

Pemrograman II

Diajukan untuk memenuhi kelulusan matakuliah
Pemrograman II Program Studi DIV Teknik Informatika

Oleh :

Naomi C.H Tampubolon
1.18.4.018



PROGRAM DIPLOMA IV TEKNIK INFORMATIKA

POLITEKNIK POS INDONESIA

BANDUNG

2019

DAFTAR ISI

DAFTAR ISI	i
DAFTAR GAMBAR	i
I Chapter I	I-1
1.1 Pengenalan Python	I-1
1.2 Python dalam Dunia Pekerjaan	I-4
1.3 Instalisasi	I-4
II Chapter II	II-16
2.1 Teori	II-16
2.2 Keterampilan Pemrograman	II-37

BAB I

Chapter I

1.1 Pengenalan Python

Python adalah Bahasa pemrograman interpretatif (dalam ilmu komputer penerjemah) multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python merupakan bahasa pemrograman yang mudah dibaca dan dapat digunakan untuk berbagai macam proyek. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python dikembangkan oleh Guido van Rossum pada tahun 1990 di Stichting Mathematisch Centrum (CWI), Amsterdam sebagai kelanjutan dari bahasa pemrograman ABC. Versi terakhir yang dikeluarkan CWI adalah 1.2. Tahun 1995, Guido pindah ke CNRI di Virginia Amerika sambil terus melanjutkan pengembangan Python. Versi terakhir yang dikeluarkan adalah 1.6. Tahun 2000, Guido dan para pengembang inti Python pindah ke BeOpen.com yang merupakan sebuah perusahaan komersial dan membentuk BeOpen PythonLabs. Python 2.0 dikeluarkan oleh BeOpen. Setelah mengeluarkan Python 2.0, Guido dan beberapa anggota tim PythonLabs pindah ke DigitalCreations

Saat ini pengembangan Python terus dilakukan oleh sekumpulan pemrogram yang dikoordinir Guido dan Python Software Foundation. Python Software Foundation adalah sebuah organisasi non-profit yang dibentuk sebagai pemegang hak cipta intelektual Python sejak versi 2.1 dan dengan demikian mencegah Python dimiliki oleh perusahaan komersial. Saat ini distribusi Python sudah mencapai versi 2.7.14 dan versi 3.6.3 Nama Python dipilih oleh Guido sebagai nama bahasa ciptaannya karena kecintaan Guido pada acara televisi Monty Python's Flying Circus. Oleh karena itu seringkali ungkapan-ungkapan khas dari acara tersebut seringkali muncul dalam korespondensi antar pengguna Python. Python memiliki beberapa versi mulai dari versi yang Python 1.0 bulan Januari 1994 sampai yang terbaru Python 3.7 bulan 27 Juni 2018.

Versi python yang 2.0 berbeda dengan python yang 3.0. Beberapa perbedaannya yaitu:

1. Syntax untuk mencetak teks atau yang lainnya.

Pada python 2 syntaxnya yaitu: bisa menggunakan tanda kurung atau tidak. Sedangkan pada python 3 syntaxnya harus menggunakan tanda kurung, jika tidak maka syntaxnya akan error. Contoh:

- Pada python 2, syntax:

```
print "tidak pakek kurung bisa"  
Print("pakek kurung juga bisa")  
Print"ini";print"mencetak satu baris"
```

Hasil :

The screenshot shows a terminal window titled 'Run tes postingan'. The command entered is '/usr/bin/python2.7 "/home/moham/Documents/untitled/tes postingan.py"'. The output shows three lines of text: 'tidak pakek kurung bisa', 'pakek kurung juga bisa', and 'ini mencetak satu baris'. At the bottom, it says 'Process finished with exit code 0'.

Figure 1.1: hasil gambar python 2

- Pada python 3, syntax: print("harus pakai kurung")

```
print("ini digunakan untuk", end="")  
print("python")
```

Hasil:

The screenshot shows a terminal window titled 'Python 3.7 (64-bit)'. It displays a command-line session where the user types 'print("harus pakai kurung")' and 'print"harus pakai kurung"'. The second line is highlighted with a cursor. An error message follows: 'SyntaxError: invalid syntax'.

Figure 1.2: hasil gambar python 3

2. Syntax untuk meminta inputan

Pada python 2 jika ingin meminta inputan kita harus menggunakan syntax seperti yang ada di bawah . Sedangkan pada python 3 kita hanya memerlukan syntax dibawah. Contoh:

- Pada python 2, syntax:

```
nama = raw_input('masukkan nama anda :')
print nama
```

Catatan: garis penghubung di atas di ubah menjadi garis bawah.

Hasil:

```
/usr/bin/python2.7 "/home/moham/Documents/untitled/tes_postingan.py"
masukkan nama anda : moh. nikmat
moh. nikmat
Process finished with exit code 0
```

Figure 1.3: hasil gambar python 2

- Pada python 3, syntax: `nama = input("masukkan nama anda: ")`

```
print(nama)
```

Hasil:

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> nama = input('masukkan nama: ')
masukkan nama: naomi
>>> print(nama)
naomi
>>> -
```

Figure 1.4: hasil gambar python 3

3. Hasil Operator Pembagian

Pada python 2 hasil pembagian $3/2$ adalah 1. Sedangkan pada python 3 yaitu 1,5. Contoh:

- Pada python 2 `print"3/2=", 3/2`

```
print"3//2=", 3//2
```

```
print"3/2.0=", 3/2.0
```

```
print"3//2.0=", 3//2.0
```

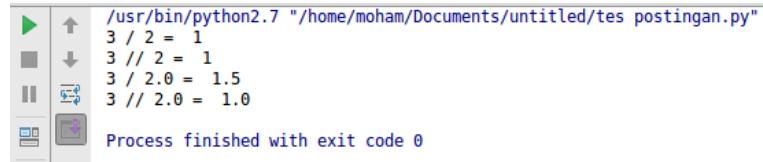
Hasil:

- Pada python 2 `print"3/2=", 3/2`

```
print"3//2=", 3//2
```

```
print"3/2.0=", 3/2.0
```

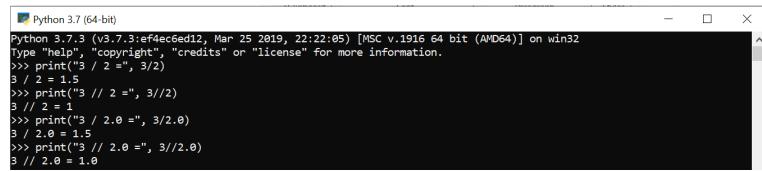
```
print"3//2.0=", 3//2.0
```



```
/usr/bin/python2.7 "/home/moham/Documents/untitled/tes_postingan.py"
3 / 2 = 1
3 // 2 = 1
3 / 2.0 = 1.5
3 // 2.0 = 1.0
Process finished with exit code 0
```

Figure 1.5: hasil gambar python 2

Hasil:



```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("3 / 2 =", 3/2)
3 / 2 = 1.5
>>> print("3 // 2 =", 3//2)
3 // 2 = 1
>>> print("3 / 2.0 =", 3/2.0)
3 / 2.0 = 1.5
>>> print("3 // 2.0 =", 3//2.0)
3 // 2.0 = 1.0
```

Figure 1.6: hasil gambar python 3

1.2 Python dalam Dunia Pekerjaan

Python dapat digunakan untuk berbagai pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. Maka dari itu Bahasa pemrograman python sangat dibutuhkan dalam dunia pekerjaan. Python sebenarnya masih jarang di Indonesia, dan itu bukan berarti Bahasa pemrograman python lemah. Sebenarnya python lebih banyak dicari di level internasional. Implementasi dan penggunaan python pada dunia pekerjaan yaitu karena python dikenal memiliki banyak kelebihan, dan manfaat yaitu untuk pengembangan web, video game, GUI desktop, hingga perangkat lunak. Dan hal-hal itulah yang paling sering dan banyak dijumpai pada dunia pekerjaan.

1.3 Instalisasi

1. Install Python

- Buka link <https://www.python.org/downloads/release/python-380/>
- Kemudian download python 3.
- Kemudian klik “Install Now”.
- Lalu tunggu instalasi sampai selesai.

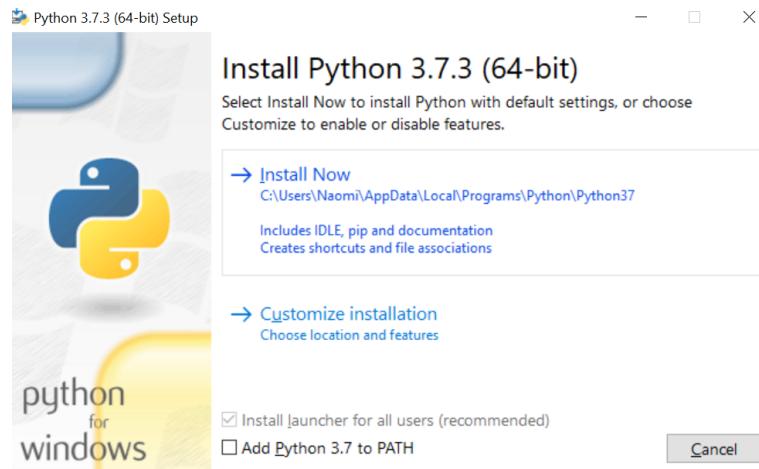


Figure 1.7: install python 3

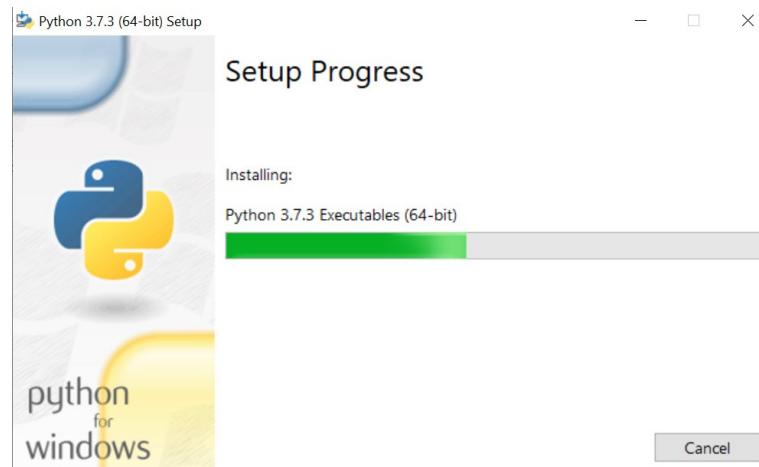


Figure 1.8: install python 3

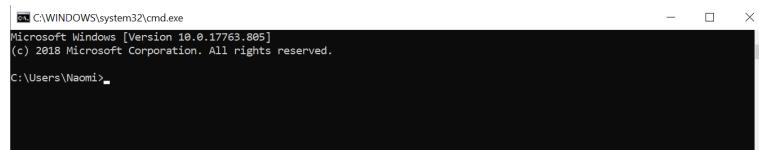


Figure 1.9: install python 3

- Jika sudah selesai maka akan muncul seperti gambar di atas, yang artinya python sudah terinstal.

2. Instal pip

- Buka CDM(Command Prompt) melalui Start atau dengan cara tekan Windows+R secara bersamaan di keyboard.

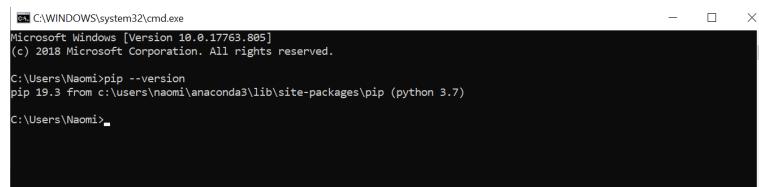


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Naomi>
```

Figure 1.10: install pip

- Kemudian ketik “pip –version” pada CMD, lalu enter. Maka akan muncul seperti pada gambar.

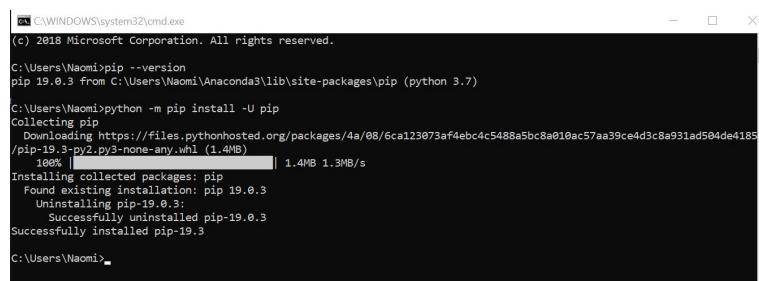


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Naomi>pip --version
pip 19.0.3 from c:\users\naomi\anaconda3\lib\site-packages\pip (python 3.7)
C:\Users\Naomi>
```

Figure 1.11: install pip

- Lalu ketik “python –m pip install –U pip” seperti pada gambar.



```
C:\WINDOWS\system32\cmd.exe
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Naomi>pip --version
pip 19.0.3 from c:\users\Naomi\Anaconda3\lib\site-packages\pip (python 3.7)

C:\Users\Naomi>python -m pip install -U pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/4a/08/6ca123073af4ebc4c5488a5bc8a010ac57aa39ce4d3c8a931ad504de4185
/pip-19.3-py2.py3-none-any.whl (1.4MB)
   100% |████████████████████████████████| 1.4MB 1.3MB/s
Installing collected packages: pip
  Found existing installation: pip 19.0.3
    Uninstalling pip-19.0.3:
      Successfully uninstalled pip-19.0.3
Successfully installed pip-19.3

C:\Users\Naomi>
```

Figure 1.12: install pip

- Pip install python selesai.

3. Cara Setting Environment

- Pertama tama buka Anaconda Navigator.
- Lalu klik Environment

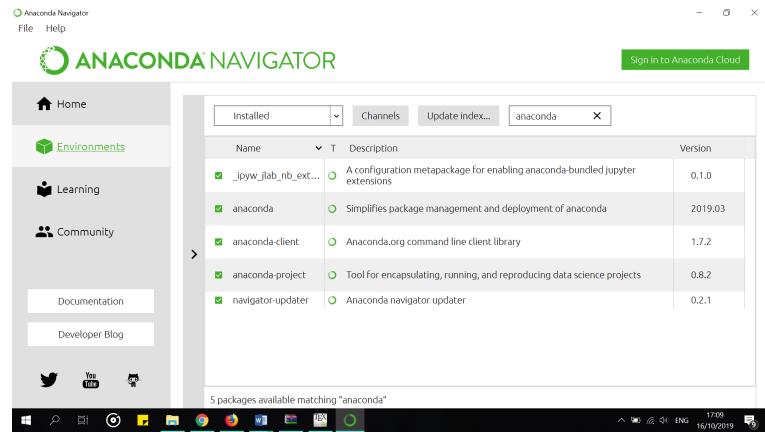


Figure 1.13: environment

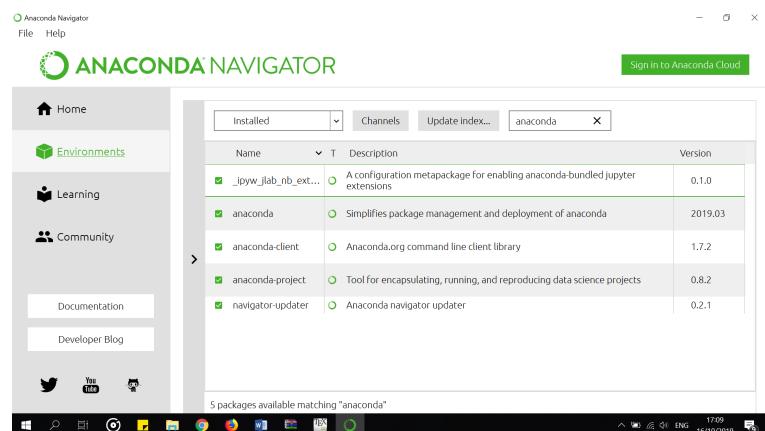


Figure 1.14: environment

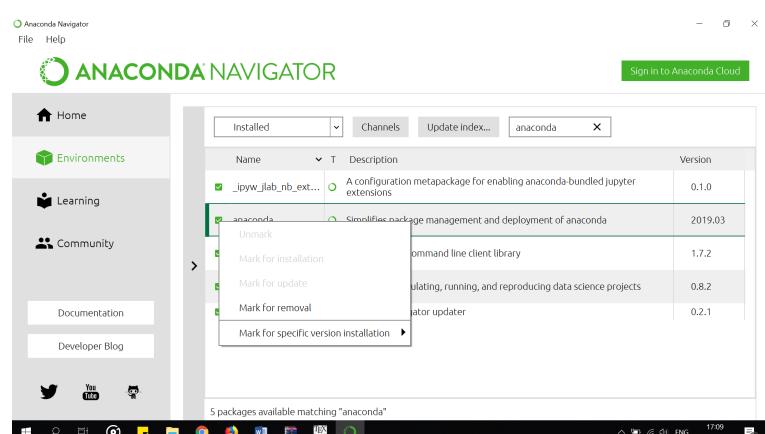


Figure 1.15: environment

- lalu klik centang hijau yang ada di sebelah kiri. seperti berikut
- Arahkan pointer ke bagian yang paling bawah

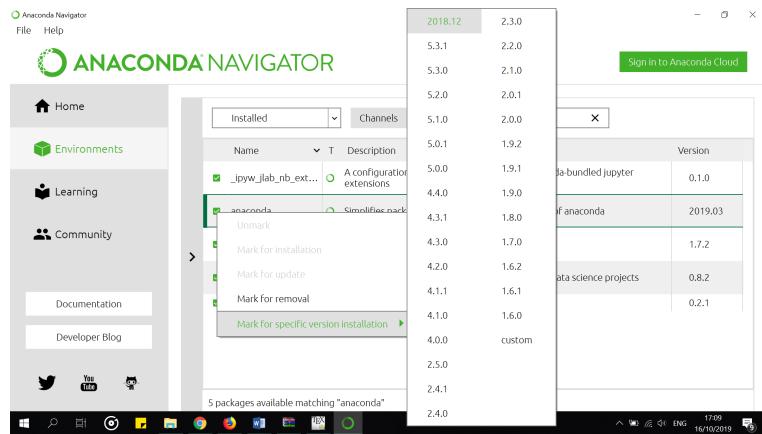


Figure 1.16: environment

- Lalu klik versi yang paling terbaru.

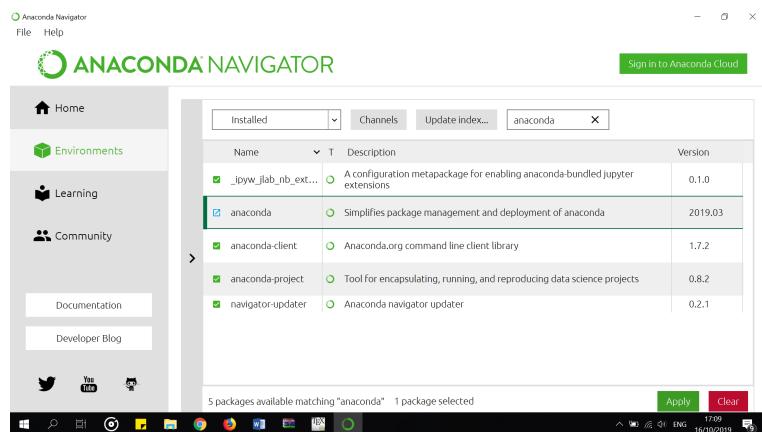


Figure 1.17: environment

- Kemudian Apply.
- Klik Apply.
- Tunggu sampai instalasi selesai.

4. Menjalankan Anaconda dan Spyder

- Install Anaconda di <https://www.anaconda.com/distribution/>
- Jika sudah didownload, buka installer Anaconda
- Klik Next.

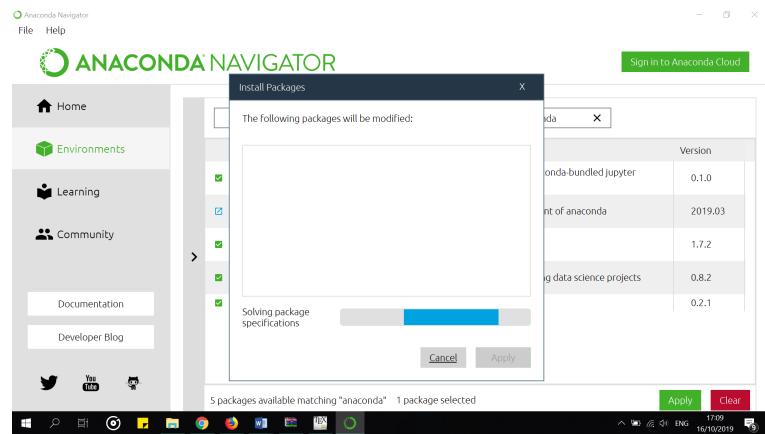


Figure 1.18: environment



Figure 1.19: environment

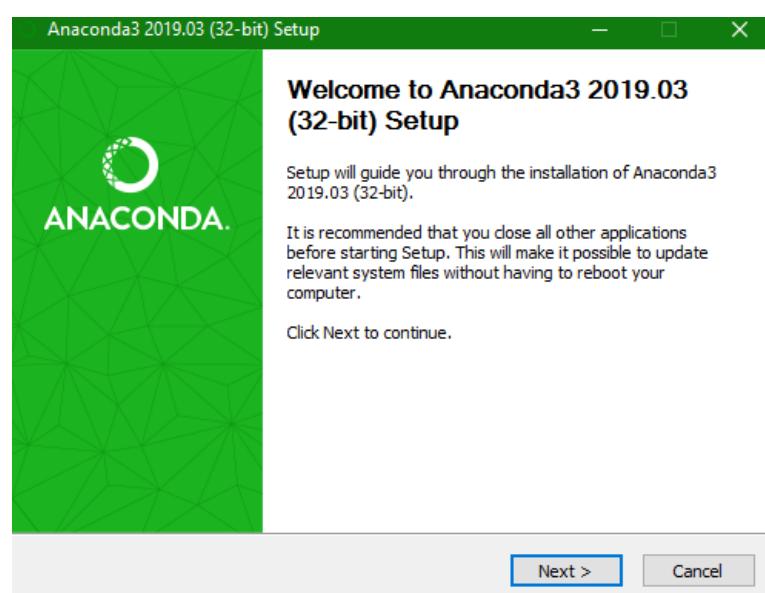


Figure 1.20: install anaconda

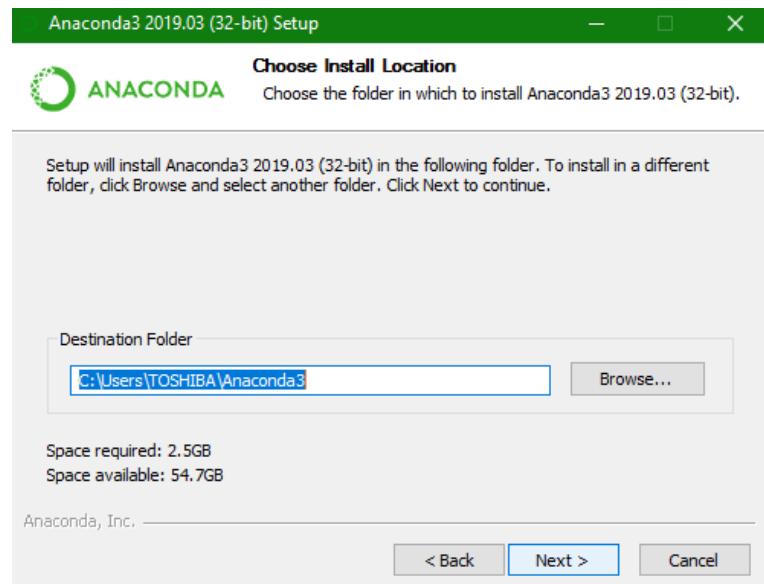


Figure 1.21: install anaconda

- Kemudian Pilih lokasi tempat penyimpanan aplikasi.
- Lalu pilih Just Me(recommended) agar sesuai dengan computer yang anda miliki.

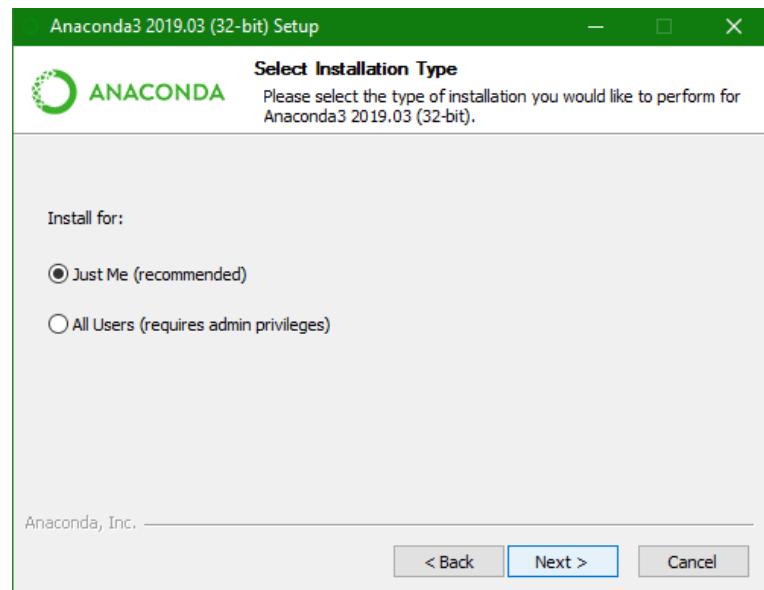


Figure 1.22: install anaconda

- Klik Next.
- Kemudian ceklis Add Anaconda to my PATH.

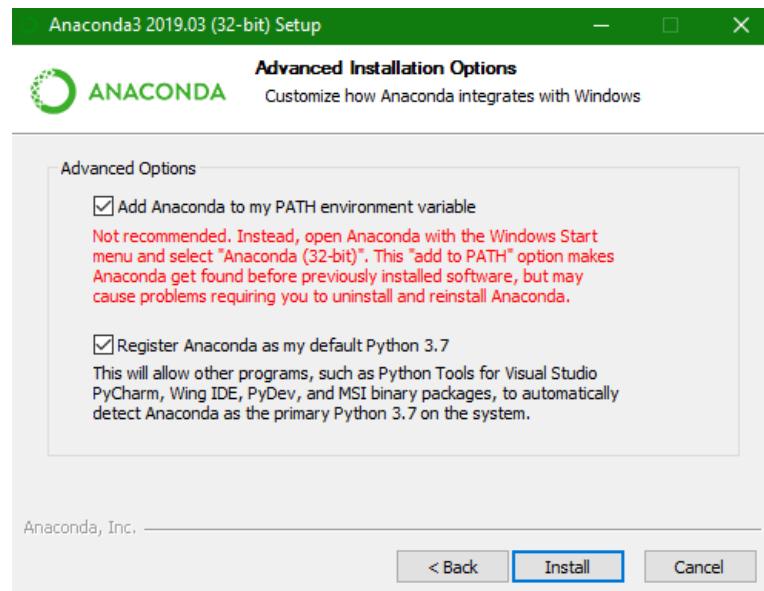


Figure 1.23: install anaconda

- Lalu anda centang Add Anaconda to my Path environment variable, agar saat mengisntall selenium langsung ke path anaconda tidak ke aplikasi yang lain.
- Jika sudah klik install, tunggu sampai selesai proses installasi selesai.

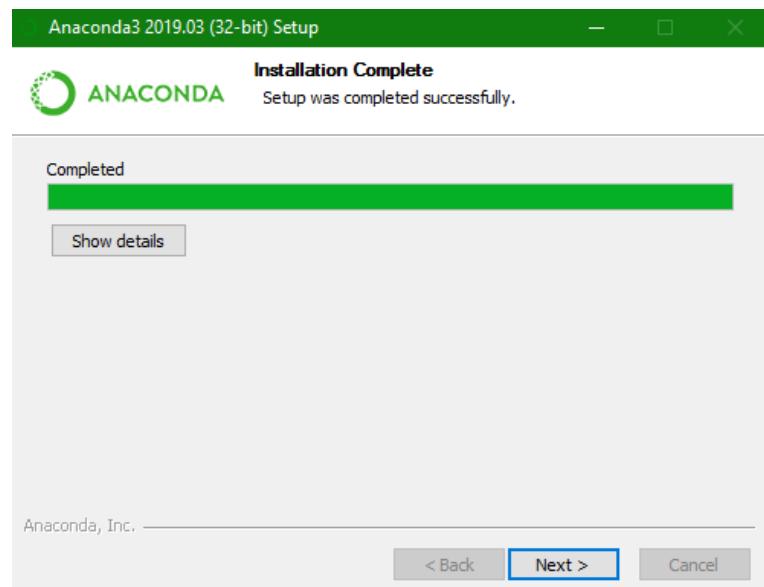


Figure 1.24: install anaconda

- Klik Next.

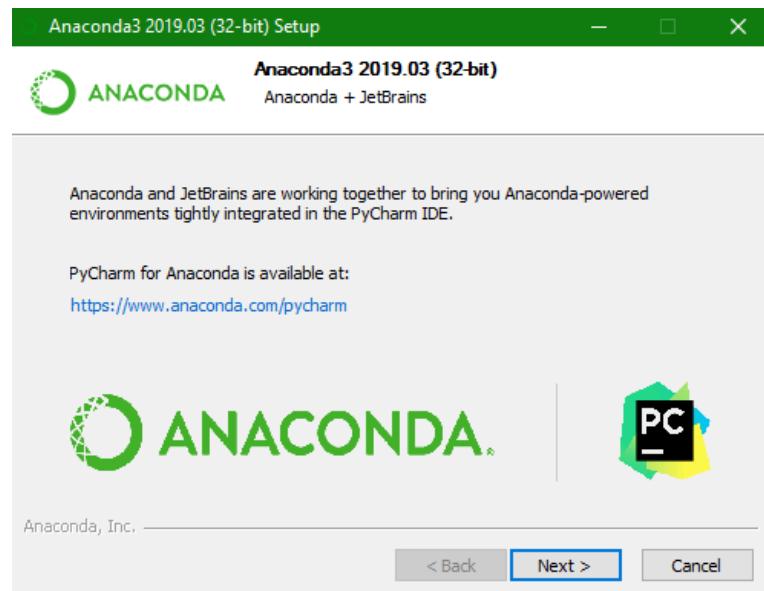


Figure 1.25: install anaconda

- Klik Next.

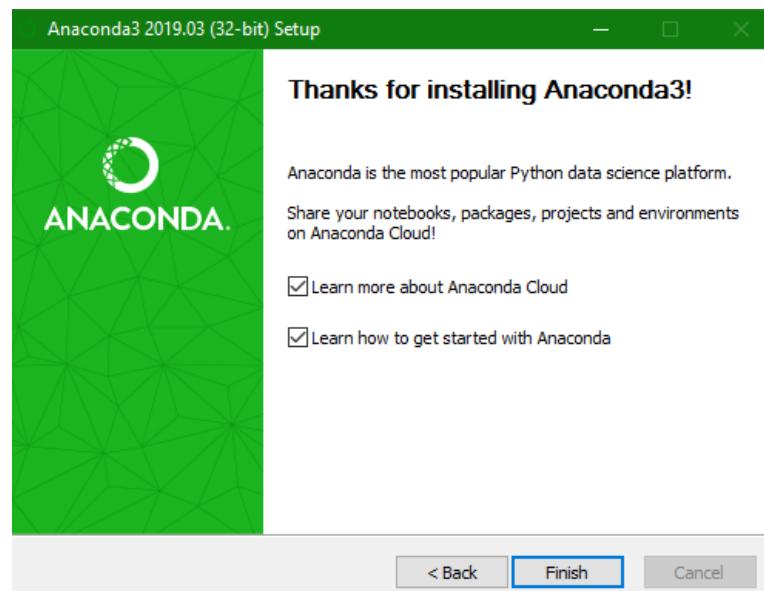


Figure 1.26: install anaconda

- Jika sudah klik Finish

5. Install Syder

- Buka Anaconda Navigator.
- Lalu scroll ke bawah untuk menemukan spyder.

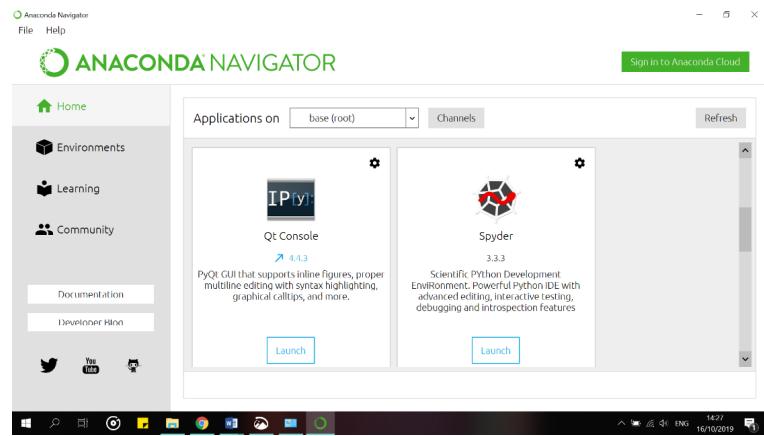


Figure 1.27: install spyder

- Lalu akan ada Spyder pada Anaconda Navigator
- Lalu lunch spyder, maka spyder akan terinstal otomatis.
- Buka Spyder yang sudah di instal.

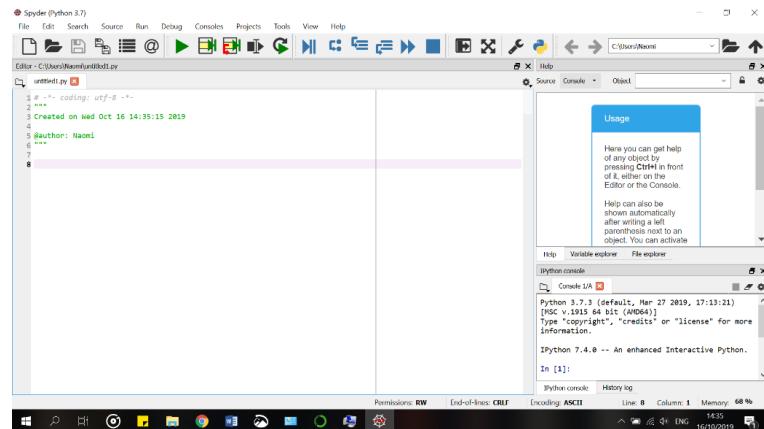


Figure 1.28: install spyder

6. Cara Menjalankan Script Otomatis Login Akademik

- Buka spyder yang sudah di install.
- Kemudian ketik codingan sebagai berikut agar bisa login ke browser akademik yang kita tuju.
- Run Codingan yang telah diketik.
- Maka secara otomatis website siap.poltekpos.ac.id akan terbuka secara otomatis.

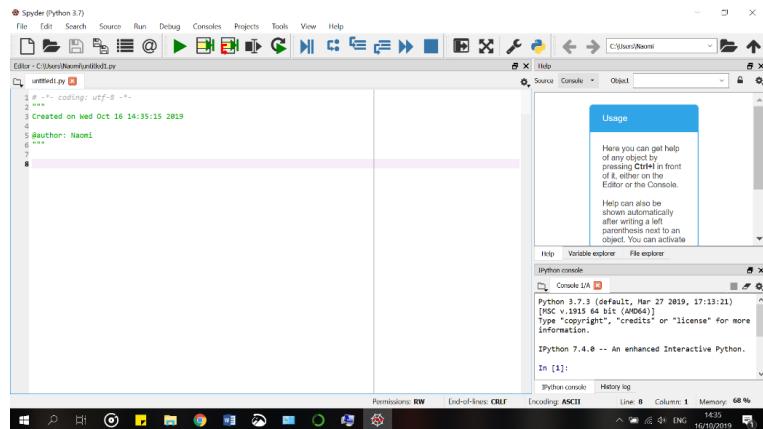


Figure 1.29: spyder

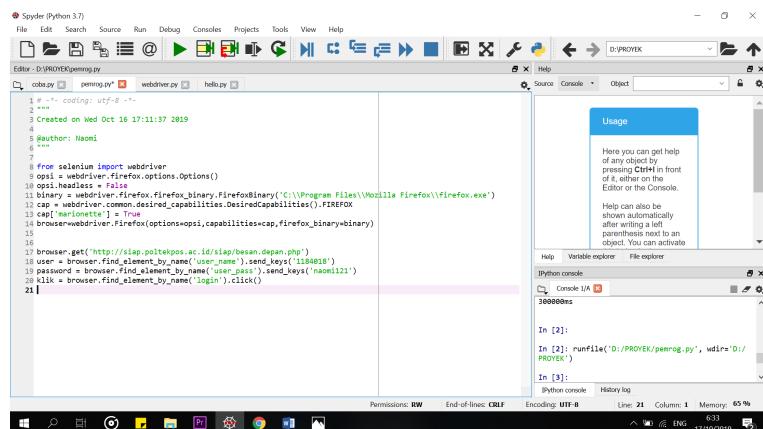


Figure 1.30: spyder

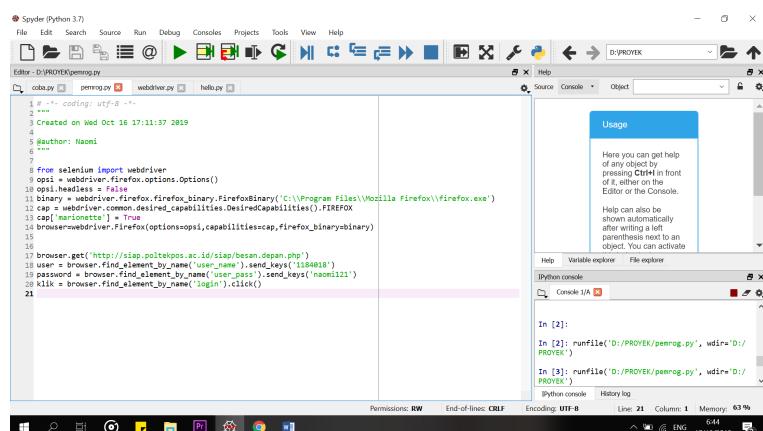


Figure 1.31: spyder

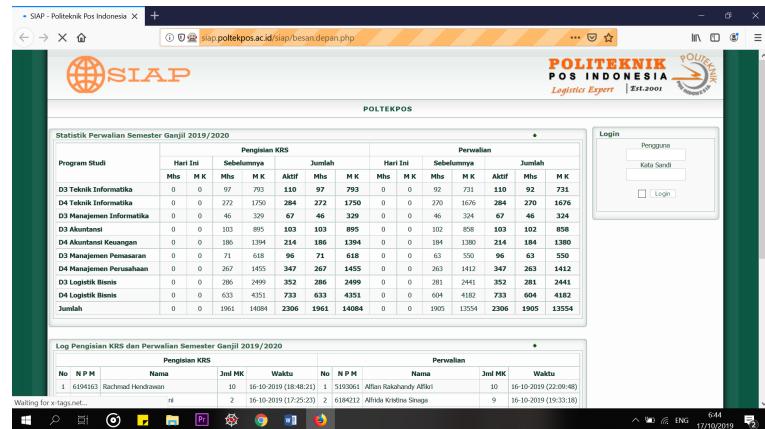


Figure 1.32: spyder

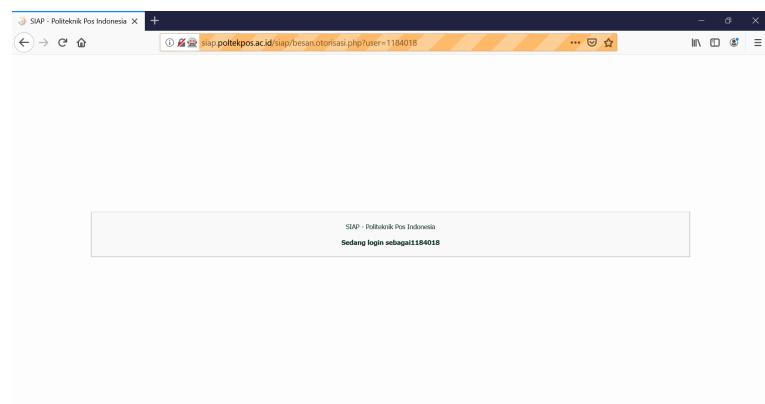


Figure 1.33: spyder

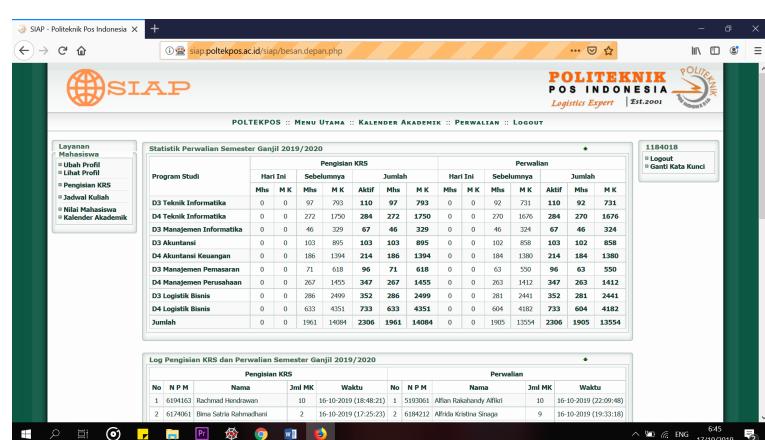


Figure 1.34: spyder

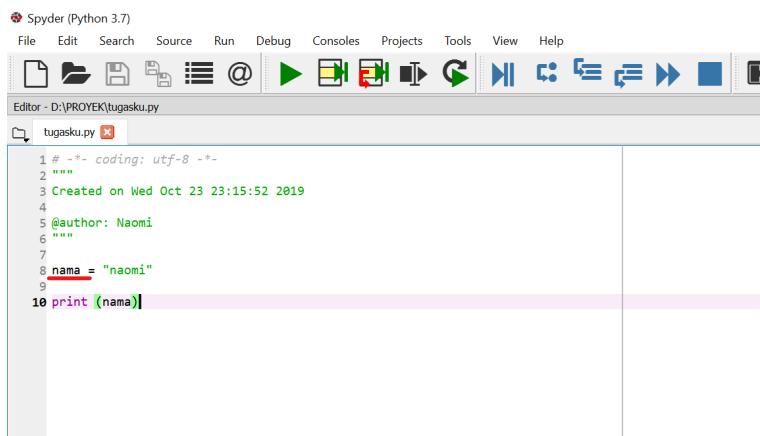
BAB II

Chapter II

2.1 Teori

1. Jenis-Jenis Variabel.

Variabel adalah lokasi di memori yang digunakan untuk menyimpan nilai. Pada saat kita membuat sebuah variabel, kita ‘memesan’ tempat di dalam memori. Tempat tersebut bisa diisi dengan data atau objek, baik itu bilangan bulat (integer), pecahan (float), karakter (string), dan lain – lain. Contoh:



```
# coding: utf-8
"""
Created on Wed Oct 23 23:15:52 2019
@author: Naomi
"""

nama = "naomi"
print(nama)
```

Figure 2.1: contoh variabel

Nama atau tanda yang diberi garis merah merupakan sebuah variabel dan “naomi” merupakan nilai dari varibel nama.

2. Tipe Data

Python memiliki enam tipe yang sering digunakan:

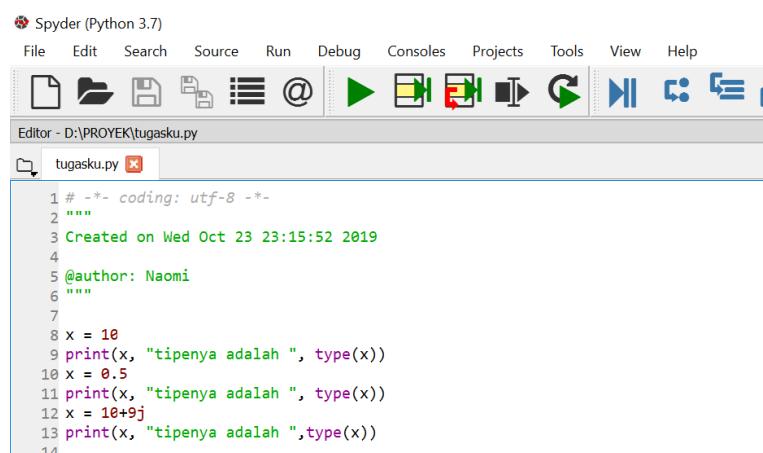
- Bilangan (Number)
- String
- List
- Set

- Dictionary

Penjelasan sebagai berikut:

- **Bilangan (Number)**

Tipe data bilangan ada dua yaitu Integer dan float. Integer ialah bilangan bulat, dan Float bilangan pecahan. Namun masih ada jenis bilangan lain yaitu bilangan kompleks yaitu bilangan yang memiliki bagian real dan imajiner. pada python tipe data bilangan diwakili kelas int, float, complex. dan fungsi yang dipakai yaitu *type()* yang digunakan untuk mengetahui tipe bilangan.



```

Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - D:\PROYEK\tugasku.py
tugasku.py x
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 x = 10
9 print(x, "tipenya adalah ", type(x))
10 x = 0.5
11 print(x, "tipenya adalah ", type(x))
12 x = 10+9j
13 print(x, "tipenya adalah ", type(x))
14

```

Figure 2.2: contoh tipe data bilangan

Hasil:

```

In [7]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
10 tipenya adalah <class 'int'>
0.5 tipenya adalah <class 'float'>
(10+9j) tipenya adalah <class 'complex'>

```

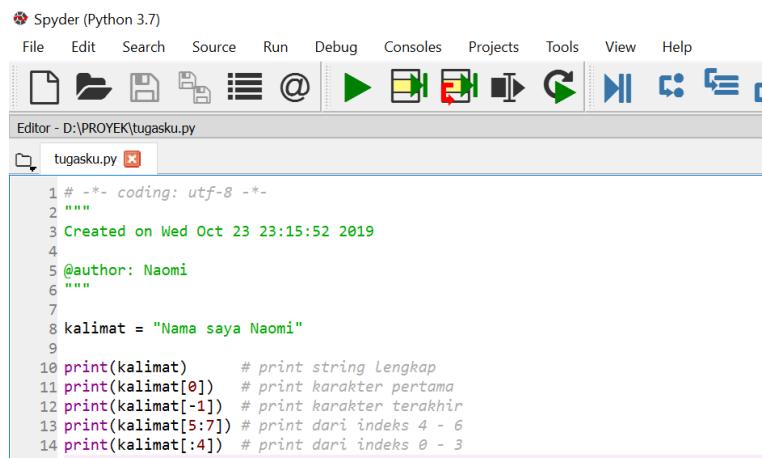
Figure 2.3: hasil contoh tipe data bilangan

- **String**

String merupakan satu atau serangkaian karakter yang diletakkan di antara tanda kutip, baik tanda kutip tunggal (‘) maupun ganda (”)

). Huruf, angka, maupun karakter lainnya yang digabung menjadi teks adalah contoh string. String adalah tipe data yang anggotanya berurut dan memiliki indeks.

Indeks dimulai dari angka 0 bila dimulai dari depan dan -1 bila diindeks dari belakang. Tiap karakter bisa dilihat dari indeksnya dengan format *namastring[indeks]*. Pada tipe data ini bisa melakukan teknik slicing (memotong nilai variabel sesuai dengan yang kita inginkan) dengan format *namastring[awal:akhir]*. Contoh:



```
# Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - D:\PROYEK\tugasku.py
tugasku.py
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 kalimat = "Nama saya Naomi"
9
10 print(kalimat)      # print string Lengkap
11 print(kalimat[0])   # print karakter pertama
12 print(kalimat[-1])  # print karakter terakhir
13 print(kalimat[5:7]) # print dari indeks 4 - 6
14 print(kalimat[:4])  # print dari indeks 0 - 3
```

Figure 2.4: contoh tipe data string

Hasil:

```
In [7]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
10 tipenya adalah <class 'int'>
0.5 tipenya adalah <class 'float'>
(10+9j) tipenya adalah <class 'complex'>
```

Figure 2.5: hasil contoh tipe data string

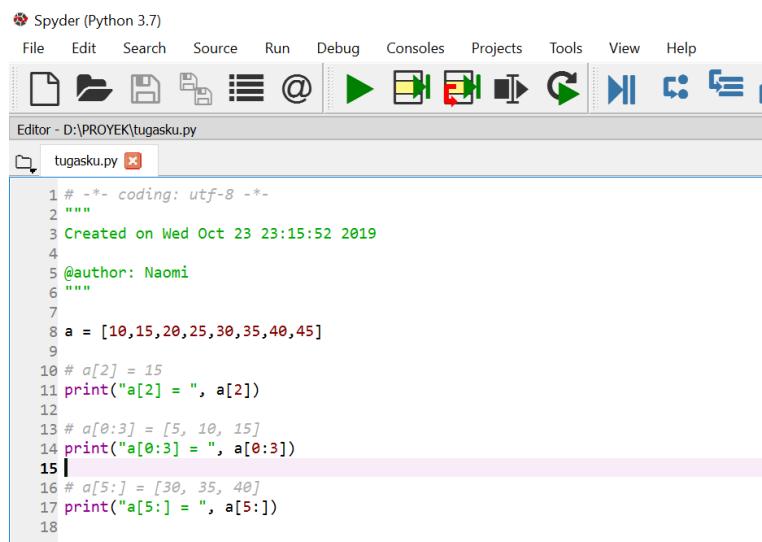
- **List**

List merupakan tipe data berisi item berurut. seperti string, tiap item (anggota) list memiliki indeks sesuai dengan urutannya masing-masing. Indeks dimulai dari awal yaitu 0 bukan 1.

List bisa digunakan dengan anggota tipe yang sama maupun beda. Untuk mendeklarasikan list, gunakan tanda kurung [] da masing masing

anggotanya di pisahkan oleh tanda koma.

Untuk mengakses item dari list caranya adalah dengan memanggil nama list diikuti indeks dari item yang bersangkutan, yaitu dengan format *namelist[index]*. Selain itu bisa juga dilakukan pengaksesan terhadap sejumlah item dari indeks ke indeks. Contoh:



```
# coding: utf-8
"""
Created on Wed Oct 23 23:15:52 2019
@author: Naomi
"""

a = [10,15,20,25,30,35,40,45]
# a[2] = 15
print("a[2] = ", a[2])
# a[0:3] = [5, 10, 15]
print("a[0:3] = ", a[0:3])
[5]
# a[5:] = [30, 35, 40]
print("a[5:] = ", a[5:])

```

Figure 2.6: contoh tipe data list

Hasil:

```
In [15]: runfile('D:/PROYEK/tugasku.py',
                   wdir='D:/PROYEK')
a[2] = 20
a[0:3] = [10, 15, 20]
a[5:] = [35, 40, 45]
```

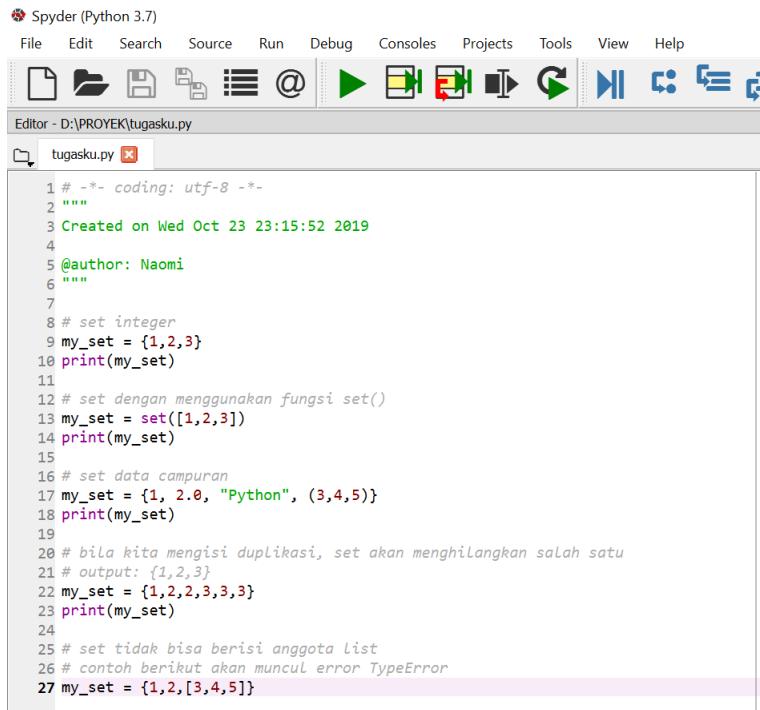
Figure 2.7: hasil contoh tipe data list

- **Set**

Set adalah satu tipe data di Python yang tidak berturut turut (unordered). Set memiliki anggota yang unik. Sandainya jika diletakkan dua anggota yang sama di dalam set, maka otomatis tipe data set akan menghilangkan salah satunya. Set bisa digunakan untuk melakukan operasi himpunan matematika seperti gabungan, irisan, selisih, dan komplemen.

Set dibuat dengan meletakkan anggota – anggotanya di dalam tanda kurung kurawal , dipisahkan menggunakan tanda koma. Kita juga bisa

membuat set dari list dengan memasukkan list ke dalam fungsi `set()`. Set bisa berisi data campuran, baik integer, float, string, dan lain sebagainya. Akan tetapi set tidak bisa berisi list, set, dan dictionary. Contoh:



```

Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - D:\PROYEK\tugasku.py
tugasku.py
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 # set integer
9 my_set = {1,2,3}
10 print(my_set)
11
12 # set dengan menggunakan fungsi set()
13 my_set = set([1,2,3])
14 print(my_set)
15
16 # set data campuran
17 my_set = {1, 2.0, "Python", (3,4,5)}
18 print(my_set)
19
20 # bila kita mengisi duplikasi, set akan menghilangkan salah satu
21 # output: {1,2,3}
22 my_set = {1,2,2,3,3,3}
23 print(my_set)
24
25 # set tidak bisa berisi anggota list
26 # contoh berikut akan muncul error TypeError
27 my_set = {1,2,[3,4,5]}

```

Figure 2.8: contoh tipe data set

Hasil:

```

In [17]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
In [18]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
{1, 2, 3}
{1, 2, 3}
{1, 2.0, 'Python', (3, 4, 5)}
{1, 2, 3}
Traceback (most recent call last):

  File "<ipython-input-18-44f63cb8d48c>", line 1, in <module>
    runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')

  File "C:\Users\Naomi\Anaconda\lib\site-packages\spyder_kernels\customize
\spydercustomize.py", line 786, in runfile
    execfile(filename, namespace)

  File "C:\Users\Naomi\Anaconda\lib\site-packages\spyder_kernels\customize
\spydercustomize.py", line 110, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

  File "D:/PROYEK/tugasku.py", line 27, in <module>
    my_set = {1,2,[3,4,5]}

TypeError: unhashable type: 'list'

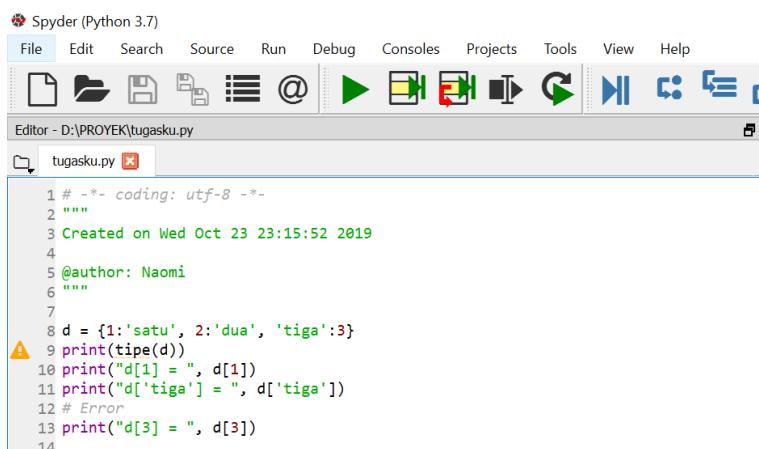
```

Figure 2.9: hasil contoh tipe data set

- **Dictionary**

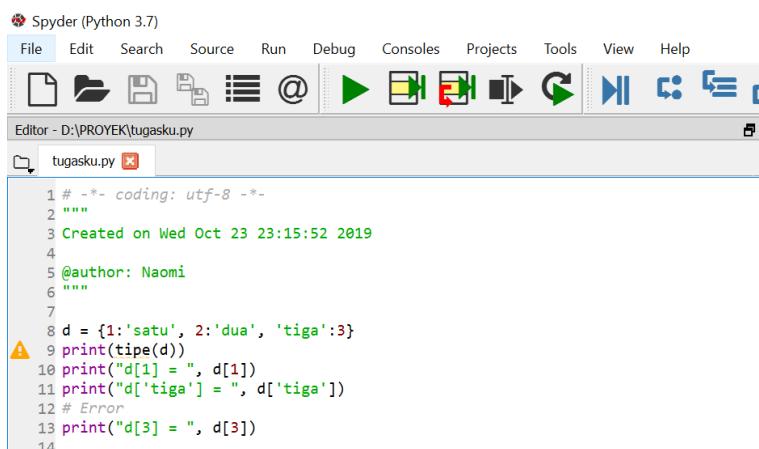
Dictionary adalah tipe data yang tiap anggotanya terdiri dari pasangan kunci-nilai (key-value). Mirip dengan kamus dimana ada kata ada arti. Dictionary umumnya dipakai untuk data yang besar dan untuk mengakses anggota data secara acak. Anggota dictionary tidak memiliki indeks. Dictionary dideklarasikan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

Untuk mengakses nilai dari anggota dictionary, kita menggunakan key-nya. Contoh: Hasil:



```
# Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - D:\PROYEK\tugasku.py
tugasku.py
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 d = {1:'satu', 2:'dua', 'tiga':3}
9 print(type(d))
10 print("d[1] = ", d[1])
11 print("d['tiga'] = ", d['tiga'])
12 # Error
13 print("d[3] = ", d[3])
14
```

Figure 2.10: contoh tipe data dictionary



```
# Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - D:\PROYEK\tugasku.py
tugasku.py
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 d = {1:'satu', 2:'dua', 'tiga':3}
9 print(type(d))
10 print("d[1] = ", d[1])
11 print("d['tiga'] = ", d['tiga'])
12 # Error
13 print("d[3] = ", d[3])
14
```

Figure 2.11: hasil contoh tipe data dictionary

- Catatan:

- Tipe data sering disebut *objek*. Pada dasarnya semua hal di python adalah objek.
- Ada tipe data lain yang umumnya dimiliki oleh bahasa Python, yaitu tipe *None*. Tipe *None* adalah sebuah tipe data spesial yang menunjukkan bahwa nilai/data suatu variabel itu belum/tidak ada (bukan nol, tapi tidak ada). Pada bahasa pemrograman lain seperti C, atau PHP, tipe data ini disebut *null*.
- Tipe data string, tuple, dan list masuk ke dalam tipe data yang disebut tipe data *berurut / ordered* atau *sekuensial / sequence*. Tipe data dictionary disebut data *tidak berurut / unordered*.

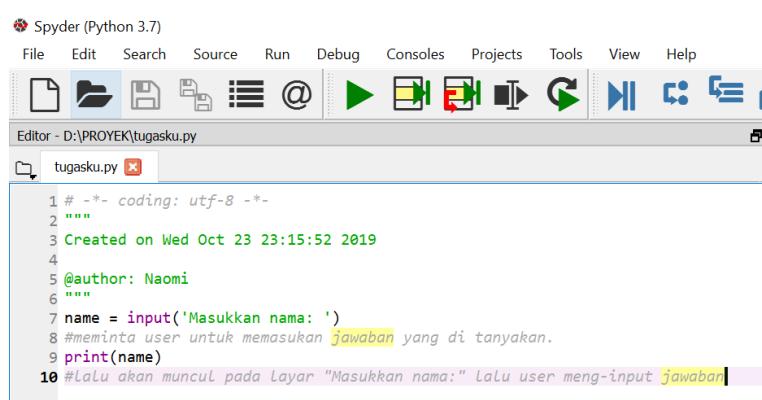
3. Input dan Output

Python menyediakan banyak fungsi yang digunakan. Salah satunya adalah fungsi *i/o* atau *input output*.

Fungsi untuk melakukan operasi output adalah *print()*, dan fungsi untuk melakukan operasi input adalah fungsi *input()*.

- Input

Python memiliki fungsi *input()*. Agar lebih memahami fungsi *input()* lihat contoh berikut:



```
# Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - D:\PROYEK\tugasku.py
tugasku.py
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7 name = input('Masukkan nama: ')
8 #meminta user untuk memasukan jawaban yang di tanyakan.
9 print(name)
10 #lalu akan muncul pada Layar "Masukkan nama:" lalu user meng-input jawaban|
```

Figure 2.12: contoh fungsi input

lalu akan muncul gambar seperti berikut:

```
In [7]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
Masukkan nama:
```

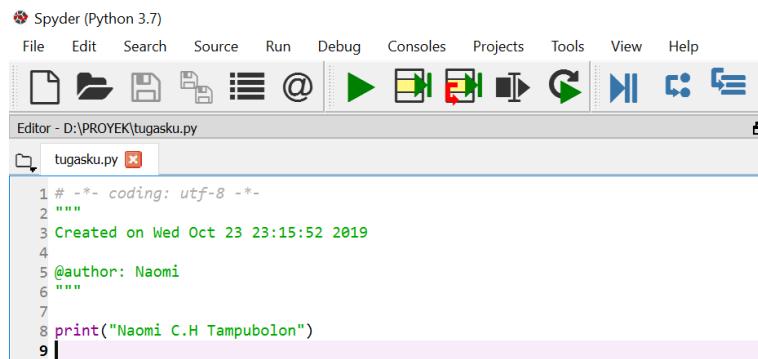
Figure 2.13: contoh fungsi input

Hasil:

```
In [7]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
Masukkan nama: Naomi Cindy Hermina Tampubolon
Naomi Cindy Hermina Tampubolon
```

Figure 2.14: hasil contoh tipe data dictionary

- Output Output menggunakan fungsi *print()*. Seperti yang sudah kita praktekkan, kita menggunakan fungsi *print()* untuk menampilkan data ke perangkat keluaran standar (layar). Contoh:



```
# Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - D:\PROYEK\tugasku.py
tugasku.py
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 print("Naomi C.H Tampubolon")
9 |
```

Figure 2.15: contoh fungsi output

Hasil:

```
In [8]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
Naomi C.H Tampubolon
```

Figure 2.16: hasil contoh output

4. Operasi dasar Aritmatika

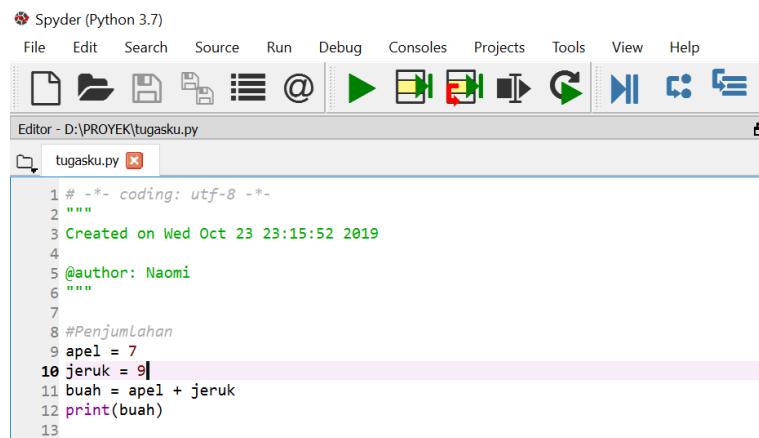
Contoh:

Operator Aritmatika

Operator	Contoh	Penjelasan
Penjumlahan +	$1 + 3 = 4$	Menjumlahkan nilai dari masing-masing operan atau bilangan
Pengurangan -	$4 - 1 = 3$	Mengurangi nilai operan di sebelah kiri menggunakan operan di sebelah kanan
Perkalian *	$2 * 4 = 8$	Mengalikan operan/bilangan
Pembagian /	$10 / 5 = 2$	Untuk membagi operan di sebelah kiri menggunakan operan di sebelah kanan
Sisa Bagi %	$11 \% 2 = 1$	Mendapatkan sisa pembagian dari operan di sebelah kiri operator ketika dibagi oleh operan di sebelah kanan
Pangkat **	$8 ** 2 = 64$	Memangkatkan operan disebelah kiri operator dengan operan di sebelah kanan operator
Pembagian Bulat //	$10 //$ $3 = 3$	Sama seperti pembagian. Hanya saja angka dibelakang koma dihilangkan

Figure 2.17: contoh Operasi dasar Aritmatika

- Penjumlahan



```
# coding: utf-8
"""
Created on Wed Oct 23 23:15:52 2019
@author: Naomi
"""

#Penjumlahan
apel = 7
jeruk = 9
buah = apel + jeruk
print(buah)
```

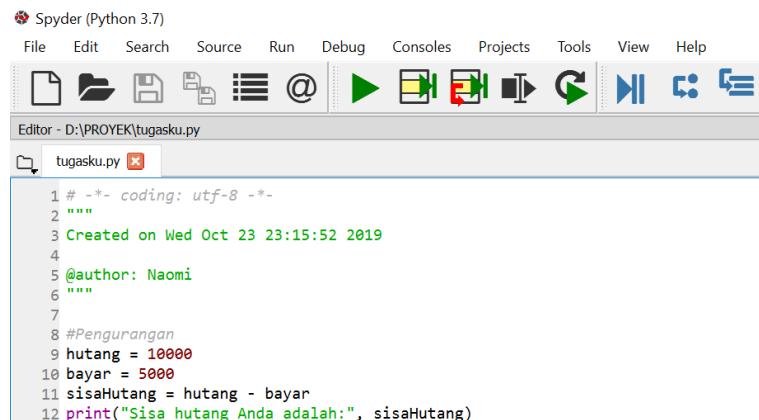
Figure 2.18: contoh operasi dasar aritmatika penjumlahan

Hasil:

```
In [10]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
16
```

Figure 2.19: hasil contoh operasi dasar aritmatika penjumlahan

- Pengurangan



```
# coding: utf-8
"""
Created on Wed Oct 23 23:15:52 2019
@author: Naomi
"""

#Pengurangan
hutang = 10000
bayar = 5000
sisaHutang = hutang - bayar
print("Sisa hutang Anda adalah:", sisaHutang)
```

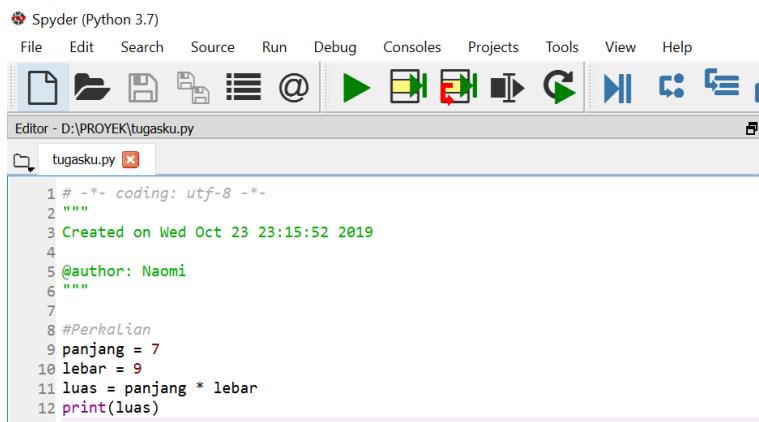
Figure 2.20: contoh operasi dasar aritmatika pengurangan

Hasil:

```
In [13]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
Sisa hutang Anda adalah: 5000
```

Figure 2.21: hasil contoh operasi dasar aritmatika pengurangan

- Perkalian



```
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - D:\PROYEK\tugasku.py
tugasku.py x
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 #Perkalian
9 panjang = 7
10 lebar = 9
11 luas = panjang * lebar
12 print(luas)
--
```

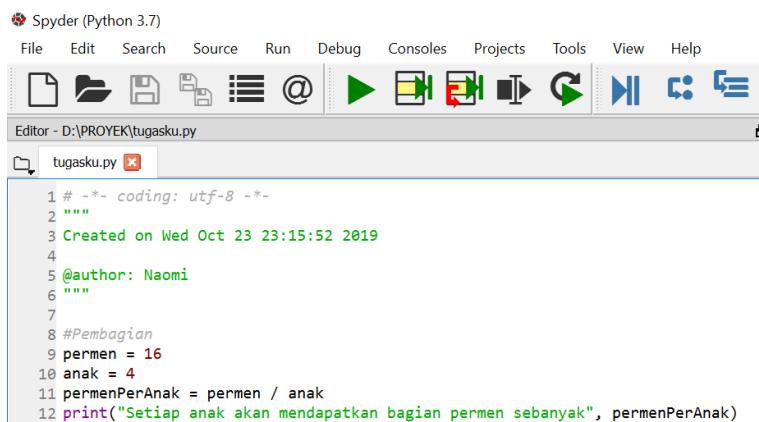
Figure 2.22: contoh operasi dasar aritmatika perkalian

Hasil:

```
In [14]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
63
```

Figure 2.23: hasil contoh operasi dasar aritmatika perkalian

- Pembagian



```
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - D:\PROYEK\tugasku.py
tugasku.py x
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 #Pembagian
9 permen = 16
10 anak = 4
11 permenPerAnak = permen / anak
12 print("Setiap anak akan mendapatkan bagian permen sebanyak", permenPerAnak)
```

Figure 2.24: contoh operasi dasar aritmatika pembagian

Hasil:

```
In [15]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
Setiap anak akan mendapatkan bagian permen sebanyak 4.0
```

Figure 2.25: hasil contoh operasi dasar aritmatika pembagian

- Sisa Bagi

```
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - D:/PROYEK/tugasku.py
tugasku.py
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 #Sisa Bagi / Modulus
9 bilangan1 = 14
10 bilangan2 = 5
11 hasil = bilangan1 % bilangan2
12 print("Sisa bagi dari bilangan ", bilangan1, " dan ", bilangan2, " adalah ", hasil)
```

Figure 2.26: contoh operasi dasar aritmatika sisa bagi

Hasil:

```
In [16]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
Sisa bagi dari bilangan 14 dan 5 adalah 4
```

Figure 2.27: hasil contoh operasi dasar aritmatika sisa bagi

- Pangkat

```
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - D:/PROYEK/tugasku.py
tugasku.py
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 #Pangkat
9 bilangan3 = 8
10 bilangan4 = 2
11 hasilPangkat = bilangan3 ** bilangan4
12 print(hasilPangkat)
```

Figure 2.28: contoh operasi dasar aritmatika pangkat

Hasil:

```
In [17]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
64
```

Figure 2.29: hasil contoh operasi dasar aritmatika pangkat

- Pembagian Bulat

```
# coding: utf-8 -*-
"""
Created on Wed Oct 23 23:15:52 2019
@author: Naomi
"""

#Pembagian Bulat
print(10//3)
#10 dibagi 3 adalah 3.3333. Karena dibulatkan maka akan menghasilkan nilai 3
```

Figure 2.30: contoh operasi dasar aritmatika pembagian bulat

Hasil:

```
In [18]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
3
```

Figure 2.31: hasil contoh operasi dasar aritmatika pembagian bulat

5. Perulangan (Looping)

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

- Perulangan For

Pengulangan *for* pada Python memiliki kemampuan untuk mengulangi item dari urutan apapun, seperti *list* atau *string*. Contoh:

The screenshot shows the Spyder Python 3.7 IDE interface. The title bar says "Spyder (Python 3.7)". The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. Below the menu is a toolbar with various icons. The main area is titled "Editor - D:\PROYEK\tugasku.py". A tab labeled "tugasku.py" is selected. The code in the editor is:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 #Contoh pengulangan for sederhana
9 angka = [1,2,3,4,5]
10 for x in angka:
11     print(x)
12
13 #Contoh pengulangan for
14 buah = ["nanas", "apel", "jeruk"]
15 for makanan in buah:
16     print("Saya suka makan", makanan)
```

Figure 2.32: contoh perulangan for

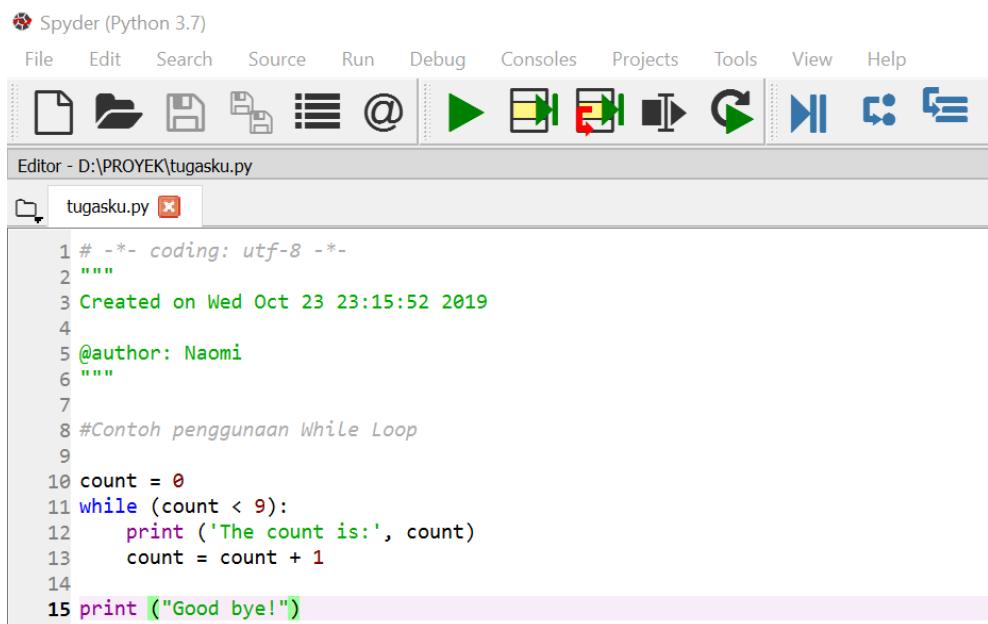
Hasil:

```
In [1]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
1
2
3
4
5
Saya suka makan nanas
Saya suka makan apel
Saya suka makan jeruk
```

Figure 2.33: hasil contoh perulang For

- Perulangan While

Pengulangan While di dalam bahasa pemrograman Python yaitu dieksekusi berkali-kali selama kondisi bernilai benar atau *True*. Contoh: Hasil:



```
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 23 23:15:52 2019
@author: Naomi
"""

#Contoh penggunaan While Loop
count = 0
while (count < 9):
    print ('The count is:', count)
    count = count + 1
print ("Good bye!")
```

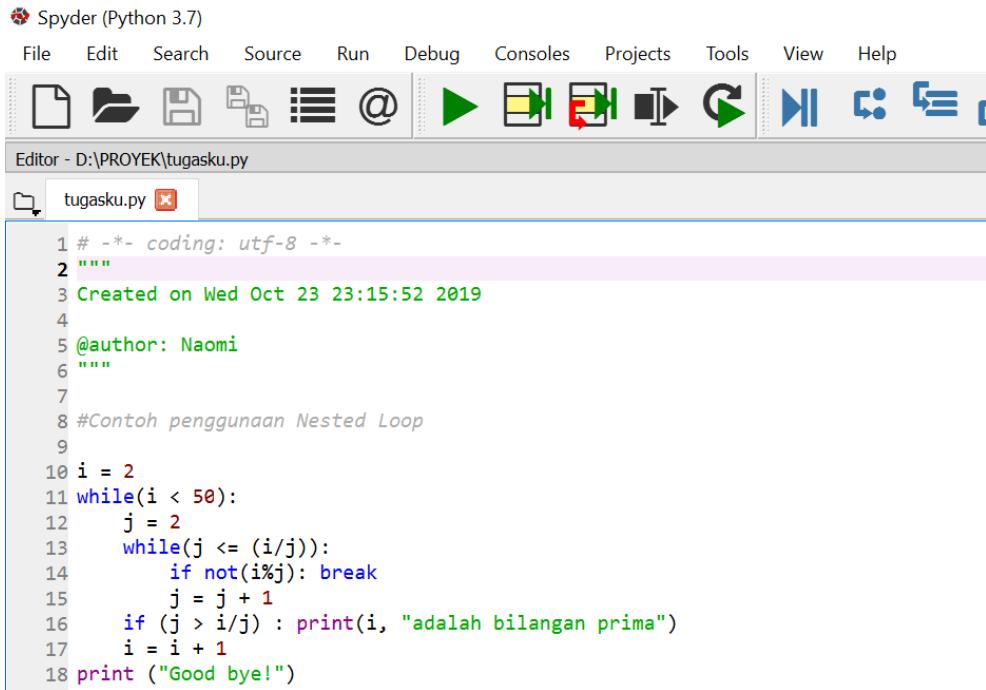
Figure 2.34: contoh perulangan while

```
In [2]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
The count is: 5
The count is: 6
The count is: 7
The count is: 8
Good bye!
```

Figure 2.35: hasil contoh perulang While

- Nasted

Nasted jarang digunakan. Contoh Nasted adalah:



The screenshot shows the Spyder Python 3.7 IDE interface. The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The toolbar has various icons for file operations and execution. The editor window title is "Editor - D:\PROYEK\tugasku.py". The code editor contains the following Python script:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 #Contoh penggunaan Nested Loop
9
10 i = 2
11 while(i < 50):
12     j = 2
13     while(j <= (i/j)):
14         if not(i%j): break
15         j = j + 1
16     if (j > i/j) : print(i, "adalah bilangan prima")
17     i = i + 1
18 print ("Good bye!")
```

Figure 2.36: contoh perulangan nasted

Hasil:

```
In [5]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
2 adalah bilangan prima
3 adalah bilangan prima
5 adalah bilangan prima
7 adalah bilangan prima
11 adalah bilangan prima
13 adalah bilangan prima
17 adalah bilangan prima
19 adalah bilangan prima
23 adalah bilangan prima
29 adalah bilangan prima
31 adalah bilangan prima
37 adalah bilangan prima
41 adalah bilangan prima
43 adalah bilangan prima
47 adalah bilangan prima
Good bye!
```

Figure 2.37: hasil contoh perulang nasted

6. Kondisi

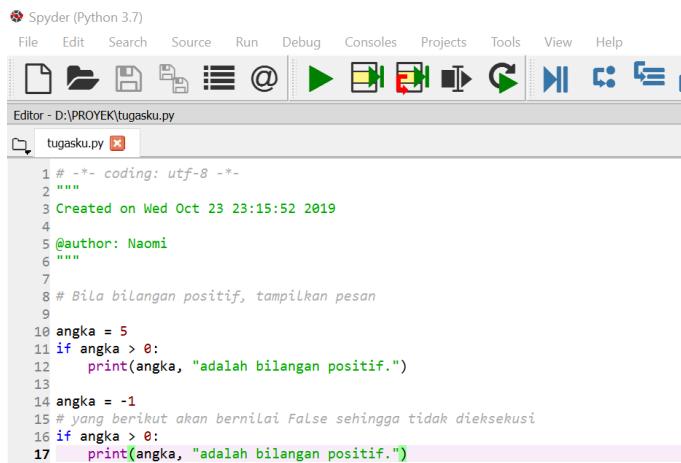
Kodisi ada 3(tiga) yaitu:

No	Pernyataan	Deskripsi
1	if	Pernyataan if terdiri dari ekspresi <i>boolean</i> diikuti oleh satu baris atau lebih pernyataan.
2	if...else	Bila pernyataan if benar, maka blok pernyataan if dieksekusi. Bila salah, maka blok pernyataan else yang dieksekusi.
3	if...elif...else	Disebut juga if bercabang. Bila ada kemungkinan beberapa kondisi bisa benar maka digunakan pernyataan if...elif atau if...elif...else

Figure 2.38: Kondisi IF

- Kondisi IF

Pernyataan if menguji satu buah kondisi. Bila hasilnya benar maka pernyataan di dalam blok if tersebut dieksekusi. Bila salah, maka pernyataan tidak dieksekusi. contoh: Hasil:



```
# Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - D:\PROYEK\tugasku.py
tugasku.py
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 # Bila bilangan positif, tampilkan pesan
9
10 angka = 5
11 if angka > 0:
12     print(angka, "adalah bilangan positif.")
13
14 angka = -1
15 # yang berikut akan bernilai False sehingga tidak dieksekusi
16 if angka > 0:
17     print(angka, "adalah bilangan positif.")
```

Figure 2.39: kondisi IF

```
In [1]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
5 adalah bilangan positif.
```

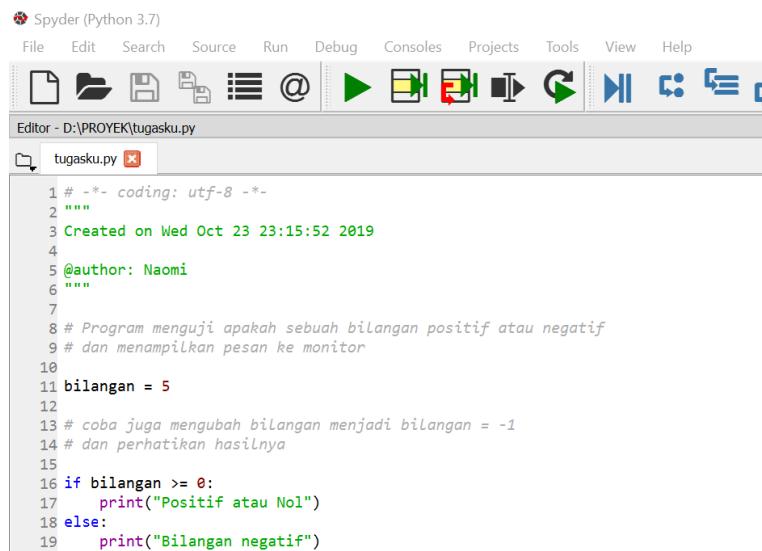
Figure 2.40: kondisi IF

Pada contoh di atas, awalnya angka berisi 5. Pada saat if yang pertama dieksekusi maka kondisinya adalah apakah 5 lebih besar dari 0? Karena hasilnya benar/True, maka statement di grup if ini dieksekusi dan menampilkan pesan 5 adalah bilangan positif.

Selanjutnya angka sudah diubah jadi -1. Untuk if yang kedua, hasil pengujian kondisinya menjadi apakah -1 lebih besar dari 0? Hasilnya salah/False. Oleh karena itu, pernyataan di dalam grupnya tidak dijalankan.

- Kondisi IF ELSE

Pernyataan if... else menguji 2 kondisi. Kondisi pertama kalau benar, dan kondisi kedua kalau salah. Contoh:



```
# coding: utf-8
"""
Created on Wed Oct 23 23:15:52 2019
@author: Naomi
"""

# Program menguji apakah sebuah bilangan positif atau negatif
# dan menampilkan pesan ke monitor
bilangan = 5
# coba juga mengubah bilangan menjadi bilangan = -1
# dan perhatikan hasilnya
if bilangan >= 0:
    print("Positif atau Nol")
else:
    print("Bilangan negatif")
```

Figure 2.41: kondisi IF ELSE

Hasil:

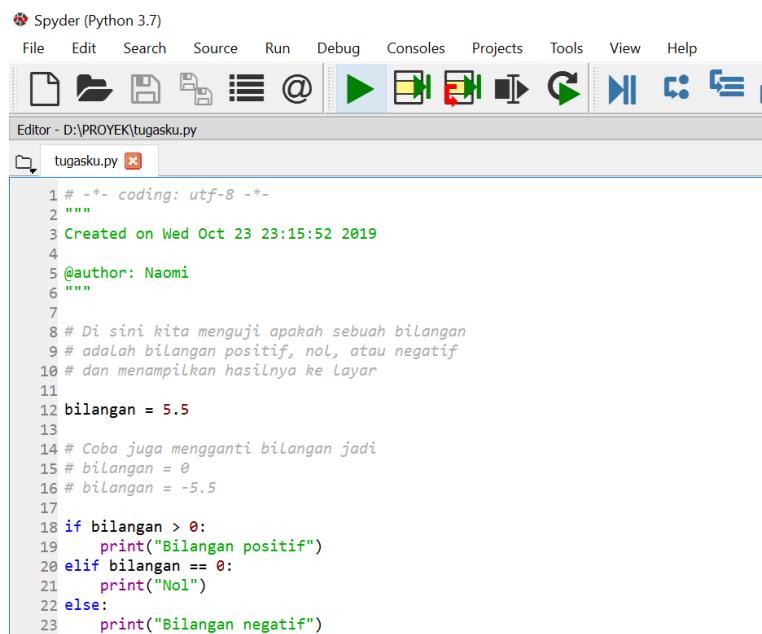
```
In [7]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
Positif atau Nol
```

Figure 2.42: kondisi IF ELSE

Pada contoh di atas, bilangan kita beri nilai 5. Kemudian pada pengujian if, kondisinya adalah apakah bilangan $i = 0$? Hasilnya adalah benar, maka hasil yang ditampilkan adalah Positif atau Nol. Seandainya kita ganti bilangan jadi -1, maka hasil pengujian if nya akan salah/False dan blok else yang akan dijalankan, yaitu menampilkan pesan Bilangan negatif

- Kondisi *IF ELIF ELSE*

Pernyataan if elif else digunakan untuk menguji lebih dari 2 kondisi. Bila kondisi pada if benar, maka pernyataan di dalamnya yang dieksekusi. Bila salah, maka masuk ke pengujian kondisi elif. Terakhir bila tidak ada if atau elif yang benar, maka yang dijalankan adalah yang di blok else. Contoh:



```

Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - D:\PROYEK\tugasku.py
tugasku.py x
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 # Di sini kita menguji apakah sebuah bilangan
9 # adalah bilangan positif, nol, atau negatif
10 # dan menampilkan hasilnya ke layar
11
12 bilangan = 5.5
13
14 # Coba juga mengganti bilangan jadi
15 # bilangan = 0
16 # bilangan = -5.5
17
18 if bilangan > 0:
19     print("Bilangan positif")
20 elif bilangan == 0:
21     print("Nol")
22 else:
23     print("Bilangan negatif")

```

Figure 2.43: kondisi IF ELIF ELSE

Hasil:

```
In [8]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
Bilangan positif
```

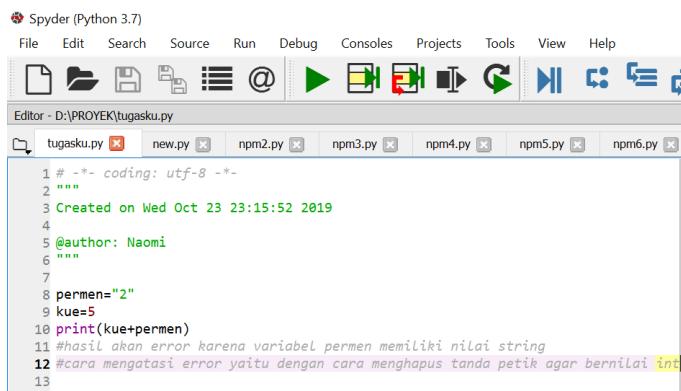
Figure 2.44: kondisi IF ELIF ELSE

Pada contoh di atas, bilangan kita beri nilai 5.5. Pada pengujian if, kondisinya adalah apakah bilangan lebih besar 0? Hasilnya benar, maka yang ditampilkan adalah pesan Bilangan positif.

Bila nilai bilangan kita ganti menjadi 0, maka yang akan bernilai benar adalah pernyataan elif. Bila kita menggantikan bilangan jadi minus, maka kondisi if dan elif salah, dan yang dijalankan adalah blok else.

- Catatan: Python mengasumsikan bahwa nilai selain nol dan selain tipe None sebagai nilai True, dan yang nilai nol dan None sebagai False.

7. Sintaks Error Contoh Sintaks yang error:



The screenshot shows the Spyder Python 3.7 IDE interface. The top menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. Below the menu is a toolbar with various icons. The main area is titled 'Editor - D:\PROYEK\tugasku.py'. There are several tabs at the top of the editor window: tugasku.py, new.py, npm2.py, npm3.py, npm4.py, npm5.py, and npm6.py. The code in the editor is as follows:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 permen="2"
9 kue=5
10 print(kue+permen)
11 #hasil akan error karena variabel permen memiliki nilai string
12 #cara mengatasi error yaitu dengan cara menghapus tanda petik agar bernilai int
13
```

Figure 2.45: Contoh Sintaks Error

Hasil:

```
In [13]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
Traceback (most recent call last):

  File "<ipython-input-13-44f63cb8d48c>", line 1, in <module>
    runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')

  File "C:\Users\Naomi\Anaconda3ok\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 704, in runfile
    execfile(filename, namespace)

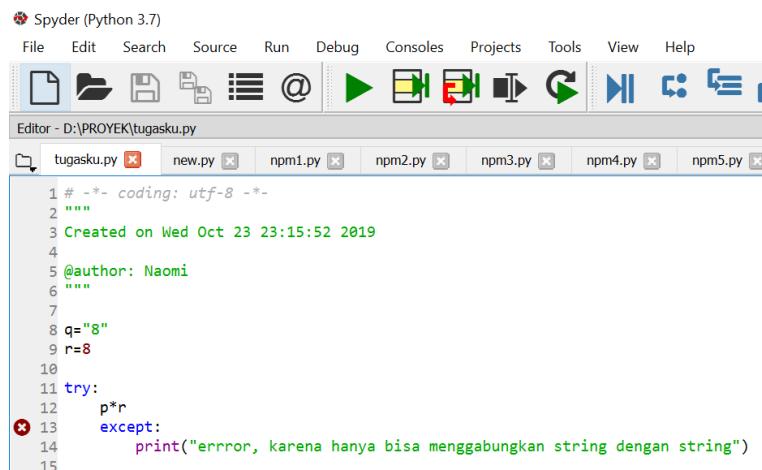
  File "C:\Users\Naomi\Anaconda3ok\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 108, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

  File "D:/PROYEK/tugasku.py", line 11, in <module>
    print(kue+permen)

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Figure 2.46: Hasil Contoh Sintaks Error

8. Try Except



The screenshot shows the Spyder Python 3.7 IDE interface. The title bar says "Spyder (Python 3.7)". The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. Below the menu is a toolbar with various icons. The main area is an "Editor" window titled "Editor - D:\PROYEK\tugasku.py". The code in the editor is:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 q="8"
9 r=8
10
11 try:
12     p*r
13 except:
14     print("error, karena hanya bisa menggabungkan string dengan string")
15
```

A red circular icon with a white exclamation mark is positioned over the line "try:" at line 11, indicating a syntax error.

Figure 2.47: Contoh Try Except

Hasil:

```
In [9]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
Traceback (most recent call last):

  File "C:\Users\Naomi\Anaconda3ok\lib\site-packages\IPython\core\interactiveshell.py", line 3267, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)

  File "<ipython-input-9-44f63cb8d48c>", line 1, in <module>
    runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')

  File "C:\Users\Naomi\Anaconda3ok\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 704, in runfile
    execfile(filename, namespace)

  File "C:\Users\Naomi\Anaconda3ok\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 108, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

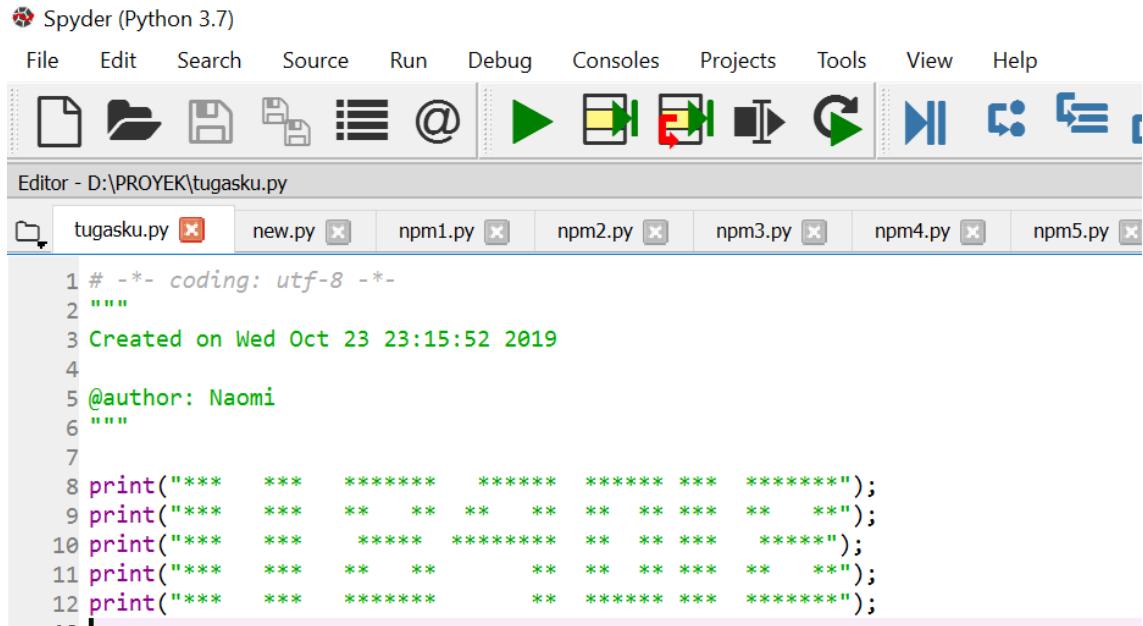
  File "D:/PROYEK/tugasku.py", line 10
    try
        ^
SyntaxError: invalid syntax
```

Figure 2.48: Hasil Contoh Try Except

2.2 Keterampilan Pemrograman

1. Membuat NPM menggunakan tanda (*)

Contoh Codingan:



The screenshot shows the Spyder Python 3.7 IDE interface. The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The toolbar has various icons for file operations like Open, Save, and Run. The main window shows an editor titled "Editor - D:\PROYEK\tugasku.py". Below the title bar are tabs for "tugasku.py", "new.py", "npm1.py", "npm2.py", "npm3.py", "npm4.py", and "npm5.py". The code in the editor is as follows:

```
1 # -*- coding: utf-8 -*-
"""
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 print("***   ***   *****   *****   ***** ***   *****");
9 print("***   ***   **  **  **  **  **  ** ***   **  **");
10 print("***   ***   ****   *****   **  **  **  **   ****");
11 print("***   ***   **  **       **  **  **  **   **  **");
12 print("***   ***   *****       **  *****   ***   *****");
13
```

Figure 2.49: Contoh NPM

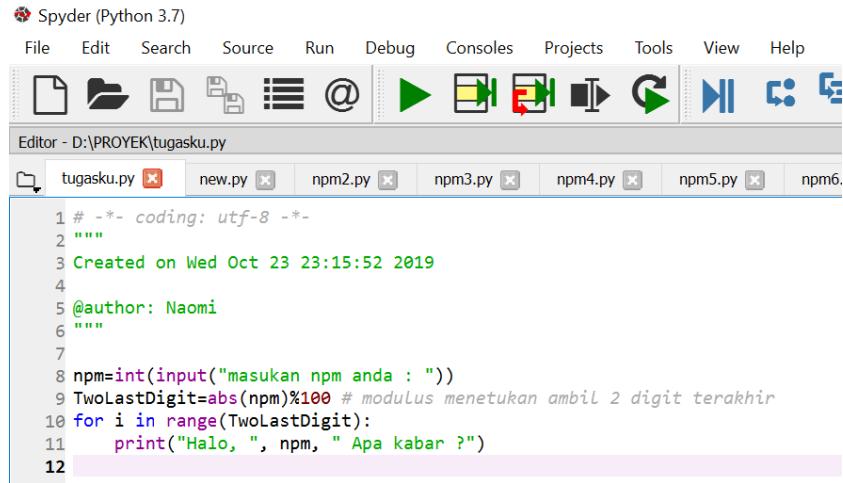
Hasil:

```
In [12]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
***   ***   *****   *****   ***** ***   *****
***   ***   **  **  **  **  **  ** ***   **  **
***   ***   ****   *****   **  **  **  **   ****
***   ***   **  **       **  **  **  **   **  **
***   ***   *****       **  *****   ***   *****
```

Figure 2.50: Hasil Contoh NPM

2. Membuat Program Hello Word dengan input NPM

Contoh Codingan: Hasil:



The screenshot shows the Spyder Python 3.7 IDE interface. The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The toolbar has icons for file operations like Open, Save, and Run. The code editor window title is 'Editor - D:\PROYEK\tugasku.py'. Below the title bar are tabs for 'tugasku.py', 'new.py', 'npm2.py', 'npm3.py', 'npm4.py', 'npm5.py', and 'npm6.'. The code in 'tugasku.py' is as follows:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 npm=int(input("masukan npm anda : "))
9 TwoLastDigit=abs(npm)%100 # modulus menentukan ambil 2 digit terakhir
10 for i in range(TwoLastDigit):
11     print("Halo, ", npm, " Apa kabar ?")
12 
```

Figure 2.51: Contoh Hello Word dengan input NPM

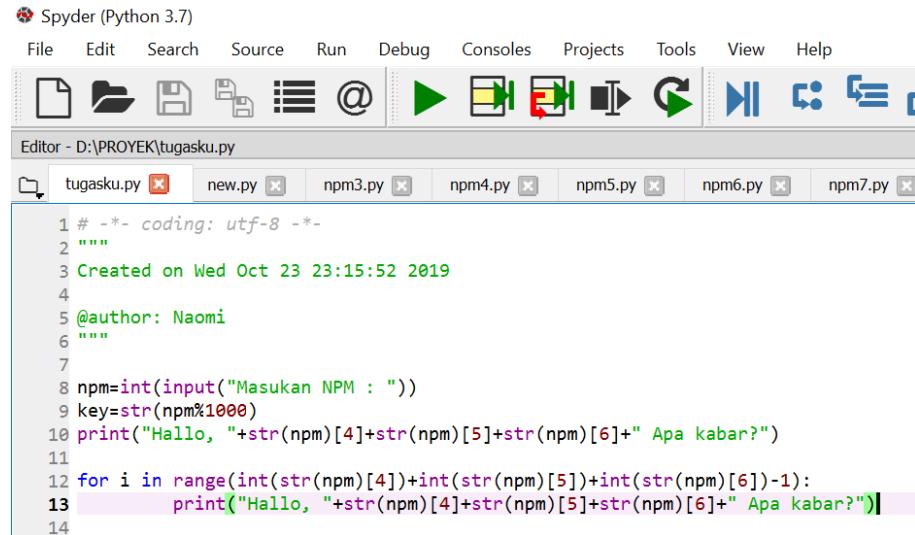
```
In [16]: runfile('D:/PROYEK/npm2.py', wdir='D:/PROYEK')
```

```
masukan npm anda : 1184018
Halo, 1184018 Apa kabar ?
```

Figure 2.52: Hasil Contoh Hello Word dengan input NPM

3. Program Hello Word menggunakan Input dan Ouput

Contoh Codingan:



```
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 23 23:15:52 2019
@author: Naomi
"""

npm=int(input("Masukan NPM : "))
key=str(npm%1000)
print("Hallo, "+str(npm)[4]+str(npm)[5]+str(npm)[6]+" Apa kabar?")
for i in range(int(str(npm)[4])+int(str(npm)[5])+int(str(npm)[6])-1):
    print("Hallo, "+str(npm)[4]+str(npm)[5]+str(npm)[6]+" Apa kabar?")
```

Figure 2.53: Contoh Hello Word Input Ouput

Hasil:

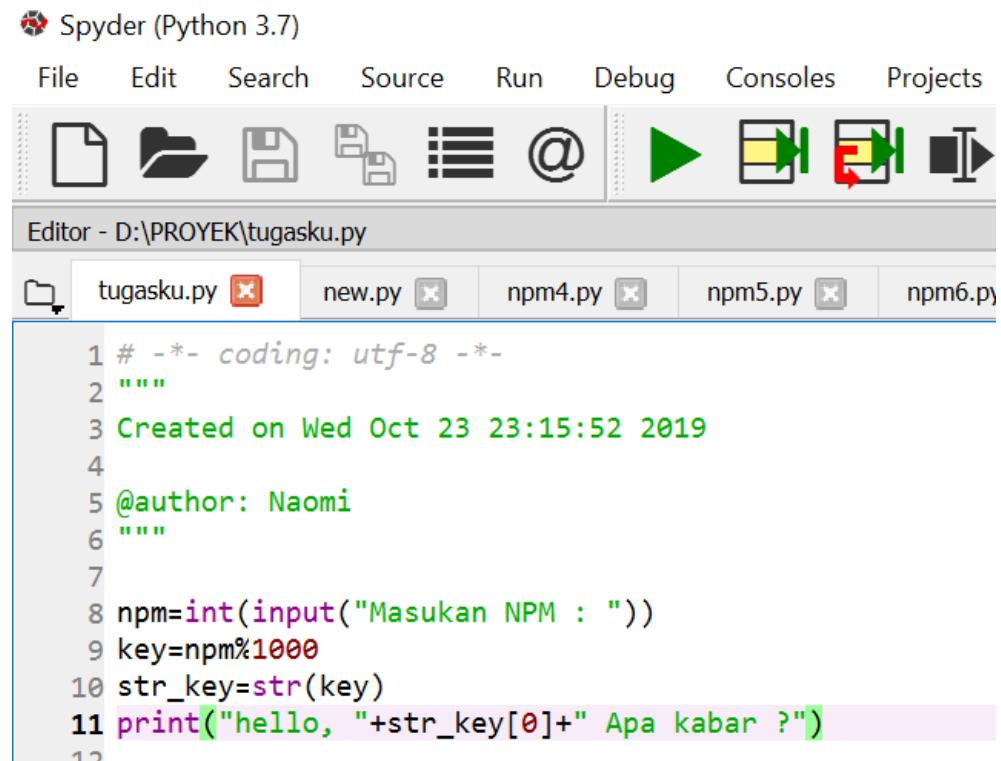
```
In [17]: runfile('D:/PROYEK/npm3.py', wdir='D:/PROYEK')

Masukan NPM : 1184018
Hallo, 018 Apa kabar?
```

Figure 2.54: Hasil Contoh Hello Input Ouput

4. Program Hello Word dengan input nama yang disimpan dalam sebuah variabel string bernama NPM dan beri luaran output berupa digit ketiga dari belakang dari variabel NPM.

Contoh Codingan:



The screenshot shows the Spyder Python IDE interface. The title bar says "Spyder (Python 3.7)". The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, and Projects. Below the menu is a toolbar with icons for file operations like Open, Save, and Run. The main window shows an "Editor - D:\PROYEK\tugasku.py" tab. In the tabs below, "tugasku.py" is active, followed by new.py, npm4.py, npm5.py, and npm6.py. The code in the editor is:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 npm=int(input("Masukan NPM : "))
9 key=npm%1000
10 str_key=str(key)
11 print("hello, "+str_key[0]+" Apa kabar ?")
12
```

Figure 2.55: Contoh Ouput digit Ketiga

Hasil:

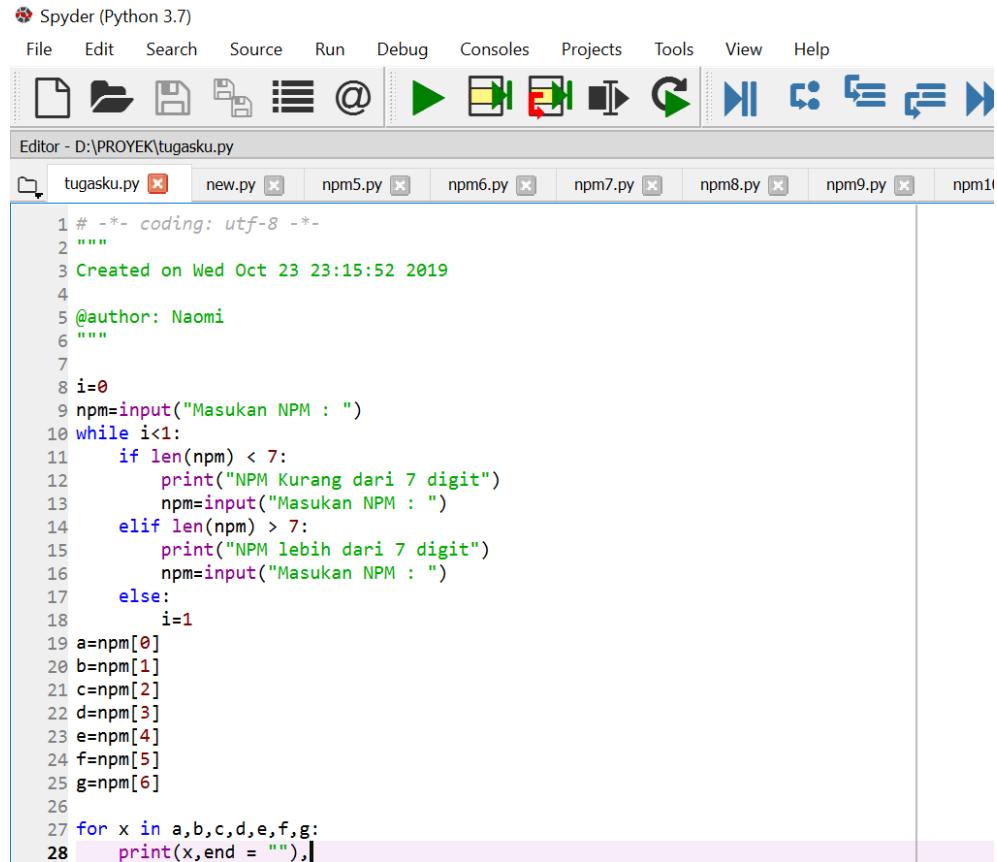
```
In [19]: runfile('D:/PROYEK/npm4.py', wdir='D:/PROYEK')

Masukan NPM : 1184018
hello, 1 Apa kabar ?
```

Figure 2.56: Hasil Contoh Ouput digit Ketiga

5. Menggunakan Perulangan dan Kondisi

Contoh Codingan:



The screenshot shows the Spyder Python 3.7 IDE interface. The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. Below the menu is a toolbar with various icons for file operations like Open, Save, and Run. The main window has tabs for 'tugasku.py' (active), 'new.py', 'npm5.py', 'npm6.py', 'npm7.py', 'npm8.py', 'npm9.py', and 'npm1'. The code editor contains the following Python script:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 i=0
9 npm=input("Masukan NPM : ")
10 while i<1:
11     if len(npm) < 7:
12         print("NPM Kurang dari 7 digit")
13         npm=input("Masukan NPM : ")
14     elif len(npm) > 7:
15         print("NPM lebih dari 7 digit")
16         npm=input("Masukan NPM : ")
17     else:
18         i=1
19 a=npm[0]
20 b=npm[1]
21 c=npm[2]
22 d=npm[3]
23 e=npm[4]
24 f=npm[5]
25 g=npm[6]
26
27 for x in a,b,c,d,e,f,g:
28     print(x,end = ""),
```

Figure 2.57: Contoh menggunakan Perulangan dan Kondisi

Hasil:

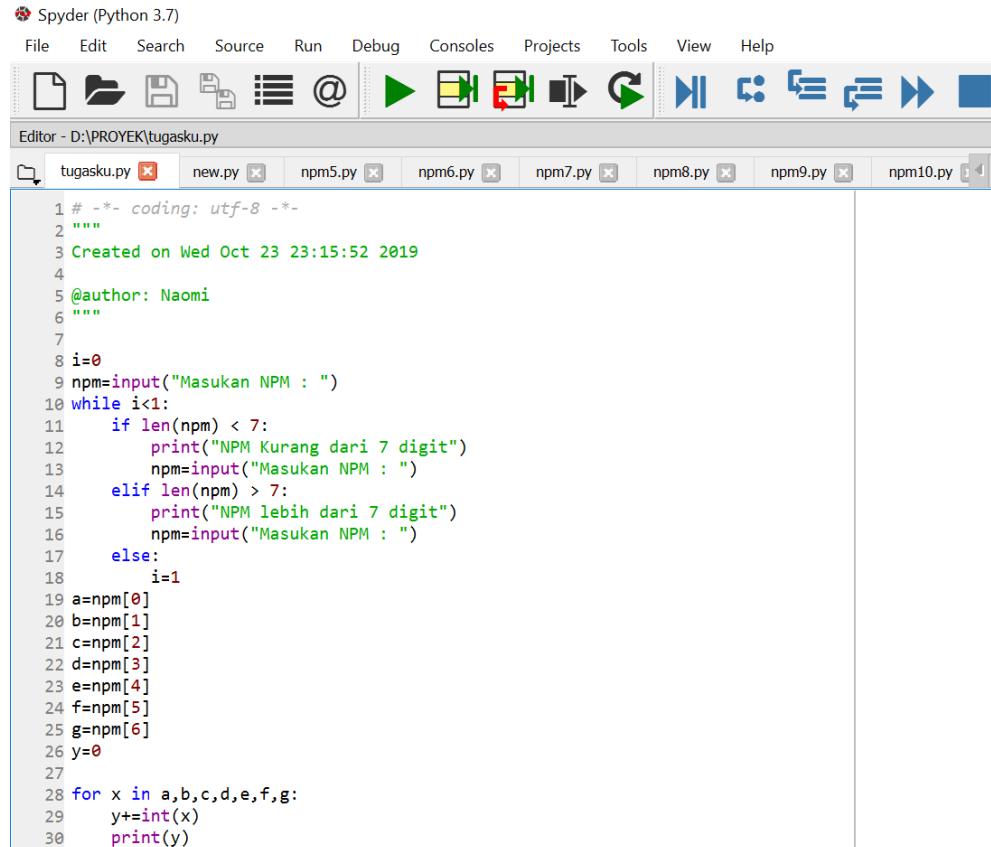
```
In [18]: runfile('D:/PROYEK/new.py', wdir='D:/PROYEK')
```

```
Masukan NPM : 1184018
1
1
8
4
0
1
8
```

Figure 2.58: Hasil Contoh menggunakan Perulangan dan Kondisi

6. penjumlahan dari seluruh variabel

Contoh Codingan:



```
# -*- coding: utf-8 -*-  
#  
# Created on Wed Oct 23 23:15:52 2019  
# @author: Naomi  
  
i=0  
npm=input("Masukan NPM : ")  
while i<1:  
    if len(npm) < 7:  
        print("NPM Kurang dari 7 digit")  
        npm=input("Masukan NPM : ")  
    elif len(npm) > 7:  
        print("NPM lebih dari 7 digit")  
        npm=input("Masukan NPM : ")  
    else:  
        i=1  
a=npm[0]  
b=npm[1]  
c=npm[2]  
d=npm[3]  
e=npm[4]  
f=npm[5]  
g=npm[6]  
y=0  
for x in a,b,c,d,e,f,g:  
    y+=int(x)  
print(y)
```

Figure 2.59: Contoh penjumlahan dari seluruh variabel

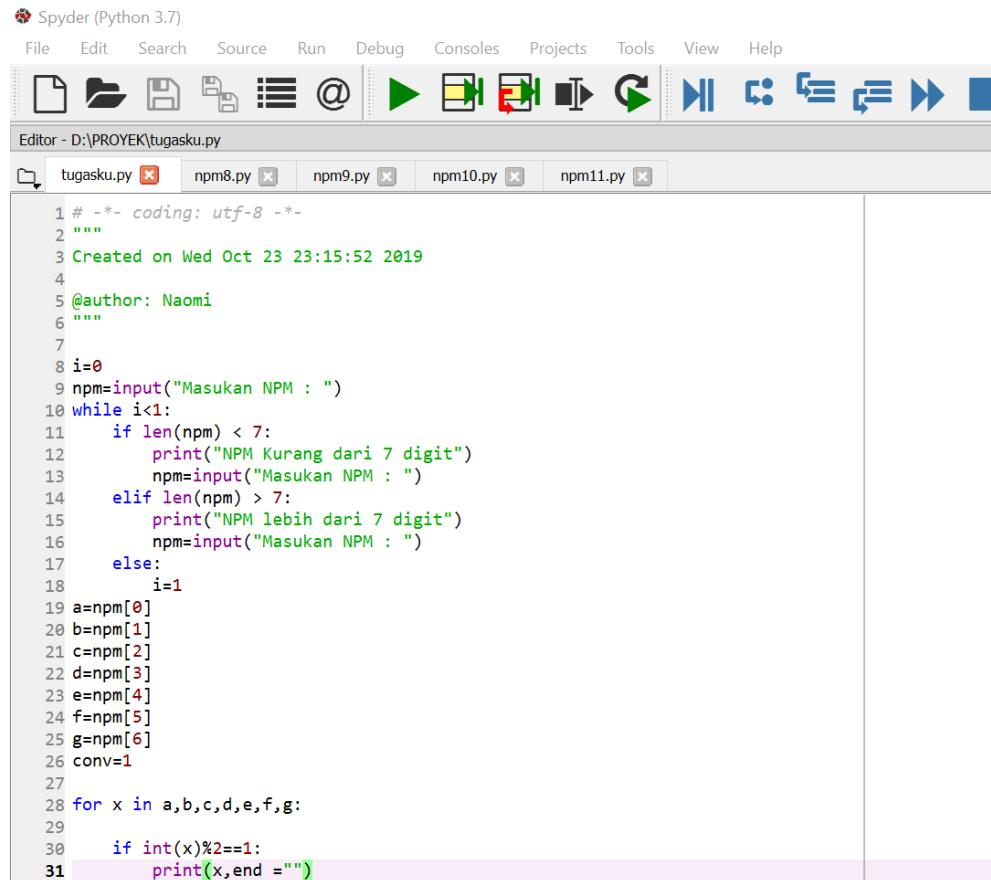
Hasil:

```
In [21]: runfile('D:/PROYEK/npm6.py', wdir='D:/PROYEK')  
  
Masukan NPM : 1184018  
1  
2  
10  
14  
14  
15  
23
```

Figure 2.60: HasilContoh penjumlahan dari seluruh variabel

7. Bilangan Ganjil

Contoh Codingan:



The screenshot shows the Spyder Python 3.7 IDE interface. The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The toolbar has various icons for file operations like Open, Save, and Run. The tabs at the top show 'tugasku.py' (active), 'npm8.py', 'npm9.py', 'npm10.py', and 'npm11.py'. The code editor contains the following Python script:

```
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 23 23:15:52 2019
@author: Naomi
"""

i=0
npm=input("Masukan NPM : ")
while i<1:
    if len(npm) < 7:
        print("NPM Kurang dari 7 digit")
        npm=input("Masukan NPM : ")
    elif len(npm) > 7:
        print("NPM lebih dari 7 digit")
        npm=input("Masukan NPM : ")
    else:
        i=1
a=npm[0]
b=npm[1]
c=npm[2]
d=npm[3]
e=npm[4]
f=npm[5]
g=npm[6]
conv=1
for x in a,b,c,d,e,f,g:
    if int(x)%2==1:
        print(x,end = "")
```

Figure 2.61: Contoh Bilangan Ganjil

Hasil:

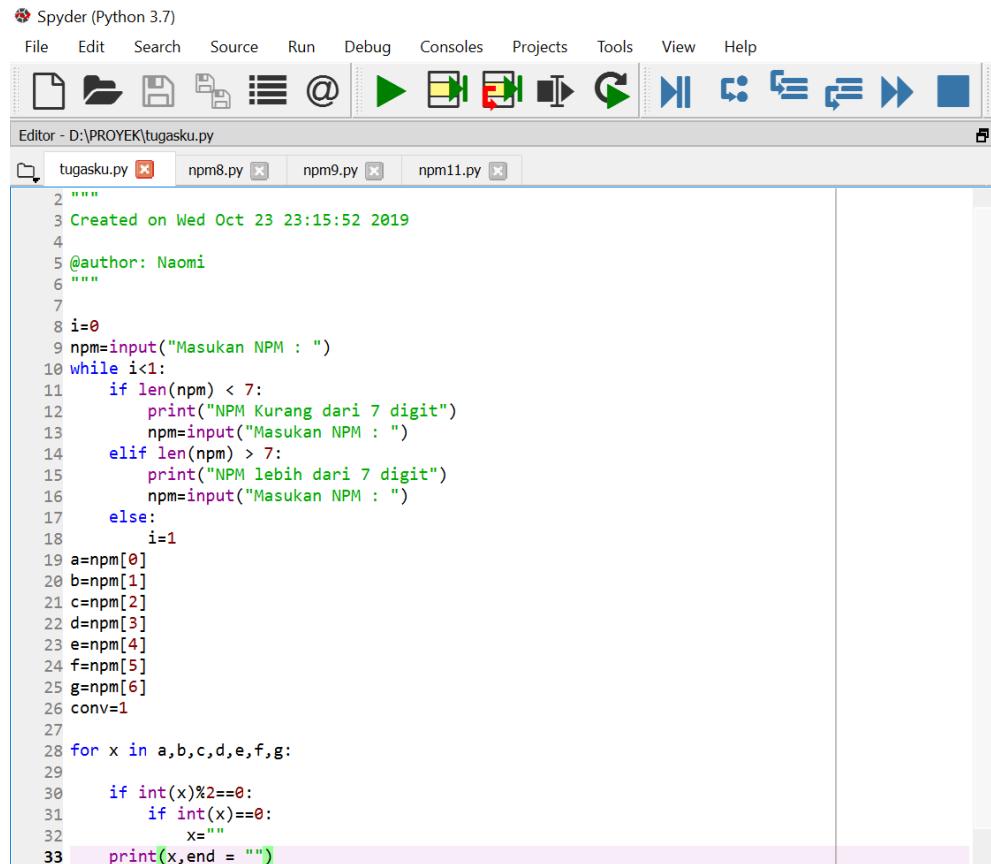
```
In [32]: runfile('D:/PROYEK/npm10.py', wdir='D:/PROYEK')

Masukan NPM : 1184018
111
```

Figure 2.62: Hasil Contoh Bilangan Ganjil

8. Bilangan Genap

Contoh Codingan:



The screenshot shows the Spyder Python 3.7 IDE interface. The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The toolbar has various icons for file operations like Open, Save, and Run. The editor tab bar shows 'tugasku.py' as the active file, along with npm8.py, npm9.py, and npm11.py. The code in 'tugasku.py' is as follows:

```
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 i=0
9 npm=input("Masukan NPM : ")
10 while i<1:
11     if len(npm) < 7:
12         print("NPM Kurang dari 7 digit")
13         npm=input("Masukan NPM : ")
14     elif len(npm) > 7:
15         print("NPM lebih dari 7 digit")
16         npm=input("Masukan NPM : ")
17     else:
18         i=1
19 a=npm[0]
20 b=npm[1]
21 c=npm[2]
22 d=npm[3]
23 e=npm[4]
24 f=npm[5]
25 g=npm[6]
26 conv=1
27
28 for x in a,b,c,d,e,f,g:
29
30     if int(x)%2==0:
31         if int(x)==0:
32             x=""
33     print(x,end = "")
```

Figure 2.63: Contoh Bilangan Genap

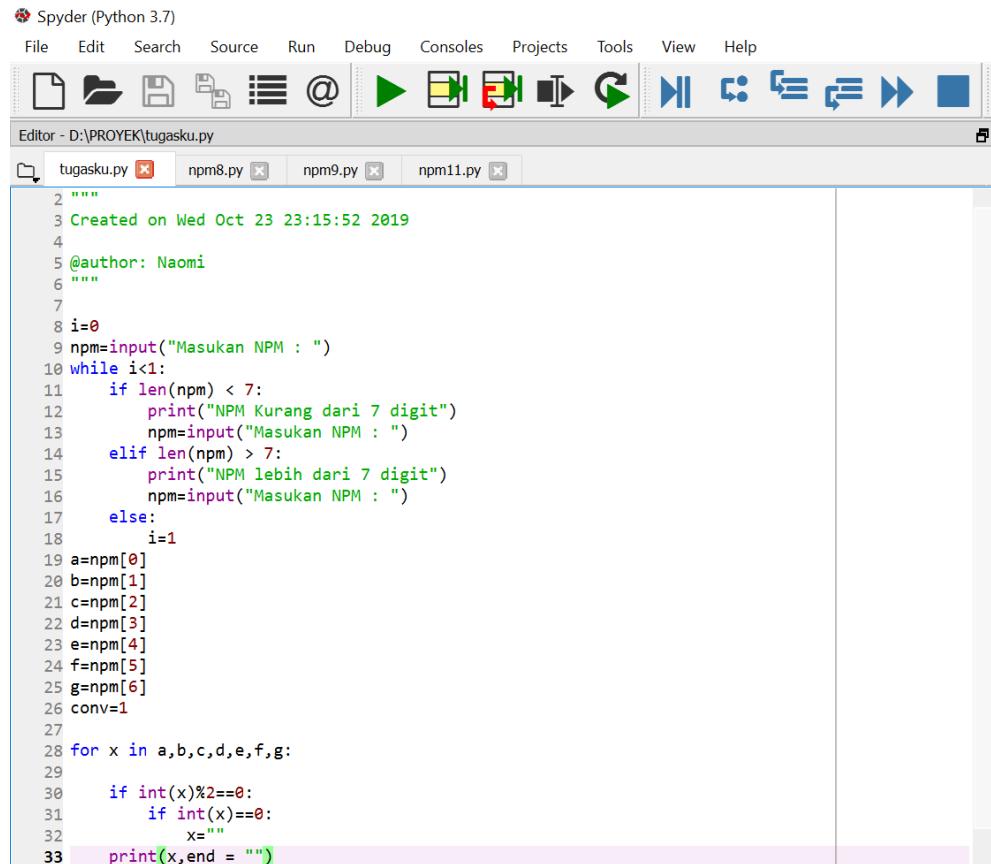
Hasil:

```
In [23]: runfile('D:/PROYEK/npm9.py', wdir='D:/PROYEK')
Masukan NPM : 1184018
118418
```

Figure 2.64: Hasil Contoh Bilangan Genap

9. Bilangan Genap

Contoh Codingan:



The screenshot shows the Spyder Python 3.7 IDE interface. The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The toolbar has various icons for file operations like Open, Save, and Run. The editor tab bar shows 'tugasku.py' as the active file, along with npm8.py, npm9.py, and npm11.py. The code in 'tugasku.py' is as follows:

```
2 """
3 Created on Wed Oct 23 23:15:52 2019
4
5 @author: Naomi
6 """
7
8 i=0
9 npm=input("Masukan NPM : ")
10 while i<1:
11     if len(npm) < 7:
12         print("NPM Kurang dari 7 digit")
13         npm=input("Masukan NPM : ")
14     elif len(npm) > 7:
15         print("NPM lebih dari 7 digit")
16         npm=input("Masukan NPM : ")
17     else:
18         i=1
19 a=npm[0]
20 b=npm[1]
21 c=npm[2]
22 d=npm[3]
23 e=npm[4]
24 f=npm[5]
25 g=npm[6]
26 conv=1
27
28 for x in a,b,c,d,e,f,g:
29
30     if int(x)%2==0:
31         if int(x)==0:
32             x=""
33     print(x,end = "")
```

Figure 2.65: Contoh Bilangan Genap

Hasil:

```
In [23]: runfile('D:/PROYEK/npm9.py', wdir='D:/PROYEK')
Masukan NPM : 1184018
118418
```

Figure 2.66: Hasil Contoh Bilangan Genap

10. Bilangan Prima

Contoh Codingan:

The screenshot shows the Spyder Python IDE interface. On the left, the code editor displays a file named 'tugasku.py' with the following content:

```
4
5 #author: Naomi
6 """
7 i=0
8 Npm = input("Npm : ")
9 while i<1:
10     if len(Npm)<7:
11         print("Npm Kurang dari 7!")
12         Npm = input("NPM : ")
13     elif len(Npm)>7:
14         print("Npm lebih dari 7!")
15         Npm = input("Npm : ")
16     else:
17         i=1
18 A=Npm[0]
19 B=Npm[1]
20 C=Npm[2]
21 D=Npm[3]
22 E=Npm[4]
23 F=Npm[5]
24 G=Npm[6]
25
26
27 X=1
28 for X in A,B,C,D,E,F,G:
29     if int(X) > 1:
30         for i in range(2,int(X)):
31             if (int(X) % i) == 0:
32                 break
33             else:
34                 print(int(X),end = ""),
```

On the right, the IPython console window shows the execution of the script. It starts with the Python version information and then runs the script. The output shows the input 'Npm : 1184018' and the resulting prime number check output.

```
In [1]: runfile('D:/PROYEK/tugasku.py', wdir='D:/PROYEK')
Npm : 1184018
In [2]:
```

Figure 2.67: Contoh Bilangan Prima