

Laporan Tugas

Instalasi dan Penggunaan

Anaconda



Dinda Majesty
NPM : 1.18.4.011

Informatics Research Center
Applied Bachelor Program of Informatics Engineering
Bandung 2019

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

Acknowledgements

Pertama-tama kami panjatkan puji dan syukur kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga Laporan Tugas Instalasi dan Penggunaan Anaconda ini dapat diselesaikan.

Abstract

Laporan Tugas Instalasi dan Penggunaan Anaconda ini dibuat dengan tujuan memenuhi tugas matakuliah Pemrograman II. Pada intinya Laporan ini menjawab secara lengkap pertanyaan yang terkait dengan silabus matakuliah Pemrograman II. Di dalamnya memuat cara instalasi anaconda hingga cara menggunakan anaconda spyder dan dijelaskan sedikit mengenai error (error indentasi).

Contents

1	Mengenal Python dan Anaconda	1
1.1	Teori	1
1.1.1	Sejarah Python	1
1.1.2	Perbedaan Python 2 dan 3	1
1.1.3	Implementasi dan Penggunaan Python di Perusahaan Dunia	2
1.2	Instalasi	3
1.2.1	Instalasi Anaconda 3	3
1.2.2	Instalasi Pip	7
1.2.3	Setting Environment	9
1.2.4	Command Line Interface/Interpreter	11
1.2.5	Update Anaconda dan Spyder	12
1.2.6	Run Script Hello World di Spyder	12
1.2.7	Automatic Login SIAP	13
1.2.8	Pemakaian Variable Explorer	16
1.3	Indentasi	17
1.3.1	Penjelasan Indentasi	17
1.3.2	Jenis-Jenis Error Indentasi	17
1.3.3	Cara Membaca Error	18
1.3.4	Cara Menangani Error	19
1.3.5	Link Youtube	20
2	Mengenal Python dan Anaconda	21
2.1	Teori	21
2.1.1	Jenis-Jenis Variabel	21
2.1.2	Input dan Output	22
2.1.3	Operator Aritmatika dan Konversi Tipe Data	22
2.1.4	Perulangan	23
2.1.5	IF Statement	24

2.1.6	Error	25
2.1.7	Try Except	27
2.2	Keterampilan Pemrograman	27
2.2.1	Modulus	27
2.2.2	Hello NPM	27
2.2.3	Hello NPM (3 Digit Belakang)	28
2.2.4	Hello NPM (Digit ke-3)	28
2.2.5	Variabel Alfabet	28
2.2.6	Penjumlahan NPM	28
2.2.7	Perkalian NPM	29
2.2.8	Print Vertical	29
2.2.9	Digit Genap NPM	29
2.2.10	Digit Ganjil NPM	29
2.2.11	Bilangan Prima NPM	30
2.3	Keterampilan Penanganan Error	30
2.3.1	Peringatan Error dan Cara Mengatasinya	30
2.3.2	Percobaan Try Except	31

List of Figures

1.1	Perbedaan Python 2 dan Python 3	2
1.2	Run Setup Anaconda	4
1.3	Setup Loading	4
1.4	Welcome to Anaconda Setup	4
1.5	<i>License Agreement</i>	5
1.6	<i>Just Me(recomended)</i>	5
1.7	<i>Pilih lokasi</i>	5
1.8	<i>Centang Anaconda to my PATH</i>	6
1.9	<i>Installation Complete</i>	6
1.10	<i>Installation Complete</i>	6
1.11	<i>Anaconda+JetBrains</i>	7
1.12	<i>Thanks for install Anaconda</i>	7
1.13	<i>Install pip</i>	8
1.14	<i>Install pip Selesai</i>	8
1.15	<i>Melihat Versi pip</i>	9
1.16	<i>Properties</i>	9
1.17	<i>Advanced system settings</i>	9
1.18	<i>Environment Variables</i>	10
1.19	<i>Path</i>	10
1.20	<i>Edit Environment Variable</i>	11
1.21	<i>CLI in Command Prompt</i>	11
1.22	<i>CLI in Command Prompt</i>	12
1.23	<i>Print Hello World</i>	13
1.24	<i>Hello World</i>	13
1.25	<i>Automatic Login SIAP</i>	14
1.26	<i>Hasil Running</i>	14
1.27	<i>Automatic Input dan Login</i>	15
1.28	<i>Automatic Input dan Login Selesai</i>	16

1.29	<i>Variable Explorer</i>	16
1.30	<i>Indentasi</i>	18
1.31	<i>Error Indentasi</i>	18
1.32	<i>Error</i>	19
1.33	<i>Syntax Error</i>	19
1.34	<i>Syntax yang Telah Diperbaiki</i>	20
2.1	<i>Try Except</i>	31

Listings

2.1	Input dan Output	22
2.2	While Loop	23
2.3	For Loop	23
2.4	Nested Loop	23
2.5	if Statement	24
2.6	Elif	24
2.7	Else	25
2.8	Nested If	25
2.9	Try Except	27
2.10	Modulus	27
2.11	Hello NPM	28
2.12	3 Digit Belakang	28
2.13	Digit ke-3	28
2.14	Variabel Alfabet	28
2.15	Penjumlahan NPM	28
2.16	Perkalian NPM	29
2.17	Print Vertical	29
2.18	Digit Genap NPM	29
2.19	Digit Ganjil NPM	29
2.20	Bilangan Prima NPM	30
2.21	Bilangan Prima NPM	31

Chapter 1

Mengenal Python dan Anaconda

1.1 Teori

1.1.1 Sejarah Python

Nama python berasal dari acara televisi Monty python's flying circus. Python merupakan bahasa pemrograman yang dikembangkan oleh Guido Van Rossum pada tahun 1990 di CWI, Amsterdam. bahasa ini merupakan lanjutan dari bahasa pemrograman ABC. pada tahun 1995, Guido pindah ke CNRI dan mengeluarkan python versi 1.6. pada tahun 2000, Guido pindah ke BeOpen dan mengeluarkan python versi 2.0 setelah itu Guido dan tim PythonLabs pindah ke DigitalCreations. saat ini Guido dan Python Software Foundation terus melakukan perkembangan hingga python versi 2.6.1 dan python versi 3.0 Python Software Foundation merupakan sebuah organisasi yang memiliki hak atas bahasa pemrograman python, hal ini dilakukan untuk mencegah bahasa pemrograman python dimiliki oleh perusahaan komersial.

1.1.2 Perbedaan Python 2 dan 3

Perbedaan Python 2 dan Python 3 dapat dilihat pada gambar berikut.

No.	Perbedaan	Python 2	Python 3
1.	Syntax Print	Print tanpa kurung dan menggunakan kurung dapat dijalankan Print"tanpa kurung" Print("dengan kurung")	Print wajib menggunakan kurung, print tanpa kurung menyebabkan error Print("wajib pakai kurung")
2.	Syntax Input	Menggunakan raw_input Nama = raw_input("Masukkan nama anda: ") Menggunakan input akan menyebabkan error	Menggunakan input Nama = input("Masukkan nama anda") Menggunakan raw_input akan menyebabkan error
3.	Hasil Operator Pembagian	3/2 = 1 3//2 = 1 3/2.0 = 1.5 3//2.0 = 1.0 Apabila tipe data float maka akan menghasilkan float, jika tidak maka akan menghasilkan integer	3/2 = 1.5 3//2 = 1 3/2.0 = 1.5 3//2.0 = 1.0 Tipe data float ataupun integer tidak memiliki pengaruh kepada hasil.

Figure 1.1: Perbedaan Python 2 dan Python 3

1.1.3 Implementasi dan Penggunaan Python di Perusahaan Dunia

1. spotify

spotify adalah suatu layanan musik streaming yang menggunakan pemrograman python untuk analisis data dan backend. pada backend spotify berkomunikasi dengan OMQ. OMQ itu sendiri adalah suatu framework dan library open source untuk networking. untuk analisis data tersebut spotify menggunakan luigi, dan modul python yang sinkron dengan hadoop.

2. Google

Google ini sudah menggunakan bahasa pemrograman python ini sudah sejak dari awal berdirinya. Dan pada saat ini bahasa pemrograman python merupakan salah satu bahasa pemrograman server-side resmi di google. Meskipun ada script yang ditulis untuk google menggunakan bahasa perl dan bash, maka nantinya script tersebut akan diubah ke python terlebih dahulu, karena kemudahan dalam perawatannya.

3. Industrial Light and Magic

Industrial Light and Magic ini merupakan studio special efek yang dibutuhkan untuk film star wars saja. Karena infrastruktur awal industrial light and magic ini menggunakan C dan C++, maka akan lebih mudah mengintegrasikan bahasa pemrograman python ketimbang bahasa pemrograman lainnya. Dengan menggunakan bahasa pemrograman python ini industrial light and magic dengan mudah membungkus komponen software dan dapat meningkatkan aplikasi grafisnya.

4. Netflix

Netflix adalah suatu layanan pemutaran film yang dapat dilakukan oleh pengguna dimanapun dan kapanpun. Pada netflix bahasa pemrograman yang digunakan adalah bahasa pemrograman python, bahasa pemrograman ini digunakan pada Central Alert Gateway yang akan me-reroute alert dan mengirimkannya pada individu yang akan melihatnya serta juga dapat secara otomatis reboot atau menghentikan proses yang dianggap bermasalah. Selain itu python juga digunakan untuk menelusuri riwayat dan perubahan pengaturan keamanan.

5. instagram

Instagram adalah suatu aplikasi mobile berbasis IOS, android dan windows phone, dimana pengguna dapat berbagi foto dan video melalui instagram ini. Pada instagram ini menggunakan bahasa pemrograman python dalam task queue-nya atau fitur dimana setiap pengguna dapat berbagi foto atau video ke beberapa social network lainnya seperti facebook, twitter, dan lain-lainnya.

1.2 Instalasi

1.2.1 Instalasi Anaconda 3

Hal yang harus diperhatikan sebelum melakukan instalasi *Anaconda Python*

1. Perhatikan versi dari sistem operasi yang digunakan (versi 32bit atau 64bit)
2. Download file anaconda yang sesuai dengan versi sistem operasi (32bit atau 64bit)
3. *Download Anaconda Python* <https://www.anaconda.com/distribution/>

Berikut langkah-langkah instalasi anaconda.

1. Buka aplikasi *installer Anaconda* tersebut lalu akan muncul gambar *installer anaconda*.

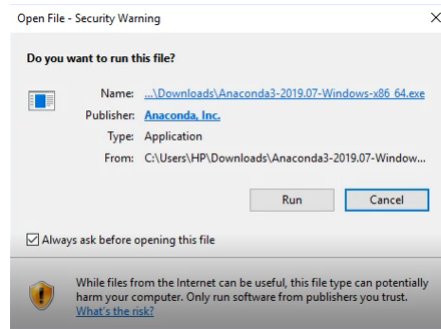


Figure 1.2: Run Setup Anaconda

2. Tunggu hingga *setup loading* selesai

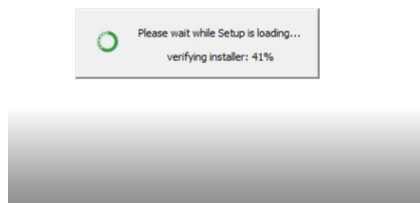


Figure 1.3: Setup Loading

3. Jika *setup loading* telah selesai, maka klik *next*

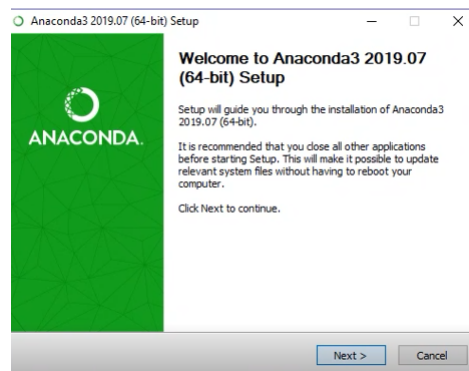


Figure 1.4: Welcome to Anaconda Setup

4. Pada *License Agreement* klik *I Agree* gambar *License Agreement*.

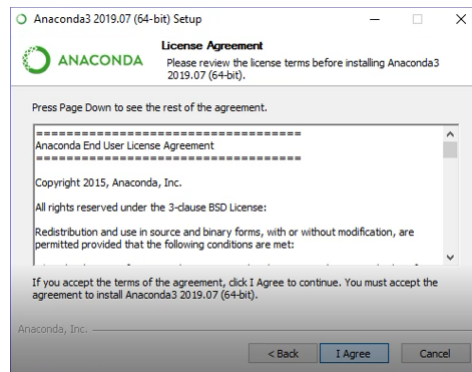


Figure 1.5: *License Agreement*

5. Kemudian pilih *Just Me(Recomended)* agar sesuai dengan komputer yang digunakan, kemudian klik *next* gambar *Just Me(recomended)*.

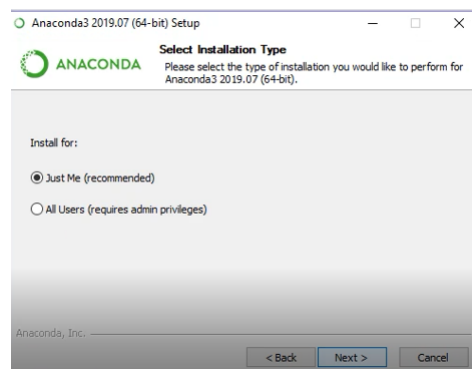


Figure 1.6: *Just Me(recomended)*

6. Kemudian pilih lokasi tempat *menginstall anaconda* gambar *Pilih lokasi*.

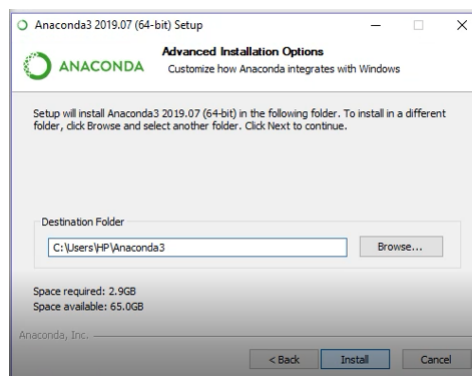


Figure 1.7: *Pilih lokasi*

7. Kemudian centang *Add Anaconda to my Path environment variable*, agar saat *install selenium* langsung ke *path anaconda* tidak ke aplikasi yang lain. Klik *install* gambar *Centang Anaconda to my PATH*.

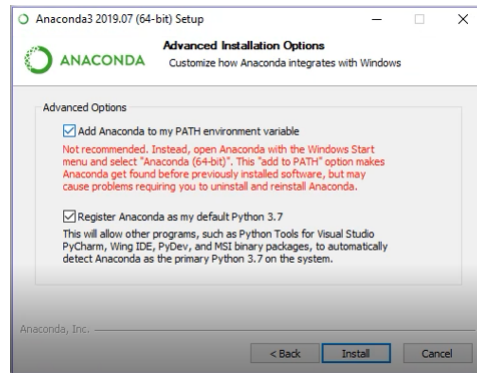


Figure 1.8: *Centang Anaconda to my PATH*

8. Tunggu sampai proses *installasi* selesai gambar *Installation Complete*.

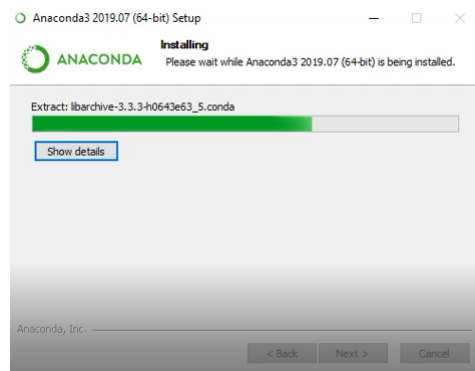


Figure 1.9: *Installation Complete*

9. Apabila instalasi telah selesai klik *next*

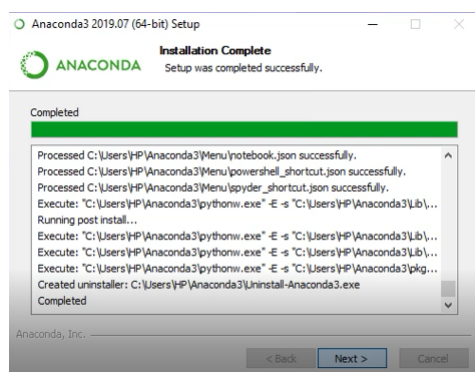


Figure 1.10: *Installation Complete*

10. klik *next*

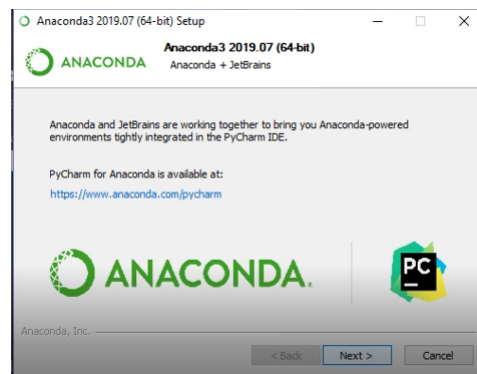


Figure 1.11: *Anaconda+JetBrains*

11. Jika sudah klik *finish* gambar *Thanks fo install Anaconda*.

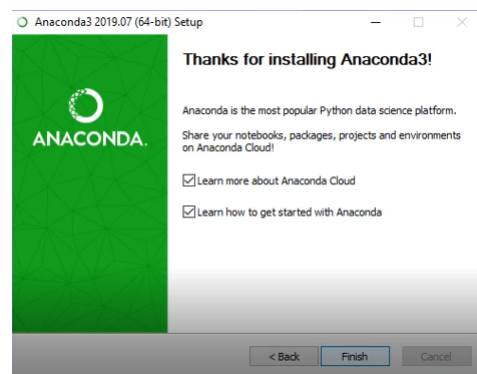


Figure 1.12: *Thanks for install Anaconda*

1.2.2 Instalasi Pip

1. buka anaconda prompt
2. ketikkan `conda install -c anaconda pip`


```
Anaconda Prompt (Anaconda3)

(base) C:\Users\trian>conda install -c anaconda pip
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.7.10
  latest version: 4.7.12

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: C:\Users\trian\Anaconda3

  added / updated specs:
    - pip

The following packages will be downloaded:

  package | build | size | channel
  -----|-----|-----|-----
  ca-certificates-2019.5.15 | 0 | 166 KB | anaconda
  certifi-2019.6.16 | py37_0 | 155 KB | anaconda
```

Figure 1.13: *Install pip*

3. ketik y, lalu enter. Tunggu hingga proses instalasi selesai.

```
Anaconda Prompt (Anaconda3)

-----
Total: 10.9 MB

The following packages will be UPDATED:

  conda                pkgs/main::conda-4.7.10-py37_0 --> anaconda::conda-4.7.12-py37_0

The following packages will be SUPERSEDED by a higher-priority channel:

  ca-certificates      pkgs/main --> anaconda
  certifi              pkgs/main --> anaconda
  openssl              pkgs/main --> anaconda
  pip                  pkgs/main --> anaconda

Proceed ([y]/n)? y

Downloading and Extracting Packages
openssl-1.1.1c      | 5.7 MB | ##### | 100%
certifi-2019.6.16  | 155 KB | ##### | 100%
ca-certificates-2019 | 166 KB | ##### | 100%
pip-19.1.1         | 1.8 MB | ##### | 100%
conda-4.7.12       | 3.0 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\trian>
```

Figure 1.14: *Install pip Selesai*

4. jika telah selesai, lakukan pengecekan versi pip dengan mengetikkan pip -V

```

Anaconda Prompt (Anaconda3)
help          Show help for commands.

General Options:
-h, --help          Show help.
--isolated          Run pip in an isolated mode, ignoring environment variables and user configuration.
-v, --verbose       Give more output. Option is additive, and can be used up to 3 times.
-V, --version       Show version and exit.
-q, --quiet         Give less output. Option is additive, and can be used up to 3 times (corresponding to
                    WARNING, ERROR, and CRITICAL logging levels).
--log <path>       Path to a verbose appending log.
--proxy <proxy>     Specify a proxy in the form [user:passwd@]proxy.server:port.
--retries <retries> Maximum number of retries each connection should attempt (default 5 times).
--timeout <sec>     Set the socket timeout (default 15 seconds).
--exists-action <action> Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup,
                    (a)bort.
--trusted-host <hostname> Mark this host as trusted, even though it does not have valid or any HTTPS.
--cert <path>       Path to alternate CA bundle.
--client-cert <path> Path to SSL client certificate, a single file containing the private key and the
                    certificate in PEM format.
--cache-dir <dir>   Store the cache data in <dir>.
--no-cache-dir      Disable the cache.
--disable-pip-version-check Don't periodically check PyPI to determine whether a new version of pip is available for
                    download. Implied with --no-index.
--no-color          Suppress colored output

(base) C:\Users\trian>pip -V
pip 19.1.1 from C:\Users\trian\Anaconda3\lib\site-packages\pip (python 3.7)
(base) C:\Users\trian>

```

Figure 1.15: *Melihat Versi pip*

1.2.3 Setting Environment

1. Buka file explorer
2. Klik kanan pada This pc, lalu pilih properties

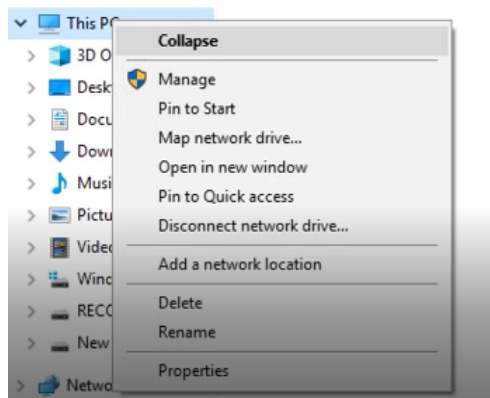


Figure 1.16: *Properties*

3. Pilih menu Advanced system settings



Figure 1.17: *Advanced system settings*

4. Pilih Environment Variables

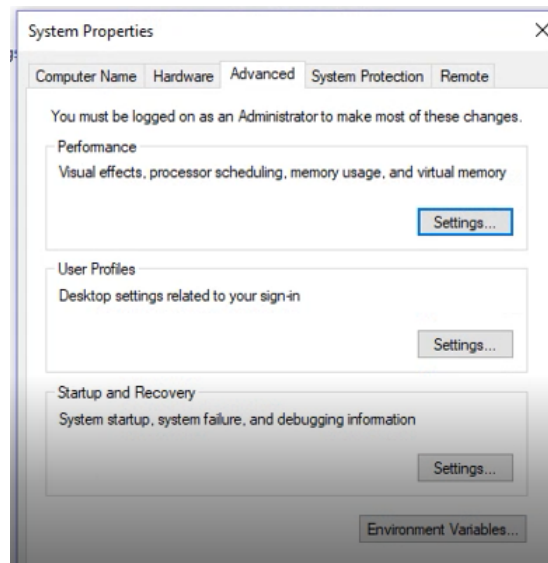


Figure 1.18: *Environment Variables*

5. Pilih Path

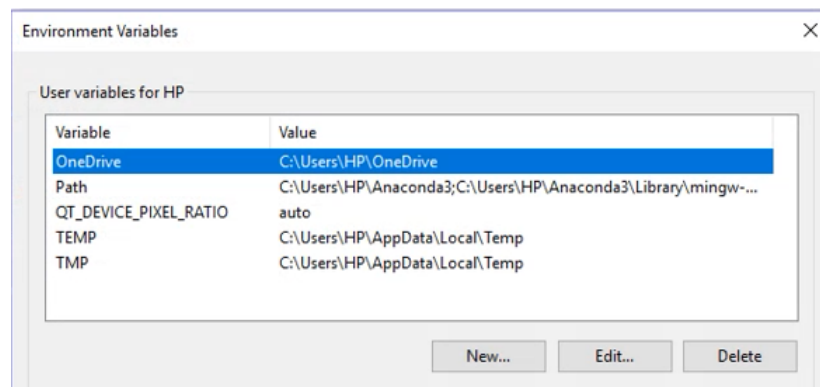


Figure 1.19: *Path*

6. lalu pilih environment variable yang ingin ditambahkan, klik OK

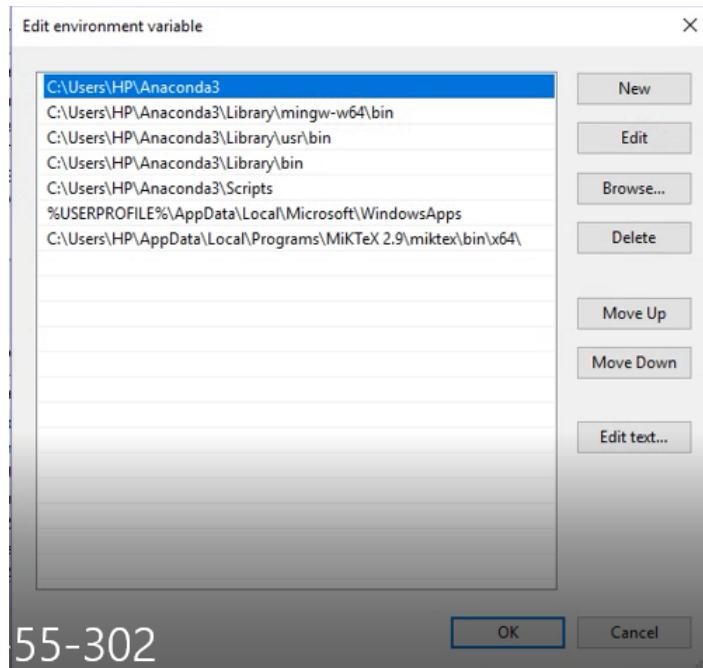


Figure 1.20: *Edit Environment Variable*

1.2.4 Command Line Interface/Interpreter

1. Buka command prompt lalu ketikkan python
2. Buatlah perintah print, input, perkalian, dan pembagian
3. Bisa juga menjalankan file .py yang telah dibuat di IDE dengan cara python namafilename.py, lalu klik enter

```

Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\trian>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Salah Apa Aku")
Salah Apa Aku
>>> input("Masukkan Nama Anda: ")
Masukkan Nama Anda: Dinda
'Dinda'
>>> 3*2
6
>>> 3/2
1.5
>>> exit()

C:\Users\trian>python python.py
Hello World

C:\Users\trian>

```

Figure 1.21: *CLI in Command Prompt*

1.2.5 Update Anaconda dan Spyder

Cara mengupdate Spyder

1. Buka anaconda prompt, lalu ketikkan perintah `conda update spyder`
2. Konfirmasi update dengan mengetikkan `y`, lalu tekan enter
3. Tunggu hingga installan selesai

Cara mengupdate Anaconda

1. Buka anaconda prompt, lalu ketikkan perintah `conda update anaconda`
2. Konfirmasi update anaconda dengan mengetikkan `y` dan kemudian tekan enter
3. Tunggu hingga installan selesai

1.2.6 Run Script Hello World di Spyder

1. Buka anaconda navigator, lalu klik launch

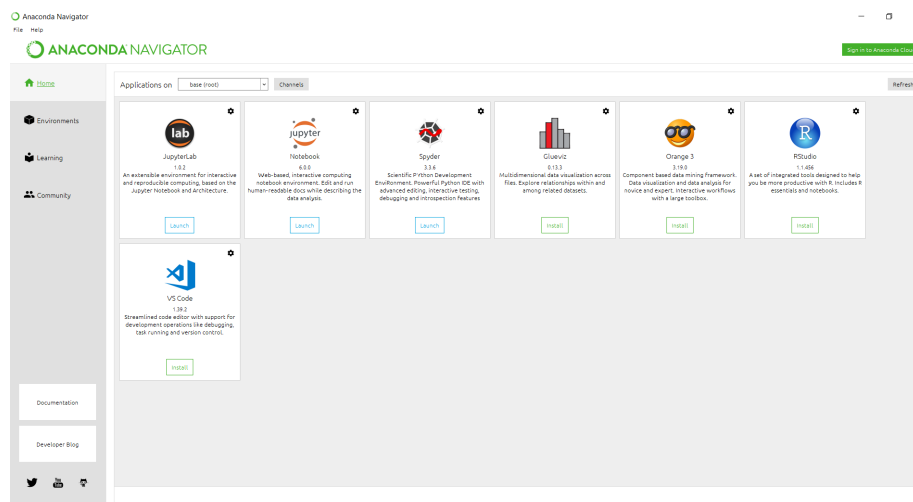


Figure 1.22: *CLI in Command Prompt*

2. ketikkan `print("Hello World")` dan run spyder

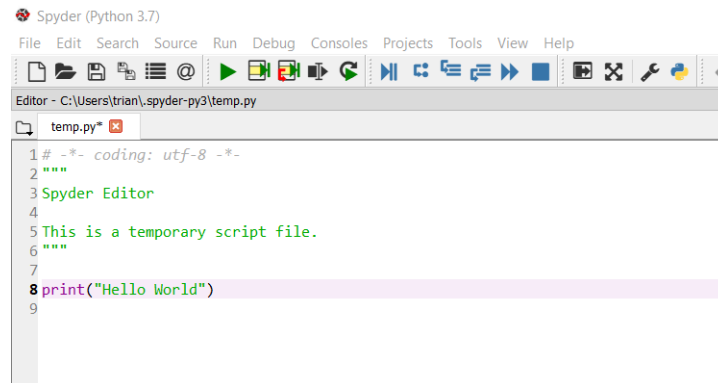


Figure 1.23: *Print Hello World*

3. hasilnya akan seperti ini

```

Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.6.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/trian/.spyder-py3/temp.py', wdir='C:/Users/trian/.spyder-
py3')
Hello World

```

Figure 1.24: *Hello World*

1.2.7 Automatic Login SIAP

Buka Spyder lalu tuliskan script sebagai berikut.

```

from selenium import webdriver

print("Masukkan NPM Anda: ")
npm = input()
print("Masukkan Password Anda: ")
password = input()

options = webdriver.ChromeOptions()
driver = webdriver.Chrome(chrome_options=options)

driver.get("http://siap.poltekpos.ac.id/siap/besan.depan.php")

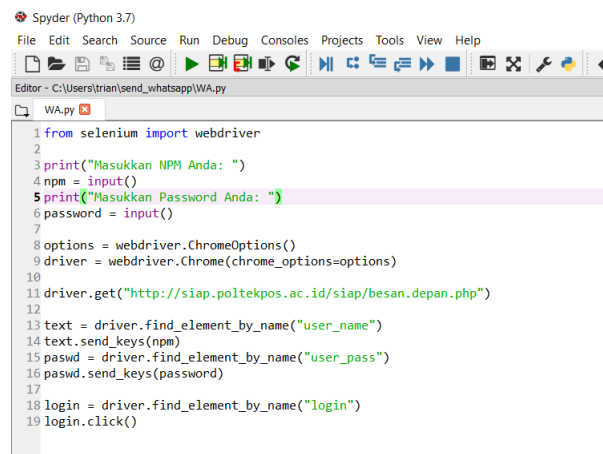
text = driver.find_element_by_name("user_name")
text.send_keys(npm)

```

```
paswd = driver.find_element_by_name("user_pass")
paswd.send_keys(password)
```

```
login = driver.find_element_by_name("login")
login.click()
```

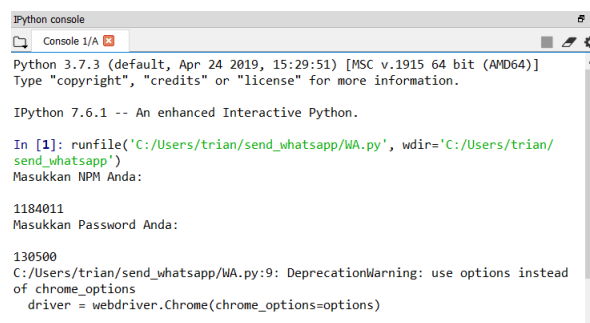
Kemudian save program dengan nama WA.py dan run program



```
1 from selenium import webdriver
2
3 print("Masukkan NPM Anda: ")
4 npm = input()
5 print("Masukkan Password Anda: ")
6 password = input()
7
8 options = webdriver.ChromeOptions()
9 driver = webdriver.Chrome(chrome_options=options)
10
11 driver.get("http://siap.poltekpos.ac.id/siap/besan.depan.php")
12
13 text = driver.find_element_by_name("user_name")
14 text.send_keys(npm)
15 paswd = driver.find_element_by_name("user_pass")
16 paswd.send_keys(password)
17
18 login = driver.find_element_by_name("login")
19 login.click()
```

Figure 1.25: *Automatic Login SIAP*

Setelah di run maka akan diminta memasukkan NPM dan Password SIAP



```
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.6.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/trian/send_whatsapp/WA.py', wdir='C:/Users/trian/
send_whatsapp')
Masukkan NPM Anda:
1184011
Masukkan Password Anda:
130500
C:/Users/trian/send_whatsapp/WA.py:9: DeprecationWarning: use options instead
of chrome_options
  driver = webdriver.Chrome(chrome_options=options)
```



Figure 1.26: *Hasil Running*

Program selanjutnya akan membuka chrome secara otomatis dan mengetikkan NPM serta password yang telah diinputkan oleh user dan kemudian mengklik login.

SIAP - Politeknik Pos Indonesia

Not secure | siap.poltekpos.ac.id/siap/besan.depan.php

Chrome is being controlled by automated test software.

POLTEKPOS

Statistik Perwalian Semester Ganjil 2019/2020

Program Studi	Pengisian KRS						Perwalian							
	Hari Ini		Sebelumnya		Jumlah		Hari Ini		Sebelumnya		Jumlah			
	Mhs	M K	Mhs	M K	Aktif	Mhs	M K	Mhs	M K	Mhs	M K	Aktif	Mhs	M K
D3 Teknik Informatika	0	0	97	792	110	97	792	0	0	92	731	110	92	731
D4 Teknik Informatika	0	0	272	1750	284	272	1750	0	0	270	1676	284	270	1676
D3 Manajemen Informatika	0	0	46	329	67	46	329	0	0	46	324	67	46	324
D3 Akuntansi	0	0	103	895	103	103	895	0	0	102	858	103	102	858
D4 Akuntansi Keuangan	0	0	186	1394	214	186	1394	0	0	184	1380	214	184	1380
D3 Manajemen Pemasaran	1	8	70	610	96	71	618	0	0	63	550	96	63	550
D4 Manajemen Perusahaan	1	6	265	1441	347	266	1447	2	12	260	1385	347	262	1397
D3 Logistik Bisnis	2	18	283	2469	352	285	2487	2	18	278	2413	352	280	2431
D4 Logistik Bisnis	13	120	618	4218	733	631	4338	26	255	577	3916	733	603	4171
Jumlah	17	152	1940	13898	2306	1957	14050	30	285	1872	13233	2306	1902	13518

Login

Pengguna
1184011

Kata Sandi
.....

Login

Log Pengisian KRS dan Perwalian Semester Ganjil 2019/2020

Pengisian KRS				Perwalian			
No	N P M	Nama	Jml MK Waktu	No	N P M	Nama	Jml MK Waktu
1	6164130	Hafiz Riady Sandira	8 16-10-2019 (14:11:02)	1	6194160	Novia Ratnasari	10 16-10-2019 (15:52:20)
2	5173137	Wira Cristian Anugrah Pardede	8 16-10-2019 (14:05:06)	2	4184117	Rafi Pratama Andito	6 16-10-2019 (14:32:13)
3	5173131	Dina Wulandari	10 16-10-2019 (14:01:12)	3	4184064	NANCY MARIA MEDELIN	6 16-10-2019 (14:31:15)
4	4184064	NANCY MARIA MEDELIN	6 16-10-2019 (13:27:34)	4	5173131	Dina Wulandari	10 16-10-2019 (14:08:50)
5	6174172	Itha Rifkiana	7 16-10-2019 (12:54:44)	5	5173137	Wira Cristian Anugrah Pardede	8 16-10-2019 (14:08:29)
6	6184170	Lutago Aldita	10 16-10-2019 (13:08:08)				

Waiting for siap.poltekpos.ac.id...

Figure 1.27: Automatic Input dan Login

Login selesai

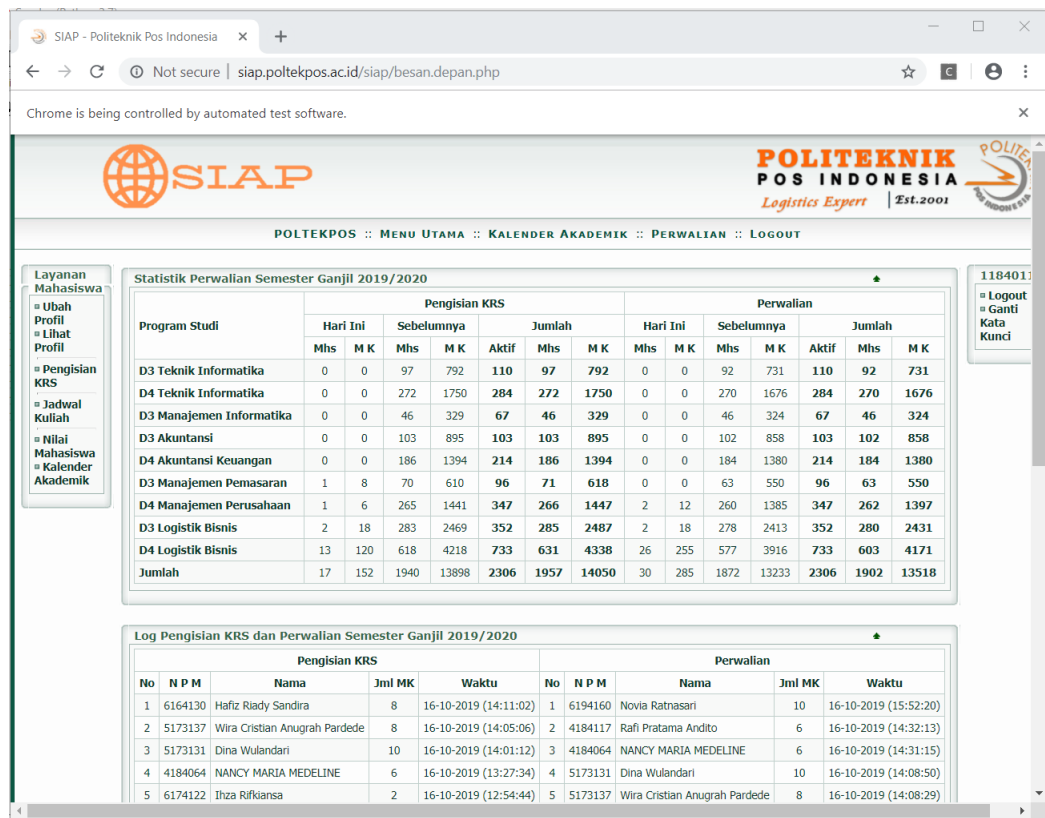


Figure 1.28: *Automatic Input dan Login Selesai*

1.2.8 Pemakaian Variable Explorer

Variable explorer akan secara otomatis terisi ketika kita membuat variable, pada variable explorer kita bisa melihat nama variable, tipe data, length, dan value dari variable tersebut.

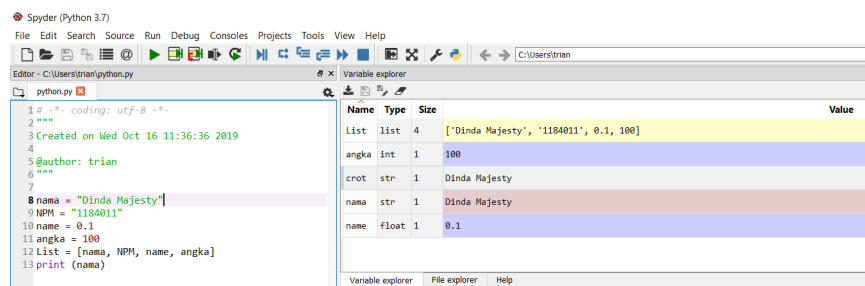


Figure 1.29: *Variable Explorer*

1.3 Indentasi

1.3.1 Penjelasan Indentasi

Indentasi adalah bagian paragraf yang menjorok ke dalam pada baris-baris paragraf. Mengatur indentasi dapat menggunakan tab atau spasi. Indentasi digunakan oleh bahasa pemrograman python sebagai pengganti briket () untuk membuka dan menutup fungsi. Error indentasi dapat terjadi apabila syntax tidak menggunakan tab atau space. Contoh yang benar (menggunakan tab/spasi sebagai indentasi):

```
# blok percabangan if
if username == 'petanikode':
    print("Selamat Datang Admin")
    print("Silahkan ambil tempat duduk")

# blok percabangan for
for i in range(10):
    print i
```

Contoh yang salah (tidak menggunakan tab/spasi):

```
# blok percabangan if
if username == 'petanikode':
print("Selamat Datang Admin")
print("Silahkan ambil tempat duduk")

# blok percabangan for
for i in range(10):
print i
```

1.3.2 Jenis-Jenis Error Indentasi

IndentationError: unexpected indent. Error diatas terjadi apabila syntax kekurangan tab atau spasi.

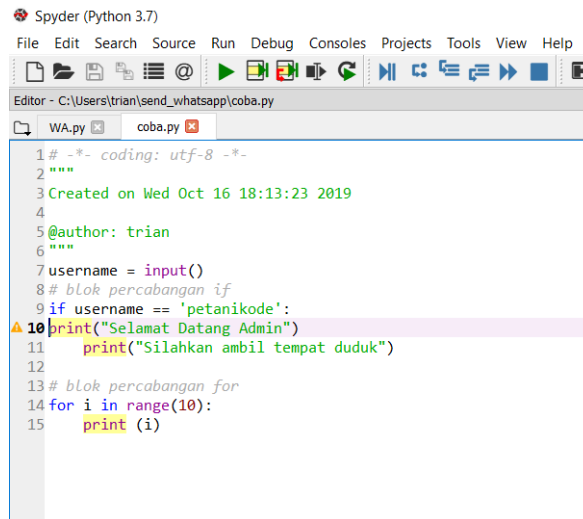


Figure 1.30: *Indentasi*

Apabila di running akan memunculkan error sebagai berikut.

```
In [6]: runfile('C:/Users/trian/send_whatsapp/coba.py', wdir='C:/Users/trian/
send_whatsapp')
Traceback (most recent call last):

  File "C:/Users/trian/Anaconda3/lib/site-packages/IPython/core
\interactiveshell.py", line 3325, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)

  File "<ipython-input-6-0a8945c5cda6>", line 1, in <module>
    runfile('C:/Users/trian/send_whatsapp/coba.py', wdir='C:/Users/trian/
send_whatsapp')

  File "C:/Users/trian/Anaconda3/lib/site-packages/spyder_kernels\customize
\spydercustomize.py", line 827, in runfile
    execfile(filename, namespace)

  File "C:/Users/trian/Anaconda3/lib/site-packages/spyder_kernels\customize
\spydercustomize.py", line 110, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

  File "C:/Users/trian/send_whatsapp/coba.py", line 10
    print("Selamat Datang Admin")
    ^
IndentationError: expected an indented block
```

Figure 1.31: *Error Indentasi*

1.3.3 Cara Membaca Error

Jika terjadi error maka cari di line berapa error terjadi, pada gambar berikut terdapat error indentasi pada line 10.

```
In [6]: runfile('C:/Users/trian/send_whatsapp/coba.py', wdir='C:/Users/trian/
send_whatsapp')
Traceback (most recent call last):

  File "C:/Users/trian/Anaconda3/lib/site-packages/IPython/core
\interactiveshell.py", line 3325, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)

  File "<ipython-input-6-0a8945c5cda6>", line 1, in <module>
    runfile('C:/Users/trian/send_whatsapp/coba.py', wdir='C:/Users/trian/
send_whatsapp')

  File "C:/Users/trian/Anaconda3/lib/site-packages/spyder_kernels\customize
\spydercustomize.py", line 827, in runfile
    execfile(filename, namespace)

  File "C:/Users/trian/Anaconda3/lib/site-packages/spyder_kernels\customize
\spydercustomize.py", line 110, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

  File "C:/Users/trian/send_whatsapp/coba.py", line 10
    print("Selamat Datang Admin")
    ^
IndentationError: expected an indented block
```

Figure 1.32: *Error*

1.3.4 Cara Menangani Error

Menangani error indentasi dapat dilakukan dengan cara menambahkan tab atau space pada line yang error. Berikut syntax yang error.

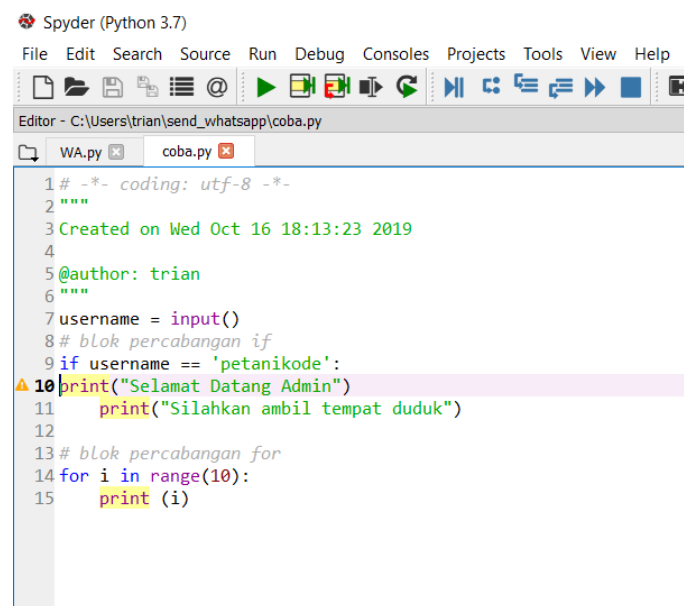
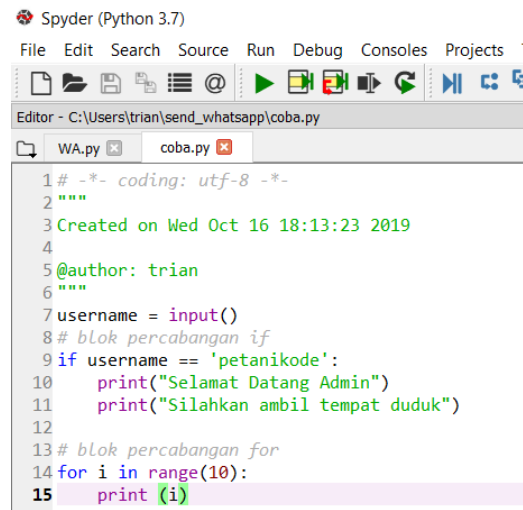


Figure 1.33: *Syntax Error*

Berikut syntax yang tidak error menggunakan tab/spasi (indentasi).



```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 16 18:13:23 2019
4
5 @author: trian
6 """
7 username = input()
8 # blok percabangan if
9 if username == 'petanikode':
10     print("Selamat Datang Admin")
11     print("Silahkan ambil tempat duduk")
12
13 # blok percabangan for
14 for i in range(10):
15     print(i)
```

Figure 1.34: *Syntax yang Telah Diperbaiki*

1.3.5 Link Youtube

Berikut link youtube yang berisi jawaban dari pertanyaan-pertanyaan diatas.

https://www.youtube.com/channel/UCBadc75-CEWvCTTWkfUJEUg?view_as=subscriber

Chapter 2

Mengenai Python dan Anaconda

2.1 Teori

2.1.1 Jenis-Jenis Variabel

Variabel merupakan tempat penyimpanan data. Tipe data merupakan jenis data yang tersimpan di dalam variabel. terdapat aturan dalam penulisan Variabel.

1. Nama variabel diawali dengan huruf atau garis bawah, contoh: nama, _nama, namaKu, nama_variabel.
2. Karakter selanjutnya dapat berupa huruf, garis bawah atau angka, contoh: _nama, nama1, p1.
3. Karakter bersifat case-sensitive (huruf besar dan huruf kecil dibedakan), contoh: Nama dan NAMA keduanya adalah variabel yang berbeda.
4. Nama variabel tidak boleh menggunakan kata kunci yang ada pada bahasa pemrograman python, contoh: if, else, while
5. Nama variabel tidak boleh diawali dengan angka

Jenis-jenis tipe data pada python. Tipe Data Primitif, dibagi menjadi 3 yaitu:

1. Tipe data integer (angka), penulisannya tidak membutuhkan tanda petik, contoh: 10 atau 15
2. Tipe data string (teks), tipe data string ditandai dengan teks yang diapit oleh tanda petik (""), contoh: "nama saya adalah dinda majesty"
3. Tipe data boolean (memiliki dua nilai yaitu true dan false atau 0 dan 1)

Contoh penulisan variabel dan tipe datanya:

1. angka = 10, angka merupakan nama variabel sedangkan 10 adalah nilai dari variabel yang tipe datanya integer.
2. nama = "Dinda Majesty", nama merupakan nama variabel sedangkan "Dinda Majesty" merupakan nilai dari variabel yang tipe datanya string, ditandai dengan adanya petik ("").
3. makan = True , makan merupakan nama variabel sedangkan True merupakan nilai dari variabel yang tipe datanya boolean.

2.1.2 Input dan Output

Berikut kode untuk meminta inputan dari user.

```
1 #input output
2 print("masukkan nama anda : ")
3 nama = input()
4 print("nama saya adalah " + nama)
```

Listing 2.1: Input dan Output

Perintah input() berguna untuk meminta inputan dari user, sehingga memungkinkan user untuk menginputkan data.

Perintah print() berguna untuk menampilkan output dari data yang diinputkan oleh user, sehingga data yang diinputkan user dapat ditampilkan ke layar.

2.1.3 Operator Aritmatika dan Konversi Tipe Data

Operator aritmatika

1. penjumlahan (+)
2. pengurangan (-)
3. perkalian (*)
4. pembagian (/)
5. sisa bagi/modulus (%)
6. pemangkatan (**)

Cara melakukan perubahan terhadap tipe data string menjadi integer, contoh: variabel = "10". Kita dapat mengubah string "10" menjadi angka 10 dengan menambahkan kode int(variabel), dengan begitu 10 yang awalnya bertipe data string akan dikonversikan menjadi integer.

Cara melakukan perubahan terhadap tipe data integer menjadi string, contoh: variabel = 150. Kita dapat mengubah integer 150 menjadi string "150" dengan menambahkan kode str(variabel), maka tipe data dari variabel akan dikonversikan menjadi string.

2.1.4 Perulangan

perulangan terdiri atas 3 kondisi.

1. While, apabila kondisinya True, maka perulangan akan terus berjalan hingga diperoleh kondisi False. Contoh penggunaan while:

```
1 #perulangan while
2 hitung = 0
3 while (hitung < 9):
4     print ('hitungan ke :', hitung)
5     hitung = hitung + 1
6
7 print ("Good bye!")
```

Listing 2.2: While Loop

2. For, perulangan for bisa melakukan perulangan terhadap item apapun seperti list atau string. Contoh penggunaan For:

```
1 #perulangan for
2 minum = ["kopi", "susu", "teh"]
3 for minuman in minum:
4     print("Saya suka minuman", minuman)
```

Listing 2.3: For Loop

3. nested, perulangan ini memungkinkan adanya perulangan didalam perulangan. Contoh penggunaan nested:

```
1 #nested loop
2 i = 2
3 while(i < 100):
4     j = 2
5     while(j <= (i/j)):
```



```

6         if not(i%j): break
7         j = j + 1
8     if (j > i/j) : print(i, " is prime")
9     i = i + 1
10
11 print ("Good bye!")

```

Listing 2.4: Nested Loop

Pada penulisan sintaks While dan For harus memperhatikan identasi (baris yang menjorok ke dalam), jika tidak diperhatikan dengan baik maka akan terjadi error terhadap identasi. Untuk menambahkan identasi dapat menggunakan spasi atau tab.

2.1.5 IF Statement

kondisi if dapat digunakan didalam looping dan dapat digunakan untuk memberikan kondisi tertentu dengan cara mengetikkan if lalu kondisi yang akan terjadi.

1. if hanya menjalankan satu kondisi dan menampilkan satu output. Contoh: kondisi dimana variabel a lebih besar dari variabel b, maka tampilkan hasil bahwa a lebih besar dari b.

```

1 #if statement
2 a = 330
3 b = 200
4 if a > b:
5     print("a lebih besar dari b")

```

Listing 2.5: if Statement

2. elif digunakan apabila kondisi pertama tidak benar maka lakukan kondisi lain (alternatif). Contoh: kondisi dimana variabel a sama dengan variabel b, maka jika b lebih besar dari a, tampilkan hasil b lebih besar dari a, namun jika a dan b bernilai sama, maka tampilkan a sama dengan b

```

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

```

Listing 2.6: Elif

3. `else` digunakan apabila kondisi yang terjadi bernilai salah, maka lakukan `else`.
Contoh: kondisi dimana variabel `a` lebih besar dari variabel `b`, maka jika `b` lebih besar dari `a`, tampilkan hasil `b` lebih besar dari `a`, jika `a` dan `b` bernilai sama, maka tampilkan `a` sama dengan `b`, jika salah maka tampilkan `a` lebih besar dari pada `b`

```
1 #else
2 a = 200
3 b = 33
4 if b > a:
5     print("b is greater than a")
6 elif a == b:
7     print("a and b are equal")
8 else:
9     print("a is greater than b")
```

Listing 2.7: Else

4. Nested if merupakan if didalam if (if bersarang), terdapat dua if didalam satu kondisi. Contoh: variabel `x` sama dengan 41, kondisi pertama yaitu jika `x` besar dari 10 maka tampilkan lebih besar dari 10, kondisi kedua yaitu jika `x` besar dari 20, maka tampilkan lebih besar dari 20, jika salah maka tampilkan tidak melebihi 20.

```
1 #nested if
2 x = 41
3
4 if x > 10:
5     print("lebih besar dari 10,")
6     if x > 20:
7         print("lebih besar dari 20!")
8     else:
9         print("tidak melebihi 20.")
```

Listing 2.8: Nested If

2.1.6 Error

1. `NameError`, terjadi apabila kode mengeksekusi nama yang tidak terdefiniskan.
Contoh:

```
nama = "Dinda Majesty"
print>Nama)
```

Maka akan menghasilkan output `NameError: name 'Nama' is not defined`. error ini dapat diatasi dengan mengubah variabel yang di print sesuai dengan variabel yang didefinisikan, karena penulisan pada python bersifat case-sensitive

2. `SyntaxError`, terjadi apabila kode python mengalami kesalahan saat penulisan. Contoh: menuliskan variabel yang didahului angka (`1nama = "Dinda Majesty"`) maka akan muncul error `SyntaxError: invalid syntax`. error ini dapat diatasi dengan memperhatikan tata cara penulisan kode pada bahasa pemrograman python.
3. `TypeError`, terjadi apabila kode melakukan operasi atau fungsi terhadap tipe data yang tidak sesuai. Contoh: melakukan penjumlahan terhadap tipe data string dan integer. error ini dapat diatasi dengan mengubah tipe data string menjadi integer.

```
a = "10"
b = 5

print(a + b)
```

Maka akan menghasilkan output error `TypeError: can only concatenate str (not "int") to str`

4. `IndentationError`, terjadi apabila kode perulangan atau pengkondisian tidak menjorok kedalam (tidak menggunakan indentasi), error ini dapat diatasi dengan menambahkan tab atau spasi. Contoh:

```
a = 200
b = 330
if b > a:
    print("b lebih besar dari a")
```

Maka akan menghasilkan output error `IndentationError: expected an indented block`

2.1.7 Try Except

Try Except merupakan perintah yang bisa digunakan dalam penanganan error pada bahasa pemrograman python. perintah ini biasanya digunakan saat penanganan error input/output, operasi database, pengaksesan indeks suatu list atau dictionary dan berbagai kasus lainnya.

Contoh sederhana penggunaan Try-Except saat menangani NameError

```
1 #try except
2 try:
3     print(x)
4 except NameError:
5     print("Variable x tidak ada")
6 except:
7     print("ada sesuatu yang salah nih")
```

Listing 2.9: Try Except

Pada contoh diatas, try except akan menghasilkan output Variabel x tidak ada, karena kita tidak mendefinisikan variabel x sebelum kita menampilkan output x ke layar. Dengan menggunakan try except kode yang kita buat terhindar dari error dan kita bisa mengetahui kesalahan yang terjadi ketika terdapat error pada kode kita.

2.2 Keterampilan Pemrograman

2.2.1 Modulus

```
1 #Modulus
2 print("Soal no 1")
3
4 print("Masukkan NPM anda: ")
5 NPM = input()
6
7 npm = int(NPM) % 3
8 print(npm)
9
10 print("###      ###      #####      ###      ###      #####      ###      ###")
11 print("###      ###      #####      ###      ###      #####      ###      ###")
12 print("###      ###      ###      ###      ###      ###      ###      ###      ###")
13 print("###      ###      ###      ###      ###      ###      ###      ###      ###")
14 print("###      ###      #####      #####      ###      ###      ###      ###")
15 print("###      ###      ###      ###      ###      ###      ###      ###      ###")
16 print("###      ###      ###      ###      ###      ###      ###      ###      ###")
17 print("###      ###      #####      ###      #####      ###      ###")
18 print("###      ###      #####      ###      #####      ###      ###")
```

Listing 2.10: Modulus

2.2.2 Hello NPM

```

1 #Hello NPM
2 print("Soal No 2")
3
4 Loop = NPM[5:7]
5
6 for x in range(int(Loop)):
7     print("Hallo " + NPM + " Apa Kabar?")

```

Listing 2.11: Hello NPM

2.2.3 Hello NPM (3 Digit Belakang)

```

1 #3 digit belakang NPM
2 print("Soal No 2")
3
4 Loop = NPM[4:7]
5
6 total = int(NPM[5]) + int(NPM[6])
7
8 for x in range(total):
9     print("Hallo " + Loop + " Apa Kabar?")

```

Listing 2.12: 3 Digit Belakang

2.2.4 Hello NPM (Digit ke-3)

```

1 #digit ke 3
2 print("Soal No 3")
3
4 Loop = NPM[4]
5 print("Hello " + Loop + " Apa Kabar?")

```

Listing 2.13: Digit ke-3

2.2.5 Variabel Alfabet

```

1 #variabel alfabet
2 print("Soal no 5")
3
4 var = "abcdefg"
5 index = 0
6
7 for i in var:
8     print(i + " = " + NPM[index])
9     index += 1

```

Listing 2.14: Variabel Alfabet

2.2.6 Penjumlahan NPM

```

1 #penjumlahan NPM
2 print("Soal no 6")
3
4 index = 0
5 angka = 0

```

```

6
7 for i in NPM:
8     jumlah = int(NPM[index]) + int(angka)
9     angka = jumlah
10    index += 1
11
12 print(jumlah)

```

Listing 2.15: Penjumlahan NPM

2.2.7 Perkalian NPM

```

1 #perkalian NPM
2 print("Soal no 7")
3
4 index = 0
5 angka = 0
6
7 for i in NPM:
8     jumlah = int(NPM[index]) * int(angka)
9     angka = jumlah
10    index += 1
11
12 print(jumlah)

```

Listing 2.16: Perkalian NPM

2.2.8 Print Vertical

```

1 #print vertical
2 print("Soal no 8")
3
4 for i in NPM:
5     print(i)

```

Listing 2.17: Print Vertical

2.2.9 Digit Genap NPM

```

1 #digit genap NPM
2 print("Soal no 9")
3
4 index = 0
5 for i in NPM:
6     if (int(NPM[index])%2 == 0) & (int(NPM[index]) != 0):
7         print(NPM[index])
8     index += 1

```

Listing 2.18: Digit Genap NPM

2.2.10 Digit Ganjil NPM

```

1 #digit ganjil NPM
2 print("Soal no 10")
3

```

```

4 index = 0
5 for i in NPM:
6     if (int(NPM[index])%2 != 0) & (int(NPM[index]) != 0):
7         print(NPM[index])
8     index += 1

```

Listing 2.19: Digit Ganjil NPM

2.2.11 Bilangan Prima NPM

```

1 #bilangan prima NPM
2 print("Soal no 11")
3 index = 0
4
5 for i in NPM:
6     prima = True
7     var=int(NPM[index])
8     if(var<=1):
9         prima=False
10    for i in range (2,var):
11        if(var%i==0):
12            prima=False
13    if(prima==True):
14        print(var,"Prima")
15    else:
16        print(var,"bukan prima")
17    index += 1

```

Listing 2.20: Bilangan Prima NPM

2.3 Keterampilan Penanganan Error

2.3.1 Peringatan Error dan Cara Mengatasinya

1. `IndentationError`: expected an indented block, terjadi apabila kode perulangan atau pengkondisian tidak menjorok kedalam (tidak menggunakan indentasi), error ini dapat diatasi dengan menambahkan tab atau spasi.
2. `TypeError`: can only concatenate str (not "int") to str, terjadi apabila kode melakukan operasi atau fungsi terhadap tipe data yang tidak sesuai. Contoh: melakukan penjumlahan terhadap tipe data string dan integer. error ini dapat diatasi dengan mengubah tipe data string menjadi integer.
3. `NameError`: name 'NOM' is not defined, error ini dapat terjadi apabila terdapat kesalahan penulisan dan saat dijalankan, tidak ditemukan name NOM didalam kode yang dituliskan. error ini dapat diatasi dengan memperhatikan penulisan kode.

4. `SyntaxError: invalid syntax`, error ini terjadi apabila kode yang dijalankan memiliki kesalahan penulisan `syntax`. Contoh: `if(prima=True):` salah karena kekurangan tanda `=`, seharusnya ditulis `if(prima==True):`, error ini dapat ditangani dengan memperhatikan penulisan kode program

2.3.2 Percobaan Try Except

```
1 var_str = "10"  
2 var_int = 10  
3  
4 try:  
5     jumlah = var_str + var_int  
6     print(jumlah)  
7 except:  
8     print("String tidak bisa ditambahkan dengan Integer!")  
9     print("Tolong perbaiki tipe data string, konversikan terlebih  
    dahulu ke integer.")
```

Listing 2.21: Bilangan Prima NPM

Output:

```
In [12]: runfile('C:/Users/trian/Dinda Majesty(1184011)/src/  
2rr.py', wdir='C:/Users/trian/Dinda Majesty(1184011)/src')  
String tidak bisa ditambahkan dengan Integer  
Tolong perbaiki tipe data string, konversikan terlebih dahulu ke  
integer
```

Figure 2.1: *Try Except*