 12
2 for u in range(t):
3     print("Ini Yang Ke : "+str(u))
4
5 while(t <= 15):
6     print("Yeyyyyyyy Betul")
7     t = t + 1

```

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. If statement

```

1 v = 12
2 if(v==12):
3     print("Dua Belas")

```

Ifelse

```

1 if(v==13):
2     print("Tiga Belas")
3 else:
4     print("Dua Belas")

```

IfNested

```

1 if(v==11):
2     print("Sebelas")
3 elif(v==12):
4     print("Sebelas")
5 else :
6     print("Tiga Belas")

```

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.
- ZeroDivisionError 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 penerapan.
- EOFError Dibesarkan bila tidak ada input dari fungsi rawinput () atau input () dan akhir file tercapai.
- ImportError Dibesarkan saat sebuah pernyataan import 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 pengenal 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 w = 10
2 x = "10"
3 try :
4     y = w+x
5     print(y)
6 except TypeError:
7     print("Gabisa Coyyyy")
```

2.12.2 Ketampilan Pemrograman

- 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("###  ###  #####  ###  ###  #####  ######")
3 print("###  ###  #####  ###  ###  #####  ######")
4 print("###  ###      ###  ###  ##  ##  ##  ##")
5 print("###  ###      ###  ###  ##  ##  ##  ##")
6 print("###  ###      ###  #####  ##  ##  #####")
7 print("###  ###      ###          ##  ##  ##  ##")
8 print("###  ###      ###          ##  ##  ##  ##")
9 print("###  ###      ###          ##  ##  ##  ##")
10 print("###  ###      ###          ##  #####  ######")
11 print("###  ###      ###          ##  #####  ######")

```

- Buatlah program hello word dengan input NPM yang disimpan dalam sebuah variabel string bernama NPM dan output sebanyak dua digit belakang NPM.

```

1 #No 2
2 npm = input("Masukan NPM :")
3 hitung = 0
4 while(hitung < 51):
5     print("Halo, " + str(npm) + " Apa kabar?")
6     hitung = hitung + 1

```

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

```

- 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("Halo, " + str(npm[-3]) + " Apa kabar?")

```

5. Buat program dengan mengisi variabel alfabet dengan nomor npm satu persatu berurut.

```

1 #No 5
2 a = 1
3 b = 2
4 c = 3
5 d = 4
6 e = 5
7 f = 6
8 g = 7
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. Dari soal no 5, Lakukan penjumlahan dari seluruh variabel tersebut.

```

1 #No 6
2 print(h)

```

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

```

1 #No 7
2 print(i)

```

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

```

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

```

1 #No 9
2 print(d,e,f)

```

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

```

1
2 #No 10
3 print(a,b,c,g)

```

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

```
1 #No 11
2 print(a,b,c,g)
```

2.12.3 Error

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar  4 11:15:33 2019
4
5 @author: Tomy
6 """
7
8 a = 3
9 b = "9"
10
11 try:
12     c = a + b
13     print(c)
14 except TypeError:
15     print("Perbedaan Tipe Data")
```

2.13 Choirul Anam

2.13.1 Teori

1. variable Python Variabel merupakan tempat menyimpan data. Dalam phython kita dapat membuat variable dengan cara sebagai berikut

```
1 nama = 'mas choi'
2 print (nama)
```

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

```
1 nama = raw_input("masukan nama anda :")
2 print (nama)
```

```
1 umur = input("tolong masukan umur anda :")
2 print (umur)
```

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 perhitungan

```
1 aku = 0
2 kamu = 1
3 kita = kamu + aku
4 print "aku kamu itu", (kita)
5
```

```

6 #perkalian
7 panjang = 10
8 lebar = 5
9 luas = panjang * lebar
10 print(luas)
11
12 #pengurangan
13 dia1 = 10
14 dia2 = 5
15 dia_siapa = dia1 - dia2
16 print(dia_siapa)
17
18 #pembagian
19 a = 14
20 b = 4
21 c = a / b
22 print(c)
23
24 #mengubah int ke str
25 kamu = 'cantik'
26 kamu2 = 1
27 kamuitu = kamu + str(kamu2)
28 print (kamuitu)
29
30 #mengubah str ke int
31 a = 5
32 b = str(a)
33 print (b)

```

4. Tuliskan dan jelaskan sintak untuk perulangan, jenis-jenisnya contoh kode dan cara pakainya di python Untuk Perulangan Pada Python ada For dan While, Untuk Contohnya bisa lihat gambar berikut :

```

1 count = 1
2 while (count < 15):
3     print count
4     count = count + 2
5
6 #for loop
7 kamu = ["cantik", "manis", "sholeha"]
8 for aku in kamu:
9     print("saya suka cewek yang", aku)
10
11 #nested loop
12 i = 1
13 while(1 < 5):
14     j = 1
15     while(j <= 5):
16         j += 1
17         i += 1

```

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 nilai = 9
2 if(nilai > 11):
3     print("selamat ulang tahun")
4 else:
5     print("selamat datang")
6
7 #kondisi di dalam kondisi
8 nilai = 9
9 if nilai > 9:
10    print("nilai A")
11 elif nilai > 7:
12    print("nilai B")
13 else:
14    print("nilai C")
```

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.
- ZeroDivisionError 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 pungasan.
- 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 menyelesaikan 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 pengenal 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 try :
```

2.13.2 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 print(1174004%3)
2
3
4 print ("+++
5      +++)
6      +++)
7      +++)
8      +++)
9      +++)
10     +++)
11    +++)
12   +++)
13  +++)
14 +++)
15 +++)
16 +++)
17 +++)
18 +++)
19 +++)
20 +++)
21 +++)
22 +++)
23 +++)
24 +++)
25 +++)
26 +++)
27 +++)
28 +++)
29 +++)
30 +++)
31 +++)
32 +++)
33 +++)
34 +++)
35 +++)
36 +++)
37 +++)
38 +++)
39 +++)
40 +++)
41 +++)
42 +++)
43 +++)
44 +++)
45 +++)
46 +++)
47 +++)
48 +++)
49 +++)
50 +++)
51 +++)
52 +++)
53 +++)
54 +++)
55 +++)
56 +++)
57 +++)
58 +++)
59 +++)
60 +++)
61 +++)
62 +++)
63 +++)
64 +++)
65 +++)
66 +++)
67 +++)
68 +++)
69 +++)
70 +++)
71 +++)
72 +++)
73 +++)
74 +++)
75 +++)
76 +++)
77 +++)
78 +++)
79 +++)
80 +++)
81 +++)
82 +++)
83 +++)
84 +++)
85 +++)
86 +++)
87 +++)
88 +++)
89 +>)
90 +>)
91 +>)
92 +>)
93 +>)
94 +>)
95 +>)
96 +>)
97 +>)
98 +>)
99 +>)
100 +>)
101 +>)
102 +>)
103 +>)
104 +>)
105 +>)
106 +>)
107 +>)
108 +>)
109 +>)
110 +>)
111 +>)
112 +>)
113 +>)
114 +>)
115 +>)
116 +>)
117 +>)
118 +>)
119 +>)
120 +>)
121 +>)
122 +>)
123 +>)
124 +>)
125 +>)
126 +>)
127 +>)
128 +>)
129 +>)
130 +>)
131 +>)
132 +>)
133 +>)
134 +>)
135 +>)
136 +>)
137 +>)
138 +>)
139 +>)
140 +>)
141 +>)
142 +>)
143 +>)
144 +>)
145 +>)
146 +>)
147 +>)
148 +>)
149 +>)
150 +>)
151 +>)
152 +>)
153 +>)
154 +>)
155 +>)
156 +>)
157 +>)
158 +>)
159 +>)
160 +>)
161 +>)
162 +>)
163 +>)
164 +>)
165 +>)
166 +>)
167 +>)
168 +>)
169 +>)
170 +>)
171 +>)
172 +>)
173 +>)
174 +>)
175 +>)
176 +>)
177 +>)
178 +>)
179 +>)
180 +>)
181 +>)
182 +>)
183 +>)
184 +>)
185 +>)
186 +>)
187 +>)
188 +>)
189 +>)
190 +>)
191 +>)
192 +>)
193 +>)
194 +>)
195 +>)
196 +>)
197 +>)
198 +>)
199 +>)
200 +>)
201 +>)
202 +>)
203 +>)
204 +>)
205 +>)
206 +>)
207 +>)
208 +>)
209 +>)
210 +>)
211 +>)
212 +>)
213 +>)
214 +>)
215 +>)
216 +>)
217 +>)
218 +>)
219 +>)
220 +>)
221 +>)
222 +>)
223 +>)
224 +>)
225 +>)
226 +>)
227 +>)
228 +>)
229 +>)
230 +>)
231 +>)
232 +>)
233 +>)
234 +>)
235 +>)
236 +>)
237 +>)
238 +>)
239 +>)
240 +>)
241 +>)
242 +>)
243 +>)
244 +>)
245 +>)
246 +>)
247 +>)
248 +>)
249 +>)
250 +>)
251 +>)
252 +>)
253 +>)
254 +>)
255 +>)
256 +>)
257 +>)
258 +>)
259 +>)
260 +>)
261 +>)
262 +>)
263 +>)
264 +>)
265 +>)
266 +>)
267 +>)
268 +>)
269 +>)
270 +>)
271 +>)
272 +>)
273 +>)
274 +>)
275 +>)
276 +>)
277 +>)
278 +>)
279 +>)
280 +>)
281 +>)
282 +>)
283 +>)
284 +>)
285 +>)
286 +>)
287 +>)
288 +>)
289 +>)
290 +>)
291 +>)
292 +>)
293 +>)
294 +>)
295 +>)
296 +>)
297 +>)
298 +>)
299 +>)
300 +>)
301 +>)
302 +>)
303 +>)
304 +>)
305 +>)
306 +>)
307 +>)
308 +>)
309 +>)
310 +>)
311 +>)
312 +>)
313 +>)
314 +>)
315 +>)
316 +>)
317 +>)
318 +>)
319 +>)
320 +>)
321 +>)
322 +>)
323 +>)
324 +>)
325 +>)
326 +>)
327 +>)
328 +>)
329 +>)
330 +>)
331 +>)
332 +>)
333 +>)
334 +>)
335 +>)
336 +>)
337 +>)
338 +>)
339 +>)
340 +>)
341 +>)
342 +>)
343 +>)
344 +>)
345 +>)
346 +>)
347 +>)
348 +>)
349 +>)
350 +>)
351 +>)
352 +>)
353 +>)
354 +>)
355 +>)
356 +>)
357 +>)
358 +>)
359 +>)
360 +>)
361 +>)
362 +>)
363 +>)
364 +>)
365 +>)
366 +>)
367 +>)
368 +>)
369 +>)
370 +>)
371 +>)
372 +>)
373 +>)
374 +>)
375 +>)
376 +>)
377 +>)
378 +>)
379 +>)
380 +>)
381 +>)
382 +>)
383 +>)
384 +>)
385 +>)
386 +>)
387 +>)
388 +>)
389 +>)
390 +>)
391 +>)
392 +>)
393 +>)
394 +>)
395 +>)
396 +>)
397 +>)
398 +>)
399 +>)
400 +>)
401 +>)
402 +>)
403 +>)
404 +>)
405 +>)
406 +>)
407 +>)
408 +>)
409 +>)
410 +>)
411 +>)
412 +>)
413 +>)
414 +>)
415 +>)
416 +>)
417 +>)
418 +>)
419 +>)
420 +>)
421 +>)
422 +>)
423 +>)
424 +>)
425 +>)
426 +>)
427 +>)
428 +>)
429 +>)
430 +>)
431 +>)
432 +>)
433 +>)
434 +>)
435 +>)
436 +>)
437 +>)
438 +>)
439 +>)
440 +>)
441 +>)
442 +>)
443 +>)
444 +>)
445 +>)
446 +>)
447 +>)
448 +>)
449 +>)
450 +>)
451 +>)
452 +>)
453 +>)
454 +>)
455 +>)
456 +>)
457 +>)
458 +>)
459 +>)
460 +>)
461 +>)
462 +>)
463 +>)
464 +>)
465 +>)
466 +>)
467 +>)
468 +>)
469 +>)
470 +>)
471 +>)
472 +>)
473 +>)
474 +>)
475 +>)
476 +>)
477 +>)
478 +>)
479 +>)
480 +>)
481 +>)
482 +>)
483 +>)
484 +>)
485 +>)
486 +>)
487 +>)
488 +>)
489 +>)
490 +>)
491 +>)
492 +>)
493 +>)
494 +>)
495 +>)
496 +>)
497 +>)
498 +>)
499 +>)
500 +>)
501 +>)
502 +>)
503 +>)
504 +>)
505 +>)
506 +>)
507 +>)
508 +>)
509 +>)
510 +>)
511 +>)
512 +>)
513 +>)
514 +>)
515 +>)
516 +>)
517 +>)
518 +>)
519 +>)
520 +>)
521 +>)
522 +>)
523 +>)
524 +>)
525 +>)
526 +>)
527 +>)
528 +>)
529 +>)
530 +>)
531 +>)
532 +>)
533 +>)
534 +>)
535 +>)
536 +>)
537 +>)
538 +>)
539 +>)
540 +>)
541 +>)
542 +>)
543 +>)
544 +>)
545 +>)
546 +>)
547 +>)
548 +>)
549 +>)
550 +>)
551 +>)
552 +>)
553 +>)
554 +>)
555 +>)
556 +>)
557 +>)
558 +>)
559 +>)
560 +>)
561 +>)
562 +>)
563 +>)
564 +>)
565 +>)
566 +>)
567 +>)
568 +>)
569 +>)
570 +>)
571 +>)
572 +>)
573 +>)
574 +>)
575 +>)
576 +>)
577 +>)
578 +>)
579 +>)
580 +>)
581 +>)
582 +>)
583 +>)
584 +>)
585 +>)
586 +>)
587 +>)
588 +>)
589 +>)
590 +>)
591 +>)
592 +>)
593 +>)
594 +>)
595 +>)
596 +>)
597 +>)
598 +>)
599 +>)
600 +>)
601 +>)
602 +>)
603 +>)
604 +>)
605 +>)
606 +>)
607 +>)
608 +>)
609 +>)
610 +>)
611 +>)
612 +>)
613 +>)
614 +>)
615 +>)
616 +>)
617 +>)
618 +>)
619 +>)
620 +>)
621 +>)
622 +>)
623 +>)
624 +>)
625 +>)
626 +>)
627 +>)
628 +>)
629 +>)
630 +>)
631 +>)
632 +>)
633 +>)
634 +>)
635 +>)
636 +>)
637 +>)
638 +>)
639 +>)
640 +>)
641 +>)
642 +>)
643 +>)
644 +>)
645 +>)
646 +>)
647 +>)
648 +>)
649 +>)
650 +>)
651 +>)
652 +>)
653 +>)
654 +>)
655 +>)
656 +>)
657 +>)
658 +>)
659 +>)
660 +>)
661 +>)
662 +>)
663 +>)
664 +>)
665 +>)
666 +>)
667 +>)
668 +>)
669 +>)
670 +>)
671 +>)
672 +>)
673 +>)
674 +>)
675 +>)
676 +>)
677 +>)
678 +>)
679 +>)
680 +>)
681 +>)
682 +>)
683 +>)
684 +>)
685 +>)
686 +>)
687 +>)
688 +>)
689 +>)
690 +>)
691 +>)
692 +>)
693 +>)
694 +>)
695 +>)
696 +>)
697 +>)
698 +>)
699 +>)
700 +>)
701 +>)
702 +>)
703 +>)
704 +>)
705 +>)
706 +>)
707 +>)
708 +>)
709 +>)
710 +>)
711 +>)
712 +>)
713 +>)
714 +>)
715 +>)
716 +>)
717 +>)
718 +>)
719 +>)
720 +>)
721 +>)
722 +>)
723 +>)
724 +>)
725 +>)
726 +>)
727 +>)
728 +>)
729 +>)
730 +>)
731 +>)
732 +>)
733 +>)
734 +>)
735 +>)
736 +>)
737 +>)
738 +>)
739 +>)
740 +>)
741 +>)
742 +>)
743 +>)
744 +>)
745 +>)
746 +>)
747 +>)
748 +>)
749 +>)
750 +>)
751 +>)
752 +>)
753 +>)
754 +>)
755 +>)
756 +>)
757 +>)
758 +>)
759 +>)
760 +>)
761 +>)
762 +>)
763 +>)
764 +>)
765 +>)
766 +>)
767 +>)
768 +>)
769 +>)
770 +>)
771 +>)
772 +>)
773 +>)
774 +>)
775 +>)
776 +>)
777 +>)
778 +>)
779 +>)
780 +>)
781 +>)
782 +>)
783 +>)
784 +>)
785 +>)
786 +>)
787 +>)
788 +>)
789 +>)
790 +>)
791 +>)
792 +>)
793 +>)
794 +>)
795 +>)
796 +>)
797 +>)
798 +>)
799 +>)
800 +>)
801 +>)
802 +>)
803 +>)
804 +>)
805 +>)
806 +>)
807 +>)
808 +>)
809 +>)
810 +>)
811 +>)
812 +>)
813 +>)
814 +>)
815 +>)
816 +>)
817 +>)
818 +>)
819 +>)
820 +>)
821 +>)
822 +>)
823 +>)
824 +>)
825 +>)
826 +>)
827 +>)
828 +>)
829 +>)
830 +>)
831 +>)
832 +>)
833 +>)
834 +>)
835 +>)
836 +>)
837 +>)
838 +>)
839 +>)
840 +>)
841 +>)
842 +>)
843 +>)
844 +>)
845 +>)
846 +>)
847 +>)
848 +>)
849 +>)
850 +>)
851 +>)
852 +>)
853 +>)
854 +>)
855 +>)
856 +>)
857 +>)
858 +>)
859 +>)
860 +>)
861 +>)
862 +>)
863 +>)
864 +>)
865 +>)
866 +>)
867 +>)
868 +>)
869 +>)
870 +>)
871 +>)
872 +>)
873 +>)
874 +>)
875 +>)
876 +>)
877 +>)
878 +>)
879 +>)
880 +>)
881 +>)
882 +>)
883 +>)
884 +>)
885 +>)
886 +>)
887 +>)
888 +>)
889 +>)
890 +>)
891 +>)
892 +>)
893 +>)
894 +>)
895 +>)
896 +>)
897 +>)
898 +>)
899 +>)
900 +>)
901 +>)
902 +>)
903 +>)
904 +>)
905 +>)
906 +>)
907 +>)
908 +>)
909 +>)
910 +>)
911 +>)
912 +>)
913 +>)
914 +>)
915 +>)
916 +>)
917 +>)
918 +>)
919 +>)
920 +>)
921 +>)
922 +>)
923 +>)
924 +>)
925 +>)
926 +>)
927 +>)
928 +>)
929 +>)
930 +>)
931 +>)
932 +>)
933 +>)
934 +>)
935 +>)
936 +>)
937 +>)
938 +>)
939 +>)
940 +>)
941 +>)
942 +>)
943 +>)
944 +>)
945 +>)
946 +>)
947 +>)
948 +>)
949 +>)
950 +>)
951 +>)
952 +>)
953 +>)
954 +>)
955 +>)
956 +>)
957 +>)
958 +>)
959 +>)
960 +>)
961 +>)
962 +>)
963 +>)
964 +>)
965 +>)
966 +>)
967 +>)
968 +>)
969 +>)
970 +>)
971 +>)
972 +>)
973 +>)
974 +>)
975 +>)
976 +>)
977 +>)
978 +>)
979 +>)
980 +>)
981 +>)
982 +>)
983 +>)
984 +>)
985 +>)
986 +>)
987 +>)
988 +>)
989 +>)
990 +>)
991 +>)
992 +>)
993 +>)
994 +>)
995 +>)
996 +>)
997 +>)
998 +>)
999 +>)
999 +>)
```

```

6 print ("+++
    +++")
7 print ("+++
    +++")
8 print ("+++
    ++++++++")
9 print ("+++
    +++")
10 print ("+++
    +++")
11 print ("+++
    +++")
12 print ("+++
    +++")
13 print ("+++
    +++")
14 print ("+++
    +++")

```

2. Buatlah program hello word dengan input NPM yang disimpan dalam sebuah variabel string bernama NPM dan output sebanyak dua digit belakang NPM.

```

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

```

5. Buat program dengan mengisi variabel alfabet dengan nomor npm satu persatu berurut.

```

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

```

```

7 g = 4
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. Dari soal no 5, Lakukan penjumlahan dari seluruh variabel tersebut.

```

1 #nomor 6
2 print(h)

```

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

```

1 #nomor 7
2 print(i)

```

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

```

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

```

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

```

1 #nomor 9
2 print(e,f,g)

```

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

```

1 #nomor 10
2 print(a,b,c,g)

```

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

```

1 #print("Tetap" + str(j),k)

```

2.13.3 Ketrampilan Penanganan Error

```

1 try :
2     print("choi")
3 except:
4     print("mas choi ganteng")
5 finally:
6     print("memang ganteng")

```

2.14 Oniwaldus Bere Mali

2.14.1 Teori

1. sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut di kode Python Variabel merupakan tempat menyimpan data. Dalam phyton kita dapat membuat variable dengan cara sebagai berikut

```

1
2 #nomor 11
3 print(a,b,c,g)
```

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

```

1 k = 1
2 print("Tetap" +str(j),k)
```

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 perhitungan

```

1 l = input("Masukan NPM Kamu :")
2 print("NPM Kamu Adalah : "+l)
3
4 m = 10
5 n = 2
6 o = j + k
7 p = j * k
8 q = j - k
9 r = j / k
10 s = "10"
```

4. Tuliskan dan jelaskan sintak untuk perulangan, jenis-jenisnya contoh kode dan cara pakainya di python Untuk Perulangan Pada Python ada For dan While, Untuk Contohnya bisa lihat gambar berikut :

```

1 print(int(s))
2 print(str(m))
3 t = 12
4 for u in range(t):
5     print("Ini Yang Ke : "+str(u))
6 while(t <= 15):
7     print(" Ini yang Betul")
```

5. Tuliskan jelaskan cara pakai sintak untuk memilih kondisi, dan bagiamana con-toh sintak kondisi di dalam kondisi. Pengambilan kondisi If yang digunakan untuk mengantisipasi kondisi yang terjadi saat program dijalankan dan menen-tukan tindakan apa yang akan diambil sesuai dengan kondisi. If statement

```

1 v = 12
2 if(v==12):
3     print("Dua Belas")
```

Ifelse

```
1 print("Tiga Belas")
2 else:
3     print("Dua Belas")
4 if(v==11):
```

IfNested

```
1 elif(v==12):
2     print("Sebelas")
3 else :
4     print("Tiga Belas")
5 w = 10
6 x = "10"
```

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.
- ZeroDivisionError 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 pungasan.
- 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 pengenal 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   y = w+x
2   print(y)
3 except TypeError:
4     print("We Are Different")

```

2.14.2 Ketampilan 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
2 #Soal Praktek
3 #nomor 1
4 print ("###    ###    #####      ###    ###    #####")
5 print ("###    ###    #####      ###    ###    #####")
6 print ("###    ###          ###    ###    ##    ##")
7 print ("###    ###          ###    ###    ##    ##")
8 print ("###    ###          ###    #####      ##    ##")
9 print ("###    ###          ###    ##    ##    ##    ##")
10 print ("###    ###          ###    ##    #####      ##    ##")
11 print ("###    ###          ###    ##    ##    ##    ##")

```

2. Buatlah program hello word dengan input NPM yang disimpan dalam sebuah variabel string bernama NPM dan output sebanyak dua digit belakang NPM.

```

1 print ("###    ###    ###      ###    #####")
2 print ("###    ###    ###      ###    #####")
3
4 #nomor 2
5 npm = input("Masukan NPM :")
6 hitung = 0

```

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     print("Halo, " + str(npm) + " Apa Kabar?")
2     hitung = hitung + 1
3
4
5 #nomor 3
6 npm = input("Masukan NPM :")

```

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 while(hitung <= 15):
2     print("Halo, " + str(npm[4:7]) + " Apa Kabar?")
3     hitung = hitung + 1

```

5. Buat program dengan mengisi variabel alfabet dengan nomor npm satu persatu berurut.

```

1 #nomor 4
2 npm = input("Masukan NPM :")
3 print("Halo, " + str(npm[-3]) + " Apa Kabar?")
4
5 #nomor 5
6 a = 1
7 b = 1
8 c = 7
9 d = 4
10 e = 0
11 f = 0
12 g = 5

```

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

```
1 i = a*b*c*d*e*f*g
```

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

```

1
2 #nomor 6

```

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

```

1
2 #nomor 7
3 print(i)
4
5 #nomor 8
6 print(a)
7 print(b)
8 print(c)

```

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

```
1 print(e)
2 print(f)
```

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

```

1
2 #nomor 9

```

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

```

1
2 #nomor 10

```

2.14.3 Ketampilan Penanganan Error

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar  4 11:15:33 2019
4
5 @author: Aegis
6 """
7
8 a = 1
9 b = "2"
10
11 try:
12     c = a + b
13     print(c)
14 except TypeError:
15     print("Perbedaan Tipe Data")

```

2.15 Habib Abdul Rasyid

2.15.1 Teori

1. sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut di kode Python. Untuk membuat variabel Python caranya sangat mudah, cukup buat nama variabel lalu diikuti oleh = dan mengisinya dengan nilai yang dinginkan. Misalnya ada variabel nama dengan nilai habib abdul rasyid. Maka penulisan variabelnya seperti ini

```

1 nama = 'habib abdul rasyid'
2 print (nama)

```

2. Input adalah masukan yang anda berikan kepada sebuah program dan sistem akan memproses lalu menampilkan outputnya. Meminta inputan dari user dan menampilkan autputnya pada python untuk meminta inputan ada dua fungsi yaitu input()

```

1 usia = input("Masukan umur anda :")
2 print (usia)

```

3. Operator python adalah simbol yang melakukan operasi pada satu atau lebih operan. Operan adalah variabel atau nilai yang digunakan untuk melakukan operasi. selain itu ada juga convert antara int dan str.

```

1 aku = 1
2 kamu = 2
3 kita = aku + kamu
4 print ("Aku dan Kamu"), (kita)
5
6 print ("pengurangan")
7 lo1 = 5
8 lo2 = 2
9 kamu_siapa = lo1 - lo2

```

```

10 print (kamu_siapa)
11
12 print ("perkalian")
13 panjang = 5
14 lebar = 2
15 luas = panjang * lebar
16 print (luas)
17
18 print ("pembagian")
19 roti = 8
20 sobek = 2
21 kue_sobek = roti / sobek
22 print (kue_sobek)
```

```

1 print ("mengubah int ke string")
2 kamu = "cantik"
3 kamu2 = 1
4 kamuitu = kamu + str(kamu2)
5 print (kamuitu)
6
7 print ("mengubah str ke int")
8 a = 5
9 b = str(a)
10 print(b)
```

4. Secara umum, perintah pada bahasa pemrograman Python akan dijalankan secara berurutan. Pernyataan pertama dalam sebuah fungsi dijalankan pertama, lalu diikuti oleh yang kedua, dan seterusnya. Tetapi akan ada situasi dimana Anda harus menulis banyak kode, dimana kode tersebut sangat banyak. Jika dilakukan secara manual maka Anda hanya akan membuang-buang waktu dan tenaga. Untuk itu Anda perlu menggunakan pengulangan atau loop di dalam bahasa pemrograman Python.

```

1 i = 1
2 while i < 9:
3     print(i)
4     i += 1
5 #while
6 kue = ["Bolu", "Brownies", "Keju"]
7 for x in kue:
8     print(x)
```

5. Pada saat pengambilan keputusan (kondisi if) digunakan untuk mengantisipasi kondisi yang akan terjadi saat program dijalankan. Dan akan menentukan tindakan apa yang akan diambil sesuai dengan kondisi program pada saat dieksekusi.

```

1 x = 100
2 y = 20
3 if x > y:
4     print("x lebih besar dari y")
5 else:
6     print("x tidak lebih besar dari y")
```

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.
- ZeroDivisionError 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 pungasan.
- 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 pengenal 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 try :
2     print("habib")
3 except :
4     print("habib abdul rasyid")
5 finally :
6     print("mantap")

```

2.15.2 Ketampilan 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 print(1174002%3)
2
3 print("***** *** ***** * *** * * * * * * * * * * * * * *")
4 print("***** *** ***** * *** * * * * * * * * * * * * * *")
5 print("***** *** * *** * *** * * * * * * * * * * * * *")
6 print("***** *** * *** * *** * *** * * * * * * * * * * *")
7 print("***** *** * *** * * * * * * * * * * * * * * * * *")
8 print("***** *** * *** * * * * * * * * * * * * * * * *")
9 print("***** *** * *** * * * * * * * * * * * * * * * *")
10 print("***** *** * *** * * * * * * * * * * * * * * *")

```

```
11 print("*** *** ***           *** ***** *****")
      *****")
```

2. Buatlah program hello word dengan input NPM yang disimpan dalam sebuah variabel string bernama NPM dan output sebanyak dua digit belakang NPM.

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

5. Buat program dengan mengisi variabel alfabet dengan nomor npm satu persatu berurut.

```
1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 0
7 g = 2
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. Dari soal no 5, Lakukan penjumlahan dari seluruh variabel tersebut.

```
1 print(h)
```

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

```
1 print(i)
```

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

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

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

```
1 print(d,g)
```

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

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

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

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

2.15.3 Ketampilan Penanganan Error

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar  4 12:03:43 2019
4
5 @author: Habib Abdul R
6 """
7
8 a = 3
9 b = "6"
10
11 try:
12     c = a + b
13     print(c)
14 except TypeError:
15     print("Tipe data nya berbeda")
```

2.16 Arjun Yuda Firwanda

2.16.1 Teori

1. Sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut di kode Python. Variabel merupakan sebuah ruang kosong untuk menyimpan suatu nilai atau data. Dalam phyton kita dapat membuat variable dengan cara sebagai berikut

```
1 print (nama)
```

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

```
1 print (nama)
```

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 perhitungan. Penjumlahan

```
1 y = 9
2 total = x + y
3 print (total)
```

Perkalian

```
1 n = 5
2 total = m * n
3 print (total)
```

Pengurangan

```
1 b = 5
2 total = a - b
3 print (total)
```

Pembagian

```
1 q = 5
```

Mengubah Integer Ke string

```
1 kamu2 = 1
2 kamuitu = akul + int(kamu2)
3 print (kamuitu)
```

Mengubah String Ke Integer

```
1 y = str(x)
2 print(y)
```

4. Tuliskan dan jelaskan sintak untuk perulangan, jenis-jenisnya contoh kode dan cara pakainya di python Untuk Perulangan Pada Python ada While Loop, For Loop dan Nested Loop, Untuk Contohnya bisa lihat gambar berikut : While Loop

```
1 while(count < 15):
2     print (count)
3     count = count + 2
```

For Loop

```
1 for aku in kamu:
2     print("Saya Suka Cowok yang", aku)
```

Nested Loop

```

1 while (i < 5):
2     j = 1
3     while (j <= 5):
4         j += 1
5     i += 1

```

5. Tuliskan jelaskan cara pakai sintak untuk memilih kondisi, dan bagaimana contoh sintak kondisi di dalam kondisi.

```

1 if ( nilai > 12):
2     print("Selamat Ya")
3 else:
4     print("Selamat Datang")

```

Kondisi Di Dalam Kondisi

```

1 if nilai > 9:
2     print("Nilai A")
3 elif nilai > 7:
4     print("Nilai C")

```

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.
- ZeroDivisionError 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 pungasan.
- 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 pengenal 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 print("Arjun")
2 except:
3     print("Arjun Ganteng")
4 finally:
5     print("Memanglah")
```

2.16.2 Keterampilan 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 print ("*** *** **** * * * * * * * * * *")
2 print ("*** *** **** * * * * * * * * * *")
3 print ("*** *** *** * * * * * * * * * *")
4 print ("*** *** *** * * * * * * * * * *")
5 print ("*** *** *** * * * * * * * * * *")
6 print ("*** *** *** * * * * * * * * * *")
7 print ("*** *** *** * * * * * * * * * *")
8 print ("*** *** *** * * * * * * * * * *")
9 print ("*** *** *** * * * * * * * * * *")
10 print ("*** *** *** * * * * * * * * * *")
11 print ("*** *** *** * * * * * * * * * *")

```

2. Buatlah program hello word dengan input NPM yang disimpan dalam sebuah variabel string bernama NPM dan output sebanyak dua digit belakang NPM.

```

1 npm = input("Masukan NPM :")
2 hitung = 0
3 while(hitung <= 87):
4     print("Halo, " + str(npm) + " Apa Kabar?")
5     hitung = hitung + 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 npm = input("Masukan NPM :")
2 hitung = 0
3 while(hitung <= 15):
4     print("Halo, " + str(npm[4:7]) + " Apa Kabar?")
5     hitung = hitung + 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 npm = input("Masukan NPM :")
2 print("Halo, " + str(npm[-3]) + " Apa Kabar?")

```

5. Buat program dengan mengisi variabel alfabet dengan nomor npm satu persatu berurut.

```
1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 2
7 g = 7
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. Dari soal no 5, Lakukan penjumlahan dari seluruh variabel tersebut.

```
1 print(h)
```

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

```
1 print(i)
```

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

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

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

```
1 print(d,g)
```

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

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

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

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

2.16.3 Keterampilan Penanganan Error

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar  4 19:51:16 2019
4
5 @author: PERSONAL
6 """
7
8 #Keterampilan Penanganan Error
9 a = 3
10 b = "6"
11
12 try:
13     c = a + b
14     print(c)
15 except TypeError:
16     print("Tipe data nya berbeda")
```

2.17 Muh. Rifky Prananda

2.17.1 Teori

1. sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut di kode Python Variabel yaitu merupakan tempat menyimpan data. Dalam phyton kita dapat membuat variable dengan cara sebagai berikut
2. tuliskan bagaimana kode untuk meminta input dari user dan tuliskan bagaimana melakukan output ke layar.
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 perhitungan
4. Tuliskan dan jelaskan sintak untuk perulangan, jenis-jenisnya contoh kode dan cara pakainya di python Untuk Perulangan Pada Python ada For dan While, Untuk Contohnya bisa lihat gambar berikut :
5. Tuliskan jelaskan cara pakai sintak untuk memilih kondisi, dan bagiamana 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

Ifelse

IfNested

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.
- ZeroDivisionError 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 pungasan.
- 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 pengenal 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.

2.17.2 Ketampilan 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 #nomor 1
2 print ("***      ***      *****      ***      ***      *****      ***")
3 print ("***      ***      *****      ***      ***      *****      ***")
4 print ("***      ***      ***      ***      ***      **      **      ***")
5 print ("***      ***      ***      ***      ***      **      **      ***")
6 print ("***      ***      ***      *****      **      **      ***")
7 print ("***      ***      ***      ***      **      **      **      ***")
8 print ("***      ***      ***      ***      **      **      **      ***")
9 print ("***      ***      ***      ***      **      **      **      ***")
10 print ("***      ***      ***      ***      *****      ***      ***")
11 print ("***      ***      ***      ***      *****      ***      ***")

```

2. Buatlah program hello word dengan input NPM yang disimpan dalam sebuah variabel string bernama NPM dan output sebanyak dua digit belakang NPM.

```

1 #nomor 2
2 npm = input("Masukan NPM :")
3 hitung = 0
4 while(hitung <= 87):
5     print("Halo, " + str(npm) + " Apa kabar ?")
6     hitung = hitung +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 dijít tersebut.

```

1 #nomor 3
2 npm = input("Masukan NPM :")
3 hitung = 0
4 while(hitung <= 15):
5     print("Halo, " + str(npm[4:7]) + " Apa kabar ?")
6     hitung = hitung +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 #nomor 4
2 npm = input("Masukan NPM :")
3 print("Halo, " + str(npm[-3]) + " Apa Kabar?")

```

5. Buat program dengan mengisi variabel alfabet dengan nomor npm satu persatu berurut.

```

1 #nomor 5
2 a = 1
3 b = 1
4 c = 7
5 d = 4
6 e = 0
7 f = 1
8 g = 7
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. Dari soal no 5, Lakukan penjumlahan dari seluruh variabel tersebut.

```

1 #nomor 6
2 print(h)

```

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

```

1 #nomor 7
2 print(i)

```

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

```

1 #nomor 8
2 print(a)
3 print(b)
4 print(c)
5 print(d)
6 print(e)
7 print(f)
8 print(g)

```

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

```

1 #nomor 9
2 print(d,g)

```

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

```

1 #nomor 10
2 print(a,b,c,f)

```

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

```

1 #nomor 11
2 print(a,b,c,)

```

2.17.3 Ketampilan Penanganan Error

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar  4 11:15:33 2019
4
5 @author: Aegis
6 """
7
8 a = 1
9 b = "2"
10
11 try :
12     c = a + b
13     print(c)
14 except TypeError :
15     print("Perbedaan Tipe Data")

```

2.18 Muhammad Fahmi

2.18.1 Teori

2.18.1.1 Variabel Variabel merupakan sebuah ruang kosong untuk menyimpan suatu nilai atau data. Pada saat anda membuat sebuah variabel berarti anda sedang memesan sebuah ruang kosong di memori. Isi dari variabel itu dapat berubah atau mutable sesuai dengan operasi yang diinginkan.

Misalnya ada variabel nama dengan nilai Muhammad Fahmi. Maka penulisan variabelnya seperti ini :

```
1 #No.1 Variabel
2 nama = 'Muhammad Fahmi'
3 print (nama)
```

Listing 2.29 Variabel

Kemudian perintahkan print untuk menampilkan isi dari variabel, hasilnya akan seperti ini :

```
In [4]: runfile('C:/Users/user/.spyder-py3/temp.py', wdir='C:/Users/u
Muhammad Fahmi
```

Jenis-Jenis Tipe Data

1. *Boolean* True atau False Menyatakan benar True yang bernilai 1, atau salah False yang bernilai 0
2. *String* Menyatakan karakter/kalimat bisa berupa huruf angka dan sebagainya lalu String juga diapit dengan tanda " atau '
3. *Integer* Menyatakan bilangan bulat
4. *Float* Menyatakan bilangan yang mempunyai koma
5. *List* Data untaian yang menyimpan berbagai tipe data dan isinya bisa diubah-ubah
6. *Tuple* Data untaian yang menyimpan berbagai tipe data tapi isinya tidak bisa diubah
7. *Dictionary* Data untaian yang menyimpan berbagai tipe data berupa pasangan penunjuk dan nilai

Contoh dari jenis-jenis variabel tersebut apabila di implementasikan kedalam Python adalah :

```
1 #tipe data Boolean
2 print(True)
3
4 #tipe data String
5 print("Muhammad Fahmi")
6 print('Politeknik Pos Indonesia')
7
8 #tipe data Integer
9 print(20)
10
11 #tipe data Float
12 print(1.5)
13
14 #tipe data List
15 print([1,2,3,4,5])
16 print(["satu", "dua", "tiga"])
```

```
17  
18 #tipe data Tuple  
19 print((1,2,3,4,5))  
20 print(("satu", "dua", "tiga"))  
21  
22 #tipe data Dictionary  
23 print({"nama": "Fahmi", 'umur': 19})
```

Listing 2.30 Tipe Data

2.18.1.2 Input Output User Input adalah masukan yang anda berikan kepada sebuah program dan sistem akan memproses lalu menampilkan outputnya. Cara mengambil input :

Di dalam python sudah terdapat fungsi input dan raw input. input digunakan untuk masukan bernilai angka sedangkan raw input untuk masukan bernilai teks.

Cara penggunaanya :

```

1  nama = input("Masukkan Nama : ")
2  print(nama)
3
4 # Mengambil input
5 nama = raw_input("nama: ")
6 umur = input("umur: ")
7
8 # Menampilkan output
9 print "Assalamualaikum",nama,"umur kamu",umur,"tahun"

```

Listing 2.31 Input Output

2.18.1.3 Operator Dasar Aritmatika Operator python adalah simbol yang melakukan operasi pada satu atau lebih operan. Operan adalah variabel atau nilai yang digunakan untuk melakukan operasi.

Operator pada Python dibagi menjadi beberapa jenis, yaitu :

```

1 #No.3 Operator Dasar Aritmatika
2 #Pertambahan
3 angka1 = 5
4 angka2 = 5
5 hasil = angka1+angka2
6 print(hasil)
7
8 #Pengurangan
9 angka1 = 5
10 angka2 = 5
11 hasil = angka1-angka2
12 print(hasil)
13
14 #Perkalian
15 angka1 = 5
16 angka2 = 5
17 hasil = angka1*angka2
18 print(hasil)
19
20 #Pembagian
21 angka1 = 5
22 angka2 = 5
23 hasil = angka1/angka2
24 print(hasil)
25
26 #Modulus
27 angka1 = 5
28 angka2 = 5
29 hasil = angka1%angka2
30 print(hasil)

```

```

31
32 #Perpangkatan
33 angka1 = 5
34 angka2 = 5
35 hasil = angka1**angka2
36 print(hasil)
37
38 #Pembulatan Pembagian
39 angka1 = 5
40 angka2 = 5
41 hasil = angka1 // angka2
42 print(hasil)
43
44 #Mengubah Tipe Data
45 #String ke Integer
46 var_str = '5'
47 var_int = int(var_str)
48 print(var_int)
49
50 #Integer ke String
51 var_int = 5
52 var_str = str(var_int)
53 print(var_str)

```

Listing 2.32 Operator Dasar Aritmatika

2.18.1.4 Sintak Untuk Perulangan Secara umum, perintah pada bahasa pemrograman Python akan dijalankan secara berurutan. Pernyataan pertama dalam sebuah fungsi dijalankan pertama, lalu diikuti oleh yang kedua, dan seterusnya. Tetapi akan ada situasi dimana Anda harus menulis banyak kode, dimana kode tersebut sangat banyak. Jika dilakukan secara manual maka Anda hanya akan membuang-buang waktu dan tenaga. Untuk itu Anda perlu menggunakan pengulangan atau loop di dalam bahasa pemrograman Python.// Cara penggunaannya adalah :

```

1 #No.4 Perulangan
2 #While Loop
3 hitung = 1
4 while (hitung < 6):
5     print (hitung)
6     hitung += 1
7
8 #For Loop
9 angka = [1 ,2 ,3 ,4 ,5]
10 for a in angka:
11     print(a)
12
13 #Nested Loop
14 i = 1
15 while(i < 5):
16     j = 1
17     while(j <= 5):
18         print('#')
19         j += 1
20     i += 1

```

Listing 2.33 Operator Dasar Aritmatika

2.18.1.5 Sintak Untuk Memilih Kondisi Di dalam Python ada beberapa cara untuk memilih kondisi didalam suatu kode, kemudian jika dijalankan maka kode tersebut akan berjalan jika sesuai kondisi yang ditentukan.

Contoh penggunaannya adalah :

```

1 #No.5 Sintak untuk memilih kondisi
2 #If
3 angka = 5
4 if angka > 0:
5     print(angka, "adalah bilangan positif.")
6
7 #If Else
8 angka = -5
9 if angka > 0:
10    print(angka, "adalah bilangan positif.")
11 else:
12    print(angka, "adalah bilangan negatif.")
13
14 #Elif
15 angka = 0
16 if angka > 0:
17    print(angka, "adalah bilangan positif.")
18 elif angka < 0:
19    print(angka, "adalah bilangan negatif.")
20 else:
21    print(angka, "adalah bilangan nol.")
22
23
24 #Kondisi di dalam kondisi
25 if angka > 0:
26    print(angka, "adalah bilangan positif.")
27    if angka > 50:
28        print(angka, "adalah bilangan lebih dari 50.")
29    else:
30        print(angka, "adalah bilangan kurang dari 50.")
31 elif angka < 0:
32    print(angka, "adalah bilangan negatif.")
33 else:
34    print(angka, "adalah bilangan nol.")

```

Listing 2.34 Operator Dasar Aritmatika

2.18.1.6 Jenis-Jenis Error Ada beberapa jenis error yang terdapat di Python :

1. Syntax Errors Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
2. 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.

3. 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.
4. Type Error TypeError adalah exception yang terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengoversi varibelnya sesuai dengan tipe data yang akan digunakan.

2.18.1.7 Try Except Python memiliki banyak exceptions bawaan yang memaksa program Anda untuk menghasilkan kesalahan ketika ada sesuatu yang salah di dalamnya. Ketika exceptions ini terjadi, itu menyebabkan proses saat ini berhenti dan meneruskannya ke proses panggilan sampai ditangani. Jika tidak ditangani, program ini akan macet. Contoh penggunaannya adalah :

```
1 #NO.7 Try Except
2 try :
3     print(te)
4 except:
5     print("Telah terjadi kesalahan dalam penulisan kode")
6 finally:
7     print("Try except telah selesai")
```

Listing 2.35 Operator Dasar Aritmatika

2.18.2 Keterampilan Pemrograman

1. Jawaban Soal 1

```

1 #No.1
2 print(1174002%3)
3
4 print("*** *** **** * *** * *** * *** * ***")
5 print("*** *** **** * *** * *** * *** * ***")
6 print("*** *** *** *** * *** * *** * ***")
7 print("*** *** *** *** * *** * *** * ***")
8 print("*** *** *** *** **** * *** * *** * ***")
9 print("*** *** *** *** * *** * *** * ***")
10 print("*** *** *** *** * *** * *** * ***")
11 print("*** *** *** *** * *** * *** * ***")
12 print("*** *** *** *** * *** * *** * ***")

```

2. Jawaban Soal 2

```

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

```

3. Jawaban Soal 3

```

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

```

4. Jawaban Soal 4

```

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

```

5. Jawaban Soal 5

```

1 #No.5
2 a = 1

```

```
3 b = 1
4 c = 7
5 d = 4
6 e = 0
7 f = 2
8 g = 1
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 6

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

7. Jawaban Soal 7

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

8. Jawaban Soal 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 9

```
1 #No.9
2 print(d,f)
```

10. Jawaban Soal 10

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

11. Jawaban Soal 11

```
1 #No.11
2 print(2)
```

2.18.3 Keterampilan Penanganan Error

1. Jawaban Soal No. 1

- (a) Syntax Errors Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- (b) 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.
- (c) 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.
- (d) 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.

2. Jawaban Soal No. 2

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

2.19 Rahmatul Ridha

2.19.1 Teori

1. jenis-jenis variable phyton dan cara pemakaianya Variable merupakan tempat untuk menyimpan data, Isi dari variabel itu dapat berubah atau mutable sesuai dengan operasi yang diinginkan. Saat program dieksekusi maka variabellah yang bertugas menyimpan data. Dimana didalam phyton terdapat beberapa variable diantaranya number, boolean, string. Dalam membuat variabel Python caranya adalah sebagai berikut

```
1 # variabel
2 c = 7
3 d = "Rahmatul Ridha"
4
5 print(c)
6 print(d)
```

```

7
8 #integer
9 x = 4
10 y = 6969787819
11 z = -456283
12
13 print(type(x))
14 print(type(y))
15 print(type(z))
16
17 #float
18 x = 1.115
19 y = 1.5
20 z = -35.60
21
22 print(type(x))
23 print(type(y))
24 print(type(z))
25
26 #complex
27 x = 5+5j
28 y = 5j
29 z = -5j
30
31 print(type(x))
32 print(type(y))
33 print(type(z))
34
35 #variabel string
36 a = "Hello , World!"
37 print(a[1])
38
39 b = "Hello , World!"
```

2. operator dasar aritmatika dimana terdapat penjumlahan,pengurangan,pembagian,perkalian nominal

```

1 print(b[2:7])
2
3 a = "Hello , World!"
4 print(a.strip())
5
6 a = "Hello , World!"
7 print(len(a))
8
9 a = "Hello , World!"
10 print(a.lower())
11
12 a = "Hello , World!"
13 print(a.upper())
14
15 #variabel boolean
16 a = 40
17 b = 300
18 if b>a:
19     print("b lebih besar dari a")
```

```
20
21 #input output
22 npm = input()
23 print(npm)
24
25 #operator
26 x = 5
27 y = 3
28
29 print(x+y)
```

3. Perulangan dalam phyton terdapat perulangan while dan for

```
1
2 x = 5
3 y = 3
4
5 print(x-y)
6
7 x = 12
8 y = 3
9
10 print(x*y)
11
12 x = 8
```

4. Dimana terdapat sintak untuk meilih kondisi didalam kondisi Untuk memilih keputusan menggunakan (kondisi if) dimana digunakan untuk mengantisipasi kondisi yang terjadi saat jalannya suatu program dan menentukan tindakan apa yang akan dilakukan sesuai dengan kondisi.

```
1 y = 2
2
3 print(x/y)
4
5 x = 5
6 y = 2
7
8 print(x%y)
9
10 #perulangan for
11 ulang = ["Motor","Mobil","Pesawat"]
12 for i in range(ulang):
13     print(i)
14
15 #perulangan while
16 coba = 1
17 while i < 6 :
18     print(i)
19     i += 1
20
21 #if
22 a = 40
23 b = 300
24 if b > a :
25     print("b lebih besar dari a")
```

5. Jenis-jenis sintak error pada python Syntax errors Jika dalam program terdapat kesalahan sintaks maka proses akan berhenti dan menampilkan pesan kesalahan. Runtime errors, disebut begitu karenakesalahan 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.

Type eror merupakan eror yang terjadi saat dilakukan eksekusi pada suatu operasi dengan type object yang tidak sesuai. ZeroDivision eror merupakan eror yang terjadi saat eksekusi program menghasilkan perhitungan matematika dengan angka 0

6. Try except cara memakai try except adalah sebagai berikut

```

1 # variabel
2 c = 7
3 d = "Rahmatul Ridha"
4
5 print(c)
6 print(d)
7
8 #integer
9 x = 4
10 y = 6969787819

```

2.19.2 praktek

1. Jawaban soal no 1

```

1 print(1144124%3)
2
3 #Nomor 1
4
5 print("###    ###    ###    ###    ###    ###    ###    ###    #####    ###    ###")
6 print("###    ###    ###    ###    ###    ###    ###    #####    ###    ###")
7 print("###    ###    ###    ###    ###    ###    ###    ")
8 print("###    ###    ###    ###    ###    ###    ")
9 print("###    ###    #####    #####    ###    #####    #####    #####")
10 print("###    ###    ")

```

2. Jawaban soal no 2

```

1 print("###    ###    ###    ###    ###    ")
2 print("###    ###    ###    ###    ###    ")
3 print("###    ###    ###    ###    #####    ")

```

```
4 print("###    ###      ###      ###    ###    #####      ###")  
5  
6 #Nomor 2
```

3. Jawaban soal no 3

```
1  
2 npm = input("Masukkan NPM :")  
3 hitung = 0  
4 while(hitung <= 51):  
5     print("Hallo, " + str(npm) + "Apa kabar?")  
6     hitung = hitung +1
```

4. Jawaban soal no 4

```
1  
2  
3 #Nomor 3
```

5. Jawaban soal no 5

```
1  
2 npm = input("Masukkan NPM :")  
3 hitung = 0  
4 while(hitung <= 6):  
5     print("Hello, " + str(npm[4:7]) + " Apa kabar?")  
6     hitung = hitung +1  
7  
8 #Nomor 4  
9 npm = input("Masukkan NPM : ")  
10 print("Hello, " + str(npm[-3]) + " Apa kabar?")
```

6. Jawaban soal no 6

```
1 #Nomor 5  
2 a = 1
```

7. Jawaban soal no 7

```
1 b = 1  
2 c = 7
```

8. Jawaban soal no 8

```
1 d = 4  
2 e = 0  
3 f = 5  
4 g = 1  
5 h = a+b+c+d+e+f+g  
6 i = a*b*c*d*e*f*g  
7  
8 print("Hasil no 5")
```

9. Jawaban soal no 9

```
1 print("NPM anda")
2 print(a, b, c, d, e, f, g, h)
```

10. Jawaban soal no 10

```
1
2 #Nomor 6
```

11. Jawaban soal no 11

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

2.19.3 Keterangan dan Penanganan eror

```
1     c = a + b
2     print(c)
3 except TypeError :
4     print("Perbedaan Tipe Data")
```

BAB 3

JUDUL BAGIAN KETIGA

3.1 Harun Ar - Rasyid

3.1.0.1 Pemahaman Teori

1. Apa itu fungsi, inputan

fungsi dan kembalian fungsi dengan contoh kode program lainnya. Fungsi adalah bagian dari program yang dapat digunakan ulang. Berikut merupakan contoh fungsi dan cara pemanggilannya

```
1 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 menspesifikasi 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_harun 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 mewakili 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 pemakaianya dengan contoh program lainnya. Cara Pemanggilannya

- 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_harun import *
2 #print( penulisan(int(input("Masukan NPM kamu : "))))
```

6. Jelaskan dengan contoh kodennya, pemakaian paket fungsi apabila
le library ada di dalam folder. Pemakaian paket adalah perkumpulan fungsi-fungsi. contoh kodennya adalah sebagai berikut :

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

```
1 #Chapter 3
```

3.1.0.2 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 #Chapter 3
2 #No 1
3 def penulisan(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": " ##### "}
```

```

10 angka5 = {"0": "##      ## ", "1": "## ## ## ", "2": "## ##### ", "3": "######
11 ###", "4": "##### ## ", "5": "##### ", "6": "##### ", "7": "#####
12 ##", "8": "##### ", "9": "##### "}
13 angka6 = {"0": "##      ## ", "1": "## ## ## ", "2": "## ##### ", "3": "######
14 ###", "4": "## ## ## ", "5": "##### ", "6": "##### ", "7": "#####
15 ##", "8": "##### ", "9": "##### "}
16 angka7 = {"0": "##      ## ", "1": "## ## ## ", "2": "## ##### ", "3": "######
17 ###", "4": "## ## ## ", "5": "##### ", "6": "##### ", "7": "#####
18 ##", "8": "##### ", "9": "##### "}
19 angka8 = {"0": "##      ## ", "1": "## ## ## ", "2": "## ##### ", "3": "######
20 ###", "4": "## ## ## ", "5": "##### ", "6": "##### ", "7": "#####
21 ##", "8": "##### ", "9": "##### "}
22 angka9 = {"0": "##### ## ", "1": "## ## ## ", "2": "## ##### ", "3": "######
23 ###", "4": "## ## ## ", "5": "##### ", "6": "##### ", "7": "#####
24 ##", "8": "##### ", "9": "##### "}
25 angka10 = {"0": "##### ## ", "1": "## ## ## ", "2": "## ##### ", "3": "######
26 ###", "4": "## ## ## ", "5": "##### ", "6": "##### ", "7": "#####
27 ##", "8": "##### ", "9": "##### "}
28
29 hasil1 = []
30 hasil2 = []
31 hasil3 = []
32 hasil4 = []
33 hasil5 = []
34 hasil6 = []
35 hasil7 = []
36 hasil8 = []
37 hasil9 = []
38 hasil10 = []
39
40 for x in npm:
41     hasil1.append(angka1[x])
42     hasil2.append(angka2[x])
43     hasil3.append(angka3[x])
44     hasil4.append(angka4[x])
45     hasil5.append(angka5[x])
46     hasil6.append(angka6[x])
47     hasil7.append(angka7[x])
48     hasil8.append(angka8[x])
49     hasil9.append(angka9[x])
50     hasil10.append(angka10[x])
51
52 print(*hasil1, sep=' ')
53 print(*hasil2, sep=' ')
54 print(*hasil3, sep=' ')
55 print(*hasil4, sep=' ')
56 print(*hasil5, sep=' ')
57 print(*hasil6, sep=' ')
58 print(*hasil7, sep=' ')
59 print(*hasil8, sep=' ')
60 print(*hasil9, sep=' ')
61 print(*hasil10, sep=' ')
62
63 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 def perulangan(npm):
2     hitung = 0
3     while(hitung < 27):
4         print("Halo , "+str(npm)+" apa kabar?")
5         hitung = hitung +1
6
7 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 < 9):
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)

```

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 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 = [1,1,7,4,0,2,7]
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 = [1,1,7,4,0,2,7]
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 : ")))

```

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_harun 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 3lib import penulisan
```

3.1.0.3 Ketrampilan Penanganan Error Error yang di dapat dari mengerjakan tugas ini adalah type error, cara menaggulaginya dengan cara mengecheck 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")

```

3.2 Evetania

3.2.0.1 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")
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 menspesifikasi 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_evie 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 mepresentasikan 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 Pemanggilannya

- 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_evie import *
2 #print(penulisan(int(input("Masukan NPM kamu : "))))
```

6. Jelaskan dengan contoh kodennya, pemakaian paket fungsi apabila le library ada di dalam folder. Pemakaian paket adalah perkumpulan fungsi-fungsi. contoh kodennya adalah sebagai berikut :
7. Jelaskan dengan contoh kodennya, pemakaian paket kelas apabila le library ada di dalam folder.

```
#Chapter 3
```

3.2.0.2 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 #Chapter 3
2 #No 1
3 def penulisan(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": " ##### "}
```

```

8     angka3 = {"0": " ##      ## ", "1": " ###  ", "2": "    ### ", "3": " "
9       "###", "4": " ###  ## ", "5": " ##      ", "6": " ##### ", "7": "
10      "##", "8": " ##### ", "9": " ##### "}
11     angka4 = {"0": " ##      ## ", "1": " ###  ", "2": "    ### ", "3": " "
12       "###", "4": " ###  ## ", "5": " ##      ", "6": " ##### ", "7": "
13      "##", "8": " ##### ", "9": " ##### "}
14     angka5 = {"0": " ##      ## ", "1": " ###  ", "2": "    ### ", "3": " "
15       "###", "4": " ##### ", "5": " ##### ", "6": " ##### ", "7": "
16     hasil1 = []
17     hasil2 = []
18     hasil3 = []
19     hasil4 = []
20     hasil5 = []
21     hasil6 = []
22     hasil7 = []
23     hasil8 = []
24     hasil9 = []
25     hasil10 = []
26
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 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 def perulangan(npm):
2     hitung = 0
3     while(hitung < 51):
4         print("Halo, "+str(npm)+" apa kabar?")
5         hitung = hitung +1
6
7 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 < 9):
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)

```

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 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 dijitur genap saja. wajib menggunakan perulangan dan atau kondisi.

```

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()

```

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

```

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()

```

10. Buatlah fungsi dengan inputan variabel NPM, Lakukan print NPM anda tapi hanya dijitur 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 : ")))

```

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

```
1 from fungsi_evie 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 mainn.py.

```
1 from 3lib import penulisan
```

3.2.0.3 Ketrampilan Penanganan Error Error yang di dapat dari mengerjakan tugas ini adalah type error, cara menaggulaginya dengan cara mengecheck 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")

```

3.3 Kadek Diva Krishna Murti

3.3.1 Pemahaman Teori

- Pengertian fungsi, inputan fungsi, dan kembalian fungsi serta contoh kode programnya.

- Fungsi adalah blok program untuk melakukan tugas-tugas tertentu yang dilakukan berulang dan dapat digunakan berulang kali dari tempat lain di dalam program.

```
1 def namaFungsi(inputanFungsi):
```

Listing 3.1 namaFungsi merupakan nama dari fungsi yang dibuat.

- Inputan fungsi adalah inputan yang berasal dari luar fungsi yang akan di proses di dalam fungsi itu sendiri.

```
1 def namaFungsi(inputanFungsi):
```

Listing 3.2 inputanFungsi merupakan nama dari inputan fungsi yang diterima dari luar fungsi namaFungsi.

- Kembalian fungsi adalah untuk mengembalikan suatu nilai ekspresi dari proses yang dilakukan fungsi.

```
1     return inputanFungsi
```

Listing 3.3 return inputanFungsi merupakan kembalian fungsi dari fungsi namaFungsi.

Penggunaan fungsi di Python

```
1 #Jawaban No. 1
2 def namaFungsi(inputanFungsi):
3     return inputanFungsi
4
5 output = namaFungsi("Kembalian Fungsi")
6 print(output)
```

Listing 3.4 Contoh penggunaan fungsi di Python.

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.namafungsinya.

Berikut ini merupakan contoh penggunaan paket atau library.

```
1 #Jawaban No. 2
2 import math
3 print("Nilai pi adalah: ", math.pi)
```

Listing 3.5 Contoh penggunaan paket atau library.

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 #Jawaban 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 tampilanProfil(self):
11        print("NPM :", self.npm)
12        print("Nama :", self.nama)
13        print()
14
15 mahasiswa1 = Mahasiswa("1174006", "Diva Krishna")
16 mahasiswa2 = Mahasiswa("1174001", "Indah Pramesti")
17
18 mahasiswa1.tampilanProfil()
19 mahasiswa2.tampilanProfil()
20
21 print("Total mahasiswa adalah ", Mahasiswa.jumlahMahasiswa)

```

Listing 3.6 Contoh penggunaan kelas di python.

- Objek Objek adalah instansi atau perwujudan dari sebuah kelas.

```

1 mahasiswa1 = Mahasiswa("1174006", "Diva Krishna")
2 mahasiswa2 = Mahasiswa("1174001", "Indah Pramesti")

```

Listing 3.7 Contoh penggunaan objek di python.

- Atribut Atribut adalah variabel yang menyimpan data yang berhubungan dengan kelas dan objeknya.

```
1 jumlahMahasiswa = 0
```

Listing 3.8 Contoh penggunaan atribut di python.

- Method Metode adalah fungsi yang didefinisikan di dalam suatu kelas.

```

1     def tampilanProfil(self):
2         print("NPM :", self.npm)
3         print("Nama :", self.nama)
4         print()

```

Listing 3.9 Contoh penggunaan method di python.

4. Cara pemanggilan library kelas, dan contoh kode programnya.

Berikut ini adalah cara pemanggilan library kelas dari instansi dan pemakaianya. Library kelasnya adalah Mahasiswa dari file Mahasiswa.py. Lalu dipanggil dengan import. Kemudian instansi dengan mhs1 dan mhs1, dengan 2 parameter.

```

1 #Jawaban No. 4
2 from Mahasiswa import Mahasiswa
3
4 mhs1 = Mahasiswa("1174006", "Diva Krishna")

```

```

5 mhs2 = Mahasiswa("1174001", "Indah Pramesti")
6
7 mhs1.tampilkanProfil()
8 mhs2.tampilkanProfil()
9
10 print("Total mahasiswa adalah ", Mahasiswa.jumlahMahasiswa)

```

Listing 3.10 Contoh pemanggilan library kelas dari instansi dan pemakaiannya. .

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.

```

1 #Jawaban No. 5
2 from kalkulator import Penambahan
3
4 hasil = Penambahan(10, 5)
5 print(hasil)

```

Listing 3.11 Contoh pemakaian paket dengan perintah from kalkulator import Penambahan.

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.

```

1 #Jawaban No. 6
2 from folder import kalkulator
3
4 a=100
5 b=50
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)

```

Listing 3.12 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 #Jawaban No. 7
2 from folder.Mahasiswa import Mahasiswa
3
4 mhs1 = Mahasiswa("1174006", "Diva Krishna")
5 mhs2 = Mahasiswa("1174001", "Indah Pramesti")
6
7 mhs1.tampilkanProfil()

```

```

8 mhs2.tampilkanProfil()
9
10 print("Total mahasiswa adalah ", Mahasiswa.jumlahMahasiswa)

```

Listing 3.13 Contoh kode pemakaian paket kelas dimana file library ada di dalam folder.

3.3.2 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": "#"
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
25     hasil1 = []
26     hasil2 = []
27     hasil3 = []
28     hasil4 = []
29     hasil5 = []
30     hasil6 = []
31
32     for x in npm:
33         hasil1.append(angka1[x])
34         hasil2.append(angka2[x])
35         hasil3.append(angka3[x])
36         hasil4.append(angka4[x])
37         hasil5.append(angka5[x])
38         hasil6.append(angka6[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=' ')

```

```
35 printNPM(input("Masukan NPM anda: "))
```

Listing 3.14 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 3.15 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 3.16 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 3.17 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)
6
7 printNPMSatuPersatu(input("Masukan NPM anda: "))
```

Listing 3.18 Jawaban soal No. 5 Ketrampilan Pemrograman.

6. Jawaban soal No. 6

```

1 #Jawaban No. 6
2 def printNPMPenjumlahan(npm):
3     npm = list(map(int, npm))
4     hasil = 0
5     for x in npm:
6         hasil += x
7
8     print(hasil)
9
10 printNPMPenjumlahan(input("Masukan NPM anda: "))

```

Listing 3.19 Jawaban soal No. 6 Ketrampilan Pemrograman.

7. Jawaban soal No. 7

```

1 #Jawaban No. 7
2 def printNPMPerkalian(npm):
3     npm = list(map(int, npm))
4     hasil = 0
5     for x in npm:
6         hasil *= x
7
8     print(hasil)
9
10 printNPMPerkalian(input("Masukan NPM anda: "))

```

Listing 3.20 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 3.21 Jawaban soal No. 8 Ketrampilan Pemrograman.

9. Jawaban soal No. 9

```

1 #Jawaban No. 9
2 def printNPMDigitGanjil(npm):
3     npm = list(map(int, npm))
4     for n in npm:
5         if(n % 2 != 0):
6             print(n, end = " ")
7
8 printNPMDigitGanjil(input("Masukan NPM anda: "))

```

Listing 3.22 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 3.23 Jawaban soal No. 10 Ketampilan Pemrograman.

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.printNMPPerkalian(npm)
12 lib.printNPMDijitGenap(npm)
13 lib.printNPMDijitGanjil(npm)
14 lib.printNPMDijitPrima(npm)

```

Listing 3.24 Jawaban soal No. 11 Ketampilan Pemrograman.

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.printNMPPerkalian()

```

```

14 k3lib.printNPMDigitGenap()
15 k3lib.printNPMDigitGanjil()
16 k3lib.printNPMDigitPrima()

```

Listing 3.25 Jawaban soal No. 12 Ketrampilan Pemrograman.

3.3.3 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 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 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 #Jawaban No. 1
2 def sapa(nama):
3     try :
4         print("Hallo , "+str(nama))
5     except:
6         print("Terjadi error")
7
8 sapa(input("Masukan nama anda : "))

```

Listing 3.26 Contoh fungsi yang menggunakan try except .

