

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	Judul Bagian Pertama	1
2	Judul Bagian Kedua	7
3	Judul Bagian Ketiga	45

DAFTAR ISI

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

1.2.3	Objective and Contribution	2
1.2.4	Scoop and Environment	2
1.3	Luthfi Muhammad Nabil_1174035	2
1.3.1	Background	2
1.3.2	Problems	3
1.3.3	Objective and Contribution	3
1.3.4	Scoop and Environment	4
1.4	Hagan Rowlenstino/1174040	4
1.4.1	Background	4
1.4.2	Problems	4
1.4.3	Objective and Contribution	4
1.4.4	Scoop and Environment	5
1.5	Rangga Putra Ramdhani_1174056	5
1.5.1	Background	5
1.5.2	Problems	5
1.5.3	Objective and Contribution	5
1.5.4	Scope and Environment	6
2	Judul Bagian Kedua	7
2.1	Alit Fajar Kurniawan 1174057	7
2.1.1	Teori	7
2.1.2	praktek	9
2.1.3	Keterampilan dan penanganan eror	11
2.2	IrvanRizkiansyah/1174043	11
2.2.1	Teori	11
2.2.2	Keterampilan Pemrograman	13
2.2.3	Keterampilan Penanganan Error	16
2.3	Hagan Rowlenstino/1174040	16
2.3.1	Teori	16
2.3.2	Keterampilan Pemrograman	17
2.3.3	Keterampilan Penanganan Error	20
2.4	Muhammad Iqbal Panggabean	21
2.4.1	Teori	21
2.4.2	praktek	22
2.4.3	Keterampilan dan penanganan eror	24
2.5	Luthfi M. Nabil/1174035	24
2.5.1	Teori	24
2.5.2	Keterampilan Pemrograman	26

2.5.3	Keterampilan Penanganan Error	31
2.6	Faisal Najib Abdullah 1174042	31
2.6.1	Teori	31
2.6.2	Keterampilan Pemrograman	33
2.6.3	Keterampilan Penanganan Error	36
2.7	Dika Sukma Pradana 1174050	36
2.7.1	Teori	36
2.7.2	Praktek	38
2.7.3	Keterampilan dan Penanganan Error	40
2.8	Ichsan Hizman Hardy	40
2.8.1	Teori	40
2.8.2	praktek	42
2.8.3	Keterampilan dan penanganan eror	44
3	Judul Bagian Ketiga	45
3.0.1	Luthfi Muhammad Nabil/1174035	45
3.0.2	Ketrampilan Pemrograman	47
3.1	Rangga Putra Ramdhani	60
3.2	Faisal Najib Abdullah	67
3.2.1	Pemahaman Teori	67
3.2.2	Ketrampilan Pemrograman	69
3.2.3	Ketrampilan Penanganan Error	73
	Daftar Pustaka	75
	Index	77

DAFTAR GAMBAR

3.1	Screenshot No 1	48
3.2	Screenshot No 2	49
3.3	Screenshot No 3	49
3.4	Screenshot No 4	50
3.5	Screenshot No 5	50
3.6	Screenshot No 6	51
3.7	Screenshot No 7	51
3.8	Screenshot No 8	52
3.9	Screenshot No 9	52
3.10	Screenshot No 10	53
3.11	Screenshot No 11	56
3.12	Screenshot No 12	59
3.13	Screenshot No 13	60
3.14	Plagiarisme	60

DAFTAR TABEL

Listings

src/chapter2/1174057_teor1.py	7
src/chapter2/1174057_teor1.py	7
src/chapter2/1174057_teor1.py	8
src/chapter2/1174057_teor1.py	8
src/chapter2/1174057_teor1.py	8
src/chapter2/1174057_teor1.py	8
src/chapter2/1174057_teor1.py	8
src/chapter2/1174057_teor1.py	9
src/chapter2/1174057_teor1.py	9
src/chapter2/1174057_praktek.py	9
src/chapter2/1174057_praktek.py	10
src/chapter2/1174057_praktek.py	10
src/chapter2/1174057_praktek.py	10
src/chapter2/1174057_praktek.py	10
src/chapter2/1174057_praktek.py	10
src/chapter2/1174057_praktek.py	10
src/chapter2/1174057_praktek.py	10
src/chapter2/1174057_praktek.py	11

src/chapter2/1174057_praktek.py	11
src/chapter2/1174057_praktek.py	11
src/errr2.py	11
src/chapter2/1174043_1.py	13
src/chapter2/1174043_2.py	13
src/chapter2/1174043_3.py	13
src/chapter2/1174043_4.py	14
src/chapter2/1174043_5.py	14
src/chapter2/1174043_6.py	14
src/chapter2/1174043_7.py	14
src/chapter2/1174043_8.py	14
src/chapter2/1174043_9.py	15
src/chapter2/1174043_10.py	15
src/chapter2/1174043_11.py	15
src/chapter2/1174043_2err.py	16
src/chapter2/1174040_1.py	17
src/chapter2/1174040_2.py	18
src/chapter2/1174040_3.py	18
src/chapter2/1174040_4.py	18
src/chapter2/1174040_5.py	18
src/chapter2/1174040_6.py	18
src/chapter2/1174040_7.py	19
src/chapter2/1174040_8.py	19
src/chapter2/1174040_9.py	19
src/chapter2/1174040_10.py	19
src/chapter2/1174040_11.py	20
src/chapter2/1174040_2err.py	20
src/1174063_teor.py	21
src/1174063_teor.py	21
src/1174063_teor.py	21
src/1174063_teor.py	21
src/1174063_teor.py	22
src/1174063_teor.py	22
src/1174063_teor.py	22
src/1174063_teor.py	22
src/1174063_teor.py	22
src/1174063_praktek.py	23
src/1174063_praktek.py	23

src/1174063_praktek.py	23
src/1174063_praktek.py	23
src/1174063_praktek.py	23
src/1174063_praktek.py	24
src/1174063_praktek.py	24
src/1174063_praktek.py	24
src/1174063_praktek.py	24
src/1174063_praktek.py	24
src/1174063_praktek.py	24
src/errr2.py	24
src/chapter2/1174035_1.py	26
src/chapter2/1174035_2.py	26
src/chapter2/1174035_3.py	27
src/chapter2/1174035_4.py	27
src/chapter2/1174035_5.py	27
src/chapter2/1174035_6.py	28
src/chapter2/1174035_7.py	28
src/chapter2/1174035_8.py	29
src/chapter2/1174035_9.py	29
src/chapter2/1174035_10.py	30
src/chapter2/1174035_11.py	30
src/chapter2/1174035_2err.py	31
src/chapter2/1174042_1,1.py	31
src/chapter2/1174042_1,2.py	32
src/chapter2/1174042_1,3.py	32
src/chapter2/1174042_1,4.py	32
src/chapter2/1174042_1,5.py	32
src/chapter2/1174042_1,7.py	33
src/chapter2/1174042_2,1.py	33
src/chapter2/1174042_2,2.py	33
src/chapter2/1174042_2,3.py	33
src/chapter2/1174042_2,4.py	33
src/chapter2/1174042_2,5.py	34
src/chapter2/1174042_2,6.py	34
src/chapter2/1174042_2,7.py	34
src/chapter2/1174042_2,8.py	34
src/chapter2/1174042_2,9.py	34
src/chapter2/1174042_2,10.py	35

src/chapter2/1174042_2,11.py	36
src/chapter2/1174042_2err.py	36
src/1174050_teor.py	36
src/1174050_teor.py	37
src/1174050_teor.py	37
src/1174050_teor.py	37
src/1174050_teor.py	37
src/1174050_teor.py	38
src/1174050_teor.py	38
src/1174050_teor.py	38
src/1174050_teor.py	38
src/1174050_praktek.py	38
src/1174050_praktek.py	39
src/1174050_praktek.py	39
src/1174050_praktek.py	39
src/1174050_praktek.py	39
src/1174050_praktek.py	39
src/1174050_praktek.py	40
src/1174050_praktek.py	40
src/1174050_praktek.py	40
src/1174050_praktek.py	40
src/1174050_praktek.py	40
src/1174050_praktek.py	40
src/error1ka.py	40
src/chapter2/1174034_ichsan_teor.py	40
src/chapter2/1174034_ichsan_teor.py	41
src/chapter2/1174034_ichsan_teor.py	41
src/chapter2/1174034_ichsan_teor.py	41
src/chapter2/1174034_ichsan_teor.py	41
src/chapter2/1174034_ichsan_teor.py	41
src/chapter2/1174034_ichsan_teor.py	42
src/chapter2/1174034_ichsan_teor.py	42
src/chapter2/1174034_ichsan_teor.py	42
src/chapter2/1174034_ichsan_praktek.py	42
src/chapter2/1174034_ichsan_praktek.py	43
src/chapter2/1174034_ichsan_praktek.py	43
src/chapter2/1174034_ichsan_praktek.py	43
src/chapter2/1174034_ichsan_praktek.py	43
src/chapter2/1174034_ichsan_praktek.py	43

src/chapter2/1174034_ichsan_praktek.py	43
src/chapter2/1174034_ichsan_praktek.py	44
src/chapter2/1174034_ichsan_praktek.py	44
src/chapter2/1174034_ichsan_praktek.py	44
src/chapter2/1174034_ichsan_praktek.py	44
src/chapter2/errorrrr.py	44
src/chapter3/chap3_1174035_teor_i.py	45
src/chapter3/chap3_1174035_teor_i.py	46
src/chapter3/chap3_1174035_teor_i.py	46
src/chapter3/chap3_1174035_teor_i.py	46
src/chapter3/chap3_1174035_teor_i.py	46
src/chapter3/chap3_1174035_teor_i.py	47
src/chapter3/chap3_1174035_teor_i.py	47
src/chapter3/chap3_1174035_1.py	47
src/chapter3/chap3_1174035_2.py	48
src/chapter3/chap3_1174035_3.py	48
src/chapter3/chap3_1174035_4.py	50
src/chapter3/chap3_1174035_5.py	50
src/chapter3/chap3_1174035_6.py	51
src/chapter3/chap3_1174035_7.py	51
src/chapter3/chap3_1174035_8.py	52
src/chapter3/chap3_1174035_9.py	52
src/chapter3/chap3_1174035_10.py	53
src/chapter3/chap3_1174035_3lib.py	53
src/chapter3/chap3_1174035_main.py	55
src/chapter3/chap3_1174035_kelas3lib.py	55
src/chapter3/chap3_1174035_main.py	58
src/chapter3/chap3_1174035_error.py	59
src/chapter3/chap3_1174035_main.py	60
src/1174056_praktek.py	61
src/1174056_praktek.py	61
src/1174056_praktek.py	61
src/1174056_praktek.py	61
src/1174056_praktek.py	61
src/1174056_praktek.py	62
src/1174056_praktek.py	62
src/1174056_praktek.py	62
src/1174056_praktek.py	62

src/1174056_praktek.py	64
src/1174056_praktek.py	64
src/1174056_praktek.py	64
src/1174056_praktek.py	65
src/1174056_praktek.py	65
src/1174056_praktek.py	65
src/1174056_praktek.py	65
src/1174056_praktek.py	65
src/1174056_praktek.py	66
src/mainn.py	66
src/mainn.py	66
src/1174056_praktek.py	66
src/chapter3/1174042_1,1.py	67
src/chapter3/1174042_1,1,1.py	67
src/chapter3/1174042_1,1,2.py	67
src/chapter3/1174042_1,2.py	67
src/chapter3/1174042_1,3.py	67
src/chapter3/1174042_1,4.py	68
src/chapter3/1174042_1,2.py	68
src/chapter3/1174042_1,6.py	68
src/chapter3/1174042_1,6.py	68
src/chapter3/1174042_2,1.py	69
src/chapter3/1174042_2,2.py	69
src/chapter3/1174042_2,3.py	69
src/chapter3/1174042_2,4.py	70
src/chapter3/1174042_2,5.py	70
src/chapter3/1174042_2,6.py	70
src/chapter3/1174042_2,7.py	71
src/chapter3/1174042_2,8.py	71
src/chapter3/1174042_2,9.py	71
src/chapter3/1174042_2,10.py	72
src/chapter3/1174042_main.py	72
src/chapter3/1174042_main.py	72
src/chapter3/1174042_2err.py	73

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat
Februari, 2019*

ACKNOWLEDGMENTS

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

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

R. M. A.

ACRONYMS

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

GLOSSARY

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

SYMBOLS

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

- \mathcal{B} Number of Beats

INTRODUCTION

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

Informatics Research Center
Bandung, Jawa Barat, Indonesia

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

Rumusnya cuman sebagai contoh aja biar keren[1].

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

BAB 1

JUDUL BAGIAN PERTAMA

1.1 Irvan Rizkiansyah

1.2 Python

1.2.1 Background

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

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

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

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

1.2.2 Problems

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

1.2.3 Objective and Contribution

1.2.3.1 Objective

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

1.2.3.2 Contribution

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

1.2.4 Scoop and Environment

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

1.3 Luthfi Muhammad Nabil_1174035

1.3.1 Background

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

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

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

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

1.3.2 Problems

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

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

1.3.3 Objective and Contribution

1.3.3.1 Objective

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

1.3.3.2 Contribution Pada materi ini, penulis menggunakan Python.

1.3.4 Scoop and Environment

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

1.4 Hagan Rowlenstino/1174040

1.4.1 Background

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

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

1.4.2 Problems

1. Banyak pemrograman yang penggunaannya kompleks

1.4.3 Objective and Contribution

1.4.3.1 Objective

1. Dapat memudahkan pemrograman dengan bahasa pemrograman yang tepat

1.4.3.2 Contribution

1. Menggunakan Python sebagai bahasa pemrograman

1.4.4 Scoop and Environment

1. Mengimplementasikan Python dalam pemrograman

1.5 Rangga Putra Ramdhani.1174056

1.5.1 Background

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

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

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

1.5.2 Problems

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

1.5.3 Objective and Contribution

1.5.3.1 Objective

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

1.5.3.2 Contribution

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

1.5.4 Scope and Environment

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

BAB 2

JUDUL BAGIAN KEDUA

2.1 Alit Fajar Kurniawan 1174057

2.1.1 Teori

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

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

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

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

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

```

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

```

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

```

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

```

```

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

```

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

```

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

```

```

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

```

```

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

```

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

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

```

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

```

2.1.2 praktek

1. Jawaban soal no 1

```

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

```

2. Jawaban soal no 2

```

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

```

3. Jawaban soal no 3

```

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

```

4. Jawaban soal no 4

```

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

```

5. Jawaban soal no 5

```

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

```

6. Jawaban soal no 6

```

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

```

7. Jawaban soal no 7

```

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

```

8. Jawaban soal no 8

```

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

```

9. Jawaban soal no 9

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

10. Jawaban soal no 10

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

11. Jawaban soal no 11

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

2.1.3 Keterampilan dan penanganan error

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

2.2 IrvanRizkiansyah/1174043

2.2.1 Teori

1. Pada python variabel tidak perlu dideklarasikan, pendeklarasian terjadi secara otomatis pada saat memberikan suatu nilai atau data ke variabel. Terdapat beberapa jenis tipe data variabel pada python, diantaranya :
 - Python Numbers, dimana akan menyimpan data yang berupa angka. Penggunaan pada python sebagai berikut : `var1 = 5 var2 = 48.9`
 - Python Text, dimana akan menyimpan data yang berupa teks ataupun karakter. Penggunaan pada python harus diapitkan oleh tanda petik ("..."), contohnya : `nama = "Irvan" jnskelamin = "L"`
 - Python Boolean, dimana yang hanya memiliki 2 nilai yaitu True dan False saja. penggunaan pada python huruf pertama harus kapital, contohnya : `var3 = True var4 = False`
2.
 - Meminta input pada user `nama = input("Masukkan Nama Anda : ")`
 - menampilkan output `print "Hello Nama Saya Adalah",nama`
3.
 - Operator tambah `a = b + c`
 - Operator kurang `a = b - c`
 - Operator kali `a = b * c`
 - Operator bagi `a = b / c`
 - Konversi integer ke string `konvVar = str(var1)`

- Konversi string ke integer `konvVar = int(var2)`

4. ▪ Pengulangan `for`, kemampuan mengulang proses data menggunakan urutan apapun, seperti `list`. contoh penggunaan pada Python dan contoh kode adalah :

```
for i in range(10):
    print(i)
```

- Pengulangan `while`, kemampuan mengulang proses data yang akan terus berlanjut jika kondisinya `True`. contoh penggunaan pada Python dan contoh kode adalah :

```
i= 0
while i < 10 :
    i=i+1
    print ("loop ke =", i)
```

5. Pengambilan keputusan berguna untuk menentukan tindakan apa yang akan diambil sesuai dengan kondisi yang ada. Contohnya :

```
nilai = 9
if(nilai > 7):
    print("Selamat Anda Lulus")
else:
    print("Maaf Anda Tidak Lulus")
```

Dan untuk kondisi di dalam kondisi contohnya :

```
gaji = 10000000
berkeluarga = True
if gaji > 30000000:
    print "Gaji sudah diatas UMR"
if berkeluarga:
    print "Wajib ikutan asuransi dan menabung untuk pensiun"
else:
    print "Tidak perlu ikutan asuransi"
else:
    print "Gaji belum UMR"
```

6. ▪ **Syntax Errors**, Salahnya dalam penulisan sintaks. cara penanganannya adalah dengan menganalisa bagian kode yang error dan memperbaiki sintaks tersebut.
- **Exceptions**, error yang terjadi karena sintaks tidak dapat dieksekusi. cara penanganannya adalah dengan menganalisa bagian kode yang error dan memperbaiki sintaks tersebut.

7. Try Except adalah cara penanganan error pada Python. Contohnya :

```
x = 0
try:
    x = 1 / 0
except Exception, e:
    print e
```

2.2.2 Keterampilan Pemrograman

```
1 NPM = 1174043
2 char = "#"
3 if NPM%3 == 0:
4     char = "*"
5 if NPM%3 == 1:
6     char = "#"
7 if NPM%3 == 2:
8     char = "+"
9 baris = []
10 baris.append("  ##      ##      #####  ##      #####  ##
      #####  ")
11 baris.append(" #####  #####  ##      ##  ##      ##      ##  ##
      ##  ##      ##  ")
12 baris.append("  ##      ##      ##      ##      ##      ##      ##
      ##      ##  ")
13 baris.append("  ##      ##      ##      ##      ##      ##      ##
      ##      #####  ")
14 baris.append("  ##      ##      ##      #####  ##      ##
      #####      ##  ")
15 baris.append("  ##      ##      ##      ##      ##      ##
      ##      ##      ")
16 baris.append(" #####  #####  ##      ##      #####
      ##      #####  ")
17 a = 0
18 for x in baris:
19     print(baris[a].replace("#", char))
20     a+=1
```

```
2 print('Masukkan NPM Anda : ')
2 NPM = input()
3 y = NPM[5:7]
4 g = int(y)
5 for i in range(g):
6     print('Halo ' + NPM, 'Apa Kabar?')
```

```
3 print('Masukkan NPM Anda : ')
2 NPM = input()
3 y = NPM[4:7]
4 x = NPM[4:5]
5 y = NPM[5:6]
6 z = NPM[6:7]
7 g = (int(x) + int(y) + int(z))
```

```

8 for i in range(g):
9     print('Halo ' + y, 'Apa Kabar?')

```

```

41 print('Masukkan NPM Anda : ')
2  NPM = input()
3  y = NPM[4:5]
4  print('Halo ' + y, 'Apa Kabar?')

```

```

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

```

```

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

```

```

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

```

```

81 a = 1
2  b = 1
3  c = 7
4  d = 4
5  e = 0
6  f = 4
7  g = 3
8  print(a)
9  print(b)
10 print(c)
11 print(d)
12 print(e)
13 print(f)
14 print(g)

```

```
9 1 a = 1
  2 b = 1
  3 c = 7
  4 d = 4
  5 e = 0
  6 f = 4
  7 g = 3
  8 array =[]
  9 array.append(a)
 10 array.append(b)
 11 array.append(c)
 12 array.append(d)
 13 array.append(e)
 14 array.append(f)
 15 array.append(g)
 16 for x in array:
 17     if x != 0:
 18         if x%2==0:
 19             print(x, end = " ")
```

```
10 1 a = 1
    2 b = 1
    3 c = 7
    4 d = 4
    5 e = 0
    6 f = 4
    7 g = 3
    8 array =[]
    9 array.append(a)
   10 array.append(b)
   11 array.append(c)
   12 array.append(d)
   13 array.append(e)
   14 array.append(f)
   15 array.append(g)
   16 for x in array:
   17     if x != 0:
   18         if x%2==1:
   19             print(x, end = " ")
```

```
11 1 a = 1
    2 b = 1
    3 c = 7
    4 d = 4
    5 e = 0
    6 f = 4
    7 g = 3
    8 array =[]
    9 array.append(a)
   10 array.append(b)
   11 array.append(c)
   12 array.append(d)
   13 array.append(e)
   14 array.append(f)
   15 array.append(g)
```

```

16 for x in array :
17     if x != 0:
18         i = 1
19         bil = 0
20         while i <= x:
21             if x%i==0:
22                 bil+=1
23             i+=1
24         if bil == 2:
25             print(x)

```

2.2.3 Keterampilan Penanganan Error

1. TypeError yaitu error di dalam tipe data saat melakukan substring dan ingin memasukkannya ke dalam kondisi for yang hanya menerima tipe int. jadi harus merubah tipe inputan yaitu string menjadi integer.

```

21 a = 2
2 b = 'tes'
3 try:
4     print(a + b)
5 except TypeError:
6     print("Integer dan String Tidak Dapat Dijumlah Karena Berbeda Tipe Data")

```

2.3 Hagan Rowlenstino/1174040

2.3.1 Teori

1. tipe data teks : ada string yaitu kumpulan karakter dan char adalah karakter. penulisannya harus diapit dengan tanda petik 1,2, ataupun 3 ('..'), (".."), (""...") , ("""... """)

tipe data angka : ada float yaitu bilangan pecahan dan integer yaitu bilangan bulat. penulisannya yaitu dengan menginisialisasikan nama variable lalu masukkan angka (x = 30)

tipe data boolean : tipe yang memiliki dua nilai yaitu true dan false. penggunaannya huruf pertamanya harus kapital True dan False.

2. input().inisialisasikan input tersebut x = input() lalu print(x)
3. +,*,-,/. misal a = '10' maka integerr = int(a) dan misal a= 10 maka stringg = string(a)
4. while : untuk perulangan yang tidak pasti

i = 0

```

while True:
    if i < 10:
        print "Saat ini i bernilai: ", i
        i = i + 1
    elif i >= 10:
        break

    for : untuk perulangan yang pasti
for i in range(0, 10):
    print i

```

5. if kondisi:
 hasil

```

    dan
    if kondisi:
hasil
if kondisi:
    hasil

```

6. type error = ubah tipe str jadi int

7. taruh try : diatas sintaks yang ingin diketahui jika terjadi error lalu enter dan tulis except: lalu tekan enter dan masukkan tulisan yang akan ditampilkan.

```

a = 2
b = 'as'
try:
    print(a + b)
except TypeError:
    print("Integer dan String Tidak Dapat  
Dijumlah Karena Berbeda Tipe")

```

2.3.2 Keterampilan Pemrograman

```

1 NPM = 1174040
2 char = ""
3 if NPM%3 == 0:
4     char = "*"
5 if NPM%3 == 1:
6     char = "#"
7 if NPM%3 == 2:
8     char = "+"
9 row = []
10 row.append("  ##      ##      #####  ##      #####  ##
      #####")

```

```

11 row.append("####      ##      ##      ##      ##      ##      ##      ##")
12 row.append("##      ##      ##      ##      ##      ##      ##      ##")
13 row.append("##      ##      ##      ##      ##      ##      ##      ##")
14 row.append("##      ##      ##      #####      ##      ##")
15 row.append("#####      ##      ##      ##      ##      ##      ##")
16 row.append("#####      #####      ##      ##      #####      ##")
17 a = 0
18 for x in row:
19     print(row[a].replace("#", char))
20     a+=1

```

```

21 print('input NPM')
2 NPM = input()
3 y = NPM[5:7]
4 g = int(y)
5 for i in range(g):
6     print('Halo ' + NPM, 'Apa Kabar?')

```

```

31 print('input NPM')
2 NPM = input()
3 y = NPM[4:7]
4 x = NPM[4:5]
5 y = NPM[5:6]
6 z = NPM[6:7]
7 g = (int(x) + int(y) + int(z))
8 for i in range(g):
9     print('Halo ' + y, 'Apa Kabar?')

```

```

41 print('input NPM')
2 NPM = input()
3 y = NPM[4:5]
4 print('Halo ' + y, 'Apa Kabar?')

```

```

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

```

```

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

```

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

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

```
81 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 4
7 g = 0
8 print(a)
9 print(b)
10 print(c)
11 print(d)
12 print(e)
13 print(f)
14 print(g)
```

```
91 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 4
7 g = 0
8 array = []
9 array.append(a)
10 array.append(b)
11 array.append(c)
12 array.append(d)
13 array.append(e)
14 array.append(f)
15 array.append(g)
16 for x in array:
17     if x != 0:
18         if x%2==0:
19             print(x, end = " ")
```

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



```

7 g = 0
8 array = []
9 array.append(a)
10 array.append(b)
11 array.append(c)
12 array.append(d)
13 array.append(e)
14 array.append(f)
15 array.append(g)
16 for x in array:
17     if x != 0:
18         if x%2==1:
19             print(x, end = " ")

```

```

11 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 4
7 g = 0
8 array = []
9 array.append(a)
10 array.append(b)
11 array.append(c)
12 array.append(d)
13 array.append(e)
14 array.append(f)
15 array.append(g)
16 for x in array:
17     if x != 0:
18         i = 1
19         bil = 0
20         while i <= x:
21             if x%i==0:
22                 bil+=1
23             i+=1
24         if bil == 2:
25             print(x)

```

2.3.3 Keterampilan Penanganan Error

1. TypeError yaitu error di dalam tipe data disaat melakukan substring dan ingin memasukkannya ke dalam kondisi for yang hanya menerima tipe int. jadi harus merubah tipe inputan yaitu string menjadi integer.

```

2 a = 2
3 b = 'as'
4 try:
5     print(a + b)
6 except TypeError:
7     print("Integer dan String Tidak Dapat Dijumlah Karena Berbeda Tipe")

```

2.4 Muhammad Iqbal Panggabean

2.4.1 Teori

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

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

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

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

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

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

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

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

```

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

```

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

```

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

```

```

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

```

```

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

```

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

```

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

```

2.4.2 praktek

1. Jawaban soal no 1

```

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

```

2. Jawaban soal no 2

```

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

```

3. Jawaban soal no 3

```

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

```

4. Jawaban soal no 4

```

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

```

5. Jawaban soal no 5

```

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

```

6. Jawaban soal no 6

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

7. Jawaban soal no 7

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

8. Jawaban soal no 8

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

9. Jawaban soal no 9

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

10. Jawaban soal no 10

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

11. Jawaban soal no 11

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

2.4.3 Keterampilan dan penanganan eror

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

2.5 Luthfi M. Nabil/1174035

2.5.1 Teori

1. Berikut merupakan jenis - jenis variabel yang terdapat pada python :
 - Jenis variabel Teks (String) : Merupakan jenis variabel untuk menampung karakter. Cara penulisannya harus diapit dengan tanda petik 1 atau 2 ('.'), ("..")

- Jenis variabel numeric(Integer, Float) : Jenis variabel ini menampung nilai berupa angka diantaranya bilangan bulat (integer) dan bilangan koma (float) Penulisannya yaitu dengan menginisialisasikan nama variable lalu masukkan angka (x = 30, x=3.3)
 - Jenis variabel pengkondisian : tipe yang memiliki dua nilai yaitu true dan false. penggunaannya huruf pertamanya harus kapital True dan False.
2. input().inisialisasikan input tersebut x = input() lalu print(x)
 3. Operator dasar aritmatika dan mengubah string ke integer dan integer ke string :
 - Jenis - jenis operator aritmatika : Penjumlahan (+),Perkalian (*), Pengurangan(-),Pembagian(/).
 - Convert int to string dan sebaliknya : misal a = '10' maka integer = int(a) dan misal a= 10 maka string = string(a)
 4. While : untuk perulangan yang memiliki kondisi lebih bebas/tidak terpaku

```

        i = 20
while True:
    print "Saat ini i bernilai: ", i
    i = i - 1

    for : untuk perulangan yang pasti
for i in range(0, 10):
    print i

```

5.


```

      if kondisi:
      hasil
      dan
      if kondisi:
      hasil
      if kondisi:
      hasil
      
```

6. type error = ubah tipe str jadi int, index error = array index tidak diketahui
7. taruh try : diatas sintaks yang ingin diketahui jika terjadi error lalu enter dan tulis except: lalu tekan enter dan masukkan tulisan yang akan ditampilkan.

```

a = 2
b = 'Coba Coba'
try:

```

```

    print(a + b)
except TypeError:
    print("Integer dan String Tidak Dapat
        Dijumlah Karena Berbeda Tipe")

```

2.5.2 Keterampilan Pemrograman

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Mar  4 20:30:07 2019
4
5  @author: Intel
6  """
7
8  NPM = 1174035
9  char = "#"
10 if NPM%3==0:
11     char = "*"
12 if NPM%3==1:
13     char = "#"
14 if NPM%3==2:
15     char = "+"
16 line = []
17 line.append(" ##          ##          #####          ##          ##
18           #####          #####")
19 line.append("#####          ##          ##          ##          ##
20           ##          ##          ##          ")
21 line.append(" ##          ##          ##          ##          ##
22           ##          ##          ")
23 line.append(" ##          ##          ##          ##          ##
24           ##          #####          ")
25 line.append(" ##          ##          ##          ##          ##
26           ##          ##          ##          ")
27 line.append("#####          ##          ##
28           #####          ##### ")
29
30 a=0
31 for x in line:
32     print(line[a].replace("#", char))
33     a+=1

```

```

2  # -*- coding: utf-8 -*-
3  """
4  Created on Mon Mar  4 19:28:47 2019
5
6  @author: Intel
7  """
8
9  print("Input : ")
10 NPM = input()
11 x=1
12 print("Output : ")
13 while x <= 87:
14     x+=1

```

```
13 print("Halo, "+NPM+" apa kabar?")
```

```
31 # -*- coding: utf-8 -*-
32 """
33 Created on Mon Mar  4 19:40:11 2019
34
35 @author: Intel
36 """
37 print("Input : ")
38 NPM = input()
39 jumlah = len(NPM)
40 a = int(NPM[jumlah-3])
41 b = int(NPM[jumlah-2])
42 c = int(NPM[jumlah-1])
43 x=1
44 while x <= (a+b+c):
45     print("Output : "+NPM[jumlah-3:])
46     x+=1
```

```
41 # -*- coding: utf-8 -*-
42 """
43 Created on Mon Mar  4 19:55:35 2019
44
45 @author: Intel
46 """
47
48 print("Input : ")
49 NPM = input()
50 jumlah = len(NPM)
51 print("Output : ")
52 print("Halo, "+NPM[jumlah-3]+" apa kabar?")
```

```
51 # -*- coding: utf-8 -*-
52 """
53 Created on Mon Mar  4 19:57:12 2019
54
55 @author: Intel
56 """
57
58 a = 1
59 b = 1
60 c = 7
61 d = 4
62 e = 0
63 f = 3
64 g = 5
65
66 arr = []
67 arr.append(a)
68 arr.append(b)
69 arr.append(c)
70 arr.append(d)
71 arr.append(e)
72 arr.append(f)
```



```

23 arr.append(g)
24 for x in arr:
25     print(x, end = " ")

```

```

61 # -*- coding: utf-8 -*-
2  """
3  Created on Mon Mar  4 20:05:50 2019
4
5  @author: Intel
6  """
7
8  a = 1
9  b = 1
10 c = 7
11 d = 4
12 e = 0
13 f = 3
14 g = 5
15
16 arr = []
17 arr.append(a)
18 arr.append(b)
19 arr.append(c)
20 arr.append(d)
21 arr.append(e)
22 arr.append(f)
23 arr.append(g)
24 jum = 0
25 for x in arr:
26     jum+=x
27 print("Hasil : " +str(jum))

```

```

71 # -*- coding: utf-8 -*-
2  """
3  Created on Mon Mar  4 20:06:55 2019
4
5  @author: Intel
6  """
7
8  a = 1
9  b = 1
10 c = 7
11 d = 4
12 e = 0
13 f = 3
14 g = 5
15
16 arr = []
17 arr.append(a)
18 arr.append(b)
19 arr.append(c)
20 arr.append(d)
21 arr.append(e)
22 arr.append(f)
23 arr.append(g)

```

```

24 jum = 1
25 for x in arr:
26     jum*=x
27 print("Hasil : "+str(jum))

```

```

81 # -*- coding: utf-8 -*-
    """
    2
    3 Created on Mon Mar  4 20:10:22 2019
    4
    5 @author: Intel
    6 """
    7
    8 a = 1
    9 b = 1
   10 c = 7
   11 d = 4
   12 e = 0
   13 f = 3
   14 g = 5
   15
   16 arr = []
   17 arr.append(a)
   18 arr.append(b)
   19 arr.append(c)
   20 arr.append(d)
   21 arr.append(e)
   22 arr.append(f)
   23 arr.append(g)
   24 for x in arr:
   25     print(x)

```

```

91 # -*- coding: utf-8 -*-
    """
    2
    3 Created on Mon Mar  4 20:10:40 2019
    4
    5 @author: Intel
    6 """
    7
    8 a = 1
    9 b = 1
   10 c = 7
   11 d = 4
   12 e = 0
   13 f = 3
   14 g = 5
   15
   16 arr = []
   17 arr.append(a)
   18 arr.append(b)
   19 arr.append(c)
   20 arr.append(d)
   21 arr.append(e)
   22 arr.append(f)
   23 arr.append(g)
   24 for x in arr:

```

```

25     if x!=0:
26         if x%2==0:
27             print(x, end = "")

```

```

101 # -*- coding: utf-8 -*-
2     """
3     Created on Mon Mar  4 20:18:04 2019
4
5     @author: Intel
6     """
7
8     a = 1
9     b = 1
10    c = 7
11    d = 4
12    e = 0
13    f = 3
14    g = 5
15
16    arr = []
17    arr.append(a)
18    arr.append(b)
19    arr.append(c)
20    arr.append(d)
21    arr.append(e)
22    arr.append(f)
23    arr.append(g)
24    for x in arr:
25        if x!=0:
26            if x%2==1:
27                print(x, end = "")

```

```

111 # -*- coding: utf-8 -*-
2     """
3     Created on Mon Mar  4 20:18:25 2019
4
5     @author: Intel
6     """
7
8     a = 1
9     b = 1
10    c = 7
11    d = 4
12    e = 0
13    f = 3
14    g = 5
15
16    arr = []
17    arr.append(a)
18    arr.append(b)
19    arr.append(c)
20    arr.append(d)
21    arr.append(e)
22    arr.append(f)
23    arr.append(g)

```

```

24 for x in arr:
25     if x!=0:
26         i = 1
27         bil = 0
28         while i <= x:
29             if x%i==0:
30                 bil+=1
31             i+=1
32         if bil == 2:
33             print(x)

```

2.5.3 Keterampilan Penanganan Error

- TypeError yaitu error di dalam variabel disaat melakukan substring dan ingin memasukkannya ke dalam kondisi for yang hanya menerima tipe int. jadi harus merubah tipe inputan yaitu string menjadi integer.
 - IndexError yaitu error saat array dengan index yang telah dipilih tidak ditemukan atau tidak memiliki nilai

```

21 # -*- coding: utf-8 -*-
22 """
23 Created on Mon Mar  4 19:28:47 2019
24
25 @author: Intel
26 """
27 a = 2
28 b = 'Coba'
29 try:
30     print(a + b)
31 except TypeError:
32     print("Integer dan String Tidak Dapat Dijumlah Karena Berbeda Tipe")

```

2.6 Faisal Najib Abdullah 1174042

2.6.1 Teori

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

```

1 x = "aku"
2 y = "sayang"
3 z = "najib"
4 a = x + y + z
5 print(z)
6
7 #hasil aku sayang najib

```

Variabel x memiliki nilai aku, variable y memiliki nilai sayang, dan variabel z memiliki nilai najib. karna memiliki type data string maka kata kata tersebut jika di tambahkan berubah menjadi sebuah kalimat

2. Input, untuk membuat kode input, pertama buat variabel `x` yang berisi input seperti pada contoh jika di run maka langsung diminta untuk memasukan NIM ketika di enter hasilnya berupa Hello, 1174042

```
1 print("NPM: ")
2 x = input()
3 print("Hello , " + x)
```

3. Untuk merubah type data dari string ke integer, tambahkan kata `int` lalu kurung buka lalu nama variabel yang akan dirubah dan kurung tutup seperti pada contoh.

```
1 x = 5
2 y = 3
3 q = "7"
4 z = "8"
5
6 print(x + y)
7 print(x - y)
8 print(x * y)
9 print(x / y)
10 print(int(q) + int(z))
11 print(str(x) + str(y))
```

4. untuk perulangan disini menggunakan `while` variabel `i` bernilai 0, kemudian `while` `i` lebih kecil dari 6 jika benar maka akan terus dilakukan pengulangan dan jika salah tidak akan dilakukan pengulangan, `i` selalu bertambah 1, dan menampilkan nilai `i`.

```
1 x = 5
2 i = 0
3 while i < 6:
4     i += 1
5     print(i)
```

5. membuat 2 variabel `a` dan `b`, variabel `a` bernilai 200 dan `b` bernilai 33 jika `b` lebihbesar dari `a` maka akan menampilkan sesuai perintah seperti contoh.

```
1 a = 200
2 b = 33
3 if b > a:
4     print("b lebih kecil dari a")
5 elif a == b:
6     print("a dan b tidak sama")
7 else:
8     print("a lebih besar dari b")
```

6. Jenis error yang sering di alami pada python

- menjumlahkan bilangan yang berbeda type data. Solosinya rubah dan sesuaikan type data yang dibutuhkan
- sepasi pada kondisi yang harus sejajar. Sejajarkan posisi sesuai kondisi

- Typo. Cek kembali agar tidak terjadi kesalahan code
7. untuk menggunakan try, pertama tuliskan coba terlebih dahulu code apakah terjadi error atau tidak. Jika terjadi error copy TypeError kemudian tuliskan try sebelum line yg error, dibawah line yg error tuliskan except dan paste typeerror yang sebelumnya sudah di copy, kemudian tuliskan kenapa bisa terjadi error menggunakan katakata sendiri.

```

1 x = 1
2 c = "7"
3
4 try :
5     d = x + c
6 except TypeError:
7     print("type data berbeda")

```

2.6.2 Keterampilan Pemrograman

```

1 print(1174042%3)
2 print("# # ### # # # #")
3 print("# # # ## # # ## #")
4 print("# # # ### # # ### #")
5 print("# # # # # # # ###")

```

```

2 print("NPM1:")
2 x = input()
3 a = 0
4 b = int(x[5:7])
5 while a < b :
6     a += 1
7     print("Hello , " + x + " Apa Kabar?")

```

```

3 print("NPM1:")
2 x = input()
3 a = 0
4 k = int(x[4])
5 p = int(x[5])
6 i = int(x[6])
7 b = x[4:7]
8 c = k+p+i
9
10 while a < c:
11     a += 1
12     print("Hello , " + b + " Apa Kabar?")

```

```

4 print("NPM1:")
2 x = input()
3 p = x[4]
4 print("Hello , " + p + " Apa Kabar?")

```

```

51 print("NPM1:")
2  a = 1
3  b = 1
4  c = 7
5  d = 4
6  e = 0
7  f = 4
8  g = 2
9  x = str(a)+str(b)+str(c)+str(d)+str(e)+str(f)+str(g)
10 print("Hello , " + x + " Apa Kabar?")

```

```

61 print("NPM1:")
2  a = 1
3  b = 1
4  c = 7
5  d = 4
6  e = 0
7  f = 4
8  g = 2
9  x = a+b+c+d+e+f+g
10 print(x)

```

```

71 print("NPM1:")
2  a = 1
3  b = 1
4  c = 7
5  d = 4
6  e = 0
7  f = 4
8  g = 2
9  x = a*b*c*d*e*f*g
10 print(x)

```

```

81 print("NPM1:")
2  a = 1
3  b = 1
4  c = 7
5  d = 4
6  e = 0
7  f = 4
8  g = 2
9  print(a)
10 print(b)
11 print(c)
12 print(d)
13 print(e)
14 print(f)
15 print(g)

```

```

91 print("NPM1:")
2  a = 1
3  b = 1
4  c = 7
5  d = 4

```

```
6 e = 0
7 f = 4
8 g = 2
9
10 if a%2 == 0:
11     print(a)
12
13 if b%2 == 0:
14     print(b)
15
16 if c%2 == 0:
17     print(c)
18
19 if d%2 == 0:
20     print(d)
21
22 if e%2 == 0:
23     print(e)
24
25 if f%2 == 0:
26     print(f)
27
28 if g%2 == 0:
29     print(g)
```

```
10. print("NPM1:")
2 a = 1
3 b = 1
4 c = 7
5 d = 4
6 e = 0
7 f = 4
8 g = 2
9
10 if a%2 == 1:
11     print(a)
12
13 if b%2 == 1:
14     print(b)
15
16 if c%2 == 1:
17     print(c)
18
19 if d%2 == 1:
20     print(d)
21
22 if e%2 == 1:
23     print(e)
24
25 if f%2 == 1:
26     print(f)
27
28 if g%2 == 1:
29     print(g)
```



```

11 a = 1
12 b = 1
13 c = 7
14 d = 4
15 e = 0
16 f = 4
17 g = 0
18 array = []
19 array.append(a)
20 array.append(b)
21 array.append(c)
22 array.append(d)
23 array.append(e)
24 array.append(f)
25 array.append(g)
26 for x in array:
27     if x != 0:
28         i = 1
29         bil = 0
30         while i <= x:
31             if x%i==0:
32                 bil+=1
33             i+=1
34         if bil == 2:
35             print(x)

```

2.6.3 Keterampilan Penanganan Error

1. Pada saat mengerjakan praktek kedua ini error hanya pada kesalahan type data yaitu `TypeError`;, solusinya yaitu merumah type data.

```

21 a = 1
22 b = 10
23 c = "7"
24
25 try :
26     d = a + c
27 except TypeError:
28     print("ini teh beda atuhh type datanya mana bisa bersatu")

```

2.7 Dika Sukma Pradana 1174050

2.7.1 Teori

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

```

1 c = 7

```

```

2 d = "Dikasukmap"
3
4 print(c)
5 print(d)

```

2. Kode input user dan melakukan output ke layar

```

1 npm = input()
2 print(npm)

```

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

```

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

```

4. Perulangan Macam perulangan di dalam python yaitu perulangan while dan perulangan for

```

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

```

```

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

```

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

```

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

```

```

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

```

```

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

```

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

7. Cara memakai try except Cara pemakaian try except :

```

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

```

2.7.2 Praktek

1. Jawaban soal no 1

```

1 print ("###      ###      #####      ###      ###      #####
2 print ("###      ###      #####      ###      ###      #####
3 print ("###      ###      ###      ###      ###      ###      ###
4 print ("###      ###      ###      ###      ###      ###      ###
5 print ("###      ###      ###      #####      ###      ###

```

```

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

```

2. Jawaban soal no 2

```

1 npm = input("Masukkan NPM :")
2 hitung = 0
3 while(hitung <= 50):
4     print("Halo , " + str(npm) + "Apa kabar?")
5     hitung = hitung + 1

```

3. Jawaban soal no 3

```

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

```

4. Jawaban soal no 4

```

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

```

5. Jawaban soal no 5

```

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

```

6. Jawaban soal no 6

```

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

```

7. Jawaban soal no 7

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

8. Jawaban soal no 8

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

9. Jawaban soal no 9

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

10. Jawaban soal no 10

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

11. Jawaban soal no 11

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

2.7.3 Keterampilan dan Penanganan Error

```
1 c = 1
2 d = "4"
3
4 try:
5     z = c - d
6     print(z)
7 except:
8     print("Error Gais, Ubah yaaaa!!")
```

2.8 Ichsan Hizman Hardy

2.8.1 Teori

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

```
1 c = 7
2 d = "Ichsan Hizman"
3
4 print(c)
5 print(d)
```

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

```

1 npm = input()
2 print(npm)

```

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

```

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

```

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

```

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

```

```

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

```

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

```

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

```

```

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

```

```

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

```

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

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

```

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

```

2.8.2 praktek

1. Jawaban soal no 1

```

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

```

```

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

```

2. Jawaban soal no 2

```

1 npm = input("Masukkan NPM :")
2 hitung = 0
3 while(hitung <= 34):
4     print("Halo , " + str(npm) + "Apa kabar?")
5     hitung = hitung +1

```

3. Jawaban soal no 3

```

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

```

4. Jawaban soal no 4

```

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

```

5. Jawaban soal no 5

```

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

```

6. Jawaban soal no 6

```

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

```

7. Jawaban soal no 7

```

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

```

8. Jawaban soal no 8


```

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

```

9. Jawaban soal no 9

```

1 print(d,e,f)

```

10. Jawaban soal no 10

```

1 print(a,b,c,g)

```

11. Jawaban soal no 11

```

1 print(a,b,c,g)

```

2.8.3 Keterampilan dan penanganan error

```

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

```

BAB 3

JUDUL BAGIAN KETIGA

3.0.1 Luthfi Muhammad Nabil/1174035

3.0.1.1 *Pemahaman Teori*

1. Fungsi adalah sintaks yang terdiri dari nama fungsi, parameter input variabel, dan variabel kembali. Pada python, nama fungsi diawali dengan def dan pada sintaks paling akhir (setelah parameter) adalah titik dua. Aturan penamaan dari fungsi sama dengan penamaan sebuah variabel yang salah satunya yaitu case sensitive. Untuk penulisan parameter tidak harus memasukan inputan dan batas untuk penulisan variabel pada parameter tidak memiliki batas atau bisa lebih dari satu dengan pemisah tanda koma. Nilai yang dapat dikembalikan oleh fungsi dapat berupa variabel yang mau dikembalikan. Berikut contoh dari coding fungsi :

```
1 ##No 1##
2 def Penambahan(a, b): #Inisiasi Fungsi dan Inputan Fungsi
3     c = a+b
4     return c #Pengembalian Fungsi
```

2. Paket merupakan sebuah file yang berisikan fungsi - fungsi yang dapat dipakai. Untuk pemanggilan fungsi diperlukan keyword import untuk memanggil paket tersebut, berikut contoh pemakaian dari paket :

```
1 ##No 2##
2 import kalkulator #Pemanggilan biasa
3 import kalkulator as kal #Pemanggilan dengan inisial
```

3. Class merupakan cetak biru dari sebuah objek yang dibuat. Objek merupakan instansi dari sebuah class. Atribut merupakan variabel atau yang menampung nilai pada sebuah objek. Fungsi adalah sebuah pembungkus kumpulan instruksi pada sebuah program. Berikut contohnya :

```
1 ##No 3##
2 #File Kalkulator.py
3 class kalkulator: #Kalkulator merupakan class
4     def __init__(self, a, b):
5         self.a = a
6         self.b = b
7     def Penambahan(self): #a dan b merupakan atribut, Penambahan
8         #merupakan fungsi
9         coba = self.a+self.b #Coba merupakan atribut
10        print(str(coba))
11 #File main.py
12 import Kalkulator
13
14 kal = Kalkulator.Kalkulator(1, 2) #Berikut merupakan objek
15 kal.penambahan()
```

4. Pemanggilan sebuah kelas diawali dari sebuah paket dipanggil terlebih dahulu, lalu kelas akan disimpan ke variabel untuk diinisiasi sebagai objek. Berikut contoh pemanggilan dari kelas :

```
1 ##No 4##
2 import Kalkulator
3
4 kal = Kalkulator
```

5. Pemakaian from kalkulator merupakan sebuah inisiasi untuk memanggil fungsi penambahan dari paket kalkulator yang dipanggil agar fungsi penambahandapat digunakan langsung tanpa menulis nama file dari paket yaitu kalkulator. Berikut Contohnya :

```
1 ##No 5##
2 from Kalkulator import Penambahan
3
4 coba = Penambahan(1, 2)
```

6. Pemakaian paket fungsi memanggil fungsi dari paket lain dan memanggil paket tersebut dengan tambahan nama asal paket dari fungsi yang akan dipanggil. Berikut pemakaiannya :

```

1 ##No 6##
2 import Kalkulator
3 hasil1 = Kalkulator.Penambahan(1, 2)
4 print(str(hasil1))

```

7. Pemakaian paket kelas sama halnya dengan fungsi hanya saja untuk paket kelas diinisiasikan terlebih dahulu lalu nilai variabel akan dikirim ke constructor dari class tersebut. Pada saat memanggil fungsi tidak perlu menggunakan inputan parameter karena nilai yang dikirim sudah disimpan pada constructor di class yang dipanggil. Berikut Contohnya :

```

1 ##No 7##
2 import Kalkulator
3
4 kal = Kalkulator.kalkulator(1,2)
5 hasil1 = kal.Penambahan()
6 print(str(hasil1))

```

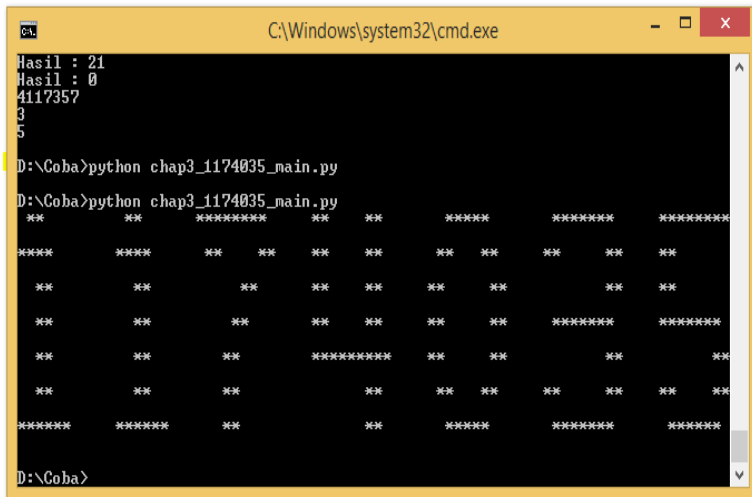
3.0.2 Ketrampilan Pemrograman

1. Membuat fungsi dengan inputan variabel NPM, dan melakukan print luaran huruf yang dirangkai dari tanda bintang, pagar, plus dari NPM kita. Untuk NPM mod 3=0 memakai bintang, NPM mod 3=1 memakai pagar, NPM mod 3 = 2 memakai tanda plus. Kodingnya :

```

1 def nomorsatu(NPM):
2     NPM = int(NPM)
3     char = ""
4     if NPM%3==0:
5         char = "*"
6     if NPM%3==1:
7         char = "#"
8     if NPM%3==2:
9         char = "+"
10    line = []
11    line.append("  ##          ##          #####          ##          ##
12    #####          #####          #####")
13    line.append("#####          ##          ##          ##          ##
14    ##          ##          ##          ")
15    line.append("  ##          ##          ##          ##          ##
16    ##          ##          ##          ")
17    line.append("  ##          ##          ##          ##          ##
18    ##          ##          ##          ")
19    line.append("  ##          ##          ##          ")
20    line.append("#####          #####          ##          ##
21    #####          #####          ##### ")
22    a=0
23    for x in line:
24        print(line[a].replace("#", char))
25        a+=1
26    return

```



Gambar 3.1 Screenshot No 1

- Membuat fungsi dengan inputan variabel NPM, dan lakukan perulangan untuk mengeluarkan print output sebanyak dua digit belakang NPM. Contoh NPM : 1174035 maka akan ada output sebanyak 35 kali dengan tulisan 'Hallo, 1174035 apa kabar?' Kodingnya :

```

1 def nomordua(NPM):
2     print("Input : "+NPM)
3     x=1
4     print("Output : ")
5     while x <= 87:
6         x+=1
7         print("Halo, "+NPM+" apa kabar?")
8     return

```

3. Membuat fungsi dengan inputan variabel NPM, dan melakukan print luaran output dengan perulangan berupa tiga karakter belakang dari NPM dijumlahkan. Lalu jumlah perulangan tersebut adalah total dari tiga karakter belakang NPM dijumlahkan. Kodingnya :

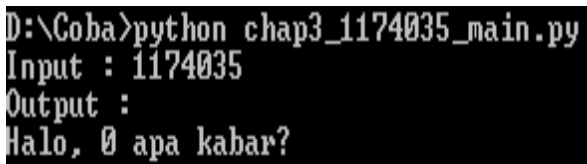
```

1 def nomortiga(NPM):
2     print("Input : "+NPM)
3     jumlah = len(NPM)
4     a = int(NPM[jumlah-3])
5     b = int(NPM[jumlah-2])
6     c = int(NPM[jumlah-1])
7     x=1
8     while x <= (a+b+c):
9         print("Output : "+NPM[jumlah-3:])
10        x+=1
11    return

```


4. Membuat fungsi dengan inputan variabel NPM, dan melakukan print hello world dan digit ketiga dari belakang dari NPM. contoh : NPM : 0, Output : Halo, 0 apa kabar? .Kodingnya :

```
1 def nomorempat(NPM):  
2     print("Input : "+NPM)  
3     jumlah = len(NPM)  
4     print("Output : ")  
5     print("Halo, "+NPM[jumlah-3]+" apa kabar?")  
6     return
```



```
D:\Coba>python chap3_1174035_main.py  
Input : 1174035  
Output :  
Halo, 0 apa kabar?
```

Gambar 3.4 Screenshot No 4

5. Membuat fungsi dengan inputan variabel NPM, dan menampilkan semua angka dari NPM tersebut secara berurutan kebawah. Kodingnya :

```
1 def nomorlima(NPM):  
2     NPM = list(NPM)  
3     for x in NPM:  
4         print(x)  
5     return
```



```
D:\Coba>python chap3_1174035_main.py  
1  
1  
7  
4  
0  
3  
5
```

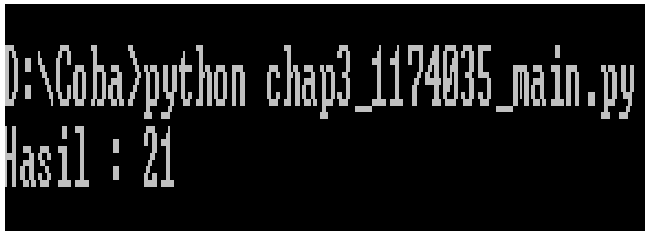
Gambar 3.5 Screenshot No 5

6. Membuat fungsi dengan inputan variabel NPM, didalamnya melakukan penjumlahan dari seluruh digit NPM tersebut. menggunakan perulangan atau kondisi. Kodingnya :

```

1 def nomorenam(NPM):
2     NPM = list(NPM)
3     jum = 0
4     for x in NPM:
5         jum+=int(x)
6     print("Hasil : "+str(jum))
7     return

```



```

D:\Coba>python chap3_1174035_main.py
Hasil : 21

```

Gambar 3.6 Screenshot No 6

7. Membuat fungsi dengan inputan variabel NPM, didalamnya melakukan perkalian dari seluruh digit NPM tersebut. menggunakan perulangan atau kondisi. Kod-
ingnya :

```

1 def nomortujuh(NPM):
2     NPM = list(NPM)
3     jum = 1
4     for x in NPM:
5         jum*=int(x)
6     print("Hasil : "+str(jum))
7     return

```



```

D:\Coba>python chap3_1174035_main.py
Hasil : 0

```

Gambar 3.7 Screenshot No 7

8. Membuat fungsi dengan inputan variabel NPM, lalu lakukan print seluruh angka genap dari setiap angka di NPM. Kodingnya :

```
1 def nomordelapan(NPM):
2     NPM = list(NPM)
3     for x in NPM:
4         if int(x) != 0:
5             if int(x) % 2 == 0:
6                 print(x, end=" ")
7     return
```

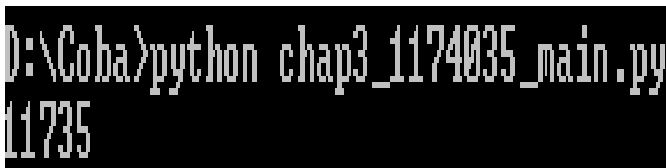


```
D:\Coba>python chap3_1174035_main.py
4
```

Gambar 3.8 Screenshot No 8

9. Membuat fungsi dengan inputan variabel NPM, lalu lakukan print seluruh angka ganjil dari setiap angka di NPM. Kodingnya :

```
1 def nomorsembilan(NPM):
2     NPM = list(NPM)
3     for x in NPM:
4         if int(x) != 0:
5             if int(x) % 2 == 1:
6                 print(x, end = " ")
7     return
```



```
D:\Coba>python chap3_1174035_main.py
11735
```

Gambar 3.9 Screenshot No 9

10. Membuat fungsi dengan inputan variabel NPM, lalu lakukan print seluruh angka prima dari setiap angka di NPM. Kodingnya :

```

1 def nomorsepuluh(NPM):
2     NPM = list(NPM)
3     for x in NPM:
4         if int(x) != 0:
5             i = 1
6             bil = 0
7             while i <= int(x):
8                 if int(x)%i == 0:
9                     bil += 1
10                    i += 1
11                if bil == 2:
12                    print(x)
13    return

```



```

D:\Coba>python chap3_1174035_main.py
7
3
5

```

Gambar 3.10 Screenshot No 10

11. Membuat Satu File library bernama 3lib.py yang berisi semua fungsi - fungsi dari setiap nomor pada soal praktek. Kodingnya :

```

1 ##File 3lib.py##
2 def nomorsatu(NPM):
3     NPM = int(NPM)
4     char = ""
5     if NPM%3==0:
6         char = "*"
7     if NPM%3==1:
8         char = "#"
9     if NPM%3==2:
10        char = "+"
11    line = []
12    line.append("  ##          ##          #####          ##          ##
13    #####          #####          #####")
14    line.append("#####          ##          ##          ##          ##
15    ##          ##          ##          ")
16    line.append("  ##          ##          ##          ##          ##
17    ##          ##          ##          ")
18    line.append("  ##          ##          ##          ##          ##
19    ##          #####          ##### ")

```

```

16 line.append(" ##          ##          ##          #####          ##
    ##          ##          ##")
17 line.append(" ##          ##          ##          ##          ##
    ##          ##          ##          ##")
18 line.append("#####          ##          ##
    #####          #####          ")
19 a=0
20 for x in line:
21     print(line[a].replace("#", char))
22     a+=1
23 return
24
25 def nomordua(NPM):
26     print("Input : "+NPM)
27     x=1
28     print("Output : ")
29     while x <= 87:
30         x+=1
31         print("Halo, "+NPM+" apa kabar?")
32     return
33
34 def nomortiga(NPM):
35     print("Input : "+NPM)
36     jumlah = len(NPM)
37     a = int(NPM[jumlah-3])
38     b = int(NPM[jumlah-2])
39     c = int(NPM[jumlah-1])
40     x=1
41     while x <= (a+b+c):
42         print("Output : "+NPM[jumlah-3:])
43         x+=1
44     return
45
46 def nomorempat(NPM):
47     print("Input : "+NPM)
48     jumlah = len(NPM)
49     print("Output : ")
50     print("Halo, "+NPM[jumlah-3]+" apa kabar?")
51     return
52
53 def nomorlima(NPM):
54     NPM = list(NPM)
55     for x in NPM:
56         print(x)
57     return
58
59 def nomorenam(NPM):
60     NPM = list(NPM)
61     jum = 0
62     for x in NPM:
63         jum+=int(x)
64     print("Hasil : "+str(jum))
65     return
66
67 def nomortujuh(NPM):
68     NPM = list(NPM)

```

```

69     jum = 1
70     for x in NPM:
71         jum*=int(x)
72     print("Hasil : "+str(jum))
73     return
74
75 def nomordelapan(NPM):
76     NPM = list(NPM)
77     for x in NPM:
78         if int(x) != 0:
79             if int(x)%2 == 0:
80                 print(x, end=" ")
81     return
82
83 def nomorsembilan(NPM):
84     NPM = list(NPM)
85     for x in NPM:
86         if int(x) != 0:
87             if int(x)%2 == 1:
88                 print(x, end = " ")
89     return
90
91 def nomorsepuluh(NPM):
92     NPM = list(NPM)
93     for x in NPM:
94         if int(x) != 0:
95             i = 1
96             bil = 0
97             while i <= int(x):
98                 if int(x)%i == 0:
99                     bil+=1
100                 i+=1
101             if bil == 2:
102                 print(x)
103     return

```

```

1  ##File Main.py##
2  import chap3_1174035_3lib as lib3
3
4  lib3.nomorsatu("1174035")
5  lib3.nomordua("1174035")
6  lib3.nomortiga("1174035")
7  lib3.nomorempat("1174035")
8  lib3.nomorlima("1174035")
9  lib3.nomorenam("1174035")
10 lib3.nomortujuh("1174035")
11 lib3.nomordelapan("1174035")
12 lib3.nomorsembilan("1174035")
13 lib3.nomorsepuluh("1174035")

```

12. Membuat Satu File library bernama kelas3lib.py yang berisi kelas yang isinya semua fungsi - fungsi dari setiap nomor yang telah dimodifikasi untuk menyesuaikan dengan kelas. Kodingnya :

```

1  ##File kelas3lib.py##
2  class kelas3lib:

```



```

3 def __init__(self, NPM):
4     self.NPM = NPM
5 def nomorsatu(self):
6     NPM = int(self.NPM)
7     char = ""
8     if NPM%3==0:
9         char = "*"
10    if NPM%3==1:
11        char = "#"
12    if NPM%3==2:
13        char = "+"
14    line = []
15    line.append(" ##          ##          #####          ##          ##
#####          #####          #####")
16    line.append("#####          ##          ##          ##          ##
##          ##          ##          ##          ")
17    line.append(" ##          ##          ##          ##          ##          ##
##          ##          ##          ")
18    line.append(" ##          ##          ##          ##          ##          ##
##          #####          ##### ")
19    line.append(" ##          ##          ##          #####          ##
##          ##          ##")
20    line.append(" ##          ##          ##          ##          ##
##          ##          ##          ##          ##")
21    line.append("#####          #####          ##          ##
#####          #####          ##### ")
22    a=0
23    for x in line:
24        print(line[a].replace("#", char))
25        a+=1
26    return
27
28 def nomordua(self):
29     print("Input : "+self.NPM)
30     x=1
31     print("Output : ")
32     while x <= 87:
33         x+=1
34         print("Halo, "+self.NPM+" apa kabar?")
35
36 def nomortiga(self):
37     print("Input : "+self.NPM)
38     jumlah = len(self.NPM)
39     a = int(self.NPM[jumlah-3])
40     b = int(self.NPM[jumlah-2])
41     c = int(self.NPM[jumlah-1])
42     x=1
43     while x <= (a+b+c):
44         print("Output : "+self.NPM[jumlah-3:])
45         x+=1
46
47 def nomorempat(self):
48     print("Input : "+self.NPM)
49     jumlah = len(self.NPM)
50     print("Output : ")
51     print("Halo, "+self.NPM[jumlah-3]+" apa kabar?")

```

```

52
53 def nomorlima(self):
54     NPM = list(self.NPM)
55     for x in NPM:
56         print(x)
57     return
58
59 def nomorenam(self):
60     NPM = list(self.NPM)
61     jum = 0
62     for x in NPM:
63         jum+=int(x)
64     print("Hasil : "+str(jum))
65     return
66
67 def nomortujuh(self):
68     NPM = list(self.NPM)
69     jum = 1
70     for x in NPM:
71         jum*=int(x)
72     print("Hasil : "+str(jum))
73     return
74
75 def nomordelapan(self):
76     NPM = list(self.NPM)
77     for x in NPM:
78         if int(x)!=0:
79             if int(x)%2==0:
80                 print(x, end="")
81     return
82
83 def nomorsembilan(self):
84     NPM = list(self.NPM)
85     for x in NPM:
86         if int(x)!=0:
87             if int(x)%2==1:
88                 print(x, end = "")
89     return
90
91 def nomorsepuluh(self):
92     NPM = list(self.NPM)
93     for x in NPM:
94         if int(x)!=0:
95             i = 1
96             bil = 0
97             while i <= int(x):
98                 if int(x)%i==0:
99                     bil+=1
100                 i+=1
101             if bil == 2:
102                 print(x)
103     return

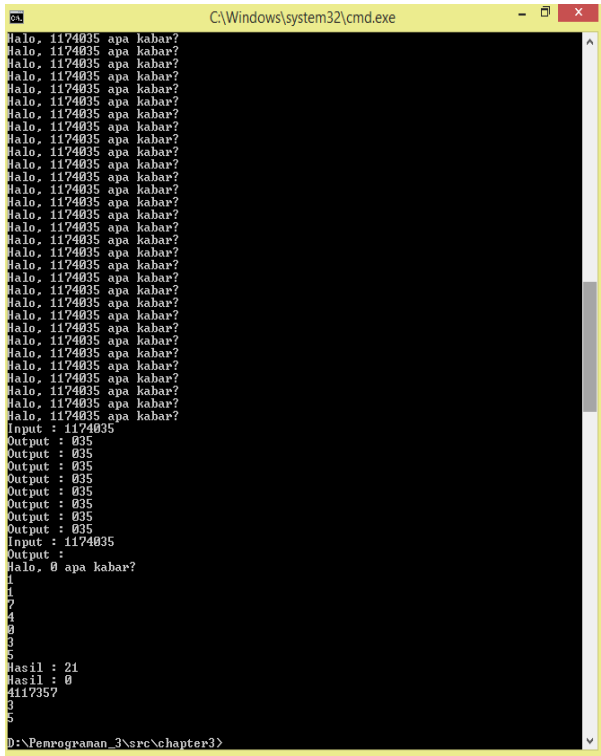
```

```
1 ##File Main.py##
```

```
2 import chap3_1174035_kelas3lib as kelas3lib
```

```
3
```

```
4 lib = kelas3lib.kelas3lib("1174035")
5 lib.nomorsatu()
6 lib.nomordua()
7 lib.nomortiga()
8 lib.nomorempat()
9 lib.nomorlima()
10 lib.nomorenam()
11 lib.nomortujuh()
12 lib.nomordelapan()
13 lib.nomorsembilan()
14 lib.nomorsepuluh()
```



Gambar 3.12 Screenshot No 12

3.0.2.1 Error

1. Tuliskan error yang terjadi saat mengerjakan section ini. Mendapat error yaitu salah konversi. Untuk handle error tersebut dapat menggunakan try catch :

```
1 def cekerror(NPM):
2     try:
3         a = int(NPM)
```



```

4     a = a+1110
5     print(str(a))
6 except ValueError:
7     print("Tidak boleh memakai huruf!!")

1 ##File Main.py##
2 import chap3_1174035_error as eror
3
4 eror.cekerror("1174035") #Percobaan saat benar
5 eror.cekerror("aaa") #percobaan saat salah

```

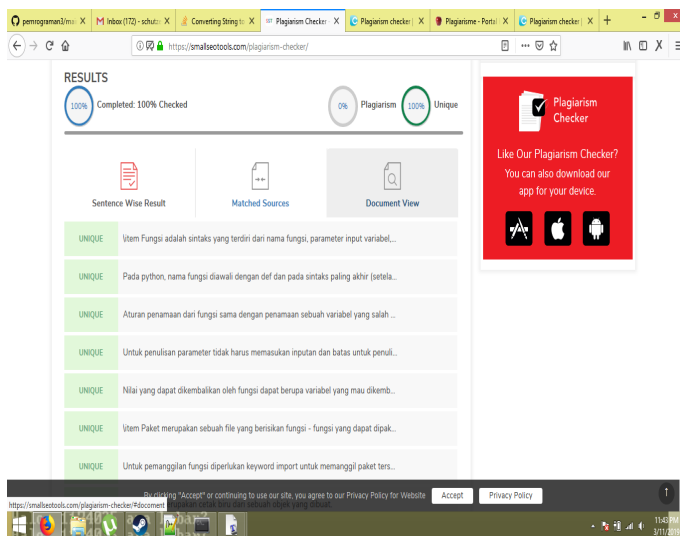
```

1175145
Tidak boleh memakai huruf!!

```

Gambar 3.13 Screenshot No 13

2. Plagiarisme



Gambar 3.14 Plagiarisme

3.1 Rangga Putra Ramdhani

3.1.0.1 Pemahaman Teori

1. Apa itu fungsi, inputan

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

```
1 def uji():
2     print("Tugas")
3
4 uji()
```

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

```
1 def uji_param(nama):
2     print("Nama saya :"+str(nama))
3
4 uji_param(input("Masukan Nama Kamu : "))
```

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

```
1 def uji_return(a,b):
2     r = a + b
3     return r
4
5 a = 10
6 b = 50
7 c = uji_return(a,b)
8 print(c)
```

2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya. Untuk memudahkan dalam pemanggilan fungsi yang dibutuhkan, agar dapat dipanggil berulang. Cara pemanggilannya

```
1 #from fungsi_evie import *
2 #print(penulisan(int(input("Masukan NPM kamu :"))))
```

3. Jelaskan Apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing. kelas merupakan sebuah blueprint yang merepresentasikan objek. objek adalah hasil cetakan dari sebuah kelas. method adalah suatu upaya yang digunakan oleh object.

```
1 #class Employee:
2 #     'Common base class for all employees'
3 #     empCount = 0
4
5 #     def __init__(self, name, salary):
6 #         self.name = name
7 #         self.salary = salary
8 #         Employee.empCount += 1
9
10 #     def displayCount(self):
11 #         print("Total Employee %d" % Employee.empCount)
```

```

12
13 #     def displayEmployee(self):
14 #         print ("Name : ", self.name, " ", Salary: ", self.salary)
15
16
17 #This would create first object of Employee class"
18 #emp1 = Employee("Zara", 2000)
19 #This would create second object of Employee class"
20 #emp2 = Employee("Manni", 5000)
21 #emp1.displayEmployee()
22 #emp2.displayEmployee()
23 #print ("Total Employee %d" % Employee.empCount)

```

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

- pertama import terlebih dahulu filenya.
- kemudian buat variabel untuk menampung datanya
- setelah itu panggil nama classnya dan panggil methodnya
- Gunakan perintah print untuk menampilkan hasilnya

```

1 #import belajar
2 #a = 100
3 #b = 50
4
5 #c = belajar.penambahan(a,b)
6 #print(c)

```

5. Jelaskan dengan contoh pemakaian paket dengan perintah from kalkulator import Penambahan disertai dengan contoh kode lainnya. Penggunaan paket from namafile import, itu berfungsi untuk memanggil file dan fungsinya

```

1 #from fungsi_evie import *
2 #print(penulisan(int(input("Masukan NPM kamu : "))))

```

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

```

1 #Chapter 3

```

3.1.0.2 Ketrampilan Pemrograman

1. Buatlah fungsi dengan inputan variabel NPM, dan melakukan print luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk NPM mod 3=0, tanda pagar untuk NPM mod 3 =1, tanda plus untuk NPM mod3=2.

```

1 #Chapter 3
2 #No 1
3 def penulisan(npm):
4     npm = list(str(npm))
5
6     angka1 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
7     angka2 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
8     angka3 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "###   ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
9     angka4 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ### ", "5": "###   ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
10    angka5 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
11    angka6 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
12    angka7 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
13    angka8 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
14    angka9 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
15    angka10 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
16
17    hasil1 = []
18    hasil2 = []
19    hasil3 = []
20    hasil4 = []
21    hasil5 = []
22    hasil6 = []
23    hasil7 = []
24    hasil8 = []
25    hasil9 = []
26    hasil10 = []
27
28    for x in npm:
29        hasil1.append(angka1[x])
30        hasil2.append(angka2[x])
31        hasil3.append(angka3[x])
32        hasil4.append(angka4[x])
33        hasil5.append(angka5[x])
34        hasil6.append(angka6[x])
35        hasil7.append(angka7[x])
36        hasil8.append(angka8[x])

```

```

37     hasil9.append(angka9[x])
38     hasil10.append(angka10[x])
39
40     print(*hasil1, sep=' ')
41     print(*hasil2, sep=' ')
42     print(*hasil3, sep=' ')
43     print(*hasil4, sep=' ')
44     print(*hasil5, sep=' ')
45     print(*hasil6, sep=' ')
46     print(*hasil7, sep=' ')
47     print(*hasil8, sep=' ')
48     print(*hasil9, sep=' ')
49     print(*hasil10, sep=' ')
50
51 penulisan(int(input("Masukan NPM :"))))

```

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

```

1 def perulangan(npm):
2     hitung = 0
3     while(hitung < 56):
4         print("Halo, "+str(npm)+" apa kabar?")
5         hitung = hitung +1
6
7 perulangan(int(input("Masukan NPM : ")))

```

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

```

1 #No 3
2 def perulangan_3_digit(npm):
3     hitung = 0
4     npm = str(npm)
5     bil = npm[4:7]
6
7     while(hitung < 9):
8         print("Halo, "+bil+" apa kabar?")
9         hitung = hitung +1
10
11 perulangan_3_digit(int(input("Masukan NPM : ")))

```

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

```

1 #No 4
2 def perulangan_3_digit_terakhir(npm):
3     npm = str(npm)
4     bil = npm[-3]
5     print("Halo, "+bil+" apa kabar?")
6
7 perulangan_3_digit_terakhir(int(input("Masukan NPM : ")))

```

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

```

1 #No 5
2 def down(npm):
3     for i in npm:
4         print (i)

```

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

```

1 #No 6
2 def penjumlahan(npm):
3     jumlah = 0
4     for i in npm:
5         jumlah += int(i)
6     print(str(jumlah)+" Adalah hasil penjumlahan")
7
8 penjumlahan(input("Masukan NPM : "))

```

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

```

1 #No 7
2 def perkalian(npm):
3     jumlah = 0
4     for i in npm:
5         jumlah *= int(i)
6     print(str(jumlah)+" Adalah hasil perkalian")
7
8 perkalian(input("Masukan NPM : "))

```

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

```

1 #No 8
2 def genap():
3     npm = [1,1,7,4,0,2,7]
4     for i in npm:
5         if (i % 2) == 0:
6             print("Bilangan Genapnya : "+str(i))
7 genap()

```

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

```

1 #No 9
2 def ganjil():
3     npm = [1,1,7,4,0,2,7]
4     for i in npm:
5         if (i%2)==1:
6             print("Bilangan Ganjilnya : "+str(i))
7 ganjil()

```

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

```

1 #No 10
2 def prima(npm):
3     npm = str(npm)
4     bil = npm[2]
5     num = int(bil)
6     if num > 1:
7         for i in range(2,num):
8             if (num%i)==0:
9                 print("Bukan Bilangan Prima")
10                break
11            else:
12                print("Bilangan Primanya :"+str(num))
13        else:
14            print("Tidak Ada Bilangan Prima")
15    prima(int(input("Masukan NPM : ")))

```

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

```

1 from fungsi_rangga import penulisan

```

12. Buatlah satu library class dengan nama le_kelas3lib.py yang merupakan mod- i kasi dari fungsi-fungsi nomor diatas dan berikan contoh cara pemanggilannya pada le_mainn.py.

```

1 from rangga import penulisan

```

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

```

1 #def penanganan_error(a,b):
2 #     try :
3 #         c = a+b
4 #         print(c)
5 #     except TypeError:
6 #         print("We Are Different")

```

3.2 Faisal Najib Abdullah

3.2.1 Pemahaman Teori

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

```
1 def uji():
2     print("Tugas Chapter 3")
3
4 uji()
```

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

```
1 def uji_param(nama):
2     print("Nama saya :"+str(nama))
3
4 uji_param(input("Masukan Nama Kamu : "))
```

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

```
1 def uji_return(a,b):
2     r = a + b
3     return r
4
5 a = 10
6 b = 50
7 c = uji_return(a,b)
8 print(c)
```

2. Apa itu paket dan cara pemanggilan paket atau library dengan contoh kode program lainnya. Untuk memudahkan dalam pemanggilan fungsi yang dibutuhkan, agar dapat dipanggil berulang. Cara pemanggilannya

```
1 #from fungsi_1174042 import *
2 #print(nol(int(input("Masukan NPM kamu : "))))
```

3. Jelaskan Apa itu kelas, apa itu objek, apa itu atribut, apa itu method dan contoh kode program lainnya masing-masing. kelas merupakan sebuah blueprint yang merepresentasikan objek. objek adalah hasil cetakan dari sebuah kelas. method adalah suatu upaya yang digunakan oleh object.

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



```

9
10 # def displayCount(self):
11 #     print ("Total Employee %d" % Employee.empCount)
12
13 # def displayEmployee(self):
14 #     print ("Name : ", self.name, ", Salary: ", self.salary)
15
16
17 #This would create first object of Employee class"
18 #emp1 = Employee("Zara", 2000)
19 #This would create second object of Employee class"
20 #emp2 = Employee("Manni", 5000)
21 #emp1.displayEmployee()
22 #emp2.displayEmployee()
23 #print ("Total Employee %d" % Employee.empCount)

```

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

- pertama import terlebih dahulu filenya.
- kemudian buat variabel untuk menampung datanya
- setelah itu panggil nama classnya dan panggil methodnya
- Gunakan perintah print untuk menampilkan hasilnya

```

1 #import belajar
2 #a = 100
3 #b = 50
4
5 #c = belajar.penambahan(a,b)
6 #print(c)

```

5. Jelaskan dengan contoh pemakaian paket dengan perintah from kalkulator import Penambahan disertai dengan contoh kode lainnya. Penggunaan paket from namafile import, itu berfungsi untuk memanggil file dan fungsinya

```

1 #from fungsi_1174042 import *
2 #print(nol(int(input("Masukan NPM kamu : "))))

```

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

```

1 #from nama_folder import nama_file
2
3 print(nol(int(input("Masukan NPM :"))))

```

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

```

1 #from nama_folder import nama_file
2
3 print(nol(int(input("Masukan NPM :"))))

```

3.2.2 Ketrampilan Pemrograman

1. Buatlah fungsi dengan inputan variabel NPM, dan melakukan print luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk NPM mod 3=0, tanda pagar untuk NPM mod 3 =1, tanda plus untuk NPM mod3=2.

```

1 def mod(x):
2     i = 3
3     if (x % i) == 1:
4         print("# # ### # # # #")
5         print("# # # ## # # ## # #")
6         print("# # # ### # # ### #")
7         print("# # # # # # # ###")
8     else:
9         print("tetot")
10    return x
11
12 x = int(input("Masukkan NPM: "))
13 c = mod(x)

```

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

```

1 def npm(x):
2     a = 0
3     b = int(x[5:7])
4     while a < b :
5         a += 1
6         print("Hello , " + x + " Apa Kabar?")
7     return x
8
9 x = input("Masukkan NPM: ")
10 c = npm(x)

```

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

```

1 def npm(x):
2     a = 0
3     k = int(x[4])
4     p = int(x[5])
5     i = int(x[6])
6     b = x[4:7]
7     c = k+p+i
8
9     while a < c:
10        a += 1
11        print("Hello , " + b + " Apa Kabar?")
12    return x
13
14 x = input("Masukkan NPM: ")
15 c = npm(x)

```

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

```

1 def npm(x):
2     p = x[4]
3     print("Hello , " + p + " Apa Kabar?")
4     return x
5
6 x = input("Masukkan NPM: ")
7 c = npm(x)

```

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

```

1 def npm(x):
2     npm = [a,b,c,d,e,f,g]
3
4     for n in npm:
5         print(n)
6     return n
7
8 a = 1
9 b = 1
10 c = 7
11 d = 4
12 e = 0
13 f = 4
14 g = 2
15
16 c = npm(x)

```

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

```

1 def npm(x):
2     npm = [a+b+c+d+e+f+g]
3
4     for n in npm:
5         print(n)
6     return n
7
8 a = 1
9 b = 1
10 c = 7
11 d = 4
12 e = 0
13 f = 4
14 g = 2
15
16 c = npm(x)

```

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

```

1 def npm(x):
2     npm = [a*b*c*d*e*f*g]
3
4     for n in npm:
5         print(n)
6     return n
7
8 a = 1
9 b = 1
10 c = 7
11 d = 4
12 e = 0
13 f = 4
14 g = 2
15
16 c = npm(x)

```

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

```

1 def npm(x):
2     npm = [a,b,c,d,e,f,g]
3
4     for n in npm:
5         if(n % 2 == 0):
6             if(n != 0):
7                 print(n, end = "")
8     return n
9
10 a = 1
11 b = 1
12 c = 7
13 d = 4
14 e = 0
15 f = 4
16 g = 2
17
18 c = npm(x)

```

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

```

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

```

```

10 b = 1
11 c = 7
12 d = 4
13 e = 0
14 f = 4
15 g = 2
16
17 c = npm(x)

```

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

```

1 def npm(x):
2     npm = [a,b,c,d,e,f,g]
3
4     for n in npm:
5         if (n % 2) == 0:
6             print(end="")
7         else:
8             print(n, end = "")
9     return n
10
11 a = 1
12 b = 1
13 c = 7
14 d = 4
15 e = 0
16 f = 4
17 g = 2
18
19 c = npm(x)

```

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

```

1 from fungsi_1174042 import nol
2 from 1174042_3lib import nol
3
4 print(nol(int(input("Masukan NPM :"))))

```

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

```

1 from fungsi_1174042 import nol
2 from 1174042_3lib import nol
3
4 print(nol(int(input("Masukan NPM :"))))

```

3.2.3 Ketrampilan Penanganan Error

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

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


DAFTAR PUSTAKA

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

Index

disruptif, xxxv
modern, xxxv