

CERDAS MENGUASAI PYTHON

CERDAS MENGUASAI PYTHON

Dalam 24 Jam

Rolly M. Awangga
Informatics Research Center



Kreatif Industri Nusantara

Penulis:

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:

M. Yusril Helmi Setyawan

Penyunting:

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

Desain sampul dan Tata letak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

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

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

CONTRIBUTORS

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

CONTENTS IN BRIEF

1	SEJARAH DAN KARAKTERISTIK PYTHON	1
2	Judul Bagian Kedua	35
	BAGIAN I KETERAMPILAN PEMROGRAMAN	
	BAGIAN II KETRAMPILAN PENANGANAN ERROR	
	BAGIAN III KETERAMPILAN PEMROGRAMAN	
	BAGIAN IV KETERAMPILAN PENANGANAN ERROR	
3	Judul Bagian Ketiga	91

DAFTAR ISI

Daftar Gambar	xi
Daftar Tabel	xiii
Foreword	xvii
Kata Pengantar	xix
Acknowledgments	xxi
Acronyms	xxiii
Glossary	xxv
List of Symbols	xxvii
Introduction	xxix
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
1 SEJARAH DAN KARAKTERISTIK PYTHON	1
1.1 Chandra Kirana Poetra	1
1.1.1 Sejarah python	1
1.1.2 Instalasi Anaconda	1
1.1.3 Spyder	2

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

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

2 Judul Bagian Kedua 35

2.1	Chandra Kirana Poetra	35
2.1.1	Teori	35
2.1.2	Praktek	38

2.1.3	Keterampilan dan penanganan error	40
2.2	Chapter 2 D. Irga B. Naufal Fakhri D4 TI 2C	40
2.2.1	Teori Praktikum	40
2.2.2	Ketrampilan Pemrograman	46
2.2.3	Ketrampilan Penanganan Error	49
2.3	Bakti Qilan Mufid	50
2.3.1	Teori	50
2.4	Fanny Shafira Damayanti	55
2.4.1	Teori	55
2.4.2	Jenis-Jenis Variable	55
2.4.3	Cara Menampilkan Kode untuk meminta Input dan output nya	56
2.4.4	Operator Dasar Aritmatika	56
2.4.5	Sintax Perulangan	57
2.4.6	Syntax Kondisi	57
2.4.7	Jenis error di Python	58
2.4.8	Cara menggunakan Try Except	58
2.5	Keterampilan Pemrograman	59
2.6	Keterampilan Penanganan Error	60
2.7	Tugas 2 Alfadian Owen	61
2.7.1	Teori	61
2.8	Muhammad Abdul Gani Wijaya	61
2.8.1	Variable Phyton	61
2.8.2	Input Phyton	62
2.8.3	Operator Aritmatika Phyton	62
2.7.2	Keterampilan Pemrograman	65
2.8.4	Pengulangan Phyton	70
2.8.5	Kondisi Phyton	70
2.8.6	Try Except Phyton	71
2.9	Keterampilan Pemograman	71
2.9.1	Program	71
2.10	Keterampilan Penanganan Error	72
2.10.1	Program	72
2.11	Tia Nur Candida	72
2.11.1	Jenis Variabel pada Python	72
2.11.2	Cara Menampilkan Output ke Layar	73
2.11.3	Operator Dasar Aritmatika	74
2.11.4	Jenis dan Syntax Perulangan	74

2.11.5	Cara Memakai Syntax untuk Memilih Kondisi	74
2.11.6	Jenis Error yang sering ditemui	75
2.11.7	TryExcept	75
2.12	Nurul Izza Hamka	76
2.12.1	Jenis-Jenis Variabel dan Pemakaian Variable di Kode Pyhton :	76
2.12.2	Tipe String	76
2.12.3	Tipe List	76
2.12.4	Tipe Tuple	77
2.13	Kode Untuk Meminta Input Dari User dan Bagaimana Melakukan Output Ke Layar Pada Python ?	77
2.14	Operator Dasar Aritmatika, Tambah, Kali, Kurang Bagi dan Bagaimana Mengubah String Ke Integer dan Integer Ke String ?	78
2.15	Sintak Perulangan , Jenis-Jenisnya Contoh dan Cara Pakainya Di Python.	78
2.16	Cara Pakai Sintak Untuk Memilih Kondisi dan Bagaimana Contoh Sintak Kondisi Di Dalam Kondisi	79
2.17	Jenis Error Yang Sering Di Temui Di Python Dalam Mengerjakan Sintak Diatas , dan Bagaimana Cara Mengatasinya ?	79
2.18	Cara Memakai Try Except	80

BAGIAN I KETERAMPILAN PEMROGRAMAN

BAGIAN II KETRAMPILAN PENANGANAN ERROR

2.19	Aulyardha Anindita	5
2.20	Jenis-jenis Variabel	5
2.20.1	Bilangan (Number)	5
2.20.2	String	6
2.20.3	List	6
2.20.4	Tuple	6
2.20.5	Set	7
2.20.6	Dictionary	7
2.21	Cara Meminta Input Kepada User dan Bagaimana Outputnya	8
2.21.1	Input	8
2.21.2	Output	8
2.22	Operator Dasar Aritmetika dan Cara Mengubah String ke Int Begitu Juga Sebaliknya	8
2.22.1	Operator Dasar Aritmetika	8

2.22.2	Cara Mengubah String Ke Int Sebaliknya	9
2.23	Perulangan	9
2.23.1	While loop	9
2.23.2	For Loop	10
2.24	Kondisi	10
2.24.1	If	10
2.24.2	If else	10
2.24.3	Elif	11
2.25	Jenis Error	11
2.25.1	Syntax errors	11
2.25.2	Runtime errors	11
2.25.3	Semantic errors	12
2.25.4	Debugging	12
2.25.5	Exceptions	12
2.26	Cara Memakai Try Except	12

BAGIAN III KETERAMPILAN PEMROGRAMAN

BAGIAN IV KETERAMPILAN PENANGANAN ERROR

2.27	Difa Al Fansha	20
2.27.1	Teori	20
2.28	Ketrampilan Pemrograman	22
2.29	Keterampilan Penanganan Error	24
2.30	Ilham Muhammad Ariq	24
2.30.1	Teori	24
2.30.2	Ketrampilan Pemrograman	28
2.30.3	Keterampilan Penanganan Error	30
2.31	Muhammad Reza Syachrani / 1174084	30
2.31.1	Teori	30
2.31.2	Keterampilan Pemrograman	32
2.31.3	Ketrampilan Penanganan Error	34
2.32	Handi Hermawan	35
2.32.1	Teori	35
2.32.2	Keterampilan Pemrograman	35
2.32.3	Keterampilan dan penanganan error	37
2.33	Mochamad Arifqi Ramadhan	37
2.33.1	Variable, pemakaian Variable dan Jenis-Jenis Type data	37
2.33.2	Ketrampilan Pemrograman	39

2.33.3	Ketrampilan Penanganan Error	41
2.34	Fernando Lorencius Sihite	41
2.34.1	Variable, pemakaian Variable dan Jenis-Jenis Type data	41
2.35	Ketrampilan Pemrograman	47
2.36	Ketrampilan Penanganan Error	49
2.37	Arrizal Furqona Gifary	49
2.37.1	Teori	49
2.37.2	Praktek	51
2.37.3	Keterampilan dan penanganan error	53
2.38	Alvan Alvanzah/1174077	54
2.38.1	Teori	54
2.38.2	Ketrampilan Pemrograman	56
2.38.3	Ketrampilan Penanganan Error	59
2.39	Kaka Kamaludin	60
2.40	Praktek	63
2.41	Praktek	64
2.42	Advent Nopele Olansi Damiahan Sihite	65
2.42.1	Variable, pemakaian Variable dan Jenis-Jenis Type data	65
2.42.2	Ketrampilan Pemrograman	68
2.42.3	Ketrampilan Penanganan Error	70
2.43	Advent nopele olansi damiahan sihite	70
2.43.1	Teori	70
2.43.2	Praktek	72
2.43.3	Keterampilan dan penanganan error	74
2.44	Ainul Filiaini	74
2.44.1	Keterampilan pemrograman	80
2.45	Sekar Jasmine	82
2.45.1	Keterampilan Pemrograman	84
2.45.2	Dini Permata Putri	84
2.46	sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut dikode phyton	84
2.47	tuliskan bagaimana kode untuk meminta input dari user dan tuliskan bagaimana melakukan output ke layar	86
2.48	tuliskan operator dasar aritmatika, tambah, kali, kurang bagi, dan bagaimana mengubah string ke integer dan integer ke string	86
2.49	tuliskan dan jelaskan sintak untuk perulangan, jenis-jenisnya contoh kode dan cara pakainya di phyton	87

2.50	tuliskan dan jelaskan cara pakai sintak untuk memilih kondisi, dan bagaimana contoh sintak kondisi didalam kondisi	87
2.51	tuliskan apa saja jenis error yang sering ditemui di python dalam mengerjakan sintak diatas dan bagaimana cara mengatasinya	88
2.52	tuliskan dan jelaskan cara memakai Try Except	88
3	Judul Bagian Ketiga	91
Index		93

DAFTAR GAMBAR

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

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

DAFTAR TABEL

Listings

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

[illegible]

src/1174066.py	49
src/2err_1174066.py	49
src/1174083.py	53
src/1174083.py	53
src/1174083.py	53
src/1174083.py	54
src/1174083.py	54
src/1174083.py	54
src/1174083.py	54
src/1174083.py	54
src/1174083.py	54
src/1174083.py	54
src/1174083.py	54
src/1174083_2err.py	55
src/1174069.py	59
src/1174069.py	59
src/1174069.py	59
src/1174069.py	59
src/1174069.py	60
src/1174069.py	60
src/1174069.py	60
src/1174069.py	60
src/1174069.py	60
src/1174069.py	60
src/1174069.py	60
src/err2_1174069.py	60
src/1174071.py	71
src/Gani2.py	72
src/1174062.py	1
src/1174062.py	1
src/1174062.py	2
src/1174062.py	2
src/1174062.py	2
src/1174062.py	2
src/1174062.py	2
src/1174062.py	3
src/1174062.py	3
src/1174062.py	3

src/1174054.py	15
src/1174054.py	16
src/1174054.py	16
src/1174054.py	16
src/1174054.py	16
src/1174054.py	16
src/1174054.py	16
src/1174054.py	17
src/1174054.py	17
src/1174054.py	17
src/1174054.py	17
src/2err_1174054.py	19
src/1174076.py	22
src/1174076.py	22
src/1174076.py	22
src/1174076.py	22
src/1174076.py	23
src/1174076.py	23
src/1174076.py	23
src/1174076.py	23
src/1174076.py	23
src/1174076.py	23
src/1174076.py	23
src/1174087.py	24
src/1174087.py	25
src/1174087.py	25
src/1174087.py	25
src/1174087.py	26
src/1174087.py	26
src/1174087.py	26
src/1174087.py	28
src/1174087.py	28
src/1174087.py	28
src/1174087.py	28
src/1174087.py	29
src/1174087.py	29
src/1174087.py	29
src/1174087.py	29

src/1174087.py	29
src/1174087.py	29
src/1174087.py	29
src/1174087.py	30
src/2erry_1174087.py	30
src/1174084.py	31
src/1174084.py	31
src/1174084.py	31
src/1174084.py	31
src/1174084.py	31
src/1174084.py	31
src/1174084.py	32
src/1174084.py	32
src/1174084.py	32
src/1174084.py	32
src/1174084.py	33
src/1174084.py	33
src/1174084.py	33
src/1174084.py	33
src/1174084.py	33
src/1174084.py	34
src/1174084.py	34
src/1174084.py	34
src/1174084.py	34
src/1174084.py	34
src/2err_1174084.py	34
src/1174074.py	39
src/1174074.py	40
src/1174074.py	40
src/1174074.py	40
src/1174074.py	40
src/1174074.py	40
src/1174074.py	40
src/1174074.py	40
src/1174074.py	40
src/1174074.py	41
src/1174074.py	41
src/1174074(2err).py	41
src/1174072.py	47

src/1174072.py	47
src/1174072.py	47
src/1174072.py	47
src/1174072.py	47
src/1174072.py	48
src/1174072.py	48
src/1174072.py	48
src/1174072.py	48
src/1174072.py	48
src/1174072.py	48
src/2err_1174072.py	49
src/teori.py	49
src/teori.py	50
src/teori.py	50
src/teori.py	50
src/teori.py	51
src/teori_1174070.py	51
src/1174070.py	51
src/1174070.py	52
src/1174070.py	52
src/1174070.py	52
src/1174070.py	52
src/1174070.py	52
src/1174070.py	53
src/1174070.py	53
src/1174070.py	53
src/1174070.py	53
src/1174070.py	53
src/error_1174070.py	53
src/1174077.py	54
src/1174077.py	54
src/1174077.py	54
src/1174077.py	54
src/1174077.py	54
src/1174077.py	55
src/1174077.py	55
src/1174077.py	55
src/1174077.py	55

src/1174077.py	55
src/1174077.py	55
src/1174077.py	56
src/1174077.py	56
src/1174077.py	56
src/1174077.py	57
src/1174077.py	57
src/1174077.py	57
src/1174077.py	57
src/1174077.py	57
src/1174077.py	58
src/1174077.py	58
src/1174077.py	58
src/1174077.py	58
src/1174077.py	58
src/1174077.py	58
src/1174077.py	59
src/2err1174077.py	59
src/1174067_teor1.py	60
src/1174067_teor1.py	60
src/1174067_teor1.py	60
src/1174067_teor1.py	61
src/1174067_teor1.py	62
src/1174067_teor1.py	62
src/1174067_praktek.py	63
src/1174067_praktek.py	63
src/1174067_praktek.py	63
src/1174067_praktek.py	63
src/1174067_praktek.py	64
src/1174067_praktek.py	64
src/1174067_praktek.py	64
src/1174067_praktek.py	64
src/1174067_praktek.py	64
src/1174067_praktek.py	64
src/1174067_2err.py	65
src/1174083.py	68
src/1174083.py	68
src/1174083.py	69

[illegible]

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat
Februari, 2019*

ACKNOWLEDGMENTS

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

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

R. M. A.

ACRONYMS

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

GLOSSARY

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

SYMBOLS

- A Amplitude
- $\&$ Propositional logic symbol
- a Filter Coefficient

- \mathcal{B} Number of Beats

INTRODUCTION

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

Informatics Research Center
Bandung, Jawa Barat, Indonesia

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

Rumusnya cuman sebagai contoh aja biar keren[?].

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

BAB 1

SEJARAH DAN KARAKTERISTIK PYTHON

1.1 Chandra Kirana Poetra

1.1.1 Sejarah python

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

1.1.2 Instalasi Anaconda

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

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

1.1.3 Spyder

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

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

1.2.1 Sejarah Python

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

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

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

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

1.2.2 Tanggal Rilis Python

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

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

1.2.3 Perbedaan Python 2 dengan Python 3

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

- Print

Pada python2, print lebih seperti statement daripada fungsi

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

sedangkan pada python3, print digunakan sebagai fungsi

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

- Pembagian pada Integer

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

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

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

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

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

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

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

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

Fitur pada Python 3 ini tidak bisa digunakan pada Python 2.7

- Dukungan Unicode

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

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

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

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

1.2.4 Penggunaan Python di perusahaan dunia

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

10. Blender, Maya, software pembuat animasi 3D terkenal, menggunakan Python sebagai salah satu bahasa skrip pemrogramannya.
11. Raspberry Pi, komputer mini yang banyak digunakan sebagai mikrokontroller, menggunakan Python sebagai bahasa utamanya.
12. ESRI, produsen terkenal pembuat software pemetaan GIS banyak menggunakan Python di produknya.

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

1.2.5 Cara menginstall Anaconda

1. Pastikan anda telah menginstall python dan anda mengetahui versi dari python yang telah anda install
2. Download Anaconda dari website www.anaconda.com/distribution
3. pilih sesuai dengan versi python anda, jika versi anda python3 maka pilih python3
4. Setelah itu buka file yang telah anda download
5. Setelah muncul gambar dibawah ini, tekan next
6. Baca license agreement lalu tekan 'I Agree'
7. Setelah itu pilih mau diinstall pada user yang sedang anda pakai atau kesemua user, direkomendasikan untuk memilih just me yaitu hanya user yang sedang dipakai saja
8. Catat tempat dimana anda akan menginstall anaconda, lalu tekan 'Next'
9. Setelah itu anda diberi pilihan, direkomendasikan untuk tidak mengubah pilihan tersebut, lalu tekan 'Install'
10. Tunggu sampai instalasi selesai
11. Setelah selesai tekan 'Next'
12. Setelah itu ada opsi untuk memilih untuk meinstall visual studio code, jika anda berminat klik 'Install VSCode' jika tidak tekan 'Skip'
13. Tekan 'Finish' untuk menyelesaikan instalasi

1.2.6 Cara menggunakan Spyder pada Anaconda

Pertama buka aplikasi Anaconda sampai muncul seperti ini

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

1.2.7 Membuat Hello World di Spyder

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

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

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

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

1.3 Bakti Qilan Mufid

1.3.1 Resume Sejarah Python

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

1.3.2 Perbedaan Python 2 dan Python 3

1.3.2.1 Python 2

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

1.3.2.2 Python 3

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

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

1.3.3 Implementasi dan penggunaan Python pada Perusahaan

daftar berikut adalah beberapa perusahaan yang menggunakan Python, diantaranya:

1. Google adalah perusahaan besar yang menggunakan banyak kode Python di dalam mesin pencariannya. Dan mesin pencari google adalah yang paling terkenal di dunia.
2. Youtube, situs video terbesar dan terpopuler di dunia, sebagian besar kodenya ditulis dalam bahasa Python.
3. Facebook, media sosial terbesar di dunia, menggunakan Tornado, sebuah framework Python untuk menampilkan timeline.
4. Instagram, siapa yang tidak kenal. Instagram menggunakan Django, framework python sebagai mesin pengolah sisi server dari aplikasinya.
5. Pinterest, banyak menggunakan python untuk membangun aplikasinya.
6. Dropbox, barangkali Anda adalah salah seorang pengguna layanan ini. Dropbox menggunakan python baik di sisi server maupun di sisi pengguna layanannya.
7. Quora, salah satu situs tanya jawab terbesar di dunia, dibangun menggunakan Python.
8. NASA, badan antariksa Amerika ini menggunakan Python untuk bidang sainsnya.
9. NSA, badan mata – mata Amerika banyak menggunakan Python untuk analisa kriptografi dan intelijen.
10. Blender, Maya, software pembuat animasi 3D terkenal, menggunakan Python sebagai salah satu bahasa skrip pemrogramannya.
11. Raspberry Pi, komputer mini yang banyak digunakan sebagai mikrokontroller, menggunakan Python sebagai bahasa utamanya.

1.4 Instalasi

1.4.1 Cara Pemakaian Script dan interpreter python

1.4.2 Cara Pemakaian spyder termasuk variable explorer

1.5 Mencoba Python

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

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

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

```
Hello world...
```

1.6 Identasi

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

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

    print("Harap memasukkan argumen.")

    sys.exit(1)
```

1.7 Resume | Kaka Kamaludin

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

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

1.8 Instalasi

1.8.1 Install Anacanda

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

1.8.2 Install Python

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

1.8.3 Install Python pip

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

1.9 MuhammadRezaSyachrani/1174084

1.9.1 Background

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

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

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Sisi utama yang membedakan Python dengan bahasa pemrograman lain adalah dalam hal aturan penulisan kode program. Programmer yang menggunakan selain python dapat dibingungkan dengan aturan indentasi, tipe data, tuple, dan dictionary. Python memiliki kelebihan tersendiri dibandingkan dengan bahasa lain terutama dalam hal penanganan modul, ini yang membuat beberapa programmer menyukai python. Selain itu python merupakan salah satu produk yang opensource, free, dan multiplatform.

1.9.2 Problems

- Bagaimana cara mengimplementasikan bahasa pemrograman python

1.9.3 Objective and Contribution

1.9.3.1 Objective

- Dapat mengimplementasikan bahasa pemrograman python

1.9.3.2 Contribution

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

1.9.4 Scoop and Environment

- Mengimplementasikan Python dalam pemrograman

1.10 Instalasi

1.10.1 Cara Pemakaian Script dan interpreter python

1.10.2 Cara Pemakaian spyder termasuk variable explorer

1.11 Mencoba Python

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

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

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

```
Hello world...
```

1.12 Identasi

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

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

    print("Harap memasukkan argumen.")

    sys.exit(1)
```

1.13 AlvanAlvanzah/1174077

1.13.1 Background

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

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

1.13.2 Problems

- Bagaimana cara agar memahami bahasa pemrograman python

1.13.3 Objective and Contribution

1.13.3.1 Objective

- Dapat memahami bahasa pemrograman Python

1.13.3.2 Contribution

- Dapat mengimplementasikan bahasa pemrograman python

1.13.4 Scoop and Environment

- Mempelajari tentang bahasa pemrograman python

1.14 Ilham Muhammad Ariq

1.14.1 Mengenal Python dan Anaconda

1. SEJARAH PYTHON

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

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

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

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

2. PERBEDAAN PYTHON 2 DAN 3

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

1.15 Cara Pemakaian Python

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

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

1.16 Instalasi Python

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

1.17 Indentasi

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

1.18 Difa Al Fansha

1.18.1 Resume Sejarah Python

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

1.18.2 Perbedaan Python 2 dan Python 3

1.18.2.1 *Python 2*

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

1.18.2.2 *Python 3*

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

1.18.3 Implementasi dan penggunaan Python pada Perusahaan

daftar berikut adalah beberapa perusahaan yang menggunakan Python, diantaranya:

1. Google adalah perusahaan besar yang menggunakan banyak kode Python di dalam mesin pencariinya. Dan mesin pencari google adalah yang paling terkenal di dunia.
2. Youtube, situs video terbesar dan terpopuler di dunia, sebagian besar kodenya ditulis dalam bahasa Python.
3. Facebook, media sosial terbesar di dunia, menggunakan Tornado, sebuah framework Python untuk menampilkan timeline.
4. Instagram, siapa yang tidak kenal. Instagram menggunakan Django, framework python sebagai mesin pengolah sisi server dari aplikasinya.
5. Pinterest, banyak menggunakan python untuk membangun aplikasinya.

6. Dropbox, barangkali Anda adalah salah seorang pengguna layanan ini. Dropbox menggunakan python baik di sisi server maupun di sisi pengguna layanannya.
7. Quora, salah satu situs tanya jawab terbesar di dunia, dibangun menggunakan Python.
8. NASA, badan antariksa Amerika ini menggunakan Python untuk bidang sainsnya.
9. NSA, badan mata – mata Amerika banyak menggunakan Python untuk analisa kriptografi dan intelijen.
10. Blender, Maya, software pembuat animasi 3D terkenal, menggunakan Python sebagai salah satu bahasa skrip pemrogramannya.
11. Raspberry Pi, komputer mini yang banyak digunakan sebagai mikrokontroler, menggunakan Python sebagai bahasa utamanya.

1.19 Instalasi

1.19.1 Cara Pemakaian Script dan interpreter python

1.19.2 Cara Pemakaian spyder termasuk variable explorer

1.20 Mencoba Python

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

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

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

```
Hello world...
```

1.21 Identasi

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

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

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

    print("Harap memasukkan argumen.")

    sys.exit(1)
```

1.22 Nurul Izza Hamka

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

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

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

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

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

Adapun perbedaan dari segi `print`:

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

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

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

1.23 Instalasi

1.23.0.1 Instalasi Python Cara Install Python di Windows

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

1.23.0.2 Instalasi Anaconda Cara Install Python di Windows

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

1.23.0.3 Script dan Interpreter Python Menggunakan Interpreter Python:

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

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

seperti :

- Integers
- Floats
- Complex numbers
- Strings

1.24 Mencoba Python

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

1.25 Identitas

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

1.26 Aulyardha Anindita

1.26.1 Sejarah Python

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

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

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

1.26.2 Perbedaan Python 2 dan Python 3

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

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

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

1.26.3 Implementasi dan Penggunaan Python di Perusahaan Dunia

1. Google adalah perusahaan besar yang menggunakan banyak kode Python di dalam mesin pencariinya. Dan mesin pencari google adalah yang paling terkenal di dunia.
2. Youtube, situs video terbesar dan terpopuler di dunia, sebagian besar kodenya ditulis dalam bahasa Python.
3. Facebook, media sosial terbesar di dunia, menggunakan Tornado, sebuah framework Python untuk menampilkan timeline.
4. Instagram, siapa yang tidak kenal. Instagram menggunakan Django, framework python sebagai mesin pengolah sisi server dari aplikasinya.
5. Pinterest, banyak menggunakan python untuk membangun aplikasinya.
6. Dropbox, barangkali Anda adalah salah seorang pengguna layanan ini. Dropbox menggunakan python baik di sisi server maupun di sisi pengguna layanannya.
7. Quora, salah satu situs tanya jawab terbesar di dunia, dibangun menggunakan Python.
8. NASA, badan antariksa Amerika ini menggunakan Python untuk bidang sainsnya.
9. NSA, badan mata – mata Amerika banyak menggunakan Python untuk analisa kriptografi dan intelijen.
10. Pixar, banyak menggunakan Python dalam animasi movie.
11. Blender, Maya, software pembuat animasi 3D terkenal, menggunakan Python sebagai salah satu bahasa skrip pemrogramannya.
12. Raspberry Pi, komputer mini yang banyak digunakan sebagai mikrokontroller, menggunakan Python sebagai bahasa utamanya.
13. ESRI, produsen terkenal pembuat software pemetaan GIS banyak menggunakan Python di produknya.

1.26.4 Instalasi

1.26.5 Proses Instalasi Python

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

1.26.6 Proses Instalasi Anaconda

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

1.26.7 Cara Pemakaian Script dan Interpreter Python

1.26.8 Script

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

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

1.26.9 Interpreter Python

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

1.26.10 Cara Pemakaian Spyder Termasuk Variable Explorer

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

1.26.11 Mencoba Phyton

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

1.26.12 Identasi

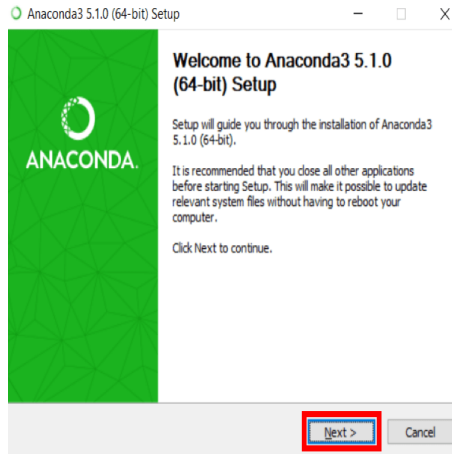
Indentasi merupakan keluarnya suatu teks/naskah dari batas kiri, batas kanan atau keduanya. Kita dapat mengatur indentasi hanya pada baris pertama (first line indent) dari paragraf atau mengatur indentasi mengandung (hanging).

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

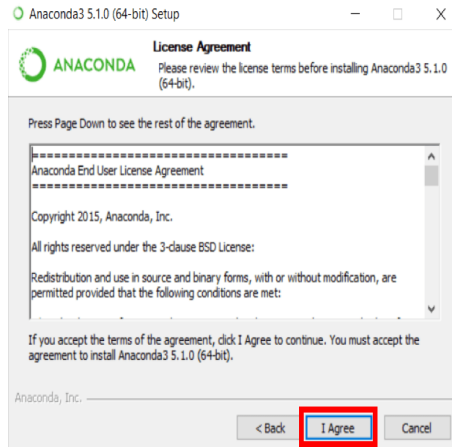
1.27 Dini Permata Putri 1174053

1.27.1 Background

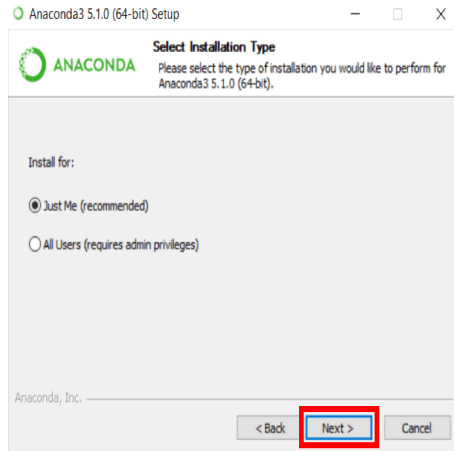
Bahasa pemrograman saat ini jumlahnya sangat banyak. Python merupakan salah satu bahasa pemrograman populer yang digunakan oleh banyak developer. Menurut survei bahasa pemrograman versi www.tiobe.com, Python berada di peringkat ke-5 pada tahun 2016. Selain itu, Python juga bisa digunakan untuk enterprise.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



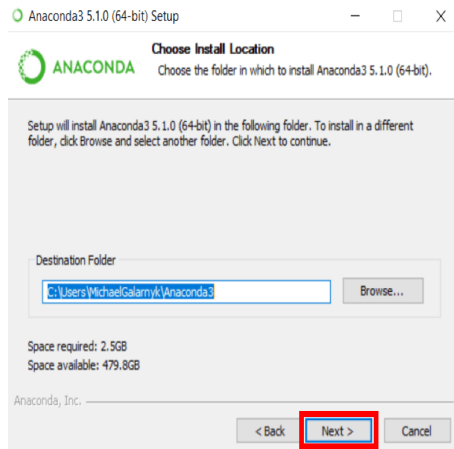
Gambar 1.1 Klik Next



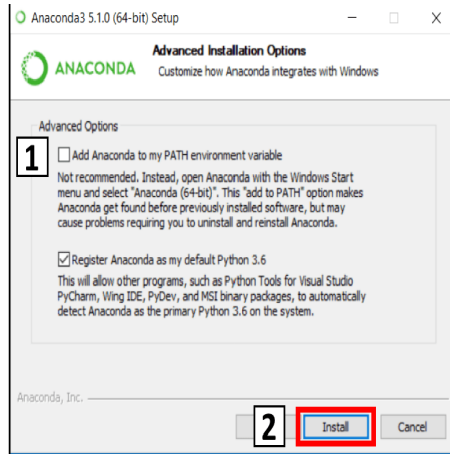
Gambar 1.2 Klik I Agree



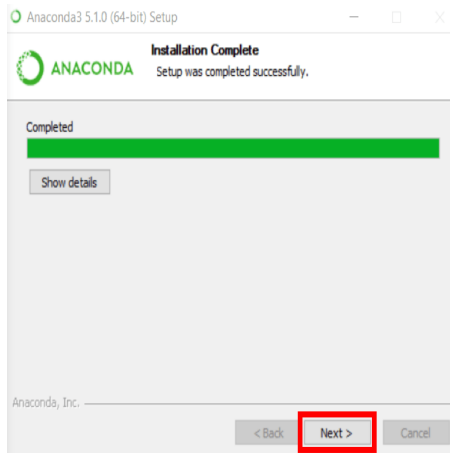
Gambar 1.3 Pilih just me saja



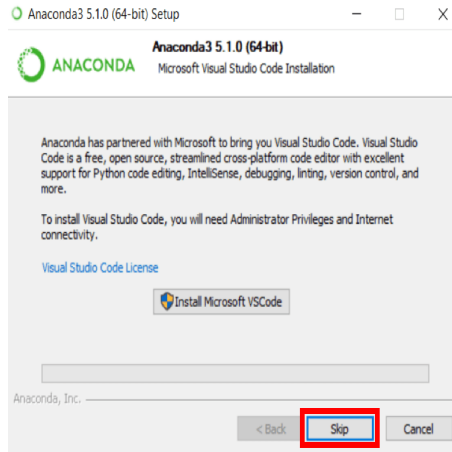
Gambar 1.4 Directory tempat anaconda akan diinstall



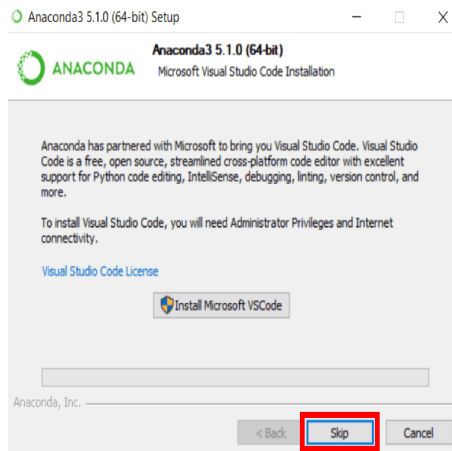
Gambar 1.5 Opsi Register



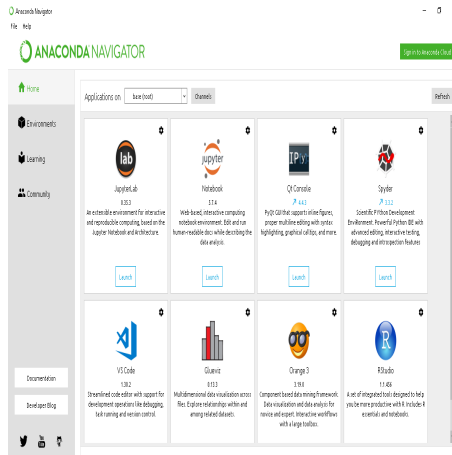
Gambar 1.6 Tunggu hingga selesai



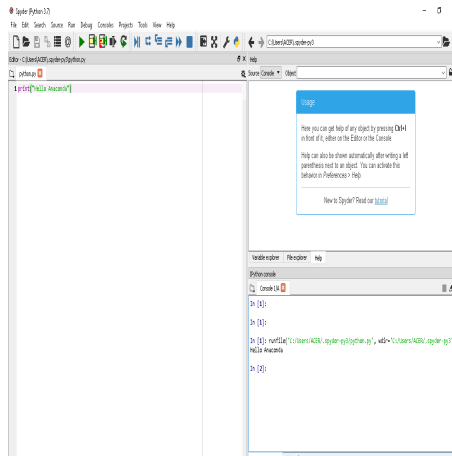
Gambar 1.7 Opsi Tambahan



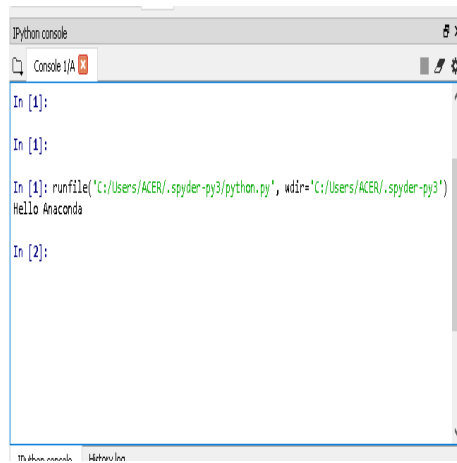
Gambar 1.8 Finish



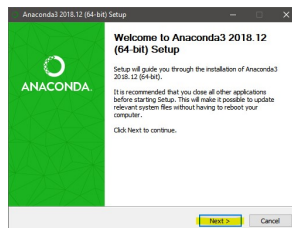
Gambar 1.9 Tampilan awal anaconda



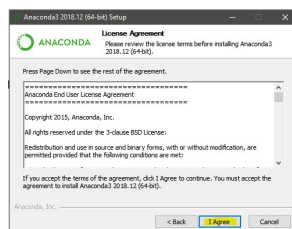
Gambar 1.10 IDE Spyder



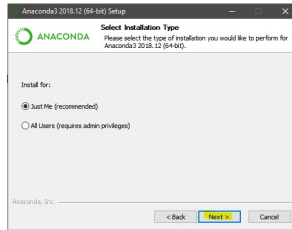
Gambar 1.11 Menu Console



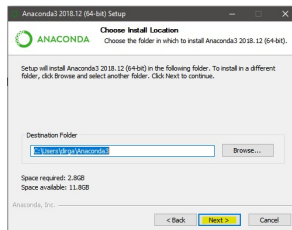
Gambar 1.12 Tampilan Instalasi 1



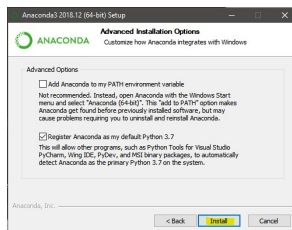
Gambar 1.13 Tampilan Instalasi 2



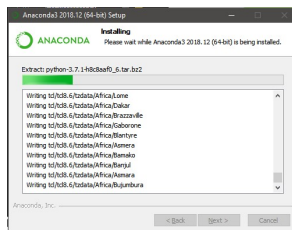
Gambar 1.14 Tampilan Instalasi 3



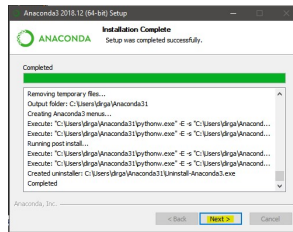
Gambar 1.15 Tampilan Instalasi 4



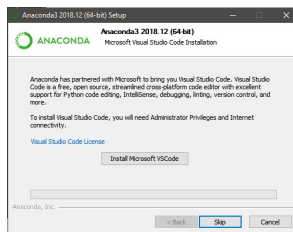
Gambar 1.16 Tampilan Instalasi 5



Gambar 1.17 Tampilan Instalasi 6



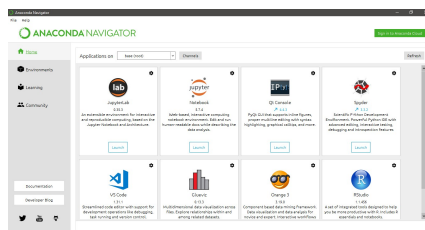
Gambar 1.18 Tampilan Instalasi 7



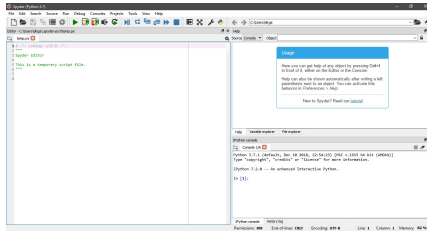
Gambar 1.19 Tampilan Instalasi 8



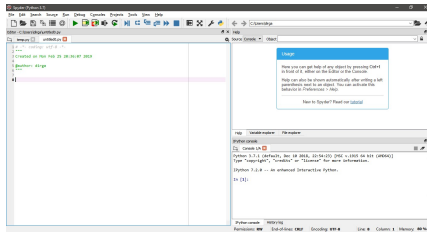
Gambar 1.20 Tampilan Instalasi 9



Gambar 1.21 Tampilan awal Anaconda



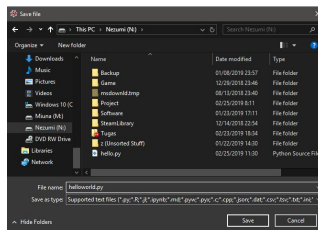
Gambar 1.22 Tampilan spider



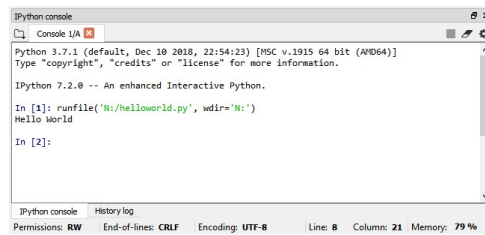
Gambar 1.23 Tampilan new file pada spider

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

Gambar 1.24 `print("Hello World")`



Gambar 1.25 Dialog simpan file



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

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

IPython 7.2.0 -- An enhanced Interactive Python.

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

In [2]:
```

At the bottom of the window, there is a status bar with the following information:

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

Gambar 1.26 Hasil Program

BAB 2

JUDUL BAGIAN KEDUA

2.1 Chandra Kirana Poetra

2.1.1 Teori

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

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

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

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

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

2.1.2 Praktek

1. Jawaban soal no 1

```

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

```

2. Jawaban soal no 2

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

3. Jawaban soal no 3

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

4. Jawaban soal no 4

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

5. Jawaban soal no 5

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

6. Jawaban soal no 6

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

7. Jawaban soal no 7

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

8. Jawaban soal no 8

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

9. Jawaban soal no 9

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

10. Jawaban soal no 10

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

11. Jawaban soal no 11

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

2.1.3 Keterampilan dan penanganan error

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

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

2.2.1 Teori Praktikum

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

(a) Boolean

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

Listing 2.1 Contoh kode variable Boolean.

(b) String

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

Listing 2.2 Contoh kode variable String.

(c) Integer

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

Listing 2.3 Contoh kode variable Integer.

(d) Float

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

Listing 2.4 Contoh kode variable Float.

(e) Hexadecimal

```

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

```

Listing 2.5 Contoh kode variable Hexadecimal.

(f) Complex

```

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

```

Listing 2.6 Contoh kode variable Complex.

(g) List

```

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

```

Listing 2.7 Contoh kode variable List.

(h) Tuple

```

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

```

Listing 2.8 Contoh kode variable Tuple.

(i) Set

```

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

```

Listing 2.9 Contoh kode variable Set.

(j) Dictionary

```

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

```

Listing 2.10 Contoh kode variable Dictionary.

2. Permintaan Input dari user dan Outputnya

```

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

```

Listing 2.11 Contoh kode input dan outputnya.

3. Operator dasar aritmatika dan perubahan tipe data variable

Operator dasar aritmatika

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

```

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

```

Listing 2.12 Contoh kode operasi pertambahan.

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

```

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

```

Listing 2.13 Contoh kode operasi pengurangan.

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

```

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

```

Listing 2.14 Contoh kode operasi perkalian.

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

```

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

```

Listing 2.15 Contoh kode operasi pembagian.

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

```

1 #Modulus
2 var1 = 7
3 var2 = 3
4 hasil = var1 % var2

```

```
5 print(hasil)
```

Listing 2.16 Contoh kode operasi modulus.

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

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

Listing 2.17 Contoh kode operasi perpangkatan.

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

```
1 #Pembulatan Kebawah Pada Hasil Pembagian
2 var1 = 11
3 var2 = 3
4 hasil = var1 // var2
5 print(hasil)
```

Listing 2.18 Contoh kode operasi pembulatan hasil pembagian kebawah.

Perubahan tipe data variable

- (a) String menjadi Integer

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

Listing 2.19 Contoh kode variable string menjadi integer.

- (b) Integer menjadi String

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

Listing 2.20 Contoh kode variable integer menjadi string.

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

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

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

Listing 2.21 Contoh kode penggunaan while loop.

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

```

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

```

Listing 2.22 Contoh kode penggunaan for loop.

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

```

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

```

Listing 2.23 Contoh kode penggunaan nested loop.

5. Sintak kondisi dan penggunaannya.

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

```

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

```

Listing 2.24 Contoh kode penggunaan if.

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

```

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

```

Listing 2.25 Contoh kode penggunaan if else.

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

```

1 #Elif
2 var = 0
3 if var > 0:

```

```

4     print(var, "adalah angka lebih dari 0")
5 elif var < 0:
6     print(var, "adalah angka kurang dari 0")
7 else:
8     print(var, "adalah angka yang sama dengan 0")

```

Listing 2.26 Contoh kode penggunaan elif.

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

```

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

```

Listing 2.27 Contoh kode penggunaan kondisi di dalam kondisi.

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

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

7. Cara pemakaian Try Except.

```

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

```

Listing 2.28 Contoh kode penggunaan try except.

2.2.2 Ketrampilan Pemrograman

1. Jawaban Soal 1

```
1 #For Loop
2 angka = [1,2,3,4,5]
3 for a in angka:
4     print(a)
5
6 #Nested Loop
7 i = 1
8 while(i < 5):
9     j = 1
10    while(j <= 5):
11        print('TEST')
12        j += 1
13    i += 1
14
15
16 #Kondisi
17 #If
18 var = 1
19 if var > 0:
20     print(var, "adalah angka lebih dari 0")
21
22 #If Else
23 var = -1
24 if var > 0:
25     print(var, "adalah angka kurang dari 0")
26 else:
27     print(var, "adalah angka lebih dari 0")
28
29 #Elif
30 var = 0
31 if var > 0:
32     print(var, "adalah angka lebih dari 0")
33 elif var < 0:
34     print(var, "adalah angka kurang dari 0")
35 else:
36     print(var, "adalah angka yang sama dengan 0")
37
38
39 #Kondisi di dalam kondisi
40 if var > 0:
41     print(var, "adalah angka lebih dari 0")
42     if var > 10:
43         print(var, "adalah angka lebih dari 10.")
44     else:
45         print(var, "adalah angka lebih dari 10.")
46 elif var < 0:
47     print(var, "adalah angka kurang dari 0")
48 else:
49     print(var, "adalah angka yang sama dengan 0")
```

```

50
51
52 #Try Except
53 try:
54     print(tidakada)
55 except:
56     print("Terjadi kesalahan penulisan kode")
57 finally:
58     print("Try except telah selesai")
59
60
61 #Jawaban No. 1
62
63 print(1174066%3)
64 #mod3 = 1
65
66 print(" ## ## #####      ## ##      #####")
67 print("#### #####      ## ##      ##")
68 print("### ##      ## ## ## ## ##")
69 print("### ##      ##### ## ##      #####")
70 print("### ##      ## ## ## ## ##")
71 print("### ##      ## #####")
72 #Cara kedua Penulisan NPM tanda pagar menurut https://
73     stackoverflow.com/questions/9632995/how-to-easily-print-ascii
74     -art-text
75 text = "1174066"
76 from PIL import Image, ImageDraw, ImageFont
77 import numpy as np
78 myfont = ImageFont.truetype("verdanab.ttf", 12)
79 size = myfont.getsize(text)
80 img = Image.new("1",size,"black")
81 draw = ImageDraw.Draw(img)
82 draw.text((0, 0), text, "white", font=myfont)
83 pixels = np.array(img, dtype=np.uint8)
84 chars = np.array([' ', '#'], dtype="UI")[pixels]
85 strings = chars.view('U' + str(chars.shape[1])).flatten()
86 print("\n".join(strings))

```

2. Jawaban Soal 2

```

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

```

3. Jawaban Soal 3

```

1 #Jawaban No. 3
2
3 npm = input("Input : ")
4 i = 1
5 akhir = int(npm[4:5]) + int(npm[5:6]) + int(npm[6:7])
6 while(i <= akhir):

```

```

7     print("Hallo , "+str(npm[4:7])+" apa kabar?")
8     i += 1

```

4. Jawaban Soal 4

```

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

```

5. Jawaban Soal 5

```

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

```

6. Jawaban Soal 6

```

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

```

7. Jawaban Soal 7

```

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

```

8. Jawaban Soal 8

```

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

```

9. Jawaban Soal 9

```

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

```

10. Jawaban Soal 10

```

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

```

11. Jawaban Soal 11

```

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

```

2.2.3 Ketrampilan Penanganan Error

1. Jawaban Soal No. 1

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

2. Jawaban Soal No. 2

```

1 a = "1"
2 b = 1
3
4 try :

```

```
5     print(a+b)
6 except TypeError:
7     print("Berbeda type datanya sehingga tidak bisa dijumlahkan")
```

2.3 Bakti Qilan Mufid

2.3.1 Teori

Praktek teori penunjang yang dikerjakan :

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

1. Pemakaian Variabel

Variabel di python dapat dibuat dengan format seperti ini:

NamaVariabel = (nilai)

Contoh:

VariabelKu = "ini isi variabel"

variabel2 = 20

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

print VariabelKu

print variabel2

(a) Aturan Penulisan Variabel

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

(b) Tipe Data

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

(c) Jenis-Jenis Tipe Data

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

2. Meminta input dan melakukan output

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

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

3. Operator dan Konvert

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

4. Perulangan di Python

- Perulangan for
contoh :
ulang = 2
for i in range(ulang):
 print ("Perulangan ke-" + str(i))

Hasil :

Perulangan ke-0

Perulangan ke-1

- Perulangan While
contoh :
jawab = 'ya'
hitung = 0
while(jawab == 'ya'):

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

5. Kondisi di Python

(a) Kondisi **If**

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

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

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

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

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

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

6. Error yang sering dialami

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

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

7. Cara memakai Try Except

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

contoh :

try:

```
print(x)
```

except:

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

2.3.1.2 Ketrampilan Pemrograman Buat program di python dengan ketentuan :

1. Jawaban

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

2. Jawaban

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

3. Jawaban

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

4. Jawaban


```

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

```

5. Jawaban

```

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

```

6. Jawaban

```

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

```

7. Jawaban

```

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

```

8. Jawaban

```

1 for n in npm:
2     print(n)

```

9. Jawaban

```

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

```

10. Jawaban

```

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

```

11. Jawaban

```

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

```

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

1. Jawaban

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

2. Jawaban

```

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

```

2.4 Fanny Shafira Damayanti

2.4.1 Teori

2.4.2 Jenis-Jenis Variable

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

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

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

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

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

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

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

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

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

2.4.3 Cara Menampilkan Kode untuk meminta Input dan output nya

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

2.4.4 Operator Dasar Aritmatika

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

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

Integer = int(a) untuk konversi string ke integer

String = str(a) untuk konversi integer ke string

2.4.5 Sintax Perulangan

1. The while Loop

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

Contoh : Print i sepanjang i adalah kurang dari 6:

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

2. Python For Loops

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

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

Contoh :

Print each fruit in a fruit list:

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

2.4.6 Syntax Kondisi

Kondisi If Else

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

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

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

```
nilai = 3
```

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

```
    print ("Selamat anda lulus")
```

```
else :
```

```
    print ("Maaf anda tidak lulus")
```

2.4.7 Jenis error di Python

1. Syntax Errors

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

```
print "Hello World"
```

SyntaxError: invalid syntax

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

untuk menanganinya cukup tambahkan tanda kurung `()` pada :

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

2. Exceptions

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

2.4.8 Cara menggunakan Try Except

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

contoh :

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

```
try:
```

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

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

```
except IOError, e:
```

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

2.4.9 Keterampilan Pemrograman

1. Jawaban Soal 1

```

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

```

2. Jawaban Soal 2

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

3. Jawaban Soal 3

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

4. Jawaban Soal 4

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

5. Jawaban Soal 5

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

6. Jawaban Soal 6

```
1 print(h)
```

7. Jawaban Soal 7

```
1 print(i)
```

8. Jawaban Soal 8

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

9. Jawaban Soal 9

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

10. Jawaban Soal 10

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

11. Jawaban Soal 11

```
1 print(c)
```

2.4.10 Keterampilan Penanganan Error

1. Jawaban Soal

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

```

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

```

2. input()

```

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

```

Gambar 2.1 input dan outputnya

2.5 Tugas 2 Alfadian Owen

2.5.1 Teori

1. Integer, Boolean, Char, Float, String

2.6 Muhammad Abdul Gani Wijaya

2.6.1 Variable Python

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


```

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

```

3. asd

Gambar 2.2 operasi aritmatika

```

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

```

2.6.2 Input Python

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

```

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

```

2.6.3 Operator Aritmatika Python

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

```

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

```

```

x10 = 5
y10 = 3

```

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

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

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

Gambar 2.3 contoh loop/pengulangan

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

```

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

```

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

Gambar 2.4 contoh syntax kondisi

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

```

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

```

memakainya :

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

```
masukan angka:
```

```
a
```

```
masukan angka ke 2 :
```

```
b
```

```
masukan angka bukan huruf
```

Gambar 2.5 contoh try except

2.5.2 Keterampilan Pemrograman

1. a

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

2.

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

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

```

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

```

3.

```

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

```

In [35]:]|

```

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

```

4.

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

5.

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

6.

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

7.

```

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

```

8.

```

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

```

9.

```

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

```

10.


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

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

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

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

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

2.6.4 Pengulangan Phyton

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

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

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

2.6.5 Kondisi Phyton

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

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

2.6.6 Try Except Phyton

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

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

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

2.6.7 Keterampilan Pemograman

2.6.8 Program

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

```

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

```

2.6.9 Keterampilan Penanganan Error

2.6.10 Program

```

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

```

2.7 Tia Nur Candida

2.7.1 Jenis Variabel pada Python

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

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

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

2.7.2 Cara Menampilkan Output ke Layar

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

2.7.3 Operator Dasar Aritmatika

1. Penjumlahan +

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

2.7.4 Jenis dan Sintax Perulangan

1. While Loop

While loop dapat mengeksekusi statement selama kondisi nya true Contoh: i = 1
 while i ≤ 6:
 print(i)
 i += 1

2. For Loop

For loop digunakan untuk mengulangi urutan (list, tuple, dictionary, set, atau string). Menggunakan for loop dapat mengeksekusi seperangkat statement, satu kali untuk setiap item dalam list, tuple, set dll.
 Contoh : fruits = ["apple", "banana", "cherry"]
 for x in fruits:
 print(x)

2.7.5 Cara Memakai Sintax untuk Memilih Kondisi

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

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

```
nilai = 3
```

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

```
if(nilai ≤ 7):
```

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

2.7.6 Jenis Error yang sering ditemui

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

```
1<<< print"Hello World"
```

2 SyntaxError: invalid syntax

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

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

Berikut beberapa contoh Exceptions:

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

2.7.7 TryExcept

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

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

```
try:
```

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

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

print orang
```

2.8 Nurul Izza Hamka

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

2.8.2 Tipe String

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

Contoh Pemakaian String pada python :

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

Contoh lain Pada penjumlahan yaitu :

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

2.8.3 Tipe List

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

Contoh :

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

2.8.4 Tipe Tuple

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

Contoh :

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

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

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

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

Contoh :

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

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

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

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

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

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

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

Macam-Macam Operator Aritmatika :

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

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

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

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

Cara mengubah Integer ke String :

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

```
a=100
```

konversi integer ke string :

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

Cara mengubah String ke Integer :

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

```
a = '1212'
```

konversi string ke integer

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

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

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

2.8.7.1 Perulangan For ulang = 5

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

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

2.8.7.2 Perulangan While jawab = 'ya'

```
hitung = 0
```

```

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

```

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

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

2.8.8.1 *If* $x = 1$

```

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

```

2.8.8.2 *If-else* If-else

```

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

```

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

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

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

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

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

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

Berikut beberapa contoh Exceptions:

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

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

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

2.8.10 Cara Memakai Try Except

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

Blok `except` memungkinkan Anda menangani kesalahan.

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

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

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

2.8.11 Keterampilan Pemrograman

1. No 1

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

2. No 2

```

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

```

3. No 3

```

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

```

4. No 4

```

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

```

5. No 5

```

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

```

6. No 6

```

1 #         print("We Are Different")

```

7. No 7

```

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

```

8. No 8

```

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

```

9. No 9

```

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

```

10. No 10

```

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

```

2.8.12 Ketrampilan Penanganan Error

2.9 Aulyardha Anindita

2.9.1 Jenis-jenis Variabel

2.9.2 Bilangan (Number)

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

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

Contoh kode python :

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

2.9.3 String

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

Contoh kode python :

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

2.9.4 List

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

Contoh kode python :

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

2.9.5 Tuple

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

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

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

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

```
print(white)
```

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

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

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

2.9.6 Set

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

Contoh Kode Python :

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

```
print(myset)
```

2.9.7 Dictionary

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

Contoh Kode Python :

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

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

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

2.9.8 Cara Meminta Input Kepada User dan Bagaimana Outputnya

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

2.9.9 Input

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

Contoh :

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

Akan menampilkan output :

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

2.9.10 Output

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

Contoh :

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

Hasil :

```
"Hello Aulyardha Anindita"
```

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

2.9.12 Operator Dasar Aritmetika

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

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

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

2.9.13 Cara Mengubah String Ke Int Sebaliknya

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

Contoh Kode :

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

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

Contoh Kode :

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

2.9.14 Perulangan

2.9.15 While loop

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

Contoh Kode :

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

```
break  
i += 1
```

2.9.16 For Loop

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

Contoh Kode :

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

2.9.17 Kondisi

2.9.18 If

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

Contoh Kode :

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

2.9.19 If else

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

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

Contoh :

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

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

2.9.20 Elif

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

Contoh Kode :

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

2.9.21 Jenis Error

2.9.22 Syntax errors

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

2.9.23 Runtime errors

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

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

2.9.24 Semantic errors

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

2.9.25 Debugging

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

2.9.26 Exceptions

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

2.9.27 Cara Memakai Try Except

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

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

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

```

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

```

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

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

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

2.9.28 Keterampilan Pemrograman

1. Jawaban soal no 1

```

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

```

```

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

```

2. Jawaban soal no 2

```

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

```

3. Jawaban soal no 3

```

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

```

4. Jawaban soal no 4

```

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

```

5. Jawaban soal no 5

```

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

```

6. Jawaban soal no 6

```

1 #No 6
2 print(h)

```

7. Jawaban soal no 7

```

1 #No 7
2 print(i)

```

8. Jawaban soal no 8

```

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

```

9. Jawaban soal no 9

```

1 #No 9
2 print(d,g)

```

10. Jawaban soal no 10

```

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

```

11. Jawaban soal no 11

```

1 #No 11
2 print(c,f)

```

2.9.29 Keterampilan Penanganan Error

```

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

```

2.10 Difa Al Fansha

2.10.1 Teori

1. Variabel merupakan tempat menyimpan data Aturan Penulisan Variabel :

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

'Contoh :'

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

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

'Contoh :'

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

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

3. Operator Dasar Aritmatika.

'Contoh :'

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


4. Perulangan

'Contoh :'

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

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

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

'Contoh :'

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

6. Error yang sering ditemui

SyntaxError: invalid syntax

Cara mengatasi

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

7. Ketika sebuah error ditemukan, untuk mengatasi itu, pernyataan try dan except.

Try berfungsi untuk menangkap error, dan except berguna untuk mengatasi error tersebut.

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

```

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

```

2.10.2 Ketrampilan Pemrograman

1. Jawaban Nomor 1

```

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

```

2. Jawaban Nomor 2

```

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

```

3. Jawaban Nomor 3

```

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

```

4. Jawaban Nomor 4

```

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

```

5. Jawaban Nomor 5

```

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

```

```

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

```

6. Jawaban Nomor 6

```

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

```

7. Jawaban Nomor 7

```

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

```

8. Jawaban Nomor 8

```

1 for z in Npm:
2     print(z)

```

9. Jawaban Nomor 9

```

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

```

10. Jawaban Nomor 10

```

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

```

11. Jawaban Nomor 11

```

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

```

2.10.3 Keterampilan Penanganan Error

1. Jawaban Nomor 1 : Error

```

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

```

2. Jawaban Nomor 2 : Solusi

```

a = 100
b = "50"

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

```

2.11 Ilham Muhammad Ariq

2.11.1 Teori

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

If Else

```

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

```

Elif

```

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

```

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

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

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

7. Tuliskan dan jelaskan cara memakai Try Except.

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

2.11.2 Ketrampilan Pemrograman

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```



```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

2.11.3 Keterampilan Penanganan Error

```

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

```

```

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

```

2.12 Muhammad Reza Syachrani / 1174084

2.12.1 Teori

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

Cara mengubah tipe data string ke integer :

```

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

```

Cara mengubah tipe data integer ke string :

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

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

▪ Perulangan while :

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

▪ Perulangan for :

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

5. sintak Untuk memilih kondisi, dan kondisi didalam kondisi

▪ If Kondisi ini dipergunakan jika penkondisiannya hanya satu.

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

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

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

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

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

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

▪ Type Error

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

▪ Syntax Error

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

▪ Name Error

Name Error adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Solusinya adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.

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

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

2.12.2 Keterampilan Pemrograman

1. Jawaban Soal 1

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

2. Jawaban Soal 2

```

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

```

3. Jawaban Soal 3

```

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

```

4. Jawaban Soal 4

```

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

```

5. Jawaban Soal 5

```

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

```

6. Jawaban Soal 6

```

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

```

7. Jawaban Soal 7

```

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

```

8. Jawaban Soal 8

```

1 for a in npm:
2     print(a)

```

9. Jawaban Soal 9

```

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

```

10. Jawaban Soal 10

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

11. Jawaban Soal 11

```
1 print(c)
```

2.12.3 Ketrampilan Penanganan Error

1. Jenis- jenis error dan penanganan error

▪ Type Error

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

▪ Syntax Error

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

▪ Name Error

Name Error adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Solusinya adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.

2. Jawaban :

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

2.13 Handi Hermawan

2.13.1 Teori

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

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

2.13.2 Keterampilan Pemrograman

```

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

```

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

2.13.3 Keterampilan dan penanganan error

1. Tuliskan peringatan error yang didapat dari mengerjakan praktek kedua ini, dan jelaskan cara penanganan error tersebut.

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

2.14 Mochamad Arifqi Ramadhan

Praktek teori penunjang yang dikerjakan :

2.14.1 Variable, pemakaian Variable dan Jenis-Jenis Type data

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

1. Pemakaian Variabel

Variabel di python dapat dibuat dengan format seperti ini:

NamaVariabel = (nilai)

Contoh:

Variabel01 = "isi variabelnya"

variabel022 = 10

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

```
print Variabel01
```

```
print variabe02
```

2. Aturan Penulisan Variabel

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

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

3. Tipe Data

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

4. Jenis-Jenis Tipe Data

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

5. Meminta input dan melakukan output

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

6. Operator dan Konversi

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

7. Perulangan di Python

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

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

2.14.2 Ketrampilan Pemrograman

Buat program di python dengan ketentuan :

(a) Jawaban 1

```

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

```

(b) Jawaban 2

```

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

```

(c) Jawaban 3

```

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

```

(d) Jawaban 4

```

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

```

(e) Jawaban 5

```

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

```

(f) Jawaban 6

```

1 #No.6
2 print(h)

```

(g) Jawaban 7

```

1 #No.7
2 print(i)

```

(h) Jawaban 8

```

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

```

(i) Jawaban 9

```

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

```

(j) Jawaban 10

```

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

```

(k) Jawaban 11

```

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

```

2.14.3 Ketrampilan Penanganan Error

Bagian Penanganan error dari script python.

(a) Jawaban

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

(b) Jawaban

```

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

```

2.15 Fernando Lorencius Sihite

Praktek teori penunjang yang dikerjakan :

2.15.1 Variable, pemakaian Variable dan Jenis-Jenis Type data

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

1. Pemakaian Variabel

Variabel di python dapat dibuat dengan format seperti ini:

NamaVariabel = (nilai)

Contoh:

a = "Fernando Lorencius S"

b = "renang"

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

print (a)

print (b)

Aturan Penulisan Variabel

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

Tipe Data

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

Jenis-Jenis Tipe Data

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

2. Meminta input dan melakukan output

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

Contoh menggunakan input :

`a = "Fernando Lorencius S"`

`b = "renang"`

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

3. Operator dan Konversi

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

Penggunaan Operasi aritmatika

Contoh penggunaan Aritmatika

`a = 10`

`b = 2`

perintah pembagian

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

perintah pertambahan

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

perintah pengurangan

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

Perintah perkalian

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

4. Perulangan di Python

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

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

Contoh Penggunaan For :

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

Contoh Penggunaan While :

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

5. Penggunaan Kondisi

Penggunaan Kondisi IF

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

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

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

```
nilai = 8\\
```

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

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

Penggunaan KOndisi IF else

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

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

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

Dibawah ini adalah contoh penggunaan kondisi if else pada Python

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

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

Penggunaan Kondisi Elif

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

```
hari = "Minggu"
```

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



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

6. Kesalahan sintaks

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

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

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

Contohnya :

```
>>> while 1 print 'Hello world'\\

File "<stdin>", line 1\\

    while 1 print 'Hello world'\\

SyntaxError: invalid syntax\\
```

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

7. Cara memakai try except

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

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

contoh :

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

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

2.15.2 Ketrampilan Pemrograman

Jawaban No.1

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

```

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

```

Jawaban no.2

```

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

```

Jawaban no.3

```

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

```

jawaban no.4

```

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

```

jawaban no.5

```

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

```

jawaban no.6

```

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

```

Jawaban no.7

```

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

```

jawaban no.8

```

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

```

jawaban no.9

```

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

```

jawaban no.10

```

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

```

jawaban no.11

```

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

```

2.15.3 Ketrampilan Penanganan Error

```

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

```

2.16 Arrizal Furqona Gifary

2.16.1 Teori

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

```
1
2 #Soal no 1 Tipe data
3
4
5 #contoh integer
6 a = 1
7 b = 1.1
8 z = 2+1j
9
10 print(type(a))
11 print(type(b))
12 print(type(z))
13
14 # contoh string
15
16 a = "Chandra Kirana Poetra"
17 print(a[1])
18 #print ini digunakan untuk mencetak huruf yang dipilih indexnya
19
20 a = "Chandra Kirana Poetra"
21 print(a[2:5])
22 #print ini digunakan untuk mencetak huruf yang dipilih indexnya
    dengan mengisi
23 #nilai index tempat kita memilih index untuk memulai mengambil,
    dan juga nilai index
24 #akhirnya
25
26 a = "Chandra Kirana Poetra"
27 print(len(a))
28 #print ini digunakan untuk menjumlahkan length yang artinya
    panjang dari string a
29
30 a = "Chandra Kirana Poetra"
31 print(a.lower())
32 #print ini digunakan untuk mencetak huruf yang kecil semua
33
34 a = "Chandra Kirana Poetra"
35 print(a.upper())
36 #print ini digunakan untuk mencetak huruf yang besar semua
37
38 #boolean
39
40 a = 10
41 b = 8
42 if (a>b):
43     print("Variabel A lebih besar dibandingkan dengan Variable B")
44     )
```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

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

2.16.2 Praktek

1. Jawaban soal no 1

```

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

```

2. Jawaban soal no 2

```

1 a = 12
2 b = 70

```

```

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

```

3. Jawaban soal no 3

```

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

```

4. Jawaban soal no 4

```

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

```

5. Jawaban soal no 5

```

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

```

6. Jawaban soal no 6

```

1 #a = 100
2 #b = 50

```

7. Jawaban soal no 7

```

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

```

8. Jawaban soal no 8

```

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

```



```

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

```

9. Jawaban soal no 9

```

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

```

10. Jawaban soal no 10

```

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

```

11. Jawaban soal no 11

```

1  hasil3 = []
2  hasil4 = []

```

2.16.3 Keterampilan dan penanganan error

```

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

```

2.17 Alvan Alvanzah/1174077

2.17.1 Teori

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

- Variabel

```

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

```

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

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

▪ Meminta input

Fungsi input () memungkinkan input pengguna.

```

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

```

▪ Output ke layar

Fungsi print () mencetak pesan yang ditentukan ke layar.

```

1 print("Hello World")

```

3. Operator dasar aritmatika pada Python.

▪ Tambah

```

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

```

▪ Kurang

```

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

```

▪ Kali

```

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

```

▪ Bagi

```

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

```

▪ String ke Integer

```

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

```

▪ Integer ke String

```

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

```

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

▪ While Loop

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

```

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

```

▪ For Loop

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

```

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

```

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

▪ Kondisi

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

"Jika pernyataan" ditulis dengan menggunakan kata kunci if.

```

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

```

▪ Kondisi dalam kondisi

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

```

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

```

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

- **Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- **Zero Division Error** ZeroDivisonError adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0). Solusinya adalah tidak membagi suatu yang hasilnya nol.
- **Name Error** NameError adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Solusinya adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.
- **Type Error** TypeError adalah exception yang terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah 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:
2     print("Hello")
3 except:
4     print("Something went wrong")
5 else:
6     print("Nothing went wrong")

```

2.17.2 Ketrampilan Pemrograman

1. Jawaban Soal 1

```

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

```

```

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

```

2. Jawaban Soal 2

```

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

```

3. Jawaban Soal 3

```

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

```

4. Jawaban Soal 4

```

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

```

5. Jawaban Soal 5

```

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

```

6. Jawaban Soal 6

```

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

```

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

7. Jawaban Soal 7

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

8. Jawaban Soal 8

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

9. Jawaban Soal 9

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

10. Jawaban Soal 10

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

11. Jawaban Soal 11

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

2.17.3 Ketrampilan Penanganan Error

1. Jawaban Soal No. 1

- Syntax Errors

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

- Zero Division Error ZeroDivisonError adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0). Solusinya adalah tidak membagi suatu yang hasilnya nol.

- Name Error NameError adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Solusinya adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.

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

2. Jawaban Soal No. 2

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

2.18 Kaka Kamaludin

1. Aturan Penulisan Variabel

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

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

Contoh penamaan variable:

- name12
- name_variable
- _variable1
- nameVarialble
- NameVariable

Contoh sytanx :

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

2. Meminta input dari user dan menampilkannya

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

contoh syntax :

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

3. penggunaan aritmatika dan convert type data

aritmatika sering dipakai dalam pemrograman terutama di python. //

Contoh aritmatika:

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



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

4. Perulangan

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

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

5. kondisi, kondisi didalam kondisi

kondisi di python menggunakan if dan else.

```

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

```

6. error yang sering ditemu di python

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

7. Try Except

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

```

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

```

2.19 Praktek

1. jawaban

```

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

```

2. jawaban

```

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

```

3. jawaban

```

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

```

4. jawaban

```

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

```

5. jawaban

```

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

```

6. jawaban

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

7. jawaban

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

8. jawaban

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

9. jawaban

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

10. jawaban

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

11. jawaban

2.20 Praktek

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

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

2.21 Advent Nopele Olansi Damiahn Sihite

Praktek teori penunjang yang dikerjakan :

2.21.1 Variable, pemakaian Variable dan Jenis-Jenis Type data

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

1. Pemakaian Variabel

Variabel di python dapat dibuat dengan format seperti ini:

```
NamaVariabel = (nilai)
```

Contoh:

```
VariabelKu = "ini isi variabel"
```

```
variabel2 = 20
```

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

```
print VariabelKu
```

```
print variabel2
```

(a) Aturan Penulisan Variabel

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

(b) Tipe Data

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

(c) Jenis-Jenis Tipe Data

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

2. Meminta input dan melakukan output

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

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

3. Operator dan Konvert

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

4. Perulangan di Python

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

5. Kodisi di Python

(a) Kondisi **If**

Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar

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

```
a = 33
```

```
b = 200
```

```
if b > a:
```

```
    print("b lebih besar dari a")
```

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

```
a = 200
```

```
b = 33
```

```
if b > a:
```

```
    print("b lebih besar dari a")
```

```
else:
```

```
    print("b bukan lebih besar dari a")
```

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

```
a = 33
```

```
b = 33
```

```
if b > a:
```

```
    print("b lebih besar dari a")
```

```
elif a == b:
```

```
    print("a sama dengan b")
```

6. Error yang sering dialami

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

7. Cara memakai Try Except

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

contoh :

try:

```
print(x)
```

except:

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

2.21.2 Ketrampilan Pemrograman

Buat program di python dengan ketentuan :

1. Jawaban

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

2. Jawaban

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

3. Jawaban

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

4. Jawaban

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

5. Jawaban


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

6. Jawaban

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

7. Jawaban

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

8. Jawaban

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

9. Jawaban

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

10. Jawaban

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

11. Jawaban

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

2.21.3 Ketrampilan Penanganan Error

Bagian Penanganan error dari script python.

1. Jawaban

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

2. Jawaban

```

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

```

2.22 Advent nopele olansi damiahan sihite

2.22.1 Teori

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

2.22.2 Praktek

1. Jawaban soal no 1

```

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

```

2. Jawaban soal no 2

```

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

```

3. Jawaban soal no 3

```

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

```

4. Jawaban soal no 4

```

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

```

5. Jawaban soal no 5

```

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

```

6. Jawaban soal no 6

```

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

```

7. Jawaban soal no 7

```

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

```

8. Jawaban soal no 8

```

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

```

9. Jawaban soal no 9

```

1 print(d)
2 print(e)

```

10. Jawaban soal no 10

```

1 print(g)

```

11. Jawaban soal no 11

```

1 print(d,e,g)

```

2.22.3 Keterampilan dan penanganan error

```

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

```

2.23 Ainul Filiaini

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

1. Variable Gabungan

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

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

Contoh :

a = "Guru"

b = 99

cetak a + b

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

a = "Guru"

b = 99

print(a + str(b))

2. Variable Local dan Global

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

perbedaan antara variable lokal dengan variable global :

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

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

Contoh :

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

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

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

3. Hapus Sebuah variable

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

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

contoh :

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

Dan Hasilnya :

Siapa naman kamu :

Berapa umur kamu :

Cara menampilkan Output:

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

contoh :

```
print ” Hello World!”
```

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

Menampilkan Variable dan Teks :

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

Hasilnya :

```
Hello PetaniKode
Antara kata hello
```

4. Operator dasar aritmatika

Tambah (+)

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

Kurang(-)

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

Kali(x)

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

Bagi(/)

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

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

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

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


Sebaliknya, untuk mengubah Integer ke String menggunakan fungsi berikut:

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

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

5. cara pakai sintak untuk memilih kondisi

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

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

Hasil Eksekusi di konsol:

```
$ python if.py
sudah beranjak dewasa
```

untuk memeriksa kondisi yang tidak memenuhi kondisi utama. maka else digunakan untuk menangani semua kondisi selain kondisi yang telah dituliskan. Berikut adalah contoh penggunaan else di dalam kondisi Python :

```
umur = 20
if umur /> 18:
print "Sudah beranjak dewasa"
else:
printl "Masih dibawah umur"
hasil eksekusi di konsol:
$ python if-else.py
Sudah beranjak dewasa
```

Bila kondisi yang akan didefinisikan cukup banyak, anda menambahkan kondisi lain dengan menggunakan elif di bawah statement

if dan sebelum statement else:

```
umur = 37
if umur < 18 and umur < 30:
    print "Sudah beranjak dewasa"
elif umur < 30 and umur < 45:
    print "Masa - masa emas"
elif umur < 45 and umur < 55:
    print "Memasuki masa paruh baya"
elif umur < 55:
    print "Masa - masa manula"
else:
```

```
    print "Masih dibawah umur"
```

Hasil eksekusi konsol :

```
$python if-elif.py
```

```
masa - masa emas
```

dan tentu saja, sebuah kondisional dapat disimpan di dalam if lain. berikut ini adalah contoh kode if bersarang di python :

```
gaji = 10000000
```

```
berkeluarga = True
```

```
punya_rumah = True
```

```
    if gaji < 3000000:
```

```
        print "Gaji sudah diatas UMR"
```

```
    if berkeluarga:
```

```
        print "Wajib ikutan asuransi dan menabung untuk pensiun"
```

```
    else
```

```
        print "Tidak perlu ikutan asuransi"
```

```
    if punya_rumah:
```

```
        print "wajib bayar pajak rumah"
```

```
    else:
```

```
        print "tidak wajib bayar pajak rumah"
```

```
    else:
```

```
        print "Gaji belum UMR"
```

Hasil eksekusi konsol :

```
$ python nested-if.py
```

```
Gaji sudah diatas UMR
```

```
Wajib ikutan asuransi dan menabung untuk pensiun
```

```
wajib bayar pajak rumah
```

6. jenis error di python :

Tidak mengenali variable :

```
$$$ d = [1, 2, 3, 4]
```

```
$$$ d[5]
```

Traceback (most recent call last):

File "jstdin", line 1, in jmodule:
 IndexError: list index out of range

~~~~

### 1. kesalahan syntax

Kesalahan ini biasanya disebabkan oleh cara mengganti program yang tidak dapat diakses kaidah menerima kode yang benar-benar sesuai dengan bahasa pemrograman yang digunakan atau memang kurang atribut. Contoh di pythor

### 2. Kesalahan Eksekusi

Kesalahan ini menyebabkan eksekusi dieksekusi. Contoh dari kesalahan ini diterbitkan adalah memasukan data yang

salah (bukan bilangan) pada fungsi input(), atau contoh lain yang disediakan indeks di luar daftar indeks daftar, atau membuka pernyataan dekorasi yang tidak ada di dalam disk kita.

### 3. Kesalahan Logika

Kesalahan ini diterapkan saat skrip sedang dalam eksekusi dieksekusi tetapi tidak dilakukan berhentinya eksekusi skrip tersebut. Kesalahan seperti ini pada umumnya karena logika dalam naskah yang salah. Programmer misalkan yang menerjemahkan kode bilangan Phi 3.14 tetapi pada kenyataanya dia menerima 31 sehingga skrip yang diperlukan mampu menghitung luas dengan benar menghasilkan angka G + sar dan salah.

### 7. Try Except

Salah satu bentuk penanganan error di Python adalah dengan menggunakan statement try..except. Mungkin kita pernah mendeteksi error dengan memanfaatkan kondisional biasa menggunakan if..else, namun hal tersebut akan lebih praktis ditangani bila dengan menggunakan try..except. Anda dapat mengurung suatu blok kode dengan try..except untuk menangani error yang mungkin kita sendiri tidak mengetahuinya. Biasanya try..except ini digunakan untuk menangani error saat penggunaan IO, operasi database, atau pengaksesan indeks suatu list atau dictionary, dan berbagai kasus lainnya.

contoh :

```
try: print("lins")
```

```
except :
```

```
print("lins kamu pinter")
```

```
finaly:
```

```
    print("memang pinter")
```

## 2.23.1 Keterampilan pemrograman

### 1. jawaban soal 1

```
1 print(1174073%3)
```

```

2 print ("+++ +++ ++++++++ +++ +++ ++++++++ ++++++++
    ++++++++")
3 print ("+++ +++ ++++++++ +++ +++ ++++++++ ++++++++
    ++++++++")
4 print ("+++ +++          +++ +++ +++ ++      ++      +++
    ++")
5 print ("+++ +++          +++ +++ +++ ++      ++      +++
    ++")
6 print ("+++ +++          +++ ++++++++ ++      ++      +++
    ++")
7 print ("+++ +++          +++          +++ ++      ++      +++
    ++++++++")
8 print ("+++ +++          +++          +++ ++      ++      +++
    ++")
9 print ("+++ +++      +++          +++ ++      ++      +++
    ++")
10 print ("+++ +++      +++          +++ ++      ++      +++
    ++")
11 print ("+++ +++      +++          +++ ++++++++ +++
    ++++++++")
12 print ("+++ +++      +++          +++ ++++++++ +++
    ++++++++")

```

## 2. jawaban soal 2

```

1 npm = input("Masukan NPM :")
2 nilai = int(npm[5:7])
3 hitung = 0
4 while(hitung < nilai):
5     print("Halo, " + str(npm) + " Apa Kabar?")
6     hitung = hitung +1

```

## 3. jawaban soal 3

```

1 npm = input("Masukan NPM :")
2 nilai = int(npm[4:5]) + int(npm[5:6]) + int(npm[6:7])
3 hitung = 0
4 while(hitung < nilai):
5     print("Halo, " + str(npm[4:7]) + " Apa Kabar?")
6     hitung = hitung +1

```

## 4. jawaban soal 4

```

1 npm = input("Masukan NPM :")
2 print("Halo, " + str(npm[-3]) + " Apa Kabar?")

```

## 5. jawaban soal 5

```

1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 7
7 g = 3

```

```

8 h = a+b+c+d+e+f+g
9 i = a*b*c*d*e*f*g
10 print(a,b,c,d,e,f,g)

```

## 6. jawab an soal 6

```

1 print(h)

```

## 7. jawaban soal 7

```

1 print(i)

```

## 8. jawaban soal 8

```

1 print(a)
2 print(b)
3 print(c)
4 print(d)
5 print(e)
6 print(f)
7 print(g)

```

## 9. jawaban soal 9

```

1 a = 1
2 b = 1
3 c = 7
4 d = 3
5 e = 0
6 f = 7
7 g = 3
8 for n in npm:
9     if(n % 2 == 0):
10         if(n != 0):
11             print(n, end = "")

```

## 10. jawaban soal 10

```

1 a = 1
2 b = 1
3 c = 7
4 d = 3
5 e = 0
6 f = 7
7 g = 3
8 for n in npm:
9     if(n % 2 != 0):
10         print(n, end = "")

```

## 11. jawaban soal 11

```

1 a = 1

```

## 2.24 Sekar Jasmine

1. Jenis-Jenis variabel dan cara pemakaian variabel tersebut di kode python

1. Variabel Gabungan

menggabungkan tipe data yang berbeda seperti string dan angka secara bersamaan . contoh , kita menggabungkan antara "guru" dengan angka "200".

tidak seperti java yang menggabungkan angka dengan string tanpa mendeklarasi angka sebagai string , python mengharuskan pendeklarasikan angka tersebut sebagai string , jika tidak ia akan menampilkan type error

Contoh :

```
a = "guru"
```

```
b = "200"
```

```
cetak a+b
```

setelah variabel dideklarasikan sebagai string , ia dapat menggabungkan kedua "siswa" + str("200") = "siswa200" di output

```
a = "siswa"
```

```
b = 200
```

```
print(a+ str(b))
```

2. Beberapa fitur yang dimiliki Python

a. memiliki aturan layout kode sumber yang memudahkan pengecekan, pembacaan kembali dan penulisan ulang kode sumber. berorientasi obyek. b. memiliki sistem pengelolaan memori otomatis (garbage collection, seperti java), mudah dikembangkan dengan menciptakan modul-modul baru; modul-modul tersebut dapat dibangun dengan bahasa Python maupun C/C++.

3. Tipe Data

A. Number Tipe data Number merepresentasikan nilai-nilai berupa angka. Python menggolongkan beberapa tipe data umum seperti, Integer (bilangan bulat) dan Floating-point (bilangan desimal) ke dalam tipe data Number. Contoh :

```
a = "siswa"
```

```
b = 200;
```

```
print(g)
```

B. String

Selain angka, python juga mampu melakukan manipulasi string, yang dapat diekspresikan dengan beberapa cara. Penulisan nilai string pada python menggunakan tanda petik satu ( ' ) atau tanda petik dua ( " ). Contoh :

```
'''Selamat Datang Siswa Baru''' "Selamat Datang Siswa Baru" '''selamat datang siswa baru" "selamat datang siswa baru"
```

Untuk menggabungkan dua buah string atau lebih dapat dilakukan dengan dua cara. Pertama, dengan menulis langsung dua buah string yang diapit dengan tanda kutip atau dengan penggunaan operator (+) Contoh :

'''Menggenal''' 'python' 'Menggenalpython' '''institusi' + 'pertanian bogor'  
'institutipertanianbogor'

### C. Dictionary

dictionary memakai key untuk merujuk pada isi variabelnya. Sifat kedua jenis data ini hanya berbeda dalam beberapa hal saja. Untuk mendeklarasikan sebuah dictionary, Python memakai tanda Contoh :

'''D = 'drink': 'spam', 'time': 8, 'color': 'purple' '''  
D 'drink': 'spam', 'color': 'purple', 'time': 8  
D['color'] 'purple'

### 4. Operator dalam Python dibagi menjadi 3 bagian, yaitu :

```

1  A. operator aritmatika : +, - , *, / , %
2      >>> x = 7 + 3 * 6
3      >>> x
4      25
5      >>> y = 200/6*5
6      >>> y
7      166
8
9
10     >>> x = ( 8 + 7 ) * 9
11     >>> x
12     71
13     >>> y = 300 / ( 5 * 4 )
14     >>> y
15     240
16
17 Menggunakan operator %
18     >>> 8 % 6
19     1
20     >>> 9 % 7
21     1
22
23 B. operator perbandingan : >= , <= , != , > , < , ==
24     Contoh :\\
25     a = input ( "Inputkan nilai a: " )
26     b = input ( "Inputkan nilai b: " )
27
28     #apakah a sama dengan b?
29     c = a == b
30     print "apakah %d == %d: %r" % (a,b,c)
31
32     #apakah a < b ?
33     c = a < b
34     print "apakah %d < %d: %r" % (a,b,c)
35
36     #apakah a > b ?
37     c = a > b
38     print "apakah %d > %d: %r" % (a,b,c)
39
40     #apakah a <= b ?
41     c
42     <= b ?
43     print "apakah %d <= %d: %r" (a,b,c)
44
45     #apakah a >= b ?

```

```

46     c = a >= b
47     print "apakah %d >= %d: %r" (a,b,c)
48 C. o= a perator penugasan : *= , /= , %= , +=, -=
49     a = input ("Inputkan nilai a: ")
50     print "Nilai a = %d" % a
51
52     # Coba kita jumlahkan nilai a dengan operator penugasan
53     a += 5
54
55     # setelah nilai a ditambah 5 , coba kita lihat isinya
56     print "Nilai setelah ditambah 5:"
57     print "a = %d" % a

```

## 2.24.1 Keterampilan Pemrograman

1. jawab soal 1
2. jawab soal 2
3. jawab soal 3
4. jawab soal 4
5. jawab soal 5
6. jawab soal 6
7. jawab soal 7
8. jawab soal 8
9. jawab soal 9
10. jawab soal 10
11. jawab soal 11

## 2.25 Dini Permata Putri

### 2.25.1 sebutkan jenis-jenis variabel dan jelaskan cara pemakaian variabel tersebut dikode python

proses memasukan data ke dalam variabel nama = "John Doe" proses mencetak variabel print(name)

nilai dan tipe data dalam variabel dapat diubah umur = 20 nilai awal print(umur) mencetak nilai umur type(umur) mengecek tipe data umur umur = "dua puluh satu" nilai setelah diubah print(umur) mencetak nilai umur type(umur) mengecek tipe data umur



```

1 namaDepan = "Budi"
2 namaBelakang = "Susanto"
3 nama = namaDepan + " " + namaBelakang
4 umur = 19
5 hobi = "Nyanyi"
6 print("Biodata\n", nama, "\n", umur, "\n", hobi)
7
8 contoh variabel lainnya
9 inivariabel = "Halo"
10 ini_juga_variabel = "Hai"
11 _inivariabeljuga = "Hi"
12 inivariabel222 = "bye"
13
14 panjang = 8
15 lebar = 4
16 luas = panjang * lebar
17 print(luas)
18
19 Membuat variabel di python
20 nama_variabel = <nilai>
21 contoh :
22 variabel_ku = "ini isi variabel"
23 variabel2 = 20
24 print variabel_ku
25 print variabel2
26
27 Tipe data
28 contoh :
29 nama_mu = "Petani Kode"
30 uumur =20
31 tinggi = 183.22
32 contoh :
33 >>> usia = 19
34 >>> type(usia)
35 <type 'int'>
36 >>> usia = "19"
37 >>> type(usia)
38 <type 'str'>
39 >>> usia = '19'
40 >>> type (usia)
41 <type 'str'>
42 >>> usia = 21.5
43 >>> type(usia)
44 <type 'float'>
45 >>>usia = true
46 Traceback (most recent cal last)
47 File "<stdin>", line 1, in <module>
48 NameError: name 'true' is not defined
49 >>> usia = true
50 >>> type(usia)
51 <type 'bool'>

```

### 2.25.2 tuliskan bagaimana kode untuk meminta input dari user dan tuliskan bagaimana melakukan output ke layar

```

1 cara mengambil input dari keyboard
2 nama_variabel = input("Sebuah Teks")
3
4 #mengambil input
5 nama + raw_input(Siapa nama kamu: ")
6
7 #menampilkan output
8 print "Hello", nama, "umur kamu adalah",umur,"tahun"
9
10 cara menampilkan output
11 print "Hello World!"
12 print nama_variabel
13 print "Gabung dengan", variabel
14
15 menampilkan variabel dan teks
16 nama = "Petanikode"
17 print "Hello", nama
18
19 menggunakan string formatting cara lama
20 menggunakan simbol %
21 nama = raw_input("inputkan nama: ")
22 print "Selamat datang %s" % nama
23 tanda % akan otomatis diganti dengan nilai yang kita inputkan ke
    variabel nama.
24 nama = raw_input("Inputkan nama: ")
25 umur = input("inputkan umur: ")
26 tinggi = input(inputkan tinggi badan: ")
27 print "Hello %5, saat ini usiamu %d tahun dan tinggi badanmu %f cm" %
    (nama, umur, tinggi)

```

### 2.25.3 tuliskan operator dasar aritmatika, tambah, kali, kurang bagi, dan bagaimana mengubah string ke integer dan integer ke string

```

1 #penjumlahan
2 print(13-2)
3 apel=7
4 jeruk=9
5 buah=apel+jeruk
6 print(buah)
7
8 pengurangan
9 hutang=8000
10 bayar=4000
11 sisahutang=hutang-bayar
12 print(*sisah utang anda adalah", sisahutang)
13
14 perkalian
15 panjang=14
16 lebar=7
17 luas=panjang*lebar
18 print(luas)
19

```

```

20 pembagian
21 kue=15
22 anak=3
23 kueperanak=kue / anak
24 print("setiap anak akan mendapatkan bagian kue sebanyak",kueperanak)
25
26 sisa bagi
27 bilangan1=12
28 bilangan2=4
29 hasil=bilangan1%bilangan2
30 print("sisa bagi dari bilangan",bilangan1,"dan",bilangan2,"adalah",..
    hasil)
31
32 cara mengubah integer ke integer
33 integer merupakan tipe data data bilangan bulat.
34 a=100#variabel/angka yang akan di konversi
35 string=str(a)konversi integer ke string
36 print(string)
37
38 cara mengubah string ke integer
39 string merupakan tipe data yang menampung sebuah teks.
40 a='1212'variabel/angka yang akan di konversi.
41 integer==int(a)konversi string ke integer
42 print(integer)

```

#### 2.25.4 tuliskan dan jelaskan sintak untuk perulangan, jenis-jenisnya contoh kode dan cara pakainya di python

```

1 -statement (pernyataan) di python
2 statement penugasan, if, for dsb.
3 if nilai <=5:
4     print("Nilai merah")
5     print(Tidak lulus")
6 else:
7     print("nilai biru")
8     print("lulus")
9
10 -tanda kutip di python
11 kata='kata'
12 kalimat="ini adalah kalimat"
13 paragraf = """ini adalah paragraf.
14 paragraf terdiri dari beberapa baris."""
15
16 -komentar di python
17 #komentar pertama
18 print("Hello World") # komentar kedua

```

#### 2.25.5 tuliskan dan jelaskan cara pakai sintak untuk memilih kondisi, dan bagaimana contoh sintak kondisi didalam kondisi

contoh penggunaan kondisi if di python :

```

1 umur = 19
2 if umur > 17:
3     print "sudah beranjak dewasa"

```

```

4  hasil eksekusi :
5  $ python if.py
6  sudah beranjak dewasa
7  contoh penggunaan kondisi else di python:
8  umur = 19
9  if umur > 17:
10     print "sudah beranjak dewasa"
11 else:
12     print "masih dibawah umur"
13 hasil eksekusi :
14 $ python if-else.py
15 sudah beranjak dewasa

```

### 2.25.6 tuliskan apa saja jenis error yang sering ditemui di python dalam mengerjakan sintak diatas dan bagaimana cara mengatasinya

```

1  if true:
2     print ("Answer")
3     print ("True")
4  else:
5     print("Answer")
6     print("False")
7  kode diatas error, karena print ("false") dan print ("answer")
   seharusnya pada blok yang sama seper
8
9
10 contoh berikut :
11 if true:
12     print ("Answer")
13     print ("True")
14
15 else:
16     print ("Answer")
17     print ("False")
18 ti

```

### 2.25.7 tuliskan dan jelaskan cara memakai Try Except

```

1  cara menangani error pembagian nol =:
2  x 0
3  try:
4     x = 1/0
5  except Exception,
6  :
7     print e
8  print x+1
9  jika kode diatas dieksekusi, maka akan muncul error seperti berikut
   ini :
10 $ python demo-1.py
11 integer division or modulo by zero
12 1
13
14 cara menangani error pembacaann file :
15 orang = {"nama": "bibb", "kota": "jepara", "umur": "20"}

```

```
16
17 try:
18     contact = open("contact.txt", 'r')
19     print orang["pekerjaan"]
20 except IOError, e:
21     print "terjadi erorr IO: ",e
22 except keyrror, e:
23     print "terjadi kesalahan pada akses list/dict/tuple:",e
24 print orang
25 jika kode diatas dieksekusi, maka akan muncul output seperti berikut:
26 $ python demo-2.py
27
28 terjadi error IO: [errno 2] No such file or directory: '/home/contact
29     .txt
30 {'nama': 'syuaib', 'umur',: '19', 'kota': 'jepara'}
```

## BAB 3

---

## JUDUL BAGIAN KETIGA

---



# Index

---

disruptif, xxxi  
modern, xxxi