

iiiiii HEAD

LLLLLL c64ab529ff89ebc66e82900d0611abe5d2422d5a

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

iiiiii HEAD

| | | |
|----------|-----------------------------|------------|
| 1 | Judul Bagian Pertama | 1 |
| 2 | Judul Bagian Kedua | 7 |
| 3 | Judul Bagian Ketiga | 49 |
| 4 | Judul Bagian Keempat | 121 |

=====

| | | |
|----------|-----------------------------|------------|
| 1 | Judul Bagian Pertama | 1 |
| 2 | Judul Bagian Kedua | 7 |
| 3 | Judul Bagian Ketiga | 49 |
| 4 | Judul Bagian Keempat | 121 |

~~~~~ c64ab529ff89ebc66e82900d0611abe5d2422d5a





# DAFTAR ISI

---

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

|          |                                  |          |
|----------|----------------------------------|----------|
| 1.2.2    | Problems                         | 2        |
| 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        |
| <b>2</b> | <b>Judul Bagian Kedua</b>        | <b>7</b> |
| 2.1      | IrvanRizkiansyah/1174043         | 7        |
| 2.1.1    | Teori                            | 7        |
| 2.1.2    | Keterampilan Pemrograman         | 9        |
| 2.1.3    | Keterampilan Penanganan Error    | 12       |
| 2.2      | Hagan Rowlenstino/1174040        | 12       |
| 2.2.1    | Teori                            | 12       |
| 2.2.2    | Keterampilan Pemrograman         | 13       |
| 2.2.3    | Keterampilan Penanganan Error    | 16       |
| 2.3      | Muhammad Iqbal Panggabean        | 17       |
| 2.3.1    | Teori                            | 17       |
| 2.3.2    | praktek                          | 19       |
| 2.3.3    | Keterampilan dan penanganan eror | 20       |
| 2.4      | Luthfi M. Nabil/1174035          | 21       |
| 2.4.1    | Teori                            | 21       |
| 2.4.2    | Keterampilan Pemrograman         | 22       |
| 2.4.3    | Keterampilan Penanganan Error    | 27       |
| 2.5      | Faisal Najib Abdullah 1174042    | 27       |
| 2.5.1    | Teori                            | 27       |

|       |                                      |    |
|-------|--------------------------------------|----|
| 2.5.2 | Keterampilan Pemrograman             | 29 |
| 2.5.3 | Keterampilan Penanganan Error        | 32 |
| 2.6   | Dika Sukma Pradana 1174050           | 33 |
| 2.6.1 | Teori                                | 33 |
| 2.6.2 | Praktek                              | 34 |
| 2.6.3 | Keterampilan dan Penanganan Error    | 36 |
| 2.7   | Ichsan Hizman Hardy                  | 36 |
| 2.7.1 | Teori                                | 36 |
| 2.7.2 | praktek                              | 38 |
| 2.7.3 | Keterampilan dan penanganan eror     | 40 |
| 2.8   | Mhd Zulfikar Akram Nasution/ 1164081 | 40 |
| 2.8.1 | Teori                                | 40 |
| 2.8.2 | Keterampilan Pemrograman             | 43 |
| 2.8.3 | Keterampilan Penanganan Error        | 47 |

### **3 Judul Bagian Ketiga 49**

|       |                                  |     |
|-------|----------------------------------|-----|
| 3.1   | IrvanRizkiansyah/1174043         | 49  |
| 3.1.1 | Pemahaman Teori                  | 49  |
| 3.1.2 | Keterampilan Pemrograman         | 51  |
| 3.1.3 | Keterampilan Penanganan Error    | 65  |
| 3.1.4 | Plagiarisme                      | 65  |
| 3.1.5 | Luthfi Muhammad Nabil/1174035    | 66  |
| 3.1.6 | Ketrampilan Pemrograman          | 67  |
| 3.2   | Rangga Putra Ramdhani            | 81  |
| 3.3   | Faisal Najib Abdullah            | 87  |
| 3.3.1 | Pemahaman Teori                  | 87  |
| 3.3.2 | Ketrampilan Pemrograman          | 89  |
| 3.3.3 | Ketrampilan Penanganan Error     | 93  |
| 3.4   | Yusniar Nur Syarif Sidiq/1164089 | 93  |
| 3.4.1 | Teori                            | 93  |
| 3.4.2 | Keterampilan Pemrograman         | 96  |
| 3.4.3 | Penanganan Error                 | 99  |
| 3.5   | Fathi Rabbani/1164074/3C         | 99  |
| 3.5.1 | Teori                            | 99  |
| 3.5.2 | Praktek Pemrograman              | 101 |
| 3.5.3 | Handling Error                   | 107 |
| 3.6   | Dika Sukma Pradana 1174050       | 108 |
| 3.6.1 | Pemahaman Teori                  | 108 |



|          |                                      |          |
|----------|--------------------------------------|----------|
| 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        |
| <b>2</b> | <b>Judul Bagian Kedua</b>            | <b>7</b> |
| 2.1      | IrvanRizkiansyah/1174043             | 7        |
| 2.1.1    | Teori                                | 7        |
| 2.1.2    | Keterampilan Pemrograman             | 9        |
| 2.1.3    | Keterampilan Penanganan Error        | 12       |
| 2.2      | Hagan Rowlenstino/1174040            | 12       |
| 2.2.1    | Teori                                | 12       |
| 2.2.2    | Keterampilan Pemrograman             | 13       |
| 2.2.3    | Keterampilan Penanganan Error        | 16       |
| 2.3      | Muhammad Iqbal Panggabean            | 17       |
| 2.3.1    | Teori                                | 17       |
| 2.3.2    | praktek                              | 19       |
| 2.3.3    | Keterampilan dan penanganan eror     | 20       |
| 2.4      | Luthfi M. Nabil/1174035              | 21       |
| 2.4.1    | Teori                                | 21       |
| 2.4.2    | Keterampilan Pemrograman             | 22       |
| 2.4.3    | Keterampilan Penanganan Error        | 27       |
| 2.5      | Faisal Najib Abdullah 1174042        | 27       |
| 2.5.1    | Teori                                | 27       |
| 2.5.2    | Keterampilan Pemrograman             | 29       |
| 2.5.3    | Keterampilan Penanganan Error        | 32       |
| 2.6      | Dika Sukma Pradana 1174050           | 33       |
| 2.6.1    | Teori                                | 33       |
| 2.6.2    | Praktek                              | 34       |
| 2.6.3    | Keterampilan dan Penanganan Error    | 36       |
| 2.7      | Ichsan Hizman Hardy                  | 36       |
| 2.7.1    | Teori                                | 36       |
| 2.7.2    | praktek                              | 38       |
| 2.7.3    | Keterampilan dan penanganan eror     | 40       |
| 2.8      | Mhd Zulfikar Akram Nasution/ 1164081 | 40       |

|          |                                     |            |
|----------|-------------------------------------|------------|
| 2.8.1    | Teori                               | 40         |
| 2.8.2    | Keterampilan Pemrograman            | 43         |
| 2.8.3    | Keterampilan Penanganan Error       | 47         |
| <b>3</b> | <b>Judul Bagian Ketiga</b>          | <b>49</b>  |
| 3.1      | IrvanRizkiansyah/1174043            | 49         |
| 3.1.1    | Pemahaman Teori                     | 49         |
| 3.1.2    | Keterampilan Pemrograman            | 51         |
| 3.1.3    | Keterampilan Penanganan Error       | 65         |
| 3.1.4    | Plagiarisme                         | 65         |
| 3.1.5    | Luthfi Muhammad Nabil/1174035       | 66         |
| 3.1.6    | Ketrampilan Pemrograman             | 67         |
| 3.2      | Rangga Putra Ramdhani               | 81         |
| 3.3      | Faisal Najib Abdullah               | 87         |
| 3.3.1    | Pemahaman Teori                     | 87         |
| 3.3.2    | Ketrampilan Pemrograman             | 89         |
| 3.3.3    | Ketrampilan Penanganan Error        | 93         |
| 3.4      | Yusniar Nur Syarif Sidiq/1164089    | 93         |
| 3.4.1    | Teori                               | 93         |
| 3.4.2    | Keterampilan Pemrograman            | 96         |
| 3.4.3    | Penanganan Error                    | 99         |
| 3.5      | Fathi Rabbani/1164074/3C            | 99         |
| 3.5.1    | Teori                               | 99         |
| 3.5.2    | Praktek Pemrograman                 | 101        |
| 3.5.3    | Handling Error                      | 107        |
| 3.6      | Dika Sukma Pradana 1174050          | 108        |
| 3.6.1    | Pemahaman Teori                     | 108        |
| 3.6.2    | Ketrampilan Pemrograman             | 110        |
| 3.6.3    | Ketrampilan Penanganan Error        | 114        |
| 3.7      | Mhd Zulfikar Akram Nasution/1164081 | 114        |
| 3.7.1    | Teori                               | 114        |
| 3.7.2    | Keterampilan Pemrograman            | 116        |
| 3.7.3    | Penanganan Error                    | 119        |
| <b>4</b> | <b>Judul Bagian Keempat</b>         | <b>121</b> |
| 4.1      | Faisal Najib Abdullah               | 121        |
| 4.1.1    | Pemahaman Teori                     | 121        |

>>>>>>> c64ab529ff89ebc66e82900d0611abe5d2422d5a

# DAFTAR GAMBAR

---

<<<<<<< HEAD

|      |                       |    |
|------|-----------------------|----|
| 3.1  | Jawaban No. 1         | 51 |
| 3.2  | Jawaban No. 2         | 52 |
| 3.3  | Jawaban No. 3         | 53 |
| 3.4  | Jawaban No. 4         | 54 |
| 3.5  | Jawaban No. 5         | 55 |
| 3.6  | Jawaban No. 6         | 56 |
| 3.7  | Jawaban No. 7         | 57 |
| 3.8  | Jawaban No. 8         | 58 |
| 3.9  | Jawaban No. 9         | 59 |
| 3.10 | Jawaban No. 10        | 60 |
| 3.11 | Jawaban No. 11        | 63 |
| 3.12 | Hasil Cek Plagiarisme | 65 |



|       |                       |     |
|-------|-----------------------|-----|
| 3.13  | Screenshot No 1       | 68  |
| 3.14  | Screenshot No 2       | 69  |
| 3.15  | Screenshot No 3       | 70  |
| 3.16  | Screenshot No 4       | 70  |
| 3.17  | Screenshot No 5       | 71  |
| 3.18  | Screenshot No 6       | 71  |
| 3.19  | Screenshot No 7       | 72  |
| 3.20  | Screenshot No 8       | 72  |
| 3.21  | Screenshot No 9       | 73  |
| 3.22  | Screenshot No 10      | 74  |
| 3.23  | Screenshot No 11      | 76  |
| 3.24  | Screenshot No 12      | 79  |
| 3.25  | Screenshot No 13      | 80  |
| 3.26  | Plagiarisme           | 80  |
| 4.1   | Contoh CSV            | 122 |
| ===== |                       |     |
| 3.1   | Jawaban No. 1         | 51  |
| 3.2   | Jawaban No. 2         | 52  |
| 3.3   | Jawaban No. 3         | 53  |
| 3.4   | Jawaban No. 4         | 54  |
| 3.5   | Jawaban No. 5         | 55  |
| 3.6   | Jawaban No. 6         | 56  |
| 3.7   | Jawaban No. 7         | 57  |
| 3.8   | Jawaban No. 8         | 58  |
| 3.9   | Jawaban No. 9         | 59  |
| 3.10  | Jawaban No. 10        | 60  |
| 3.11  | Jawaban No. 11        | 63  |
| 3.12  | Hasil Cek Plagiarisme | 65  |

|        |                                          |    |
|--------|------------------------------------------|----|
| 3.13   | Screenshot No 1                          | 68 |
| 3.14   | Screenshot No 2                          | 69 |
| 3.15   | Screenshot No 3                          | 70 |
| 3.16   | Screenshot No 4                          | 70 |
| 3.17   | Screenshot No 5                          | 71 |
| 3.18   | Screenshot No 6                          | 71 |
| 3.19   | Screenshot No 7                          | 72 |
| 3.20   | Screenshot No 8                          | 72 |
| 3.21   | Screenshot No 9                          | 73 |
| 3.22   | Screenshot No 10                         | 74 |
| 3.23   | Screenshot No 11                         | 76 |
| 3.24   | Screenshot No 12                         | 79 |
| 3.25   | Screenshot No 13                         | 80 |
| 3.26   | Plagiarisme                              | 80 |
| >>>>>> | c64ab529ff89ebc66e82900d0611abe5d2422d5a |    |



# DAFTAR TABEL

---



# Listings

---

|                              |    |
|------------------------------|----|
| <<<<<<< HEAD                 |    |
| src/chapter2/1174043_1.py    | 9  |
| src/chapter2/1174043_2.py    | 9  |
| src/chapter2/1174043_3.py    | 10 |
| src/chapter2/1174043_4.py    | 10 |
| src/chapter2/1174043_5.py    | 10 |
| src/chapter2/1174043_6.py    | 10 |
| src/chapter2/1174043_7.py    | 10 |
| src/chapter2/1174043_8.py    | 10 |
| src/chapter2/1174043_9.py    | 11 |
| src/chapter2/1174043_10.py   | 11 |
| src/chapter2/1174043_11.py   | 11 |
| src/chapter2/1174043_2err.py | 12 |
| src/chapter2/1174040_1.py    | 13 |
| src/chapter2/1174040_2.py    | 14 |
| src/chapter2/1174040_3.py    | 14 |
| src/chapter2/1174040_4.py    | 14 |
| src/chapter2/1174040_5.py    | 14 |

|                              |    |
|------------------------------|----|
| src/chapter2/1174040_6.py    | 15 |
| src/chapter2/1174040_7.py    | 15 |
| src/chapter2/1174040_8.py    | 15 |
| src/chapter2/1174040_9.py    | 15 |
| src/chapter2/1174040_10.py   | 16 |
| src/chapter2/1174040_11.py   | 16 |
| src/chapter2/1174040_2err.py | 16 |
| src/1174063_teor.py          | 17 |
| src/1174063_teor.py          | 17 |
| src/1174063_teor.py          | 17 |
| src/1174063_teor.py          | 18 |
| src/1174063_teor.py          | 18 |
| src/1174063_teor.py          | 18 |
| src/1174063_teor.py          | 18 |
| src/1174063_teor.py          | 18 |
| src/1174063_teor.py          | 19 |
| src/1174063_praktek.py       | 19 |
| src/1174063_praktek.py       | 19 |
| src/1174063_praktek.py       | 19 |
| src/1174063_praktek.py       | 19 |
| src/1174063_praktek.py       | 20 |
| src/1174063_praktek.py       | 20 |
| src/1174063_praktek.py       | 20 |
| src/1174063_praktek.py       | 20 |
| src/1174063_praktek.py       | 20 |
| src/1174063_praktek.py       | 20 |
| src/err2.py                  | 20 |
| src/chapter2/1174035_1.py    | 22 |
| src/chapter2/1174035_2.py    | 22 |
| src/chapter2/1174035_3.py    | 23 |
| src/chapter2/1174035_4.py    | 23 |
| src/chapter2/1174035_5.py    | 23 |
| src/chapter2/1174035_6.py    | 24 |
| src/chapter2/1174035_7.py    | 24 |
| src/chapter2/1174035_8.py    | 25 |
| src/chapter2/1174035_9.py    | 25 |
| src/chapter2/1174035_10.py   | 26 |

|                              |    |
|------------------------------|----|
| src/chapter2/1174035_11.py   | 26 |
| src/chapter2/1174035_2err.py | 27 |
| src/chapter2/1174042_1,1.py  | 28 |
| src/chapter2/1174042_1,2.py  | 28 |
| src/chapter2/1174042_1,3.py  | 28 |
| src/chapter2/1174042_1,4.py  | 28 |
| src/chapter2/1174042_1,5.py  | 28 |
| src/chapter2/1174042_1,7.py  | 29 |
| src/chapter2/1174042_2,1.py  | 29 |
| src/chapter2/1174042_2,2.py  | 29 |
| src/chapter2/1174042_2,3.py  | 29 |
| src/chapter2/1174042_2,4.py  | 30 |
| src/chapter2/1174042_2,5.py  | 30 |
| src/chapter2/1174042_2,6.py  | 30 |
| src/chapter2/1174042_2,7.py  | 30 |
| src/chapter2/1174042_2,8.py  | 30 |
| src/chapter2/1174042_2,9.py  | 31 |
| src/chapter2/1174042_2,10.py | 31 |
| src/chapter2/1174042_2,11.py | 32 |
| src/chapter2/1174042_2err.py | 32 |
| src/1174050_teor.py          | 33 |
| src/1174050_teor.py          | 33 |
| src/1174050_teor.py          | 33 |
| src/1174050_teor.py          | 33 |
| src/1174050_teor.py          | 34 |
| src/1174050_teor.py          | 34 |
| src/1174050_teor.py          | 34 |
| src/1174050_teor.py          | 34 |
| src/1174050_teor.py          | 34 |
| src/1174050_praktek.py       | 35 |
| src/1174050_praktek.py       | 35 |
| src/1174050_praktek.py       | 35 |
| src/1174050_praktek.py       | 35 |
| src/1174050_praktek.py       | 35 |
| src/1174050_praktek.py       | 36 |
| src/1174050_praktek.py       | 36 |
| src/1174050_praktek.py       | 36 |
| src/1174050_praktek.py       | 36 |



|                                        |    |
|----------------------------------------|----|
| src/1174050_praktek.py                 | 36 |
| src/1174050_praktek.py                 | 36 |
| src/error1ka.py                        | 36 |
| src/chapter2/1174034_ichsan.teori.py   | 37 |
| src/chapter2/1174034_ichsan.teori.py   | 37 |
| src/chapter2/1174034_ichsan.teori.py   | 37 |
| src/chapter2/1174034_ichsan.teori.py   | 37 |
| src/chapter2/1174034_ichsan.teori.py   | 37 |
| src/chapter2/1174034_ichsan.teori.py   | 38 |
| src/chapter2/1174034_ichsan.teori.py   | 38 |
| src/chapter2/1174034_ichsan.teori.py   | 38 |
| src/chapter2/1174034_ichsan.teori.py   | 38 |
| src/chapter2/1174034_ichsan.praktek.py | 38 |
| src/chapter2/1174034_ichsan.praktek.py | 39 |
| src/chapter2/1174034_ichsan.praktek.py | 39 |
| src/chapter2/1174034_ichsan.praktek.py | 39 |
| src/chapter2/1174034_ichsan.praktek.py | 39 |
| src/chapter2/1174034_ichsan.praktek.py | 39 |
| src/chapter2/1174034_ichsan.praktek.py | 40 |
| src/chapter2/1174034_ichsan.praktek.py | 40 |
| src/chapter2/1174034_ichsan.praktek.py | 40 |
| src/chapter2/1174034_ichsan.praktek.py | 40 |
| src/chapter2/1174034_ichsan.praktek.py | 40 |
| src/chapter2/errorrr.py                | 40 |
| src/chapter2/1164081_1.py              | 43 |
| src/chapter2/1164081_2.py              | 44 |
| src/chapter2/1164081_3.py              | 44 |
| src/chapter2/1164081_4.py              | 44 |
| src/chapter2/1164081_5.py              | 44 |
| src/chapter2/1164081_6.py              | 44 |
| src/chapter2/1164081_7.py              | 45 |
| src/chapter2/1164081_8.py              | 45 |
| src/chapter2/1164081_9.py              | 45 |
| src/chapter2/1164081_10.py             | 46 |
| src/chapter2/1164081_11.py             | 46 |
| src/chapter2/1164081_2err.py           | 47 |
| src/chapter3/teori_1174043_chap3.py    | 49 |
| src/chapter3/teori_1174043_chap3.py    | 50 |

|                                         |    |
|-----------------------------------------|----|
| src/chapter3/teori_1174043_chap3.py     | 50 |
| src/chapter3/teori_1174043_chap3.py     | 50 |
| src/chapter3/teori_1174043_chap3.py     | 50 |
| src/chapter3/teori_1174043_chap3.py     | 50 |
| src/chapter3/teori_1174043_chap3.py     | 51 |
| src/chapter3/chap3_1174043_no1.py       | 51 |
| src/chapter3/chap3_1174043_no2.py       | 52 |
| src/chapter3/chap3_1174043_no3.py       | 52 |
| src/chapter3/chap3_1174043_no4.py       | 53 |
| src/chapter3/chap3_1174043_no5.py       | 54 |
| src/chapter3/chap3_1174043_no6.py       | 55 |
| src/chapter3/chap3_1174043_no7.py       | 56 |
| src/chapter3/chap3_1174043_no8.py       | 57 |
| src/chapter3/chap3_1174043_no9.py       | 57 |
| src/chapter3/chap3_1174043_no10.py      | 58 |
| src/chapter3/chap3_1174043_3lib.py      | 59 |
| src/chapter3/chap3_1174043_main.py      | 62 |
| src/chapter3/chap3_1174043_kelas3lib.py | 62 |
| src/chapter3/chap3_1174043_3err.py      | 65 |
| src/chapter3/chap3_1174035_teor_i.py    | 66 |
| src/chapter3/chap3_1174035_teor_i.py    | 66 |
| src/chapter3/chap3_1174035_teor_i.py    | 66 |
| src/chapter3/chap3_1174035_teor_i.py    | 67 |
| src/chapter3/chap3_1174035_teor_i.py    | 67 |
| src/chapter3/chap3_1174035_teor_i.py    | 67 |
| src/chapter3/chap3_1174035_teor_i.py    | 67 |
| src/chapter3/chap3_1174035_1.py         | 67 |
| src/chapter3/chap3_1174035_2.py         | 68 |
| src/chapter3/chap3_1174035_3.py         | 69 |
| src/chapter3/chap3_1174035_4.py         | 69 |
| src/chapter3/chap3_1174035_5.py         | 70 |
| src/chapter3/chap3_1174035_6.py         | 71 |
| src/chapter3/chap3_1174035_7.py         | 71 |
| src/chapter3/chap3_1174035_8.py         | 72 |
| src/chapter3/chap3_1174035_9.py         | 72 |
| src/chapter3/chap3_1174035_10.py        | 73 |
| src/chapter3/chap3_1174035_3lib.py      | 73 |
| src/chapter3/chap3_1174035_main.py      | 76 |

|                                         |    |
|-----------------------------------------|----|
| src/chapter3/chap3_1174035_kelas3lib.py | 77 |
| src/chapter3/chap3_1174035_main.py      | 78 |
| src/chapter3/chap3_1174035_error.py     | 79 |
| src/chapter3/chap3_1174035_main.py      | 80 |
| src/1174056_praktek.py                  | 81 |
| src/1174056_praktek.py                  | 81 |
| src/1174056_praktek.py                  | 81 |
| src/1174056_praktek.py                  | 81 |
| src/1174056_praktek.py                  | 81 |
| src/1174056_praktek.py                  | 82 |
| src/1174056_praktek.py                  | 82 |
| src/1174056_praktek.py                  | 82 |
| src/1174056_praktek.py                  | 83 |
| src/1174056_praktek.py                  | 84 |
| src/1174056_praktek.py                  | 84 |
| src/1174056_praktek.py                  | 85 |
| src/1174056_praktek.py                  | 85 |
| src/1174056_praktek.py                  | 85 |
| src/1174056_praktek.py                  | 85 |
| src/1174056_praktek.py                  | 85 |
| src/1174056_praktek.py                  | 86 |
| src/1174056_praktek.py                  | 86 |
| src/mainn.py                            | 86 |
| src/mainn.py                            | 86 |
| src/1174056_praktek.py                  | 87 |
| src/chapter3/1174042_1,1.py             | 87 |
| src/chapter3/1174042_1,1,1.py           | 87 |
| src/chapter3/1174042_1,1,2.py           | 87 |
| src/chapter3/1174042_1,2.py             | 87 |
| src/chapter3/1174042_1,3.py             | 88 |
| src/chapter3/1174042_1,4.py             | 88 |
| src/chapter3/1174042_1,2.py             | 88 |
| src/chapter3/1174042_1,6.py             | 88 |
| src/chapter3/1174042_1,6.py             | 89 |
| src/chapter3/1174042_2,1.py             | 89 |
| src/chapter3/1174042_2,2.py             | 89 |
| src/chapter3/1174042_2,3.py             | 89 |
| src/chapter3/1174042_2,4.py             | 90 |

[illegible]

|                                   |     |
|-----------------------------------|-----|
| src/chapter3/1164074/praktek.py   | 101 |
| src/chapter3/1164074/praktek.py   | 101 |
| src/chapter3/1164074/praktek.py   | 101 |
| src/chapter3/1164074/praktek.py   | 101 |
| src/chapter3/1164074/praktek.py   | 101 |
| src/chapter3/1164074/praktek.py   | 102 |
| src/chapter3/1164074/praktek.py   | 103 |
| src/chapter3/1164074/praktek.py   | 103 |
| src/chapter3/1164074/praktek.py   | 103 |
| src/chapter3/1164074/praktek.py   | 103 |
| src/chapter3/1164074/praktek.py   | 103 |
| src/chapter3/1164074/praktek.py   | 104 |
| src/chapter3/1164074/praktek.py   | 104 |
| src/chapter3/1164074/praktek.py   | 104 |
| src/chapter3/1164074/praktek.py   | 104 |
| src/chapter3/1164074/3lib.py      | 104 |
| src/chapter3/1164074/main.py      | 107 |
| src/chapter3/1164074/praktek.py   | 107 |
| src/chapter3/1174050_praktek.py   | 108 |
| src/chapter3/1174050_praktek.py   | 108 |
| src/chapter3/1174050_praktek.py   | 108 |
| src/chapter3/1174050_praktek.py   | 108 |
| src/chapter3/1174050_praktek.py   | 108 |
| src/chapter3/1174050_praktek.py   | 109 |
| src/chapter3/1174050_praktek.py   | 109 |
| src/chapter3/1174050_praktek.py   | 109 |
| src/chapter3/1174050_praktek.py   | 110 |
| src/chapter3/1174050_praktek.py   | 111 |
| src/chapter3/1174050_praktek.py   | 111 |
| src/chapter3/1174050_praktek.py   | 112 |
| src/chapter3/1174050_praktek.py   | 112 |
| src/chapter3/1174050_praktek.py   | 112 |
| src/chapter3/1174050_praktek.py   | 112 |
| src/chapter3/1174050_praktek.py   | 112 |
| src/chapter3/1174050_praktek.py   | 113 |
| src/chapter3/1174050_praktek.py   | 113 |
| src/chapter3/1174050_maiiinn.py   | 113 |
| src/chapter3/1174050_errord1ka.py | 113 |

|                                    |     |
|------------------------------------|-----|
| src/chapter3/1174050_errord1ka.py  | 114 |
| src/chapter3/1164081/1164081_1.py  | 114 |
| src/chapter3/1164081/Matematika.py | 114 |
| src/chapter3/1164081/1164081_2.py  | 114 |
| src/chapter3/1164081/contoh.py     | 115 |
| src/chapter3/1164081/1164081_3.py  | 115 |
| src/chapter3/1164081/1164081_4.py  | 115 |
| src/chapter3/1164081/1164081_5.py  | 116 |
| src/chapter3/1164081/1164081_21.py | 116 |
| src/chapter3/1164081/1164081_22.py | 117 |
| src/chapter3/1164081/1164081_23.py | 117 |
| src/chapter3/1164081/1164081_24.py | 117 |
| src/chapter3/1164081/1164081_25.py | 117 |
| src/chapter3/1164081/1164081_26.py | 117 |
| src/chapter3/1164081/1164081_27.py | 118 |
| src/chapter3/1164081/1164081_28.py | 118 |
| src/chapter3/1164081/1164081_29.py | 118 |
| src/chapter3/1164081/1164081_30.py | 118 |
| src/chapter3/1164081/1164081_31.py | 118 |
| src/chapter3/1164081/1164081_32.py | 119 |
| src/chapter3/1164081/1164081_33.py | 119 |

=====

|                              |    |
|------------------------------|----|
| src/chapter2/1174043_1.py    | 9  |
| src/chapter2/1174043_2.py    | 9  |
| src/chapter2/1174043_3.py    | 10 |
| src/chapter2/1174043_4.py    | 10 |
| src/chapter2/1174043_5.py    | 10 |
| src/chapter2/1174043_6.py    | 10 |
| src/chapter2/1174043_7.py    | 10 |
| src/chapter2/1174043_8.py    | 10 |
| src/chapter2/1174043_9.py    | 11 |
| src/chapter2/1174043_10.py   | 11 |
| src/chapter2/1174043_11.py   | 11 |
| src/chapter2/1174043_2err.py | 12 |
| src/chapter2/1174040_1.py    | 13 |
| src/chapter2/1174040_2.py    | 14 |
| src/chapter2/1174040_3.py    | 14 |
| src/chapter2/1174040_4.py    | 14 |

|                              |    |
|------------------------------|----|
| src/chapter2/1174040_5.py    | 14 |
| src/chapter2/1174040_6.py    | 15 |
| src/chapter2/1174040_7.py    | 15 |
| src/chapter2/1174040_8.py    | 15 |
| src/chapter2/1174040_9.py    | 15 |
| src/chapter2/1174040_10.py   | 16 |
| src/chapter2/1174040_11.py   | 16 |
| src/chapter2/1174040_2err.py | 16 |
| src/1174063_teor.py          | 17 |
| src/1174063_teor.py          | 17 |
| src/1174063_teor.py          | 17 |
| src/1174063_teor.py          | 18 |
| src/1174063_teor.py          | 18 |
| src/1174063_teor.py          | 18 |
| src/1174063_teor.py          | 18 |
| src/1174063_teor.py          | 18 |
| src/1174063_teor.py          | 19 |
| src/1174063_praktek.py       | 19 |
| src/1174063_praktek.py       | 19 |
| src/1174063_praktek.py       | 19 |
| src/1174063_praktek.py       | 19 |
| src/1174063_praktek.py       | 20 |
| src/1174063_praktek.py       | 20 |
| src/1174063_praktek.py       | 20 |
| src/1174063_praktek.py       | 20 |
| src/1174063_praktek.py       | 20 |
| src/1174063_praktek.py       | 20 |
| src/errr2.py                 | 20 |
| src/chapter2/1174035_1.py    | 22 |
| src/chapter2/1174035_2.py    | 22 |
| src/chapter2/1174035_3.py    | 23 |
| src/chapter2/1174035_4.py    | 23 |
| src/chapter2/1174035_5.py    | 23 |
| src/chapter2/1174035_6.py    | 24 |
| src/chapter2/1174035_7.py    | 24 |
| src/chapter2/1174035_8.py    | 25 |
| src/chapter2/1174035_9.py    | 25 |

|                              |    |
|------------------------------|----|
| src/chapter2/1174035_10.py   | 26 |
| src/chapter2/1174035_11.py   | 26 |
| src/chapter2/1174035_2err.py | 27 |
| src/chapter2/1174042_1,1.py  | 28 |
| src/chapter2/1174042_1,2.py  | 28 |
| src/chapter2/1174042_1,3.py  | 28 |
| src/chapter2/1174042_1,4.py  | 28 |
| src/chapter2/1174042_1,5.py  | 28 |
| src/chapter2/1174042_1,7.py  | 29 |
| src/chapter2/1174042_2,1.py  | 29 |
| src/chapter2/1174042_2,2.py  | 29 |
| src/chapter2/1174042_2,3.py  | 29 |
| src/chapter2/1174042_2,4.py  | 30 |
| src/chapter2/1174042_2,5.py  | 30 |
| src/chapter2/1174042_2,6.py  | 30 |
| src/chapter2/1174042_2,7.py  | 30 |
| src/chapter2/1174042_2,8.py  | 30 |
| src/chapter2/1174042_2,9.py  | 31 |
| src/chapter2/1174042_2,10.py | 31 |
| src/chapter2/1174042_2,11.py | 32 |
| src/chapter2/1174042_2err.py | 32 |
| src/1174050_teor_i.py        | 33 |
| src/1174050_teor_i.py        | 33 |
| src/1174050_teor_i.py        | 33 |
| src/1174050_teor_i.py        | 33 |
| src/1174050_teor_i.py        | 34 |
| src/1174050_teor_i.py        | 34 |
| src/1174050_teor_i.py        | 34 |
| src/1174050_teor_i.py        | 34 |
| src/1174050_teor_i.py        | 34 |
| src/1174050_praktek.py       | 35 |
| src/1174050_praktek.py       | 35 |
| src/1174050_praktek.py       | 35 |
| src/1174050_praktek.py       | 35 |
| src/1174050_praktek.py       | 35 |
| src/1174050_praktek.py       | 36 |
| src/1174050_praktek.py       | 36 |
| src/1174050_praktek.py       | 36 |



|                                        |    |
|----------------------------------------|----|
| src/1174050_praktek.py                 | 36 |
| src/1174050_praktek.py                 | 36 |
| src/1174050_praktek.py                 | 36 |
| src/error1ka.py                        | 36 |
| src/chapter2/1174034_ichsan.teori.py   | 37 |
| src/chapter2/1174034_ichsan.teori.py   | 37 |
| src/chapter2/1174034_ichsan.teori.py   | 37 |
| src/chapter2/1174034_ichsan.teori.py   | 37 |
| src/chapter2/1174034_ichsan.teori.py   | 37 |
| src/chapter2/1174034_ichsan.teori.py   | 38 |
| src/chapter2/1174034_ichsan.teori.py   | 38 |
| src/chapter2/1174034_ichsan.teori.py   | 38 |
| src/chapter2/1174034_ichsan.teori.py   | 38 |
| src/chapter2/1174034_ichsan.praktek.py | 38 |
| src/chapter2/1174034_ichsan.praktek.py | 39 |
| src/chapter2/1174034_ichsan.praktek.py | 39 |
| src/chapter2/1174034_ichsan.praktek.py | 39 |
| src/chapter2/1174034_ichsan.praktek.py | 39 |
| src/chapter2/1174034_ichsan.praktek.py | 39 |
| src/chapter2/1174034_ichsan.praktek.py | 40 |
| src/chapter2/1174034_ichsan.praktek.py | 40 |
| src/chapter2/1174034_ichsan.praktek.py | 40 |
| src/chapter2/1174034_ichsan.praktek.py | 40 |
| src/chapter2/1174034_ichsan.praktek.py | 40 |
| src/chapter2/errorrr.py                | 40 |
| src/chapter2/1164081_1.py              | 43 |
| src/chapter2/1164081_2.py              | 44 |
| src/chapter2/1164081_3.py              | 44 |
| src/chapter2/1164081_4.py              | 44 |
| src/chapter2/1164081_5.py              | 44 |
| src/chapter2/1164081_6.py              | 44 |
| src/chapter2/1164081_7.py              | 45 |
| src/chapter2/1164081_8.py              | 45 |
| src/chapter2/1164081_9.py              | 45 |
| src/chapter2/1164081_10.py             | 46 |
| src/chapter2/1164081_11.py             | 46 |
| src/chapter2/1164081_2err.py           | 47 |
| src/chapter3/teori_1174043_chap3.py    | 49 |

|                                         |    |
|-----------------------------------------|----|
| src/chapter3/teori_1174043_chap3.py     | 50 |
| src/chapter3/teori_1174043_chap3.py     | 50 |
| src/chapter3/teori_1174043_chap3.py     | 50 |
| src/chapter3/teori_1174043_chap3.py     | 50 |
| src/chapter3/teori_1174043_chap3.py     | 50 |
| src/chapter3/teori_1174043_chap3.py     | 51 |
| src/chapter3/chap3_1174043_no1.py       | 51 |
| src/chapter3/chap3_1174043_no2.py       | 52 |
| src/chapter3/chap3_1174043_no3.py       | 52 |
| src/chapter3/chap3_1174043_no4.py       | 53 |
| src/chapter3/chap3_1174043_no5.py       | 54 |
| src/chapter3/chap3_1174043_no6.py       | 55 |
| src/chapter3/chap3_1174043_no7.py       | 56 |
| src/chapter3/chap3_1174043_no8.py       | 57 |
| src/chapter3/chap3_1174043_no9.py       | 57 |
| src/chapter3/chap3_1174043_no10.py      | 58 |
| src/chapter3/chap3_1174043_3lib.py      | 59 |
| src/chapter3/chap3_1174043_main.py      | 62 |
| src/chapter3/chap3_1174043_kelas3lib.py | 62 |
| src/chapter3/chap3_1174043_3err.py      | 65 |
| src/chapter3/chap3_1174035_teor_i.py    | 66 |
| src/chapter3/chap3_1174035_teor_i.py    | 66 |
| src/chapter3/chap3_1174035_teor_i.py    | 66 |
| src/chapter3/chap3_1174035_teor_i.py    | 67 |
| src/chapter3/chap3_1174035_teor_i.py    | 67 |
| src/chapter3/chap3_1174035_teor_i.py    | 67 |
| src/chapter3/chap3_1174035_teor_i.py    | 67 |
| src/chapter3/chap3_1174035_1.py         | 67 |
| src/chapter3/chap3_1174035_2.py         | 68 |
| src/chapter3/chap3_1174035_3.py         | 69 |
| src/chapter3/chap3_1174035_4.py         | 69 |
| src/chapter3/chap3_1174035_5.py         | 70 |
| src/chapter3/chap3_1174035_6.py         | 71 |
| src/chapter3/chap3_1174035_7.py         | 71 |
| src/chapter3/chap3_1174035_8.py         | 72 |
| src/chapter3/chap3_1174035_9.py         | 72 |
| src/chapter3/chap3_1174035_10.py        | 73 |
| src/chapter3/chap3_1174035_3lib.py      | 73 |

|                                         |    |
|-----------------------------------------|----|
| src/chapter3/chap3_1174035_main.py      | 76 |
| src/chapter3/chap3_1174035_kelas3lib.py | 77 |
| src/chapter3/chap3_1174035_main.py      | 78 |
| src/chapter3/chap3_1174035_error.py     | 79 |
| src/chapter3/chap3_1174035_main.py      | 80 |
| src/1174056_praktek.py                  | 81 |
| src/1174056_praktek.py                  | 81 |
| src/1174056_praktek.py                  | 81 |
| src/1174056_praktek.py                  | 81 |
| src/1174056_praktek.py                  | 81 |
| src/1174056_praktek.py                  | 82 |
| src/1174056_praktek.py                  | 82 |
| src/1174056_praktek.py                  | 82 |
| src/1174056_praktek.py                  | 83 |
| src/1174056_praktek.py                  | 84 |
| src/1174056_praktek.py                  | 84 |
| src/1174056_praktek.py                  | 85 |
| src/1174056_praktek.py                  | 85 |
| src/1174056_praktek.py                  | 85 |
| src/1174056_praktek.py                  | 85 |
| src/1174056_praktek.py                  | 85 |
| src/1174056_praktek.py                  | 86 |
| src/1174056_praktek.py                  | 86 |
| src/mainn.py                            | 86 |
| src/mainn.py                            | 86 |
| src/1174056_praktek.py                  | 87 |
| src/chapter3/1174042_1,1.py             | 87 |
| src/chapter3/1174042_1,1,1.py           | 87 |
| src/chapter3/1174042_1,1,2.py           | 87 |
| src/chapter3/1174042_1,2.py             | 87 |
| src/chapter3/1174042_1,3.py             | 88 |
| src/chapter3/1174042_1,4.py             | 88 |
| src/chapter3/1174042_1,2.py             | 88 |
| src/chapter3/1174042_1,6.py             | 88 |
| src/chapter3/1174042_1,6.py             | 89 |
| src/chapter3/1174042_2,1.py             | 89 |
| src/chapter3/1174042_2,2.py             | 89 |
| src/chapter3/1174042_2,3.py             | 89 |

|                                    |     |
|------------------------------------|-----|
| src/chapter3/1174042_2,4.py        | 90  |
| src/chapter3/1174042_2,5.py        | 90  |
| src/chapter3/1174042_2,6.py        | 90  |
| src/chapter3/1174042_2,7.py        | 91  |
| src/chapter3/1174042_2,8.py        | 91  |
| src/chapter3/1174042_2,9.py        | 92  |
| src/chapter3/1174042_2,10.py       | 92  |
| src/chapter3/1174042_main.py       | 92  |
| src/chapter3/1174042_main.py       | 93  |
| src/chapter3/1174042_err.py        | 93  |
| src/chapter2/1164089/1164089_1.py  | 93  |
| src/chapter2/1164089/Matematika.py | 93  |
| src/chapter2/1164089/1164089_2.py  | 94  |
| src/chapter2/1164089/mtk.py        | 94  |
| src/chapter2/1164089/1164089_3.py  | 94  |
| src/chapter2/1164089/1164089_4.py  | 95  |
| src/chapter2/1164089/1164089_5.py  | 95  |
| src/chapter2/1164089/1164089_21.py | 96  |
| src/chapter2/1164089/1164089_22.py | 97  |
| src/chapter2/1164089/1164089_23.py | 97  |
| src/chapter2/1164089/1164089_24.py | 97  |
| src/chapter2/1164089/1164089_25.py | 97  |
| src/chapter2/1164089/1164089_26.py | 97  |
| src/chapter2/1164089/1164089_27.py | 97  |
| src/chapter2/1164089/1164089_28.py | 98  |
| src/chapter2/1164089/1164089_29.py | 98  |
| src/chapter2/1164089/1164089_30.py | 98  |
| src/chapter2/1164089/1164089_31.py | 98  |
| src/chapter2/1164089/1164089_32.py | 98  |
| src/chapter2/1164089/1164089_33.py | 99  |
| src/chapter3/1164074/praktek.py    | 99  |
| src/chapter3/1164074/praktek.py    | 99  |
| src/chapter3/1164074/praktek.py    | 99  |
| src/chapter3/1164074/praktek.py    | 100 |
| src/chapter3/1164074/praktek.py    | 100 |
| src/chapter3/1164074/praktek.py    | 100 |
| src/chapter3/1164074/praktek.py    | 100 |
| src/chapter3/1164074/praktek.py    | 100 |

|                                 |     |
|---------------------------------|-----|
| src/chapter3/1164074/praktek.py | 101 |
| src/chapter3/1164074/praktek.py | 101 |
| src/chapter3/1164074/praktek.py | 101 |
| src/chapter3/1164074/praktek.py | 101 |
| src/chapter3/1164074/praktek.py | 101 |
| src/chapter3/1164074/praktek.py | 101 |
| src/chapter3/1164074/praktek.py | 102 |
| src/chapter3/1164074/praktek.py | 103 |
| src/chapter3/1164074/praktek.py | 103 |
| src/chapter3/1164074/praktek.py | 103 |
| src/chapter3/1164074/praktek.py | 103 |
| src/chapter3/1164074/praktek.py | 103 |
| src/chapter3/1164074/praktek.py | 104 |
| src/chapter3/1164074/praktek.py | 104 |
| src/chapter3/1164074/praktek.py | 104 |
| src/chapter3/1164074/praktek.py | 104 |
| src/chapter3/1164074/3lib.py    | 104 |
| src/chapter3/1164074/main.py    | 107 |
| src/chapter3/1164074/praktek.py | 107 |
| src/chapter3/1174050_praktek.py | 108 |
| src/chapter3/1174050_praktek.py | 108 |
| src/chapter3/1174050_praktek.py | 108 |
| src/chapter3/1174050_praktek.py | 108 |
| src/chapter3/1174050_praktek.py | 108 |
| src/chapter3/1174050_praktek.py | 109 |
| src/chapter3/1174050_praktek.py | 109 |
| src/chapter3/1174050_praktek.py | 109 |
| src/chapter3/1174050_praktek.py | 110 |
| src/chapter3/1174050_praktek.py | 111 |
| src/chapter3/1174050_praktek.py | 111 |
| src/chapter3/1174050_praktek.py | 112 |
| src/chapter3/1174050_praktek.py | 112 |
| src/chapter3/1174050_praktek.py | 112 |
| src/chapter3/1174050_praktek.py | 112 |
| src/chapter3/1174050_praktek.py | 112 |
| src/chapter3/1174050_praktek.py | 112 |
| src/chapter3/1174050_praktek.py | 113 |
| src/chapter3/1174050_praktek.py | 113 |
| src/chapter3/1174050_maiinn.py  | 113 |

|                                                 |     |
|-------------------------------------------------|-----|
| src/chapter3/1174050_errord1ka.py               | 113 |
| src/chapter3/1174050_errord1ka.py               | 114 |
| src/chapter3/1164081/1164081_1.py               | 114 |
| src/chapter3/1164081/Matematika.py              | 114 |
| src/chapter3/1164081/1164081_2.py               | 114 |
| src/chapter3/1164081/contoh.py                  | 115 |
| src/chapter3/1164081/1164081_3.py               | 115 |
| src/chapter3/1164081/1164081_4.py               | 115 |
| src/chapter3/1164081/1164081_5.py               | 116 |
| src/chapter3/1164081/1164081_21.py              | 116 |
| src/chapter3/1164081/1164081_22.py              | 117 |
| src/chapter3/1164081/1164081_23.py              | 117 |
| src/chapter3/1164081/1164081_24.py              | 117 |
| src/chapter3/1164081/1164081_25.py              | 117 |
| src/chapter3/1164081/1164081_26.py              | 117 |
| src/chapter3/1164081/1164081_27.py              | 118 |
| src/chapter3/1164081/1164081_28.py              | 118 |
| src/chapter3/1164081/1164081_29.py              | 118 |
| src/chapter3/1164081/1164081_30.py              | 118 |
| src/chapter3/1164081/1164081_31.py              | 118 |
| src/chapter3/1164081/1164081_32.py              | 119 |
| src/chapter3/1164081/1164081_33.py              | 119 |
| src/chapter4/1174042_csv.py                     | 123 |
| src/chapter4/1174042_csv.py                     | 123 |
| src/chapter4/1174042_csv.py                     | 123 |
| src/chapter4/1174042_csv.py                     | 123 |
| >>>>>> c64ab529ff89ebc66e82900d0611abe5d2422d5a |     |



# FOREWORD

---

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa





# KATA PENGANTAR

---

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat  
Februari, 2019*



# ACKNOWLEDGMENTS

---

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

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

R. M. A.



# ACRONYMS

---

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



# GLOSSARY

---

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





# SYMBOLS

---

$A$  Amplitude

$\&$  Propositional logic symbol

$a$  Filter Coefficient

$\mathcal{B}$  Number of Beats



# INTRODUCTION

---

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

Informatics Research Center  
Bandung, Jawa Barat, Indonesia

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

Rumusnya cuman sebagai contoh aja biar keren[?].

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



# BAB 1

---

## 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 [?].

Python memiliki kepastakaan 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 [?].

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 [?].

### **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 kalan-

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 IrvanRizkiansyah/1174043

#### 2.1.1 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 > 3000000:
    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
```

### 2.1.2 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
```

```
21 print('Masukkan NPM Anda : ')
22 NPM = input()
23 y = NPM[5:7]
24 g = int(y)
25 for i in range(g):
26     print('Halo ' + NPM, 'Apa Kabar?')
```

```

31 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. a = 1
   b = 1
   c = 7
   d = 4
   e = 0
   f = 4
   g = 3
   array = []
   array.append(a)
   array.append(b)
   array.append(c)
   array.append(d)
   array.append(e)
   array.append(f)
   array.append(g)
   for x in array:
       if x != 0:
           if x%2==0:
               print(x, end = " ")
```

```
10. a = 1
    b = 1
    c = 7
    d = 4
    e = 0
    f = 4
    g = 3
    array = []
    array.append(a)
    array.append(b)
    array.append(c)
    array.append(d)
    array.append(e)
    array.append(f)
    array.append(g)
    for x in array:
        if x != 0:
            if x%2==1:
                print(x, end = " ")
```

```
11. a = 1
    b = 1
    c = 7
    d = 4
    e = 0
    f = 4
    g = 3
    array = []
    array.append(a)
    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)

```

### 2.1.3 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")

```

## 2.2 Hagan Rowlenstino/1174040

### 2.2.1 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 tulisaan yang akan ditampilkan.

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

## 2.2.2 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))

```

```
6: 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)
```

```
7: 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)
```

```
8: 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)
```

```
9: 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 = " ")
```

```

10_1 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_1 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)

```

### 2.2.3 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_1 a = 2
    2 b = 'as'

```

```

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

```

## 2.3 Muhammad Iqbal Panggabean

### 2.3.1 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 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")

```

## 2.3.2 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("Hallo , " + str(npm) + "Apa kabar?")
5     hitung = hitung +1

```

### 3. Jawaban soal no 3

```

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

```

### 4. Jawaban soal no 4

```

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

```



## 5. Jawaban soal no 5

```

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

```

## 2.3.3 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")

```

## 2.4 Luthfi M. Nabil/1174035

### 2.4.1 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")
```

## 2.4.2 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
```

```
21 # -*- coding: utf-8 -*-
22 """
```

```

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

```

```

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, 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))

```

```

8. # -*- 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)

```

```

9. # -*- 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 = " ")

```

```

10. # -*- 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 = " ")

```

```

11. # -*- 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)

```

### 2.4.3 Keterampilan Penanganan Error

1.
  - 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")

```

## 2.5 Faisal Najib Abdullah 1174042

### 2.5.1 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 jadi 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")

```

## 2.5.2 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?")

```

```

41 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)
```

```
9. 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)

```

### 2.5.3 Keterampilan Penanganan Error

1. Pada saat mengerjakan praktek kedua ini error hanya pada kesalahan type data yaitu `TypeError`., solusinya yaitu merubah 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")

```

## 2.6 Dika Sukma Pradana 1174050

### 2.6.1 Teori

1. Jenis-jenis variable python dan cara pemakaiannya Variabel merupakan tempat menyimpan data. Dalam Python terdapat beberapa variabel dengan berbagai type data diantaranya adalah variabel dengan type data number, string, dan boolean. Dalam python 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")

```

## 2.6.2 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)
```

## 2.6.3 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!!")
```

## 2.7 Ichsan Hizman Hardy

### 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 = "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 python 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")
```

### 2.7.2 praktik

1. Jawaban soal no 1

```

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

```

```

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 <= 34):
4     print("Hallo , " + str(npm) + "Apa kabar?")
5     hitung = hitung +1

```

## 3. Jawaban soal no 3

```

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

```

## 4. Jawaban soal no 4

```

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

```

## 5. Jawaban soal no 5

```

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

### 2.7.3 Keterampilan dan penanganan eror

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

## 2.8 Mhd Zulfikar Akram Nasution/ 1164081

### 2.8.1 Teori

#### 1. Python memiliki dua Variabel yaitu integer dan string

- Contoh Variabel String

```
x="Zulfikar"
y="D4 Teknik Informatika"
z=x + y
print(z)
```

Output dari source code tersebut adalah Zulfikar D4 Teknik Informatika

- Contoh Variabel Integer

```
x=5
y=10
print (x+y)
```

Output dari source code tersebut adalah 15

## 2. Cara melakukan proses input

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

Dimana ketika kita running akan meminta form inputan dan apabila saya berikan inputan Zulfikar maka output yang keluar adalah Hello Zulfikar di karenakan variabel x adalah inputan yang kita berikan.

## 3. Operator dasar aritmatika

### ▪ Aritmatika Pertambahan

```
x=5
y=3
print (x+y)
```

Output yang keluar dari source code tersebut adalah 8 di karenakan 5 ditambah 3 sama dengan 8.

### ▪ Aritmatika Pengurangan

```
x=5
y=3
print (x-y)
```

Output yang keluar dari source code tersebut adalah 2.

### ▪ Aritmatika Perkalian

```
x=5
y=3
print (x*y)
```

Output yang keluar dari source code tersebut adalah 15.

### ▪ Aritmatika Pembagian

```
x=5
```

```
y=3
print (x/y)
```

Output yang keluar dari source code tersebut adalah 1,67.

- Merubah Integer Ke String

```
x=5
y=3
z=str(x)+str(y)
```

- Merubah String Ke Integer

```
x='5'
y='3'
z=int(x)+int(y)
```

#### 4. Python memiliki dua type pengulangan yaitu While Looping dan For Looping

- Contoh While Looping

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

Output yang akan keluar adalah memunculkan angka 1 samapi 5 dikarenakan adanya source code

```
while i<6
```

- Contoh For Looping

```
for x in "banana":
    print(x)
```

Output yang keluar adalah mengulang huruf b a n a n a secara vertikal.

#### 5. Contoh source code condition adalah seperti berikut

```
a = 33
b = 33
if b > a:
    print("b is greater than a")
elif a == b:
    print("a and b are equal")
```

Output dari source code tersebut adalah a and b are equal di karenakan nilai vaiabel a dan b sama.

## 6. Jenis Errorr yang ditemukan

dalam kasus ini errorr yang saya temukan adalah source code pada nomor aritmatika pembagian.

```
x=5
y='3'
print (x/y)
```

pada source code tersebut akan terjadi errorr dikarenakan integer dan string tidak dapat di persatukan. Untu penyelesaiannya adalah sebagai berikut.

```
x=5
y=3
print (x/y)
```

## 7. Try except

```
x = 3
try:
    print(x)
except NameError:
    print("Variable x is not defined")
except:
    print("Something else went wrong")
```

Output yang dikeluarkan adalah 3 dikarenakan variabel dari x bernilai 3.

## 2.8.2 Keterampilan Pemrograman

```
1 NPM = 1164081
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 row.append("  ##  ")
18 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
19 row.append("  ##  ")
20 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
21 row.append("  ##  ")
22 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
23 row.append("  ##  ")
24 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
25 row.append("  ##  ")
26 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
27 row.append("  ##  ")
28 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
29 row.append("  ##  ")
30 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
31 row.append("  ##  ")
32 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
33 row.append("  ##  ")
34 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
35 row.append("  ##  ")
36 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
37 row.append("  ##  ")
38 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
39 row.append("  ##  ")
40 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
41 row.append("  ##  ")
42 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
43 row.append("  ##  ")
44 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
45 row.append("  ##  ")
46 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
47 row.append("  ##  ")
48 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
49 row.append("  ##  ")
50 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
51 row.append("  ##  ")
52 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
53 row.append("  ##  ")
54 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
55 row.append("  ##  ")
56 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
57 row.append("  ##  ")
58 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
59 row.append("  ##  ")
60 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
61 row.append("  ##  ")
62 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
63 row.append("  ##  ")
64 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
65 row.append("  ##  ")
66 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
67 row.append("  ##  ")
68 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
69 row.append("  ##  ")
70 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
71 row.append("  ##  ")
72 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
73 row.append("  ##  ")
74 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
75 row.append("  ##  ")
76 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
77 row.append("  ##  ")
78 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
79 row.append("  ##  ")
80 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
81 row.append("  ##  ")
82 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
83 row.append("  ##  ")
84 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
85 row.append("  ##  ")
86 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
87 row.append("  ##  ")
88 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
89 row.append("  ##  ")
90 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
91 row.append("  ##  ")
92 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
93 row.append("  ##  ")
94 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
95 row.append("  ##  ")
96 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
97 row.append("  ##  ")
98 row.append("  ##          ##  #####  ##          ##  ##          ##  #####")
99 row.append("  ##  ")
100 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 a = NPM[5:7]
4 b = int(a)
5 for i in range(b):
6     print('Halo ' + NPM, 'Apa Kabar?')

```

```

31 print("Masukkan NPM:")
2 x = input()
3 a = 0
4 b = int(x[4])
5 c = int(x[5])
6 d = int(x[6])
7 e = x[4:7]
8 f = b+c+d
9
10 while a < f:
11     a += 1
12     print("Hello , " + e + " Apa Kabar?")

```

```

41 print("Masukkan Npm:")
2 x = input()
3 y = x[4]
4 print("Hello , " + y + " Apa Kabar?")

```

```

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

```

```

61 print("NPM Anda:")
2 a = 1
3 b = 1
4 c = 6
5 d = 4
6 e = 0

```

```
7 f = 8
8 g = 1
9 x = a+b+c+d+e+f+g
10 print(x)
```

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

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

```
91 print("NPM Anda:")
2 a = 1
3 b = 1
4 c = 6
5 d = 4
6 e = 0
7 f = 8
8 g = 1
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("NPM Anda:")
2  a = 1
3  b = 1
4  c = 6
5  d = 4
6  e = 0
7  f = 8
8  g = 1
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
2  b = 1
3  c = 6
4  d = 4
5  e = 0
6  f = 8
7  g = 1
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)

```

### 2.8.3 Keterampilan Penanganan Error

1. Contoh error yang saya dapat berada pada nomor 5. Dalam no 5 terdapat 7 variabel dan akan di tampilkan output berupa string. Pada awalnya saya membuat source code seperti berikut: Print(a,b,c,d,e,f,g) Sehingga output yang dikeluarkan adalah 1 1 6 4 0 8 1 dan masih berupa integer. Untuk menampilkan output String maka tambahkan source code str seperti contoh berikut : print(str(a)+str(b)+str(c)+s  
Output yang di keluarkan nantinya adalah 1164081 dengan type data String

```

21 a = 8
2 b = "ahoy"
3
4 try :
5     print = a + b
6 except TypeError:
7     print("Terjadi Erorr di karenakan string dan integer tidak
    dapat di satukan")

```



## BAB 3

---

# JUDUL BAGIAN KETIGA

---

### 3.1 IrvanRizkiansyah/1174043

#### 3.1.1 Pemahaman Teori

- Fungsi adalah bagian dari program yang berupa blok kode yang diberikan nama dan nama tersebut berguna untuk memanggil fungsi tersebut.
- Inputan fungsi adalah sebuah fungsi yang telah di sediakan pada library python, yang berguna untuk menerima inputan dari user.
- Kembalian fungsi adalah sebuah nilai balikan yang diberikan oleh sebuah fungsi yang dibuat.

Contoh Program :

```
1 def LuasPersegi(s): ##pendefinisian Fungsi
2     x = s * s
3     return x ##Nilai kembalian fungsi
4     print("Masukkan Sisi Persegi = ")
5     sisi = input() ##Fungsi input
6     s = int(sisi)
7     print(LuasPersegi(s))
```

2. paket adalah sebuah cara yang dilakukan untuk memanggil file script python, yang nantinya akan digunakan fungsi fungsi yang terdapat pada file script yang dipanggil tersebut. cara pemanggilan paket dengan cara :

```
import scriptFilePython
```

Contoh Program :

```
1 import kalkulator
```

3.
  - Kelas merupakan sebuah cetakan atau Blueprint yang berguna untuk mencetak objek.
  - Objek merupakan sebuah objek yang dari proses hasil dari cetakan atau blueprint.
  - Atribut merupakan penggambaran data yang bisa memberikan sebuah informasi kelas atau objek dimana atribut tersebut berada.
  - Method merupakan fungsi atau prosedur yang bergabung dengan sebuah objek dan juga atribut.

Contoh Program :

```
1 class MyClass: ##pendefinisian Kelas
2     a = 10 ##atribut
3     def func(self): ##method
4         print('Hello')
```

4. cara pemanggilan library kelas dari instansiasi, adalah dengan cara mengubah library kelas yang dipanggil menjadi sebuah objek.

Contoh Program :

```
1 import kalkulator
2
3 a = kalkulator ##instansiasi library kelas
```

5. Jadi pemakaian paket dengan perintah from kalkulator import penambahan berguna untuk menghemat memori pemakaian pada program, karena hanya memanggil fungsi yang diperlukan saja pada library yang terpanggil. cara memanggilnya dengan cara :

```
from scripFilePython import namaFungsi, namaFungsi2
```

Contoh Program :

```
1 from kalkulator import penambahan, pengurangan
```

6.
 

```
1 import kalkulator
2 a = 10
3 b = 20
4
5 c = kalkulator.penambahan(a,b)
6 print(c)
```

```

74 import kalkulator
2
3 x = kalkulator.Hitung(3,4)
4 hasil = x.Penambahan()
5 print(str(hasil))

```

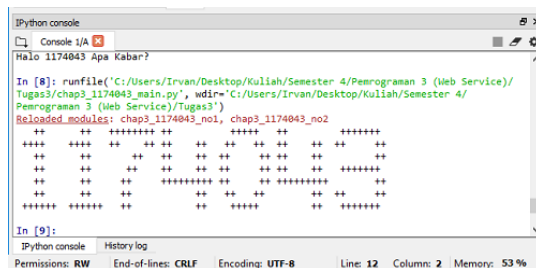
### 3.1.2 Keterampilan Pemrograman

#### 1. Jawaban soal no 1

```

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    baris = []
11    baris.append("    ##          ## ##### ##          ##
12                  ##### ")
13    baris.append("    ##          ##          ##          ##          ##          ##
14                  ##          ")
15    baris.append("    ##          ##          ##          ##          ##          ##
16                  ##          ")
17    baris.append("    ##          ##          ##          ##          ##          ##
18                  ##          ")
19    baris.append("    ##          ##          ##          ##          ##          ##
20                  ##          ")
21    baris.append("    ##          ##          ##          ##          ##          ##
22                  ##          ")
23    a = 0
24    for x in baris:
25        print(x.replace("#", char))
26        a+=1
27    nomorSatu("1174043")

```



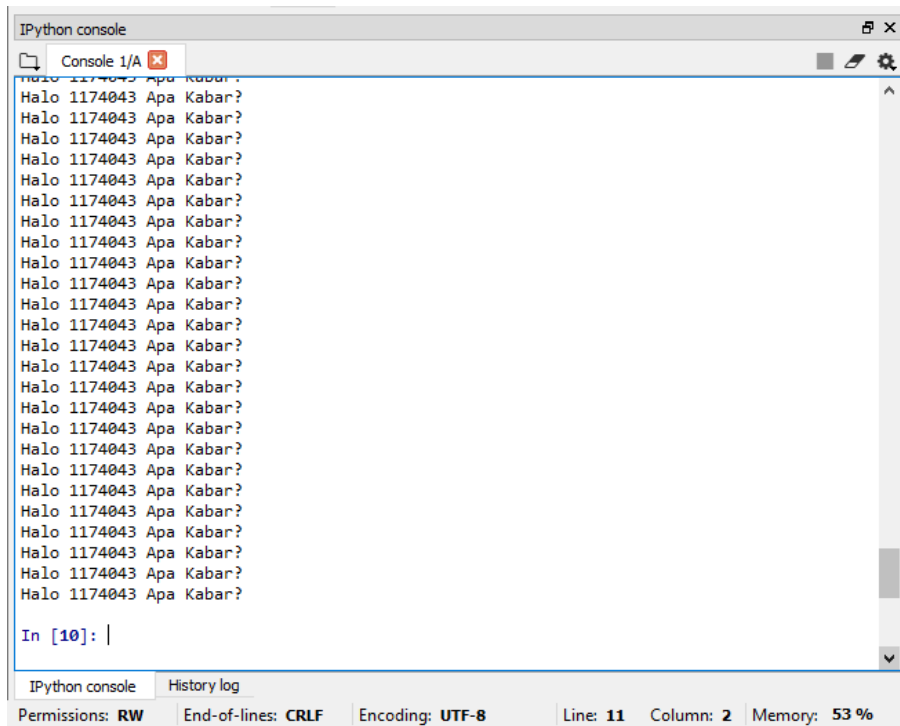
Gambar 3.1 Jawaban No. 1



??

## 2. Jawaban soal no 2

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 20:18:18 2019
4
5 @author: Irvan
6 """
7
8 def nomorDua(NPM):
9     y = NPM[5:7]
10     g = int(y)
11     for i in range(g):
12         print(' Halo ' + NPM, 'Apa Kabar?')
13
14 nomorDua("1174043")
```



**Gambar 3.2** Jawaban No. 2

??

### 3. Jawaban soal no 3

```
1 # -*- coding: utf-8 -*-
```

```

2  """
3  Created on Mon Mar 11 20:23:54 2019
4
5  @author: Irvan
6  """
7
8  def nomorTiga(NPM):
9      y = NPM[4:7]
10     x = NPM[4:5]
11     y = NPM[5:6]
12     z = NPM[6:7]
13     g = (int(x) + int(y) + int(z))
14     for i in range(g):
15         print('Halo ' + y, 'Apa Kabar?')
16
17 nomorTiga("1174043")

```

```

IPython console
Console 1/A
\spydercustomize.py, line 704, in runfile
execfile(filename, namespace)

File "C:\Users\Irvan\Anaconda3\lib\site-packages\spyder_kernels\customize
\spydercustomize.py", line 108, in execfile
exec(compile(f.read(), filename, 'exec'), namespace)

File "C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/Tugas3/
chap3_1174043_main.py", line 14, in <module>
chap3_1174043_no2.nomorTiga("1174043")

AttributeError: module 'chap3_1174043_no2' has no attribute 'nomorTiga'

In [12]:

In [12]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/
Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/
Pemrograman 3 (Web Service)/Tugas3')
AttributeError: module 'chap3_1174043_no1' has no attribute 'nomorTiga'
AttributeError: module 'chap3_1174043_no2' has no attribute 'nomorTiga'
AttributeError: module 'chap3_1174043_no3' has no attribute 'nomorTiga'
Halo 4 Apa Kabar?
Halo 4 Apa Kabar?
Halo 4 Apa Kabar?
Halo 4 Apa Kabar?
Halo 4 Apa Kabar?
Halo 4 Apa Kabar?
Halo 4 Apa Kabar?
In [13]:

IPython console History log
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 14 Column: 18 Memory: 52 %

```

Gambar 3.3 Jawaban No. 3

??

## 4. Jawaban soal no 4

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Mar 11 20:25:58 2019

```

```

4
5 @author: Irvan
6 """
7
8 def nomorEmpat(NPM):
9     y = NPM[4:5]
10     print('Halo ' + y, 'Apa Kabar?')
11
12 nomorEmpat("1174043")

```

The screenshot shows an IPython console window with the following content:

```

File "<ipython-input-13-639e7eb2fa16>", line 1, in <module>
    runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/Tugas3')

File "C:\Users\Irvan\Anaconda3\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 704, in runfile
    execfile(filename, namespace)

File "C:\Users\Irvan\Anaconda3\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 108, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

File "C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/Tugas3/chap3_1174043_main.py", line 16
    import chap3_1174043_no3.nomorEmpat("1174043")
                                         ^
SyntaxError: invalid syntax

In [14]:

In [14]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/Tugas3')
Halo 0 Apa Kabar?

In [15]:

```

The status bar at the bottom indicates: Permissions: RW, End-of-lines: CRLF, Encoding: UTF-8, Line: 16, Column: 1, Memory: 53 %.

Gambar 3.4 Jawaban No. 4

??

## 5. Jawaban soal no 5

```

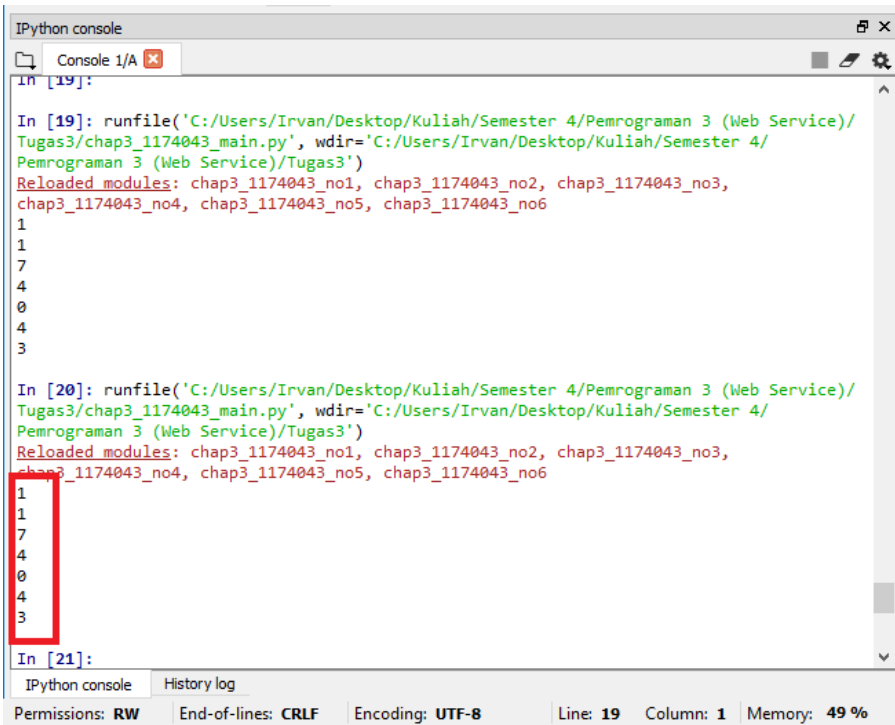
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 20:27:39 2019
4
5 @author: Irvan
6 """
7
8 def nomorLima(NPM):
9     NPM = list(NPM)
10     for x in NPM:

```

```

11     print(x)
12
13 nomorLima("1174043")

```



```

IPython console
Console 1/A
In [19]:
In [19]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/Tugas3')
Reloaded modules: chap3_1174043_no1, chap3_1174043_no2, chap3_1174043_no3, chap3_1174043_no4, chap3_1174043_no5, chap3_1174043_no6
1
1
7
4
0
4
3

In [20]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/Tugas3')
Reloaded modules: chap3_1174043_no1, chap3_1174043_no2, chap3_1174043_no3, chap3_1174043_no4, chap3_1174043_no5, chap3_1174043_no6
1
1
7
4
0
4
3

In [21]:
IPython console History log
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 19 Column: 1 Memory: 49 %

```

**Gambar 3.5** Jawaban No. 5

??

## 6. Jawaban soal no 6

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 20:36:07 2019
4
5 @author: Irvan
6 """
7
8 def nomorEnam(NPM):
9     NPM = list(NPM)
10     jmlh = 0
11     for x in NPM:
12         jmlh = jmlh + int(x)
13     print("Hasilnya : " + str(jmlh))
14
15 nomorEnam("1174043")

```

```

IPython console
Console 1/A

1
1
7
4
0
4
3

In [20]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/
Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/
Pemrograman 3 (Web Service)/Tugas3')
Reloaded modules: chap3_1174043_no1, chap3_1174043_no2, chap3_1174043_no3,
chap3_1174043_no4, chap3_1174043_no5, chap3_1174043_no6

1
1
7
4
0
4
3

In [21]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/
Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/
Pemrograman 3 (Web Service)/Tugas3')
Reloaded modules: chap3_1174043_no1, chap3_1174043_no2, chap3_1174043_no3,
chap3_1174043_no4, chap3_1174043_no5, chap3_1174043_no6
Hasilnya : 20

In [22]:

```

IPython console History log

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 20 Column: 1 Memory: 49 %

Gambar 3.6 Jawaban No. 6

??

## 7. Jawaban soal no 7

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 20:43:26 2019
4
5 @author: Irvan
6 """
7
8 def nomorTujuh(NPM):
9     NPM = list(NPM)
10    jmlh = 0
11    for x in NPM:
12        jmlh = jmlh * int(x)
13    print("Hasilnya : " + str(jmlh))
14
15 nomorTujuh("1174043")

```

??

## 8. Jawaban soal no 8

```

IPython console
Console 1/A

In [20]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/
Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/
Pemrograman 3 (Web Service)/Tugas3')
Reloaded modules: chap3_1174043_no1, chap3_1174043_no2, chap3_1174043_no3,
chap3_1174043_no4, chap3_1174043_no5, chap3_1174043_no6
1
1
7
4
0
3

In [21]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/
Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/
Pemrograman 3 (Web Service)/Tugas3')
Reloaded modules: chap3_1174043_no1, chap3_1174043_no2, chap3_1174043_no3,
chap3_1174043_no4, chap3_1174043_no5, chap3_1174043_no6
Hasilnya : 20

In [22]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/
Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/
Pemrograman 3 (Web Service)/Tugas3')
Reloaded modules: chap3_1174043_no1, chap3_1174043_no2, chap3_1174043_no3,
chap3_1174043_no4, chap3_1174043_no5, chap3_1174043_no6
Hasilnya : 0

In [23]:

```

IPython console    History log

Permissions: RW    End-of-lines: CRLF    Encoding: UTF-8    Line: 22    Column: 1    Memory: 49 %

Gambar 3.7    Jawaban No. 7

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 20:45:48 2019
4
5 @author: Irvan
6 """
7
8 def nomorDelapan(NPM):
9     NPM = list(NPM)
10    for x in NPM:
11        if int(x) != 0:
12            if int(x)%2==0:
13                print(x, end = "")
14
15 nomorDelapan("1174043")

```

??

## 9. Jawaban soal no 9

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 20:53:00 2019
4

```

```

IPython console
Console 1/A
\spydercustomize.py, line 704, in runfile
execfile(filename, namespace)

File "C:\Users\Irvan\Anaconda3\lib\site-packages\spyder_kernels\customize
\spydercustomize.py", line 108, in execfile
exec(compile(f.read(), filename, 'exec'), namespace)

File "C:\Users\Irvan\Desktop\Kuliah\Semester 4\Pemrograman 3 (Web Service)\Tugas3/
chap3_1174043_main.py", line 24, in <module>
chap3_1174043_no8.nomorDelapan("1174043")

File "C:\Users\Irvan\Desktop\Kuliah\Semester 4\Pemrograman 3 (Web Service)
\Tugas3\chap3_1174043_no8.py", line 12, in nomorDelapan
if x%2==0:

TypeError: not all arguments converted during string formatting

In [29]:

In [29]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/
Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/
Pemrograman 3 (Web Service)/Tugas3')
Reloaded modules: chap3_1174043_no1, chap3_1174043_no2, chap3_1174043_no3,
chap3_1174043_no4, chap3_1174043_no5, chap3_1174043_no6, chap3_1174043_no7,
chap3_1174043_no8
44
In [30]:

```

IPython console History log

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 24 Column: 1 Memory: 48 %

Gambar 3.8 Jawaban No. 8

```

5 @author: Irvan
6 """
7
8 def nomorSembilan(NPM):
9     for x in NPM:
10         if int(x) != 0:
11             if int(x)%2==1:
12                 print(x, end = "")
13
14 nomorSembilan("1174043")

```

??

## 10. Jawaban soal no 10

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 20:59:17 2019
4
5 @author: Irvan
6 """
7
8 def nomorSepuluh(NPM):
9     for x in NPM:

```

The screenshot shows an IPython console window with the following content:

```

chapg3_1174043_main.py , line 24, in <module>
    chap3_1174043_no8.nomorDelapan("1174043")

File "C:\Users\Irvan\Desktop\Kuliah\Semester 4\Pemrograman 3 (Web Service)
\tugas3\chap3_1174043_no8.py", line 12, in nomorDelapan
    if x%2==0:

TypeError: not all arguments converted during string formatting

In [29]:

In [29]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/
Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/
Pemrograman 3 (Web Service)/Tugas3')
Reloaded modules: chap3_1174043_no1, chap3_1174043_no2, chap3_1174043_no3,
chap3_1174043_no4, chap3_1174043_no5, chap3_1174043_no6, chap3_1174043_no7,
chap3_1174043_no8
44

In [30]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/
Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/
Pemrograman 3 (Web Service)/Tugas3')
Reloaded modules: chap3_1174043_no1, chap3_1174043_no2, chap3_1174043_no3,
chap3_1174043_no4, chap3_1174043_no5, chap3_1174043_no6, chap3_1174043_no7,
chap3_1174043_no8
1173

In [31]:

```

The status bar at the bottom indicates: IPython console, History log, Permissions: RW, End-of-lines: CRLF, Encoding: UTF-8, Line: 26, Column: 32, Memory: 49 %.

Gambar 3.9 Jawaban No. 9

```

10         if int(x) != 0:
11             i = 1
12             bil = 0
13             while i <= int(x):
14                 if int(x)%i==0:
15                     bil+=1
16                     i+=1
17             if bil == 2:
18                 print(x)
19
20 nomorSepuluh("1174043")

```

??

## 11. Jawaban soal no 11

File 3lib.py

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 22:38:15 2019
4
5 @author: Irvan
6 """

```





```

IPython console
Console 1/A

In [29]:

In [29]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/
Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/
Pemrograman 3 (Web Service)/Tugas3')
Reloaded modules: chap3_1174043_no1, chap3_1174043_no2, chap3_1174043_no3,
chap3_1174043_no4, chap3_1174043_no5, chap3_1174043_no6, chap3_1174043_no7,
chap3_1174043_no8
44

In [30]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/
Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/
Pemrograman 3 (Web Service)/Tugas3')
Reloaded modules: chap3_1174043_no1, chap3_1174043_no2, chap3_1174043_no3,
chap3_1174043_no4, chap3_1174043_no5, chap3_1174043_no6, chap3_1174043_no7,
chap3_1174043_no8
1173

In [31]: runfile('C:/Users/Irvan/Desktop/Kuliah/Semester 4/Pemrograman 3 (Web Service)/
Tugas3/chap3_1174043_main.py', wdir='C:/Users/Irvan/Desktop/Kuliah/Semester 4/
Pemrograman 3 (Web Service)/Tugas3')
Reloaded modules: chap3_1174043_no1, chap3_1174043_no2, chap3_1174043_no3,
chap3_1174043_no4, chap3_1174043_no5, chap3_1174043_no6, chap3_1174043_no7,
chap3_1174043_no8, chap3_1174043_no9
7
3

In [32]:

```

Gambar 3.10 Jawaban No. 10

```

7
8 def nomorSatu(NPM):
9     NPM = int(NPM)
10    char = ""
11    if NPM%3 == 0:
12        char = "*"
13    if NPM%3 == 1:
14        char = "#"
15    if NPM%3 == 2:
16        char = "+"
17    baris = []
18    baris.append("    ##          ## ##### ##          ##          ##
19                ##### ")
20    baris.append("    #####          ##          ##          ##          ##
21                ##          ## ")
22    baris.append("    ##          ##          ##          ##          ##          ##
23                ##          ")
24    baris.append("    ##          ##          ##          ##          ##          ##
25                #####          ## ")
26    baris.append("    ##          ##          ##          ##          ##
27                ##          ## ")
28    baris.append("    #####          ##          ##          ##          ##
29                ##### ")
30    a = 0

```

```

26     for x in baris:
27         print(baris[a].replace("#", char))
28         a+=1
29
30 def nomorDua(NPM):
31     y = NPM[5:7]
32     g = int(y)
33     for i in range(g):
34         print('Halo ' + NPM, 'Apa Kabar?')
35
36 def nomorTiga(NPM):
37     y = NPM[4:7]
38     x = NPM[4:5]
39     y = NPM[5:6]
40     z = NPM[6:7]
41     g = (int(x) + int(y) + int(z))
42     for i in range(g):
43         print('Halo ' + y, 'Apa Kabar?')
44
45 def nomorEmpat(NPM):
46     y = NPM[4:5]
47     print('Halo ' + y, 'Apa Kabar?')
48
49 def nomorLima(NPM):
50     NPM = list(NPM)
51     for x in NPM:
52         print(x)
53
54 def nomorEnam(NPM):
55     NPM = list(NPM)
56     jmlh = 0
57     for x in NPM:
58         jmlh = jmlh + int(x)
59     print("Hasilnya : " +str(jmlh))
60
61 def nomorTujuh(NPM):
62     NPM = list(NPM)
63     jmlh = 0
64     for x in NPM:
65         jmlh = jmlh * int(x)
66     print("Hasilnya : " +str(jmlh))
67
68 def nomorDelapan(NPM):
69     NPM = list(NPM)
70     for x in NPM:
71         if int(x) != 0:
72             if int(x)%2==0:
73                 print(x, end = "")
74
75 def nomorSembilan(NPM):
76     for x in NPM:
77         if int(x) != 0:
78             if int(x)%2==1:
79                 print(x, end = "")
80
81 def nomorSepuluh(NPM):

```

```

82     for x in NPM:
83         if int(x) != 0:
84             i = 1
85             bil = 0
86             while i <= int(x):
87                 if int(x)%i==0:
88                     bil+=1
89                 i+=1
90             if bil == 2:
91                 print(x)

```

### File main.py

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Mar 11 20:14:27 2019
4
5  @author: Irvan
6  """
7
8  import chap3_1174043_3lib
9
10
11 chap3_1174043_3lib.nomorSatu("1174043")
12 chap3_1174043_3lib.nomorDua("1174043")
13 chap3_1174043_3lib.nomorTiga("1174043")
14 chap3_1174043_3lib.nomorEmpat("1174043")
15 chap3_1174043_3lib.nomorLima("1174043")
16 chap3_1174043_3lib.nomorEnam("1174043")
17 chap3_1174043_3lib.nomorTujuh("1174043")
18 chap3_1174043_3lib.nomorDelapan("1174043")
19 chap3_1174043_3lib.nomorSembilan("1174043")
20 chap3_1174043_3lib.nomorSepuluh("1174043")

```

??

## 12. Jawaban soal no 12

```

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             #####          #####          ")
17             line.append("#####          ##          ##          ##          ##
18             ##          ##          ##          ##          ")

```

```

IPython console
Console 1/A
Halo 1174043 Apa Kabar?
Halo 1174043 Apa Kabar?
Halo 1174043 Apa Kabar?
Halo 1174043 Apa Kabar?
Halo 1174043 Apa Kabar?
Halo 1174043 Apa Kabar?
Halo 1174043 Apa Kabar?
Halo 4 Apa Kabar?
Halo 4 Apa Kabar?
Halo 4 Apa Kabar?
Halo 4 Apa Kabar?
Halo 4 Apa Kabar?
Halo 4 Apa Kabar?
Halo 4 Apa Kabar?
Halo 0 Apa Kabar?
1
1
7
4
0
4
3
Hasilnya : 20
Hasilnya : 0
4411737
3

In [48]:
IPython console History log
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 20 Column: 1 Memory: 48 %

```

Gambar 3.11 Jawaban No. 11

```

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

```

### 3.1.3 Keterampilan Penanganan Error

```

1 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")

```

### 3.1.4 Plagiarisme

**RESULTS**

100% Completed: 100% Checked      0% Plagiarism      100% Unique

**Sentence Wise Result**      **Matched Sources**      **Document View**

|        |                                                                                      |
|--------|--------------------------------------------------------------------------------------|
| UNIQUE | Item Fungsi adalah bagian dari program yang berupa blok kode yang diberikan na...    |
| UNIQUE | Item Inputan fungsi adalah sebuah fungsi yang telah di sediakan pada library pyth... |
| UNIQUE | Item Kembalian fungsi adalah sebuah nilai balikan yang diberikan oleh sebuah fun...  |
| UNIQUE | Item paket adalah sebuah cara yang dilakukan untuk memanggil file script python...   |
| UNIQUE | Item Kelas merupakan sebuah cetakan atau Blueprint yang berguna untuk mencet...      |
| UNIQUE | Item Objek merupakan sebuah objek yang dari proses hasil dari cetakan atau blue...   |

By clicking "Accept" or continuing to use our site, you agree to our Privacy Policy for Website

Accept Privacy Policy

**Gambar 3.12** Hasil Cek Plagiarisme

### 3.1.5 Luthfi Muhammad Nabil/1174035

#### 3.1.5.1 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 kodung 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
          merupakan fungsi
8          coba = self.a+self.b #Coba merupakan atribut
9          print(str(coba))
10
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))

```

### 3.1.6 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  $NPM \bmod 3 = 0$  memakai bintang,  $NPM \bmod 3 = 1$  memakai pagar,  $NPM \bmod 3 = 2$  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 = ""#""

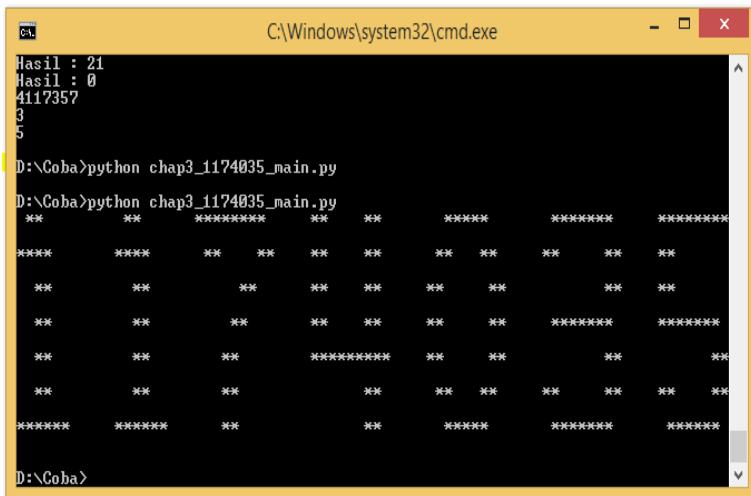
```



```

8  if NPM%3==2:
9      char = "+"
10     line = []
11     line.append(" ##          ##          #####          ##          ##
          #####          #####")
12     line.append("####          ##          ##          ##          ##
          ##          ##          ##          ")
13     line.append(" ##          ##          ##          ##          ##
          ##          ##          ")
14     line.append(" ##          ##          ##          ##          ##
          ##          #####          ")
15     line.append(" ##          ##          ##          ##          #####          ##
          ##          ##          ")
16     line.append(" ##          ##          ##          ##          ##          ##
          ##          ##          ##          ")
17     line.append("#####          #####          ##          ##
          #####          #####          ")
18     a=0
19     for x in line:
20         print(line[a].replace("#", char))
21         a+=1
22     return

```



**Gambar 3.13** Screenshot No 1

2. 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 :

```

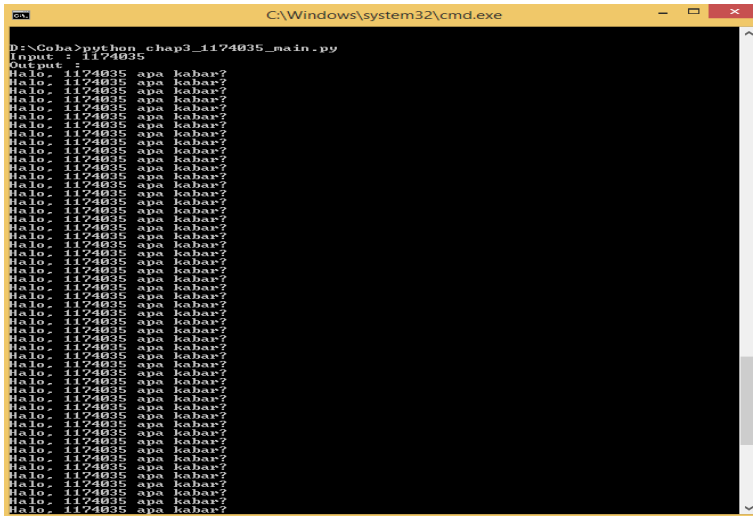
1  def nomordua(NPM):
2      print("Input : "+NPM)

```

```

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

```



**Gambar 3.14** Screenshot No 2

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

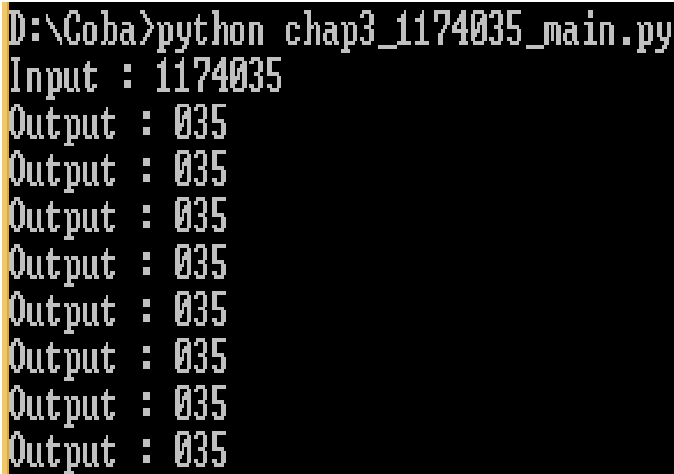
```

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):

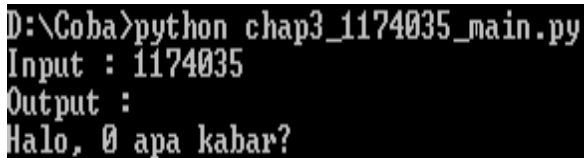
```



```
D:\Coba>python chap3_1174035_main.py
Input : 1174035
Output : 035
Output : 035
Output : 035
Output : 035
Output : 035
Output : 035
Output : 035
Output : 035
```

**Gambar 3.15** Screenshot No 3

```
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 3.16** Screenshot No 4

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

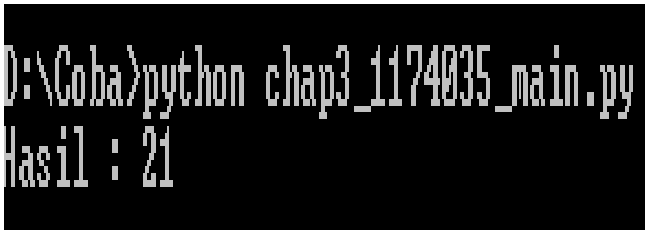
```
1 def nomorlima(NPM):
2     NPM = list(NPM)
3     for x in NPM:
4         print(x)
5     return
```



Gambar 3.17 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
```



Gambar 3.18 Screenshot No 6

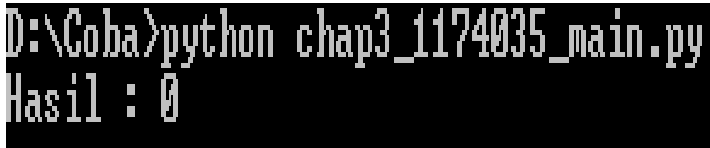
7. Membuat fungsi dengan inputan variabel NPM, didalamnya melakukan perkalian dari seluruh digit NPM tersebut. menggunakan perulangan atau kondisi. Kodingnya :

```
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 3.19** 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 3.20** Screenshot No 8

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

```

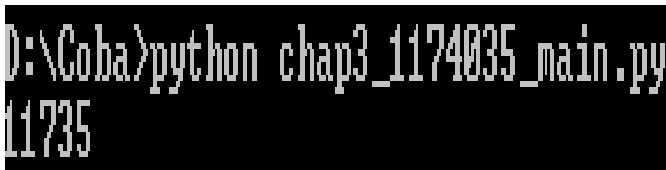
1  def nomorsembilan(NPM):
2      NPM = list(NPM)

```

```

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 3.21** 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

```

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

```



Gambar 3.22 Screenshot No 10

```

13 line.append("####      ####      ##      ##      ##      ##      ##
    ##      ##      ##      ##      ")
14 line.append("  ##      ##      ##      ##      ##      ##      ##
    ##      ##      ##      ")
15 line.append("  ##      ##      ##      ##      ##      ##      ##
    ##      #####      ##### ")
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

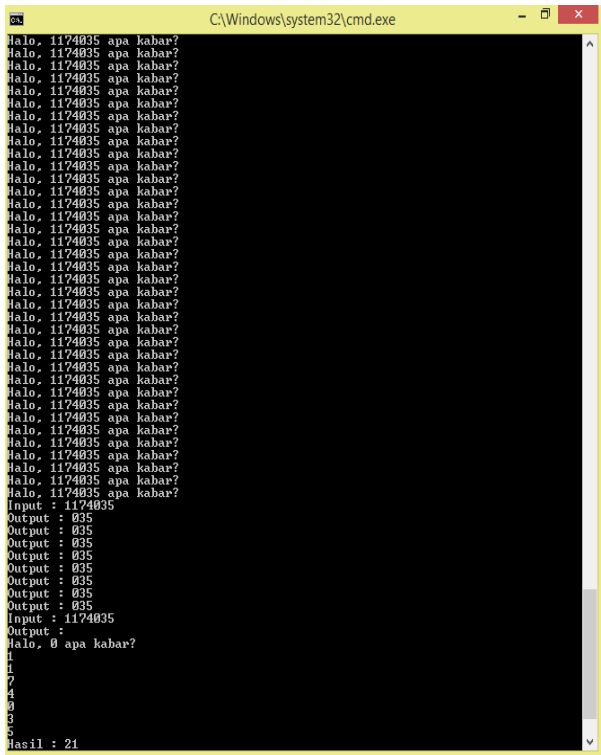
```

---

```

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")

```



**Gambar 3.23** Screenshot No 11

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         #####          #####          #####")
17         line.append("#####          ##          ##          ##
18         ##          ##          ##          ")
19         line.append(" ##          ##          ##          ##          ##
20         ##          ##          ##          ")
21         line.append(" ##          ##          ##          ##          ##
22         ##          #####          ")
23         line.append(" ##          ##          ##          ##
24         ##          ##          ##          ##")
25         line.append("#####          ##          ##
26         #####          #####          ")
27         a=0
28         for x in line:
29             print(line[a].replace("#", char))
30             a+=1
31         return
32
33     def nomordua(self):
34         print("Input : "+self.NPM)
35         x=1
36         print("Output : ")
37         while x <= 87:
38             x+=1
39             print("Halo, "+self.NPM+" apa kabar?")
40
41     def nomortiga(self):
42         print("Input : "+self.NPM)
43         jumlah = len(self.NPM)
44         a = int(self.NPM[jumlah-3])
45         b = int(self.NPM[jumlah-2])
46         c = int(self.NPM[jumlah-1])
47         x=1
48         while x <= (a+b+c):
49             print("Output : "+self.NPM[jumlah-3:])
50             x+=1
51
52     def nomorempat(self):
53         print("Input : "+self.NPM)
54         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

```



```

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

```



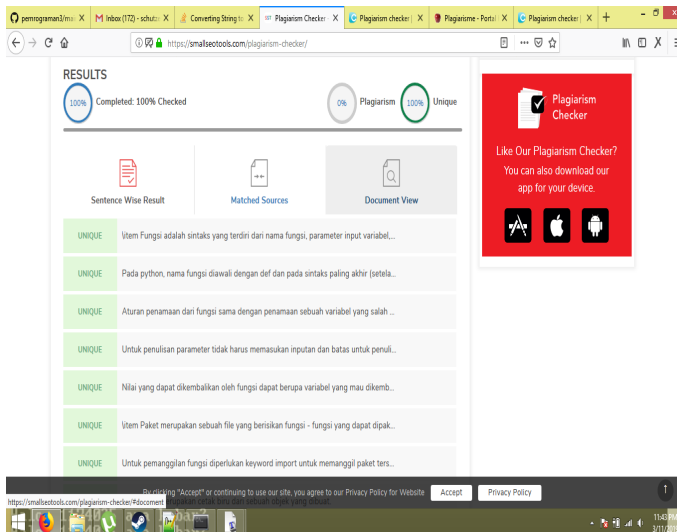
```

1175145
Tidak boleh memakai huruf!!

```

**Gambar 3.25** Screenshot No 13

## 2. Plagiarisme



**RESULTS**

100% Completed: 100% Checked

0% Plagiarism 100% Unique

Sentence Wise Result | Matched Sources | Document View

|        |                                                                                       |
|--------|---------------------------------------------------------------------------------------|
| UNIQUE | Item Fungsi adalah sintaks yang terdiri dari nama fungsi, parameter input variabel... |
| UNIQUE | Pada python, nama fungsi diawali dengan def dan pada sintaks paling akhir (setela...  |
| UNIQUE | Aturan penamaan dari fungsi sama dengan penamaan sebuah variabel yang salah...        |
| UNIQUE | Untuk penulisan parameter tidak harus memasukan inputan dan batas untuk penul...      |
| UNIQUE | Nilai yang dapat dikembalikan oleh fungsi dapat berupa variabel yang mau dikemb...    |
| UNIQUE | Item Paket mengunakan sebuah file yang berisikan fungsi - fungsi yang dapat dipak...  |
| UNIQUE | Untuk pemanggilan fungsi diperlukan keyword import untuk memanggil paket ters...      |

Like Our Plagiarism Checker? You can also download our app for your device.

Accept Privacy Policy

**Gambar 3.26** Plagiarisme

## 3.2 Rangga Putra Ramdhani

### 3.2.0.1 Pemahaman Teori

#### 1. Apa itu fungsi, inputan

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

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

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

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

Statemen return digunakan untuk keluar dari fungsi. Kita juga dapat 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

```

### 3.2.0.2 Ketrampilan Pemrograman

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

```

1 #Chapter 3
2 #No 1
3 def penulisan(npm):
4     npm = list(str(npm))
5
6     angka1 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
7     ##### ", "4": " ###   ## ", "5": " ##### ", "6": " ##### ", "7"
8     : " ##### ", "8": " ##### ", "9": " ##### "}
9     angka2 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
10    ##### ", "4": " ###   ## ", "5": " ##### ", "6": " ##### ", "7"
11    : " ##### ", "8": " ##### ", "9": " ##### "}
12    angka3 = {"0": " ##   ## ", "1": " ### ", "2": "   ## ", "3": "
13    ##### ", "4": " ###   ## ", "5": "   ## ", "6": "   ## ", "7"
14    : "   ## ", "8": "   ## ", "9": "   ## "}
15    angka4 = {"0": " ##   ## ", "1": " ### ", "2": "   ## ", "3": "
16    ##### ", "4": " ###   ## ", "5": "   ## ", "6": "   ## ", "7"
17    : "   ## ", "8": "   ## ", "9": "   ## "}
18    angka5 = {"0": " ##   ## ", "1": " ### ", "2": "   ## ", "3": "
19    ##### ", "4": " ###   ## ", "5": "   ## ", "6": "   ## ", "7"
20    : "   ## ", "8": "   ## ", "9": "   ## "}
21    angka6 = {"0": " ##   ## ", "1": " ### ", "2": "   ## ", "3": "
22    ##### ", "4": " ###   ## ", "5": "   ## ", "6": "   ## ", "7"
23    : "   ## ", "8": "   ## ", "9": "   ## "}
24    angka7 = {"0": " ##   ## ", "1": " ### ", "2": "   ## ", "3": "
25    ##### ", "4": " ###   ## ", "5": "   ## ", "6": "   ## ", "7"
26    : "   ## ", "8": "   ## ", "9": "   ## "}
27    angka8 = {"0": " ##   ## ", "1": " ### ", "2": "   ## ", "3": "
28    ##### ", "4": " ###   ## ", "5": "   ## ", "6": "   ## ", "7"
29    : "   ## ", "8": "   ## ", "9": "   ## "}
30    angka9 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
31    ##### ", "4": "   ## ", "5": " ##### ", "6": " ##### ", "7"
32    : "   ## ", "8": "   ## ", "9": "   ## "}
33    angka10 = {"0": " ##### ", "1": " ### ", "2": " ##### ", "3": "
34    ##### ", "4": "   ## ", "5": " ##### ", "6": " ##### ", "7"
35    : "   ## ", "8": "   ## ", "9": "   ## "}
36
37    hasil1 = []
38    hasil2 = []
39    hasil3 = []
40    hasil4 = []
41    hasil5 = []
42    hasil6 = []
43    hasil7 = []
44    hasil8 = []
45    hasil9 = []
46    hasil10 = []
47
48    for x in npm:
49        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 luaran output dengan perulangan berupa tiga karakter belakang dari NPM sebanyak penjumlahan tiga digit tersebut.

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

7. Buatlah fungsi dengan inputan variabel NPM, didalamnya melakukan 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 dijit 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 dijit 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

```

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

```

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

```

### 3.3 Faisal Najib Abdullah

#### 3.3.1 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 :"))))

```

### 3.3.2 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 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)

```

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 digit 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 :"))))

```

### 3.3.3 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")

```

## 3.4 Yusniar Nur Syarif Sidiq/1164089

### 3.4.1 Teori

1. Fungsi merupakan sebuah bagian dari program yang dapat digunakan ulang dan memiliki inputan variabel serta nilai yang akan di kembalikannya. Contohnya adalah source code berikut ini,

```

1 def memanggil(a,b):
2     x = a + b
3     return x
4
5 a='Hallo'
6 b=' 1164089'
7 c=memanggil(a,b)

```

Dalam dalam source code tersebut akan mengeluarkan output Hallo 1164089 ketika kita running di dalam spyder.

2. Library dalam python disini merupakan kumpulan dari fungsi dan cara pemanggilannya adalah dengan melakukan import file librarynya. Sebagai contoh, buatlah Matematika.py dan 1164089\_2.py, simpan dalam satu folder. Untuk Matematika.py isikan fungsi sebagai berikut

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 16:44:02 2019
4
5 @author: NS
6 """

```

```

7
8 def matematika(a,b,c):
9     r = a + b / c
10    return r

```

Untuk memanggil fungsi tersebut adalah dengan melakukan import Matematika.py pada 1164089\_2.py adalah sebagai berikut,

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 16:47:53 2019
4
5 @author: NS
6 """
7
8 import Matematika
9
10 a=50
11 b=50
12 c=2
13 hasil=Matematika.matematika(a,b,c)

```

3. Class merupakan salah satu cara untuk membuat sebuah kode yang mempunyai objek serta atribut tertentu sehingga akan lebih mudah dalam mengorganisasi berbagai fungsi dan statenya. Objek disini merupakan instansiasi atau perwujudan dari sebuah class. Untuk membuat class yang memiliki objek serta atribut dapat dilihat pada source code berikut ini, dimana kita akan membuat file bernama mtk.py

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 17:10:12 2019
4
5 @author: NS
6 """
7
8 class Mtk:
9     def __init__(self , a , b , c):
10         self.a = a
11         self.b = b
12         self.c = c
13     def Hasil(self):
14         r = self.a + self.b / self.c
15         return r

```

Self tersebut berfungsi untuk menunjukkan variabel lokal dari class tersebut. Untuk memanggil class tersebut kita akan membuat file bernama 1164089\_3.py dan kita akan melakukan import mtk.py pada file tersebut, untuk source codenya dapat dilihat seperti berikut,

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 17:16:05 2019
4

```

```

5 @author: NS
6 """
7
8 import mtk
9
10 a=50
11 b=50
12 c=2
13
14 hasilnya = mtk.Mtk(a,b,c)
15
16 jadi=hasilnya.Hasil()

```

#### 4. Cara memanggilnya yaitu

- Pertama import terlebih dahulu filenya
- Buat variabel yang berfungsi menampung data
- Panggil nama classnya dan methodnya
- Gunakan perintah print untuk menampilkannya

Sebagai contoh perhatikan source code ini,

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 17:45:02 2019
4
5 @author: NS
6 """
7
8 import Matematika
9
10 a=50
11 b=50
12 c=2
13 hasil=Matematika.matematika(a,b,c)
14 print(hasil)

```

5. Dimana kita akan melakukan membuka library Matematika.py dan akan melakukan import dari fungsi di dalamnya yaitu matematika, sehingga akan lebih simple dalam penulisan source codenya adalah sebagai berikut,

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 11 17:51:40 2019
4
5 @author: NS
6 """
7
8 from Matematika import matematika
9 a=50
10 b=50
11 c=2
12
13 hasil=matematika(a,b,c)

```

6.

7.

### 3.4.2 Keterampilan Pemrograman

#### 1. Soal No 1

```

1 def NPM(npm):
2
3     npm = list(str(npm))
4
5     angka1 = {"0": "+++++ ", "1": " ++", "2": "++++++ ", "3": "
6         ++++++ ", "4": "      ++", "5": "+++++++", "6": "++++++ ", "
7         7": "+++++++", "8": "++++++ ", "9": "++++++"}
8     angka2 = {"0": "+++ ++", "1": "++++", "2": "++      ++", "3": "
9         ++      ++", "4": "      +++++", "5": "++      ", "6": "+++      ",
10        "7": "      ++ ", "8": "+++ ++", "9": "++      ++"}
11    angka3 = {"0": "+++ ++", "1": " ++", "2": "      ++ ", "3": "
12        +++++ ", "4": "+++ ++", "5": "++++++ ", "6": "++++++ ", "
13        7": "      ++ ", "8": "+++ ++", "9": "++      ++"}
14    angka4 = {"0": "+++ ++", "1": " ++", "2": "      ++ ", "3": "
15        +++++ ", "4": "+++++++", "5": "      ++", "6": "+++ ++", "
16        7": "      ++ ", "8": "+++++++", "9": "++++++"}
17    angka5 = {"0": "+++ ++", "1": " ++", "2": "      ++ ", "3": "
18        ++      ++", "4": "      ++", "5": "++      ++", "6": "+++ ++",
19        "7": "      ++ ", "8": "+++ ++", "9": "      ++"}
20    angka6 = {"0": "+++++ ", "1": " ++", "2": "+++++++", "3": "
21        ++++++", "4": "      ++", "5": "++++++ ", "6": "++++++ ",
22        "7": "      ++ ", "8": "++++++ ", "9": "++++++"}
23
24    hasil1 = []
25    hasil2 = []
26    hasil3 = []
27    hasil4 = []
28    hasil5 = []
29    hasil6 = []
30
31    for x in npm:
32        hasil1.append(angka1[x])
33        hasil2.append(angka2[x])
34        hasil3.append(angka3[x])
35        hasil4.append(angka4[x])
36        hasil5.append(angka5[x])
37        hasil6.append(angka6[x])
38
39    print(*hasil1, sep=' ')
40    print(*hasil2, sep=' ')
41    print(*hasil3, sep=' ')
42    print(*hasil4, sep=' ')
43    print(*hasil5, sep=' ')
44    print(*hasil6, sep=' ')
45
46    NPM(input("Masukan NPM anda: "))

```

#### 2. Soal No 2

```

1 def pengulangan(npm):
2     ulang = 1
3     while(ulang <= 89):
4         print("Halo, "+str(npm)+" apa kabar?")
5         ulang += 1
6
7 pengulangan(int(input("Masukan NPM : ")))

```

### 3. Soal No 3

```

1 def pengulangan_089(npm):
2     hitung = 0
3     npm = str(npm)
4     bilangan = npm[4:7]
5
6     while(hitung < 17):
7         print("Halo, "+bilangan+" apa kabar?")
8         hitung = hitung +1
9
10 pengulangan_089(int(input("Masukan NPM : ")))

```

### 4. Soal No 4

```

1 def perulangan_0(npm):
2     npm = str(npm)
3     bilangan = npm[-3]
4     print("Halo, "+bilangan+" apa kabar?")
5
6 perulangan_0(int(input("Masukan NPM : ")))

```

### 5. Soal No 5

```

1 def down(npm):
2     for i in npm:
3         print (i)
4 down(input("Masukan NPM : "))

```

### 6. Soal No 6

```

1 def pertambahan(npm):
2     nilai = 0
3     for i in npm:
4         nilai += int(i)
5     print(str(nilai)+" Merupakan Hasil Dari Pertambahan NPM Anda")
6
7 pertambahan(input("Masukan NPM : "))

```

### 7. Soal No 7

```

1 def perkalian(npm):
2     nilai = 0
3     for i in npm:
4         nilai *= int(i)
5     print(str(nilai)+" Merupakan Hasil Nilai Perkalian NPM Anda")
6
7 perkalian(input("Masukan NPM : "))

```

## 8. Soal No 8

```

1 def bilgenap():
2     npm = [1,1,6,4,0,8,9]
3     for i in npm:
4         if (i % 2) == 0:
5             print("Bilangan Genapnya Dari NPM : "+str(i))
6 bilgenap()

```

## 9. Soal No 9

```

1 def bilganjil():
2     npm = [1,1,6,4,0,8,9]
3     for i in npm:
4         if (i%2)==1:
5             print("Bilangan Ganjilnya Dari Npm : "+str(i))
6 bilganjil()

```

## 10. Soal No 10

```

1 def bilprima(npm):
2     npm = str(npm)
3     nilai = npm[2]
4     num = int(nilai)
5     if num > 1:
6         for i in range(2,num):
7             if (num%i)==0:
8                 print("Bukan Bilangan Prima")
9                 break
10            else:
11                print("Merupakan Bilangan Primanya :"+str(num))
12    else:
13        print("Tidak Ada Bilangan Prima")
14    bilprima(int(input("Masukan NPM : ")))

```

## 11. Soal No 11

```

1 #contoh salah satu saja
2 lib = __import__('3lib')
3
4 npm = "1164089"
5
6 lib.NPM(npm)
7 lib.Pengulangan(npm)
8
9 print()

```

## 12. Soal No 12

```

1 #contoh salah satu saja
2 from Kelas3lib import kelas3lib
3
4 npm = "1164089"
5
6 klib = kelas3lib(npm)
7
8 klib.NPM()

```

### 3.4.3 Penanganan Error

1. Error yang saya temui di antaranya adalah Syntax Error, dimana suatu keadaan script python mengalami kesalahan dalam penulisannya dan solusi dari permasalahan ini adalah dengan memperbaiki script penulisan yang salah. Untuk contoh fungsi try except dapat dilihat pada source code berikut ini,

```

1 def name(nama):
2     try:
3         print("Hai, " + str(nama))
4     except:
5         print("Error")
6
7 name(input("Siapa namamu: "))

```

## 3.5 Fathi Rabbani/1164074/3C

### 3.5.1 Teori

1. Fungsi, Inputan dan Return

- Fungsi adalah sebuah blok code yang digunakan untuk melempar parameter kedalam blok code yang berbeda.

```

1 def coba():
2     print("coba")
3     coba()

```

- Inputan fungsi adalah sebuah fungsi yang memiliki parameter berupa inputan atau data yang bias diinputkan.

```

1 def coba2(nm):
2     print("Namaku : " + str(nm))
3     coba2(input("NAMA: "))

```

- Pengembalian fungsi atau sering juga disebut sebagai return merupakan sebuah pengembalian nilai dari pengekseskuan data pada parameter yang terdapat difungsi.

```

1 def coba3(a,b):
2     x = a+b
3     return x
4
5 a = 7
6 b = 5
7 y = coba3(a,b)
8 print(y)

```

2. Paket atau Library Fungsi

paket merupakan sebuah penggunaan library dengan maksud mempermudah dalam eksekusi dan pemanggilan fungsi



```

1 def tambah2(a,b):
2     r = a+b
3     return r

```

berikut ini adalah fungsi yang digunakan untuk memanggil paket atau library

```

1 import praktek
2 a = 1
3 b = 3
4 c = praktek.tambah2(a,b)

```

### 3. Kelas, Objek, Atribut dan Method

- kelas merupakan sebuah blueprint dari objek
- objek merupakan sebuah data hasil eksekusi dari kelas
- atribut merupakan nilai data yang terdapat didalam objek
- method merupakan operasi atau eksekusi yang dilakukan dengan data dari objek

```

1 class fthr:
2     def __init__(self, a, b):
3         self.a = a
4         self.b = b
5     def tambah(self):
6         r = self.a + self.b
7         return r

```

### 4. Library Kelas

- Penggunaan kelas dan datanya

```

1 class fthr:
2     def __init__(self, a, b):
3         self.a = a
4         self.b = b
5     def tambah(self):
6         r = self.a + self.b
7         return r

```

- Pemanggilan library kelas

```

1 import fthr
2
3 a = 1
4 b = 3
5 c = lib.fthr(a,b)
6 result=c.tambah2()

```

### 5. Pemanggilan Library Kalkulator

- data Kalkulator

```

1 def penambahan(a,b):
2     r = a+b
3     return r

```

#### ▪ data pemanggilan

```

1 from kalkulator import penambahan
2
3 a = 1
4 b = 3
5
6 result = penambahan(a,b)

```

## 6. Penggunaan Paket Fungsi

#### ▪ data Fungsi Kalkulator

```

1 def penambahan(a,b):
2     r = a+b
3     return r

```

#### ▪ data Pemanggil Fungsi

```

1 import kalkulator
2 a = 1
3 b = 3
4 result = kalkulator.penambahan(a,b)

```

## 7. Penggunaan Paket Kelas

#### ▪ data Kelas fthr dari file praktek

```

1 class fthr:
2     def __init__(self, a, b):
3         self.a = a
4         self.b = b
5     def tambah(self):
6         r = self.a + self.b
7         return r

```

#### ▪ data pemanggil

```

1 a = 1
2 b = 3
3 c = lib.fthr(a,b)
4 result=c.tambah2()
5
6 #library kalkulator

```

## 3.5.2 Praktek Pemrograman

```

1 def coba4(n):
2     n = list(str(n))
3
4     angka1 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ## ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
5     angka2 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "###   ## ", "5": "##### ", "6": "##  ## ", "7": "##### ", "8": "##  ## ", "9": "##  ## "}
6     angka3 = {"0": "##   ## ", "1": "### ", "2": "##   ## ", "3": "### ", "4": "###   ## ", "5": "###   ", "6": "##   ", "7": "###   ", "8": "##  ## ", "9": "##  ## "}
7     angka4 = {"0": "##   ## ", "1": "###   ## ", "2": "##   ## ", "3": "### ", "4": "###   ## ", "5": "###   ", "6": "##   ", "7": "###   ", "8": "##  ## ", "9": "##  ## "}
8     angka5 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
9     angka6 = {"0": "##   ## ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
10    angka7 = {"0": "##   ## ", "1": "### ", "2": "##   ## ", "3": "### ", "4": "###   ## ", "5": "###   ## ", "6": "##  ## ", "7": "###   ", "8": "##  ## ", "9": "##  ## "}
11    angka8 = {"0": "##   ## ", "1": "### ", "2": "##   ## ", "3": "### ", "4": "###   ## ", "5": "###   ## ", "6": "##  ## ", "7": "###   ", "8": "##  ## ", "9": "##  ## "}
12    angka9 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##  ## ", "7": "##### ", "8": "##  ## ", "9": "##  ## "}
13    angka10 = {"0": "##### ", "1": "### ", "2": "##### ", "3": "##### ", "4": "##### ", "5": "##### ", "6": "##### ", "7": "##### ", "8": "##### ", "9": "##### "}
14
15    hasil1 = []
16    hasil2 = []
17    hasil3 = []
18    hasil4 = []
19    hasil5 = []
20    hasil6 = []
21    hasil7 = []
22    hasil8 = []
23    hasil9 = []
24    hasil10 = []
25
26    for x in n:
27        hasil1.append(angka1[x])
28        hasil2.append(angka2[x])
29        hasil3.append(angka3[x])
30        hasil4.append(angka4[x])
31        hasil5.append(angka5[x])
32        hasil6.append(angka6[x])
33        hasil7.append(angka7[x])
34        hasil8.append(angka8[x])
35        hasil9.append(angka9[x])
36        hasil10.append(angka10[x])

```

```

37
38     print(*hasil1 , sep=' ')
39     print(*hasil2 , sep=' ')
40     print(*hasil3 , sep=' ')
41     print(*hasil4 , sep=' ')
42     print(*hasil5 , sep=' ')
43     print(*hasil6 , sep=' ')
44     print(*hasil7 , sep=' ')
45     print(*hasil8 , sep=' ')
46     print(*hasil9 , sep=' ')
47     print(*hasil10 , sep=' ')
48 coba4(input("NPM:"))

```

```

1 def loop(n):
2     calc = 0
3     while (calc < 74):
4         print("Hello , "+str(n)+" Apa Kabar?")
5         calc = calc +1
6     loop(int(input("NPM:")))

```

```

2. def loop3(x):
3     y = 0
4     a = int(x[4])
5     b = int(x[5])
6     c = int(x[6])
7     d = x[4:7]
8     e = a+b+c
9     while y < e:
10         y += 1
11         print("Hello , " + d + " Apa Kabar?")
12     return x
13 loop3(input("NPM:"))

```

```

1 def value4(x):
2     n = x[4]
3     print("Hello , " + n + " Apa Kabar?")
4     return x
5     value4(input("NPM:"))

```

```

3. def loopdown(x):
2     for i in x:
3         print(i)
4     return i
5     loopdown(input("NPM:"))

```

```

1 def tambah(n):
2     hasil = 0
3     for i in n:
4         hasil += int(i)
5     print(str(hasil))
6
7     tambah(input("NPM:"))

```

```

0. def kali(n):
    2     hasil = 0
    3     for i in n:
    4         hasil *= int(i)
    5     print(str(hasil))
    6
    7 kali(input("NPM: "))

```

```

1 def nilaigenap(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     nilaigenap(input("NPM: "))

```

```

9. def nilaiganjil(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     nilaiganjil(input("NPM: "))

```

```

1 def prim(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     prim(input("NPM: "))

```

## 10. ■ data 3lib.py

```

1
2 #1
3 def coba4(n):
4     n = list(str(n))
5
6     angka1 = {"0": "##### ", "1": "### ", "2": "##### ", "3":
7 : "####", "4": "###   ", "5": "##### ", "6": "#####
8 : "7": "##### ", "8": "##### ", "9": "##### "}
9     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 n:
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
52  #2
53  def loop(n):

```

```
54     calc = 0
55     while (calc < 74):
56         print("Hello , "+str(n)+" Apa Kabar?")
57         calc = calc +1
58
59 loop(int(input("NPM:")))
60
61 #3
62 def loop3(x):
63     y = 0
64     a = int(x[4])
65     b = int(x[5])
66     c = int(x[6])
67     d = x[4:7]
68     e = a+b+c
69
70     while y < e:
71         y += 1
72         print("Hello , " + d + " Apa Kabar?")
73     return x
74 loop3(input("NPM:"))
75
76 #4
77 def value4(x):
78     n = x[4]
79     print("Hello , " + n + " Apa Kabar?")
80     return x
81 value4(input("NPM:"))
82
83 #5
84 def loopdown(x):
85     for i in x:
86         print(i)
87     return i
88 loopdown(input("NPM:"))
89
90 #6
91 def tambah(n):
92     hasil = 0
93     for i in n:
94         hasil += int(i)
95     print(str(hasil))
96
97 tambah(input("NPM:"))
98
99 #7
100 def kali(n):
101     hasil = 0
102     for i in n:
103         hasil *= int(i)
104     print(str(hasil))
105
106 kali(input("NPM:"))
107
108 #8
109 def nilaigenap(x):
```

```

110     npm = [a,b,c,d,e,f,g]
111
112     for n in npm:
113         if(n % 2 == 0):
114             if(n != 0):
115                 print(n, end = "")
116 nilaigenap(input("NPM:"))
117
118 #9
119 def nilaiganjil(x):
120     npm = [a,b,c,d,e,f,g]
121
122     for n in npm:
123         if(n % 2 != 0):
124             print(n, end = "")
125 nilaiganjil(input("NPM:"))
126
127 #10
128 def prim(x):
129     npm = [a,b,c,d,e,f,g]
130
131     for n in npm:
132         if(n % 2) == 0:
133             print(end="")
134         else:
135             print(n, end = "")
136 prim(input("NPM:"))

```

#### ▪ data main.py

```

1 import praktek
2
3 a = 1
4 b = 3
5 result = praktek.coba3(a,b)

```

### 3.5.3 Handling Error

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

```

1 def error(a,b):
2     try :
3         c = a+b
4         print(c)
5     except TypeError:
6         print("We Are Different")
7 a = 10
8 b = 2

```



### 3.6 Dika Sukma Pradana 1174050

#### 3.6.1 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 file 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 file library ada di dalam folder.

```

1 #Chapter 3

```

### 3.6.2 Ketrampilan Pemrograman

1. Buatlah fungsi dengan inputan variabel NPM, dan melakukan print luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk  $\text{NPM} \bmod 3 = 0$ , tanda pagar untuk  $\text{NPM} \bmod 3 = 1$ , tanda plus untuk  $\text{NPM} \bmod 3 = 2$ .

```

1 #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 < 50):
4         print("Halo, "+str(npm)+" apa kabar?")
5         hitung = hitung +1
6
7 perulangan(int(input("Masukan NPM : ")))

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

7. Buatlah fungsi dengan inputan variabel NPM, didalamnya melakukan 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 = [1,1,7,4,0,5,0]
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 = [1,1,7,4,0,5,0]
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 file fungsi\_d1ka\_1174050.py dan berikan contoh cara pemanggilannya pada file 1174050\_maiinn.py.

```

1 from fungsi_d1ka_1174050 import penulisan

```

12. Buatlah satu library class dengan nama file kelas3lib\_1174050.py yang merupakan mod- ifikasi dari fungsi-fungsi nomor diatas dan berikan contoh cara pemanggilannya pada file 1174050\_maiinn.py.

```

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

```

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

```

## 3.7 Mhd Zulfikar Akram Nasution/1164081

### 3.7.1 Teori

1. Fungsi merupakan sebuah bagian dari program yang dapat digunakan ulang dan memiliki inputan variable serta nilai yang akan di kembalikan. Contohnya adalah source code berikut ini,

```

1 def memanggil(a,b):
2     x = a + b
3     return x
4
5 a='Namaste'
6 b=' 1164081'
7 c=memanggil(a,b)

```

Dalam dalam source code tersebut akan mengeluarkan output Namaste 1164081 ketika kita running di dalam spyder.

2. Library dalam pythoni merupakan kumpulan dari fungsi, cara pemanggilannya adalah dengan melakukan import file librarynya. Sebagai contoh, buatlah Matematika.py dan 1164081\_2.py, simpan dalam satu folder. Untuk Matematika.py isikan fungsi sebagai berikut

```

1 def matematika(a,b,c):
2     r = a + b / c
3     return r

```

Untuk memanggil fungsi tersebut adalah dengan melakukan import Matematika.py pada 1164081\_2.py adalah sebagai berikut,

```

1 import Matematika
2
3 a=20
4 b=30
5 c=2
6 hasil=Matematika.matematika(a,b,c)

```

3. Class merupakan salah satu cara untuk membuat sebuah kode yang mempunyai objek dan atribut tertentu sehingga akan lebih mudah dalam mengorganisasikan berbagai fungsi dan statenya. Objek disini merupakan instansiasi atau perwujudan dari sebuah class. Untuk membuat class yang memiliki objek dan atribut dapat dilihat pada source code berikut ini, dimana kita akan membuat file bernama contoh.py

```

1 class Contoh:
2     def __init__(self , a , b , c):
3         self.a = a
4         self.b = b
5         self.c = c
6     def Hasil(self):
7         r = self.a + self.b / self.c
8         return r

```

Self tersebut berfungsi untuk menunjukkan variabel lokal dari class tersebut. Untuk memanggil class tersebut kita akan membuat file bernama 1164081\_3.py dan kita akan melakukan import contoh.py pada file tersebut, untuk source codenya dapat dilihat seperti berikut,

```

1 import contoh
2
3 a=20
4 b=30
5 c=2
6
7 hasilnya = contoh.Contoh(a,b,c)
8
9 jadi=hasilnya.Hasil()

```

4. Cara memanggilnya yaitu

- Pertama import terlebih dahulu filenya
- Kemudian buat variabel yang berfungsi untuk menampung data
- Lalu panggil nama class dan methodnya
- Setelah memanggil class dan methodnya, gunakan perintah print untuk menampilkannya

Sebagai contoh perhatikan source code ini,

```

1 import Matematika
2
3 a=20
4 b=30
5 c=2
6 hasil=Matematika.matematika(a,b,c)
7 print(hasil)

```

5. Penggunaan paket from namafile import, itu berfungsi untuk memanggil file dan fungsinya. Contoh penulisan source codenya adalah sebagai berikut,



```

1 from Matematika import matematika
2 a=20
3 b=30
4 c=2
5
6 hasil=matematika(a,b,c)

```

6.

7.

### 3.7.2 Keterampilan Pemrograman

#### 1. Soal No 1

```

1 def NPM(npm):
2
3     npm = list(str(npm))
4
5     angka1 = {"0": "***** ", "1": "   **", "2": "***** ", "3": "*****
6             ", "4": "   ****", "5": "***** ", "6": "***** ", "
7             7": "***** ", "8": "***** ", "9": "***** "}
8     angka2 = {"0": "***   **", "1": "*****", "2": "*****", "3": "*****
9             ", "4": "*****", "5": "*****", "6": "*****", "
10            7": "*****", "8": "*****", "9": "*****"}
11
12     angka3 = {"0": "***   **", "1": "*****", "2": "*****", "3": "*****
13             ", "4": "*****", "5": "*****", "6": "*****", "
14            7": "*****", "8": "*****", "9": "*****"}
15
16     angka4 = {"0": "*****", "1": "*****", "2": "*****", "3": "*****
17             ", "4": "*****", "5": "*****", "6": "*****", "
18            7": "*****", "8": "*****", "9": "*****"}
19
20     angka5 = {"0": "*****", "1": "*****", "2": "*****", "3": "*****
21             ", "4": "*****", "5": "*****", "6": "*****", "
22            7": "*****", "8": "*****", "9": "*****"}
23
24     angka6 = {"0": "*****", "1": "*****", "2": "*****", "3": "*****
25             ", "4": "*****", "5": "*****", "6": "*****", "
26            7": "*****", "8": "*****", "9": "*****"}
27
28     hasil1 = []
29     hasil2 = []
30     hasil3 = []
31     hasil4 = []
32     hasil5 = []
33     hasil6 = []
34
35     for x in npm:
36         hasil1.append(angka1[x])
37         hasil2.append(angka2[x])
38         hasil3.append(angka3[x])
39         hasil4.append(angka4[x])
40         hasil5.append(angka5[x])
41         hasil6.append(angka6[x])
42
43     print(*hasil1, sep=' ')
44     print(*hasil2, sep=' ')

```

```

29     print(*hasil3 , sep=' ')
30     print(*hasil4 , sep=' ')
31     print(*hasil5 , sep=' ')
32     print(*hasil6 , sep=' ')
33
34 NPM(input("Masukan NPM anda: "))

```

## 2. Soal No 2

```

1 def pengulangan(npm):
2     ulang = 1
3     while(ulang <= 81):
4         print("Halo, "+str(npm)+" apa kabar?")
5         ulang += 1
6
7 pengulangan(int(input("Masukan NPM : ")))

```

## 3. Soal No 3

```

1 def pengulangan_081(npm):
2     hitung = 0
3     npm = str(npm)
4     bilangan = npm[4:7]
5
6     while(hitung < 17):
7         print("Halo, "+bilangan+" apa kabar?")
8         hitung = hitung +1
9
10 pengulangan_081(int(input("Masukan NPM : ")))

```

## 4. Soal No 4

```

1 def perulangan_0(npm):
2     npm = str(npm)
3     bilangan = npm[-3]
4     print("Halo, "+bilangan+" apa kabar?")
5
6 perulangan_0(int(input("Masukan NPM : ")))

```

## 5. Soal No 5

```

1 def down(npm):
2     for i in npm:
3         print(i)
4 down(input("Masukan NPM : "))

```

## 6. Soal No 6

```

1 def pertambahan(npm):
2     nilai = 0
3     for i in npm:
4         nilai += int(i)
5     print(str(nilai)+" Merupakan Hasil Dari Pertambahan NPM Anda")
6
7 pertambahan(input("Masukan NPM : "))

```

## 7. Soal No 7

```

1 def perkalian(npm):
2     nilai = 0
3     for i in npm:
4         nilai *= int(i)
5     print(str(nilai)+" Merupakan Hasil Nilai Perkalian NPM Anda")
6
7 perkalian(input("Masukan NPM : "))

```

## 8. Soal No 8

```

1 def bilgenap():
2     npm = [1,1,6,4,0,8,1]
3     for i in npm:
4         if (i % 2) == 0:
5             print("Bilangan Genapnya Dari NPM : "+str(i))
6 bilgenap()

```

## 9. Soal No 9

```

1 def bilganjil():
2     npm = [1,1,6,4,0,8,1]
3     for i in npm:
4         if (i%2)==1:
5             print("Bilangan Ganjilnya Dari Npm : "+str(i))
6 bilganjil()

```

## 10. Soal No 10

```

1 def bilprima(npm):
2     npm = str(npm)
3     nilai = npm[2]
4     num = int(nilai)
5     if num > 1:
6         for i in range(2,num):
7             if (num%i)==0:
8                 print("Bukan Bilangan Prima")
9                 break
10            else:
11                print("Merupakan Bilangan Primanya :"+str(num))
12    else:
13        print("Tidak Ada Bilangan Prima")
14 bilprima(int(input("Masukan NPM : ")))

```

## 11. Soal No 11

```

1 #contoh salah satu saja
2 lib = __import__('1164081_3lib')
3
4 npm = "1164081"
5
6 lib.NPM(npm)
7 lib.bilprima(npm)
8
9 print()

```

## 12. Soal No 12

```
1 #contoh salah satu saja
2 from kelas3lib import kelas3lib
3
4 npm = "1164081"
5
6 klib = kelas3lib(npm)
7
8 klib.NPM()
```

### 3.7.3 Penanganan Errorr

1. Errorr yang saya temui di antaranya adalah Systax Errorr, dimana suatu keadaan script python mengalami kesalahan dalam penulisannya dan solusi dari permasalahan ini adalah dengan memperbaiki script penulisan yang salah. Untuk contoh fungsi `try except` dapat dilihat pada source code berikut ini,

```
1 def name(nama):
2     try :
3         print("Hai, "+str(nama))
4     except:
5         print("Errorr")
6
7 name(input("Siapa namamu: "))
```



## BAB 4

---

# JUDUL BAGIAN KEEMPAT

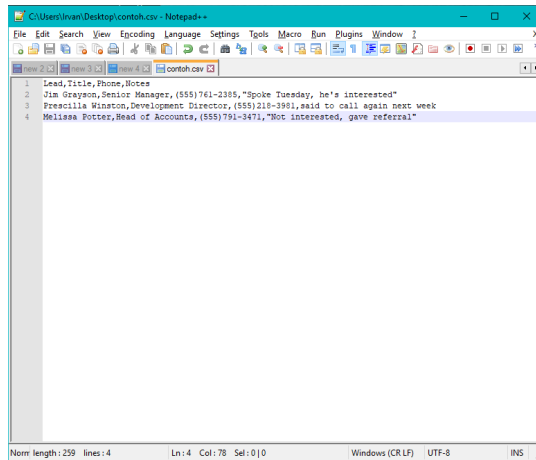
---

|||||| HEAD

### 4.1 IrvanRizkiansyah/1174043

#### 4.1.1 Pemahaman Teori

- Fungsi : File csv berfungsi untuk pencarian data akan menjadi lebih mudah dan cepat, dan juga mempermudah penginputan data ke dalam database secara sederhana.
  - Sejarah : File csv muncul pertama kali sekitar 10 tahun sebelum Personal Computer (PC) pertama didunia yaitu sejak sekitar tahun 1972, akan tetapi sebutan file csv digunakan pertama kali pada tahun 1983.
  - Contoh :  
??
2. Ada banyak aplikasi yang dapat membuat file berformat CSV, diantaranya adalah :



**Gambar 4.1** Contoh CSV

- Notepad
  - Notepad++
  - Microsoft Excel
  - Corel Quatro Pro
  - Apache Open Office, dan masih banyak yang lainnya.
3. Cara menulis file csv menggunakan Excel :
    - (a) Buka aplikasi Microsoft Excel kemudian buat dokumen baru
    - (b) Tulis judul kolom untuk setiap informasi yang ingin di rekam atau catat, kemudian tulis informasi - informasi dalam kolom dengan sesuai.
    - (c) Jika sudah selesai maka save dengan cara pilih menubar File lalu pilih Save As
    - (d) Lalu isikan nama file tersebut dan rubah dengan memilih format file yang tersedia tersebut menjadi .csv
    - (e) File csv sudah berhasil terbuat menggunakan Microsoft Excel
  4. Cara membaca file csv menggunakan Excel :
    - (a) Buka aplikasi Microsoft Excel kemudian pilih menu Open
    - (b) Cari tempat file csv yang ingin dibuka, kemudian pilih Open
    - (c) File csv sudah berhasil dibaca menggunakan Microsoft Excel
  5. Pada file csv, tanda baca koma diartikan sebagai pembatas suatu kolom. List-directed input output didefinisikan dalam FORTRAN 77. List-directed input menggunakan tanda baca koma atau spasi sebagi pembatas, sehingga karakter

yang tidak dikutip tidak dapat mengandung tanda baca koma ataupun spasi. Hal tersebut yang diadopsi oleh file csv. format csv didukung dengan library untuk banyak bahasa pemrograman, kebanyakan yang menspesifikasikan pembatas field, pemisah desimal, pengkodean karakter, dan yang lainnya.

6. Pada tahun 2008, pengembangan pandas dimulai oleh AQR Capital Management. Pada akhir tahun 2009 pandas menjadi Open Sourced, dimana disupport oleh banyak komunitas atau individu di dunia untuk mengembangkan pandas. Sejak tahun 2015, pandas menjadi NumFOCUS proyek sponsor, ini juga membantu suksesnya pengembangan dari pandas itu sendiri. pandas merupakan struktur data dan data analysis tools untuk bahasa pemrograman Python, dan merupakan BSD-licensed library yang menjadikannya memiliki performa yang tinggi.
7.
  - Tanda baca koma : Menjadi pemisah antar kolom
  - Tanda baca kutip dua : Menjadi cara untuk memasukan sebuah kalimat atau untuk memasukan karakter spasi sebagai data pada kolom informasi
  - Inputan pada baris pertama akan menjadi Header, dimana akan menjadi nama sebuah kolom, dan masih banyak yang lainnya
8. Pada pandas sedikit berbeda, dimana inputan data berbentuk seperti peng-inputan pada variabel pada umumnya, hanya saja menggunakan tanda kutip satu untuk menandakan sebuah informasi pada kolom kemudian tanda kurung kotak yang didalamnya berisi informasi data dari kolom tersebut. dan lain sebagainya.

## 4.1.2 Luthfi Muhammad Nabil/1174035

### 4.1.2.1 Fungsi, Sejarah, dan Contoh file CSV

#### ▪ Fungsi

File CSV (Comma Separated Values) adalah tipe file khusus yang menyimpan informasi dengan metode dipisahkan dengan koma. File CSV berfungsi untuk menjadi perantara untuk beberapa aplikasi yang memiliki basis data saat mengirim data. CSV dapat dibuka di berbagai text editor yang ada. Dengan bentuk filenya yang dinamis memungkinkan file CSV dapat dimanipulasi dan dapat menyimpan informasi dengan skala besar.

#### ▪ Sejarah

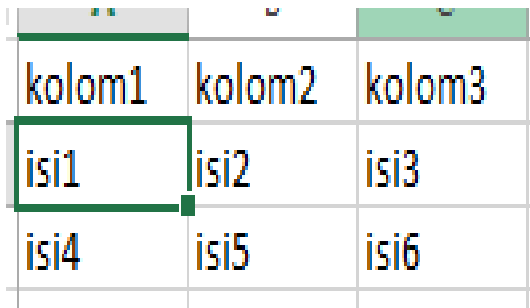
CSV sudah digunakan sejak tahun 1972 yang dapat dikompilasi pada bahasa pemrograman IBM Fortran. Saat itu, data yang dipisahkan oleh koma jika isinya memiliki spasi maka harus diberi tanda petik di awal dan akhir isi dari data tersebut. Nama CSV baru mulai digunakan pada tahun 1983. Pada panduan dari Osborne Executive Computer mendokumentasikan kutipan yang membolehkan isi karakter memiliki koma. Pada tahun 2005 dengan RFC4180, CSV didefinisikan sebagai MIME Content Type. lalu pada tahun 2013, defisiensi dari RFC4180 dipecahkan oleh rekomendasi dari W3C. Pada tahun 2014, IETF



mempublikasi RFC7111 yang mendeskripsikan pecahan Uniform Resource Identifier(URI) ke dokumen CSV. RFC7111 menjelaskan bagaimana baris, kolom dapat dipilih dalam dokumen CSV menggunakan indeks posisi. Pada Tahun 2015, W3C mempublikasikan draft rekomendasi untuk CSV-metadata standards yang dimulai dengan rekomendasi pada bulan Desember dengan tahun yang sama.

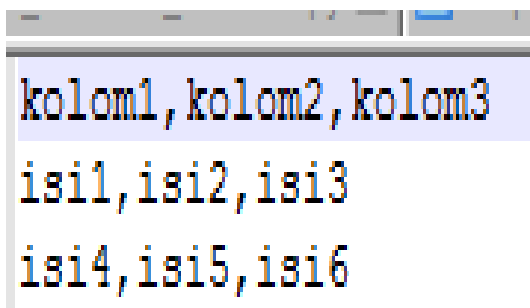
- Contoh File CSV

- CSV pada Excel ??



| kolom1 | kolom2 | kolom3 |
|--------|--------|--------|
| isi1   | isi2   | isi3   |
| isi4   | isi5   | isi6   |

**Gambar 4.2** Contoh CSV Pada Excel



```
kolom1,kolom2,kolom3
isi1,isi2,isi3
isi4,isi5,isi6
```

**Gambar 4.3** Contoh CSV Pada Text

- CSV pada Text Editor ??

**4.1.2.2 Aplikasi Yang dapat membuat file CSV** Berikut file yang dapat membuat file CSV

- Spreadsheet

Spreadsheet merupakan aplikasi yang dapat membuat CSV hanya dengan memasukkan data sesuai baris dan kolom yang diinginkan. Contoh spreadsheet seperti Google Spreadsheet, Microsoft Excel, dan aplikasi lainnya.

- **Bahasa** Pemrograman  
Bahasa pemrograman merupakan media yang dapat untuk membuat aplikasi yang dapat membuat file CSV khusus untuk bahasa pemrograman yang support dengan pembuatan file CSV. Seperti Python, C Sharp, dan lain sebagainya.
- **Text** Editor  
Text editor juga dapat membuat file CSV, untuk membuat dengan Text Editor cukup dengan membuat file sesuai format CSV dan save file tersebut dengan ekstensi .CSV.

#### 4.1.2.3 **Menulis dan Membaca file CSV** Berikut cara menulis dan membaca file CSV :

- **Menulis :**
  1. Buka file CSV dengan spreadsheet
  2. Klik Cell yang mau diisi
  3. Masukan data yang mau diisi pada cell tersebut
  4. Lalu save file dengan format .CSV
- **Membaca :**
  1. Buka file CSV dengan spreadsheet

**4.1.2.4 *Sejarah Library CSV Python*** Library CSV pada python merupakan library yang paling umum untuk import export data pada spreadsheet dan basis data dengan format sesuai dengan standarisasi RFC4180. Seiring dengan lahirnya bahasa pemrograman python, library mulai dibuat dan dikembangkan sampai akhirnya pada tahun 2003, pembuatnya Kevin Altis dan lainnya telah merilis versi final untuk library Python CSV.

**4.1.2.5 *Sejarah Library Pandas Python*** Pandas (Python Data Analysis Library) adalah library open source yang digunakan untuk melakukan data manajemen dan data analysis. Pandas diciptakan pada tahun 2008 oleh Wes McKinney dan diperbaharui oleh Sien Chang pada tahun 2010. Inspirasi dari pembuatan pandas muncul pada komunitas yang membutuhkan library khusus untuk analisis data.

#### 4.1.2.6 **Fungsi - fungsi yang terdapat di library CSV**

- `csv.reader(csvfile, dialect='excel', **fmtparams)`

Untuk mengembalikan object reader yang akan mengambil setiap line pada csv yang diambil. Data setiap baris diambil saat next() dipanggil. Berikut contohnya :

```

1 #CSV Reader
2 import csv
3 with open('cobain.csv', newline='') as csvfile:
4     spamreader = csv.reader(csvfile, delimiter=',', quotechar='\"')
5     for row in spamreader:
6         print(', '.join(row))

```

- `csv.writer(csvfile, dialect='excel', **fmtparams)`

Mengembalikan file pembuat object untuk dapat mengkonversi data pada python ke file CSV yang akan dibuat. Berikut contoh penggunaan `csv.writer` :

```
1 #CSV Writer
2 import csv
3 with open('cobain.csv', 'w', newline='') as csvfile:
4     spamwriter = csv.writer(csvfile, delimiter=' ',
5                             quotechar=' ', quoting=csv.
6                             QUOTE_MINIMAL)
7     spamwriter.writerow(['Eh coba'] * 5 + ['Coba'])
8     spamwriter.writerow(['Isi', 'Isi2', 'Isi3'])
```

- `csv.register_dialect(name[, dialect[, **fmtparams]])`

Mengasosiasikan dialek dengan nama, nama yang dimasukkan harus berupa karakter.

- `csv.unregister_dialect(name)`

Menghapus asosiasi dialek dengan nama pada registry dialek.

- `csv.get_dialect(name)`

Mengambil dialek yang telah diasosiasikan dengan nama.

- `csv.list_dialects()`

Mengembalikan dialek yang telah diregistrasi.

- `csv.field_size_limit([new_limit])`

Mengembalikan maksimal kolom data yang diperbolehkan oleh pembaca.

#### 4.1.2.7 Fungsi - fungsi yang terdapat di library Pandas

- `pandas.read_csv(filepath_or_buffer[, sep, ...])`

Untuk membaca file CSV dan menyimpannya ke DataFrame

- `pandas.read_excel(io[, sheet_name, header, names, ...])`

Membaca file excel dan menyimpannya ke DataFrame

- `to_csv([path, index, sep, na_rep, ...])`

Untuk membuat file CSV dari data yang ada

=====

## 4.2 Faisal Najib Abdullah

### 4.2.1 Pemahaman Teori

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

File CSV Nilai Berbatas Koma adalah tipe file khusus yang dapat Anda buat atau edit di Excel. File CSV menyimpan informasi yang dipisahkan oleh koma, bukan menyimpan informasi dalam kolom. Saat teks dan angka disimpan dalam file CSV, mudah untuk memindahkannya dari satu program ke program lain.

File CSV dibuat oleh program yang menangani sejumlah data yang besar. CSV merupakan cara yang nyaman untuk mengekspor data dari spreadsheet dan basis data serta mengimpor atau menggunakannya dalam program lain. Misalnya, Anda dapat mengekspor hasil program penambangan data ke file CSV dan kemudian mengimpornya ke dalam spreadsheet untuk menganalisis data, menghasilkan grafik untuk presentasi, atau menyiapkan laporan untuk publikasi.

Contohnya, Anda dapat mengekspor kontak dari Google ke dalam file CSV, kemudian mengimpornya ke Outlook.

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

- Microsoft Excel 2013
- Microsoft Works
- CCorel Quattro Pro
- Apache OpenOffice
- LibreOffice
- Microsoft Notepad
- Intuit Quicken 2015
- GenScriber

Pada Mac OS

- Microsoft Excel 2011
- Planamesa NeoOffice
- Apache OpenOffice
- LibreOffice
- GenScriber

Pada Linux

- Apache OpenOffice
- LibreOffice
- GenScriber

3. Jelaskan bagaimana cara menulis dan membaca file csv di excel atau spreadsheet?
  - Cara menulis file csv harus berupa baris dan kolom atau bisa juga di sebut berupa tabel.
  - Untuk membacanya file csv dipisahkannya menggunakan koma atau titik koma.
4. Jelaskan sejarah library csv? Library csv menyediakan fungsionalitas untuk membaca dan menulis ke file CSV. Dirancang untuk bekerja di luar kotak dengan file CSV yang dihasilkan Excel, memudahkan untuk bekerja dengan berbagai format CSV. Library csv berisi objek dan kode lain untuk membaca, menulis, dan memproses data ke file CSV.
5. Jelaskan sejarah library pandas? panda adalah pustaka Python open-source yang menyediakan alat analisis data kinerja tinggi dan struktur data yang mudah digunakan. panda tersedia untuk semua instalasi Python, tetapi itu adalah bagian penting dari distribusi Anaconda dan bekerja sangat baik di notebook Jupyter untuk berbagi data, kode, hasil analisis, visualisasi, dan teks naratif.
6. Jelaskan fungsi-fungsi yang terdapat di library csv? Terdapat 2 fungsi yang bisa digunakan oleh library csv Pertama, fungsi membaca file csv. fungsi ini bisa menggunakan list dan dictionary Dengan list :

```

1 def bacacsv():
2     with open('1174042.csv') as csv_file:
3         csv_reader = csv.reader(csv_file, delimiter=',')
4         line_count = 0
5         for row in csv_reader:
6             if line_count == 0:
7                 print(f'Column names are {", ".join(row)}')
8                 line_count += 1
9             else:
10                print(f'\t NPM : {row[0]} Nama : {row[1]} Kelas :
                {row[2]}')
11                line_count += 1

```

Dengan dictionary :

```

1 def bacadictionary():
2     with open('1174042.csv', mode='r') as csv_file:
3         csv_reader = csv.DictReader(csv_file)
4         line_count = 0
5         for row in csv_reader:
6             if line_count == 0:
7                 print(f'Column names are {", ".join(row)}')
8                 line_count += 1
9             print(f'\t NPM : {row["npm"]} Nama : {row["nama"]}
                Kelas : {row["kelas"]}')
10                line_count += 1

```

Kedua, fungsi menulis file csv.

```
1 def nulis():
2     with open('test-tulis.csv', mode='w') as employee_file:
3         employee_writer = csv.writer(employee_file, delimiter=',',
4             , quotechar='"', quoting=csv.QUOTE_MINIMAL)
5         employee_writer.writerow(['Ucok', 'Tukang', 'November'])
```

7. Jelaskan fungsi-fungsi yang terdapat di library pandas Hampir sama dengan library csv, tp library pandas penulisannya lebih sederhana dan terlihat lebih rapih dari pada library csv.

```
1 def bacapandas():
2     df = pandas.read_csv('1174042.csv')
```

LLLLLL c64ab529ff89ebc66e82900d0611abe5d2422d5a

