

# Tugas Chapter Pemrograman II



Aditya Rahman

1184021

Applied Bachelor of Informatics Engineering

Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering

*Politeknik Pos Indonesia*

Bandung 2019

## **Acknowledgements**

Pertama-tama kami panjatkan puji dan syukur kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga Modul Praktikum ini dapat diselesaikan.

## **Abstract**

Modul Praktikum ini dibuat dengan tujuan memberikan acuan, bagi mahasiswa dan dosen Pengajar Mata Kuliah. Pada intinya buku ini menjelaskan secara lengkap tentang Standar penilaian mata kuliah pemrograman II di Program Studi D4 Teknik Informatika, dan juga mengatur mekanisme, teknik penulisan, serta penilaiannya. Dengan demikian diharapkan semua pihak yang terlibat dalam aktivitas belajar dan mengajar berjalan lancar dan sesuai dengan standar.

# Contents

<b>1</b>	<b>Teori, Sejarah, &amp; Instalasi Python</b>	<b>1</b>
1.1	Teori . . . . .	1
1.2	Sejarah Python . . . . .	1
1.2.0.1	Penggunaan Python di Perusahaan . . . . .	1
1.2.0.2	Perbedaan Python 2 dan 3 . . . . .	2
1.3	Instalasi . . . . .	2
1.3.1	Instalasi Anaconda . . . . .	2
1.3.2	Intalasi PIP . . . . .	5
1.4	Setting Environment . . . . .	5
1.5	Entrepreter atau CLI melalui terminal atau cmd windows . . . . .	8
1.6	Mengupdate Spyder . . . . .	8
1.7	Menjalankan Hello World . . . . .	9
1.8	Menjalankan Script Otomatis Login Aplikasi Akademik . . . . .	9
1.9	Variable Explorer . . . . .	11
1.10	Identasi . . . . .	11
<b>2</b>	<b>Pemrograman Dasar</b>	<b>13</b>
2.1	Teori . . . . .	13
2.2	Ketrampilan Pemrograman . . . . .	16
2.3	Ketrampilan Penanganan Error . . . . .	26
	<b>Bibliography</b>	<b>28</b>

# List of Figures

1.1	Tahap Instalasi 1 . . . . .	3
1.2	Tahapan Instalasi 2 . . . . .	3
1.3	Tahapan Instalasi 3 . . . . .	3
1.4	Tahapan Instalasi 4 . . . . .	4
1.5	Tahapan Instalasi 5 . . . . .	4
1.6	Tahapan Instalasi 6 . . . . .	4
1.7	Tahapan Instalasi 7 . . . . .	4
1.8	Tahapan Instalasi 8 . . . . .	5
1.9	Tahapan Instalasi 9 . . . . .	5
1.10	Ketik perintah : conda install -c anaconda pip . . . . .	5
1.11	Tunggu hingga selesai . . . . .	6
1.12	pilih advanced system setting . . . . .	6
1.13	pilih envirotmet variabel . . . . .	7
1.14	setting environment . . . . .	7
1.15	Tahapan Cek python . . . . .	8
1.16	Tahapan pengudatetan . . . . .	8
1.17	Tahapan update spyder . . . . .	8
1.18	Syntak hello world . . . . .	9
1.19	menampilkan hello world . . . . .	9
1.20	Install selenium . . . . .	9
1.21	Tahap instalasi selenium . . . . .	10
1.22	Syntak pada spyder untuk otomatis login . . . . .	10
1.23	hasil . . . . .	10
1.24	cek variabel explorer . . . . .	11
1.25	syntak eror identasi . . . . .	11
1.26	cek variabel explorer . . . . .	12

# Chapter 1

## Teori, Sejarah, & Instalasi Python

### 1.1 Teori

Python adalah bahasa pemrograman interpretatif multiguna, python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks.

### 1.2 Sejarah Python

Python diciptakan oleh Guido van Rossum pertama kali di centrum wiskunde & informatica (CWI) di Belanda pada awal tahun 1990-an. bahasa yang terinspirasi dari bahasa pemograman ABC, hingga sampai saat ini Guido van Rossum menjadi penulis utama untuk phyton. Tahun 1995 masih melanjutkan pembuatan phyton di Corporation for National Research Initiative (CNRI) di Virginia Amerika yang meriliskan beberapa bahasa phyton. Diantaranya :

1. Python 1.0  
Diliris pada januari tahun 1994
2. Python 2.0  
Diliris pada 16 Oktober tahun 2000
3. Python 3.0  
Diliris pada 3 Desember tahun 2008

#### 1.2.0.1 Penggunaan Python di Perusahaan

Salah satu bahasa yang banyak dipakai dalam sebuah perusahaan hingga saat ini yaitu bahasa pemograman python, contoh penggunaan dalam perusahaan yaitu :

1. Facebook  
Menggunakan framework python "Tornado" yang digunakan untuk menampilkan timeline
2. Instagram  
Menggunakan framework python "Django" yang digunakan sebagai mesin pengelola sisi server dari aplikasi
3. Rasberry pi  
Merupakan perangkat komputer mini yang digunakan sebagai mikrokontroler, bahasa yang digunakannya adalah python
4. NASA  
Badan antariksa Amerika ini menggunakan Python untuk bidang sainsnya.

#### **1.2.0.2 Perbedaan Python 2 dan 3**

Python 2 dipublikasikan pada akhir tahun 2000, dinilai lebih transparan dan inklusif untuk pengembangan software ketimbang versi sebelumnya. didukung dengan adanya PEP – Python Enhancement Proposal, dan dilengkapi dengan berbagai fitur programatikal seperti cycle-detecting garbage collector untuk mengotomasi manajemen memori.

Python 3 merupakan versi yang saat ini dibuat masih aktif, versi ini banyak perubahan yang dirilis akhir tahun 2008. Fokus dari Python 3 itu sendiri adalah untuk melakukan perapian pada codebase dan menghapuskan duplikasi (redundancy). Python 3 mengalami hambatan pada pengadopsiannya, yang mengakibatkan tidak adanya backwards compatibility dengan Python 2.  
perbedaan yang mencolok terletak pada :

1. Sintak
2. Pembagian pada integer

## **1.3 Instalasi**

### **1.3.1 Instalasi Anaconda**

Berikut ini merupakan tutorial cara menginstalasi Anaconda, yang telah di download di [www.anaconda.com](http://www.anaconda.com) setelah itu ikuti langkah-langkah dibawah ini.



Figure 1.1: Tahap Instalasi 1



Figure 1.2: Tahapan Instalasi 2



Figure 1.3: Tahapan Instalasi 3



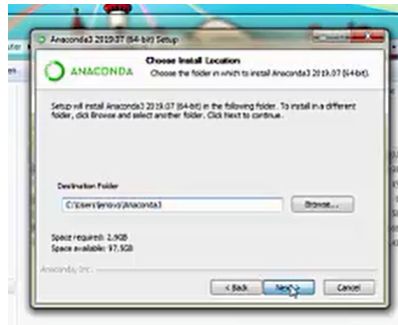


Figure 1.4: Tahapan Instalasi 4



Figure 1.5: Tahapan Instalasi 5

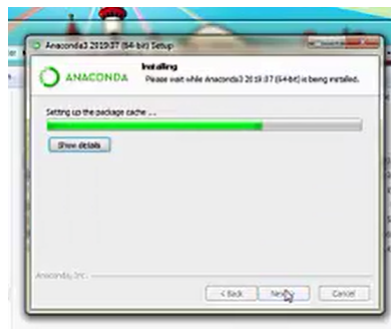


Figure 1.6: Tahapan Instalasi 6



Figure 1.7: Tahapan Instalasi 7



Figure 1.8: Tahapan Instalasi 8

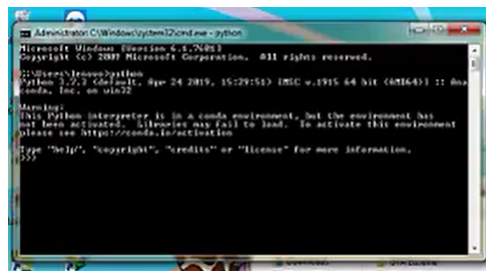


Figure 1.9: Tahapan Instalasi 9

### 1.3.2 Intalasi PIP

Langkah-langkah mengisntall PIP

## 1.4 Setting Environment

Langkah-langkah seperti ini :

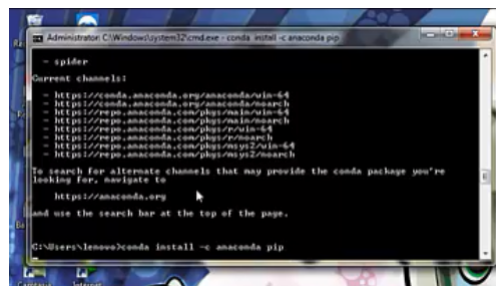


Figure 1.10: Ketik perintah : conda install -c anaconda pip



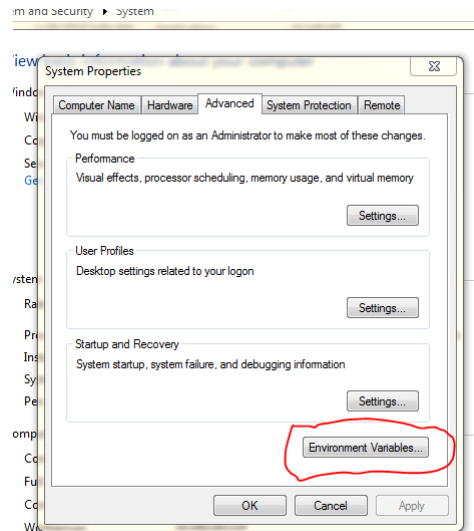


Figure 1.13: pilih envirotnet variabel

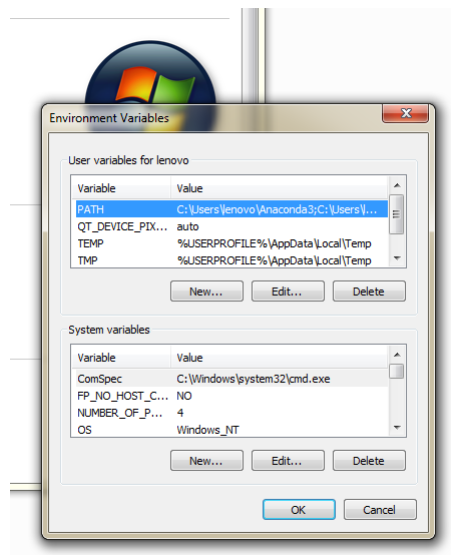


Figure 1.14: setting environment

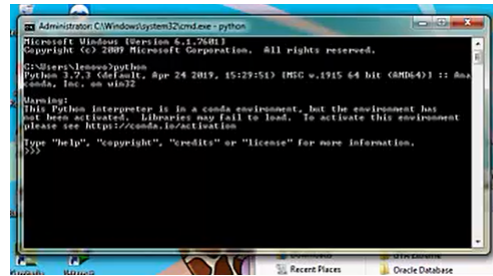


Figure 1.15: Tahapan Cek python

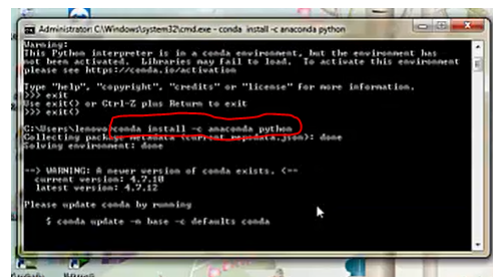


Figure 1.16: Tahapan pengudatetan

## 1.5 Entrepreneter atau CLI melalui terminal atau cmd windows

pada tahap ini, dibutuhkananya cmd sebagai bahan pembelajaran dari mulai cek status python yang sudah terbaru, hingga proses pengupdatetan seperti contoh dibawah ini:

## 1.6 Mengupdate Spyder

Pada langkah ini, dibutuhkan sebuah command Prompt dengan megetikkan conda install -c anaconda spyder

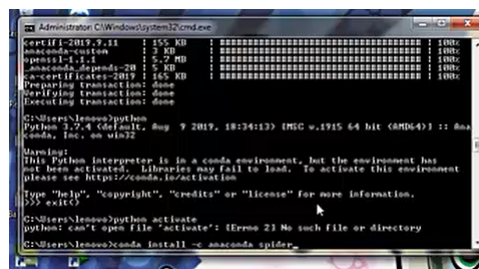


Figure 1.17: Tahapan update spyder

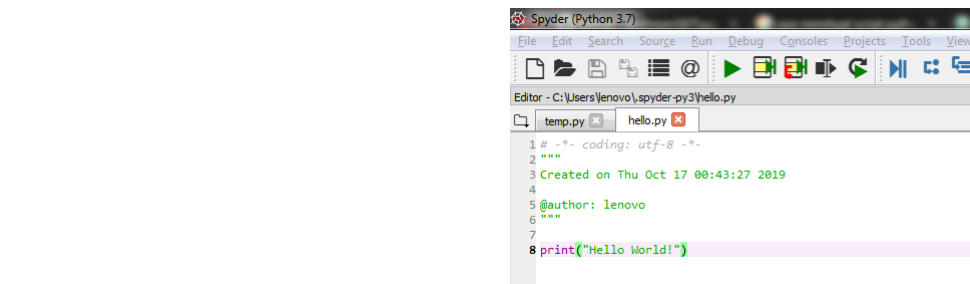


Figure 1.18: Syntak hello world

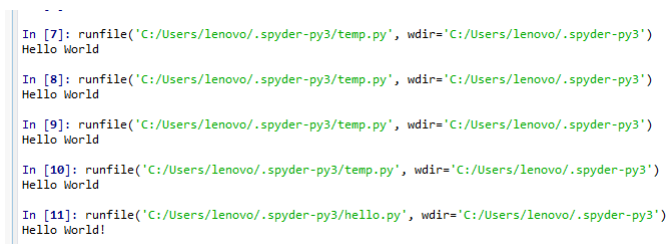


Figure 1.19: menampilkan hello world

## 1.7 Menjalankan Hello World

Pada langkah ini harus di siapkan spyder yang digunakan sebagai text editor yang membantu menerjemahkan bahasa pemrograman python, diantaranya sebagai berikut :

## 1.8 Menjalankan Script Otomatis Login Aplikasi Akademik

Pada langkah pertama instal selenium terlebih dahulu seperti contoh ini :

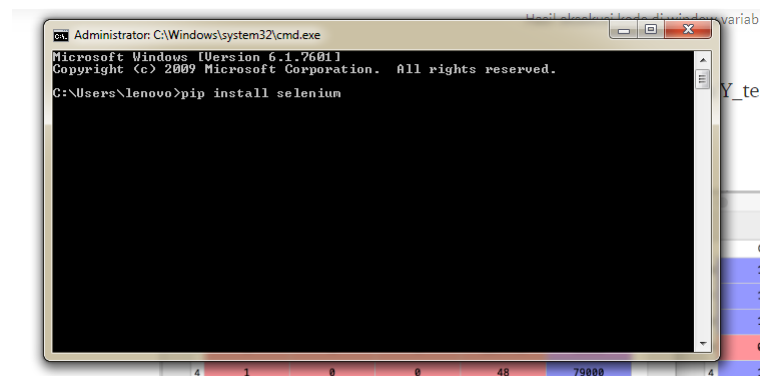


Figure 1.20: Install selenium

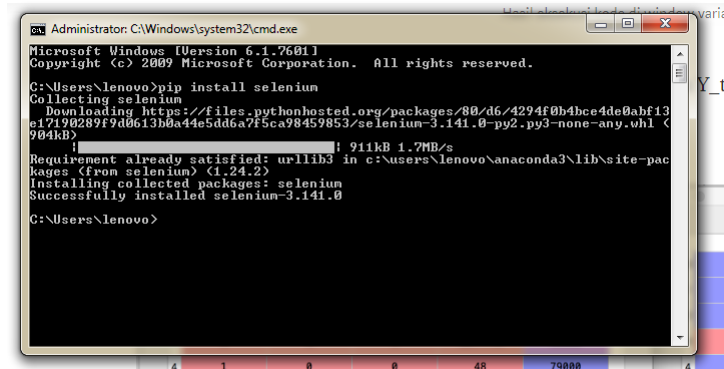


Figure 1.21: Tahap instalasi selenium

```

1 #-*- coding: utf-8 -*-
2 """
3 Created on Thu Oct 17 08:11:00 2019
4
5 @author: lenovo
6 """
7
8 from selenium import webdriver
9 driver = webdriver.Firefox()
10 driver.maximize_window()
11 driver.get("http://siap.poltekpos.ac.id/siap/besan.depan.php")
12 driver.find_element_by_name('user_name').send_keys('1184021')
13 driver.find_element_by_name('user_pass').send_keys('cirebon48')
14 driver.find_element_by_name('login').click()
  
```

Figure 1.22: Sintak pada spyder untuk otomatis login

SIAP - Politeknik Pos Indonesia

← → ↺

siap.poltekpos.ac.id/siap/besan.depan.php

🔍 🗄 📄 🏠

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

🌐

🔒

🔍

📶

<

Figure 1.23: hasil

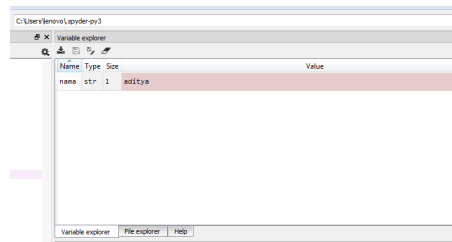


Figure 1.24: cek variabel explorer

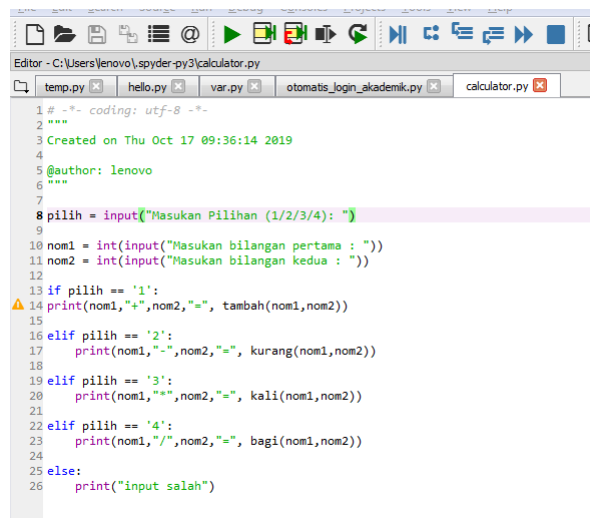


Figure 1.25: sintak eror identasi

## 1.9 Variable Explorer

Variabel explorer digunakan sebagai bawaan untuk mengedit daftar, string, kamus, array NumPy, Pandas DataFrames, dan banyak lagi, dan dapat juga histogram, plot, atau bahkan menampilkan beberapa di antaranya sebagai gambar RGB. Bisa dicek dengan mengklik variabel explorer pada spyder.

## 1.10 Identasi

Indentasi adalah bagian paragraf yang menjorok ke dalam pada baris-baris paragraf, penulisan kode python tidak memakai curly brackets "" sehingga cara membedakan blok program digunakan identasi. jenis error identasi yaitu IndentationError: expected an indented block. artinya ini berarti fungsi if memerlukan indentasi untuk membedakan blok kode.



```

32 pilih = input("Masukan pilihan (1/2/3/4): ")
33
34 nom1 = int(input("Masukkan bilangan pertama : "))
35 nom2 = int(input("Masukkan bilangan kedua : "))
36
37 if pilih == '1':
38     print(nom1,"+",nom2,"=", tambah(nom1,nom2))
39
40 elif pilih == '2':
41     print(nom1,"-",nom2,"=", kurang(nom1,nom2))
42
43 elif pilih == '3':
44     print(nom1,"*",nom2,"=", kali(nom1,nom2))
45
46 elif pilih == '4':
47     print(nom1,"/",nom2,"=", bagi(nom1,nom2))
48 else:
49     print("Input salah")

```

Figure 1.26: cek variabel explorer

# Chapter 2

## Pemrograman Dasar

### 2.1 Teori

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

- (a) Variabel global, yaitu variabel yang bisa diakses dengan semua fungsi
- (b) Variabel local, yaitu variabel yang bisa diakses dalam fungsi dan tempat variabel berada.
- (c) Variabel build-in, yaitu variabel yang sudah termasuk ada dalam python

2. Dengan menuliskan script seperti ini :

```
A=input("aditya")
```

maka untuk menampilkan ketikan perintah dibawah ini :

```
print("halo", A "Apa Kabar")
```

3. Operator dasar matematika adalah :

1. + (pertambahan)
2. - (pengurangan)
3. / (pembagian)
4. x (perkalian)

- (a) Untuk merubahnya menjadi integer gunakanlah kode `int()`

Sebagai contoh :

```
b=876
```

```
integer = int(b) #script konversi dari string ke integer
```

```
print(integer) ;untuk mencetak hasilnya;
```

- (b) Untuk merubahnya menjadi string gunakanlah kode `str()` sebagai contoh :  
`r=747`  
`string = str(r)` ;sricpt konversi dari integer ke string;  
`print(string)` ;untuk mencetak hasilnya;

4. Ada 2 perulangan yaitu while loop dan for loop.

- (a) While loop adalah perulangan yang selalu dieksekusi selama kondisi bernilai benar(true).
- (b) For loop adalah perulangan yang memiliki kemampuan untuk mengulangi item dari urutan apapun seperti list atau string

1. Contoh sricpt pada while loop :

```
Count = 0  
while (count < 9):  
    print('the count is:',count)  
    Count = count+1  
print("Good Bye!")
```

2. Contoh penerapan for loop :

```
Nomer=[1,2,3,4,5]  
For x in nomer:  
    Print(x)
```

5. Ada 3 macam kondisi diantaranya :

- (a) If  
adalah suatu kondisi yang bernilai benar atau salah, jika dalam statmennt bernilai benar akan dijalankan, tetapi sebaliknya jika statmennt bernilai salah maka tidak akan dijalankan (eror).

(b) If - Else

adalah suatu kondisi bernilai benar maka statment didalam if akan dieksekusi dan jika bernilai false maka statment yang dieksekusi adalah statment didalam else

(c) If - Elif - Else

adalah suatu kondisi Elif, lanjutan dari percabangan if dengan kondisi ini menyebabkan beberapa kemungkinan statment yang terjadi.

1. Sebagai contoh script IF yang bernilai true.

```
x=1
```

```
IF x > 0;
```

```
Print("Nilai %x adalah besar dari 0"%x) #Nilai 1 adalah besar dari 0
```

Kondisi diatas bernilai true, karena nilai x(1) lebih besar dari 0.

Sedangkan script di bawah ini merupakan contoh dari suatu kondisi bernilai false.

```
x=1
```

```
IF x > 2;
```

```
Print("Nilai %x adalah besar dari 0"%x)
```

Kondisi diatas bernilai false, maka outputan tidak akan muncul.

2. Sebagai contoh script IF-Else :

```
x=1
```

```
IF x > 5:
```

```
Print("Nilai %d adalah besar dari 5"%x)
```

```
Else: Print("Nilai %d adalah kecil dari 5"%x)
```

```
# Nilai 1 adalah kecil dari 5
```

sebaliknya, ubahlah nilai x menjadi 20.

```
x=20
```

```
IF x > 5:
```

```
Print("Nilai %d adalah besar dari 5"%x)
```

```
Else:
```

```
Print("Nilai %d adalah kecil dari 5"%x)
```

3. Sebagai contoh script IF ELIF ELSE :

```
x=5
```

```

if x < 5:
    Print("Nilai %d adalah kecil dari 5"%x)
elif x==5:
    Print("Nilai %d adalah sama dengan 5"%x)
else:
    Print("Nilai %d adalah besar dari 5"%x)

```

6. diantara sintak-sintak error yang ditemui antara lain :

- (a) `TypeError:unsupported operand type(s) for +:'int' and 'str'`  
penangan error ini bisa ditandai menggunakan casting operand kedua menjadi integer
- (b) `TypeError:can only concatenate str(not'int') to str`  
penanganan ini ditandai dengan menggunakan casting operand kedua menjadi string.

7. Try Except adalah bentuk penanganan error yang terdapat dalam bahasa pemrograman (python). Contoh penanganannya :

Setiap bilangan yang dibagi 0 akan terjadi error karena ini merupakan ketentuan dari awal dan tidak bisa diesekusi, tetapi dengan menggunakan try except dapat ditangani walaupun akan terjadi error dibawah ini :

```

x=0
Try:
x=9/0
Except exception,e;
print e
print x=1

```

Maka akan muncul peringatan error integer devision or modulo by zero 1

## 2.2 Ketrampilan Pemrograman

Buat program di python dengan ketentuan:

1. Berikut ini merupakan soal praktikum no 1 dengan soal sebagai berikut :  
Buatlah luaran huruf yang dirangkai dari tanda bintang, pagar atau plus dari NPM kita. Tanda bintang untuk  $NPM \bmod 3 = 0$ , tanda pagar untuk  $NPM \bmod 3 = 1$ , tanda plus untuk  $NPM \bmod 3 = 2$ . Contoh script :

```
# -*- coding: utf-8 -*-
"""
Created on Tue Oct 22 21:29:47 2019
@author: lenovo
"""

print("***      ***      *****      *****      *****      *****      ***");
print("***      ***      **      **      **      **      **      **      **      ***");
print("***      ***      ****      *****      **      **      *****      ***");
print("***      ***      **      **      **      **      **      **      ***");
print("***      ***      *****      **      *****      *****      ***");
```

2. Berikut ini hasil script dari soal no 2 praktikum

```
# -*- coding: utf-8 -*-
"""
Created on Tue Oct 22 21:04:07 2019
@author: lenovo
"""

npm=int(input("masukan npm anda : "))
TwoLastDigit=abs(npm)%100 # modulus menentukan ambil 2 digit terakhir
for i in range(TwoLastDigit):
    print("Halo, ", npm, " Apa kabar ?")
```

Maka hasil Outputnya adalah menampilkan 'Halo, 1184021 Apa kabar ?' sebanyak 21x tampilan.

```
masukan npm anda : 1184021
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
```

```

Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?
Halo, 1184021 Apa kabar ?

```

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

```

# -*- coding: utf-8 -*-
"""
Created on Tue Oct 22 22:19:12 2019
@author: lenovo
"""

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?")

```

Maka hasilnya adalah 3x muncul "Hallo, 021 Apa Kabar?"

```

Masukan NPM : 1184021
Hallo, 021 Apa kabar?
Hallo, 021 Apa kabar?
Hallo, 021 Apa kabar?

```

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

```
# -*- coding: utf-8 -*-
"""
Created on Tue Oct 22 22:43:26 2019
@author: lenovo
"""
npm=int(input("Masukan NPM : "))
key=npm%1000
str_key=str(key)
print("hello, "+str_key[0]+" Apa kabar ?")
```

Maka Hasil nya adalah sebagai berikut :

```
Masukan NPM : 1184021
hello, 2 Apa kabar ?
```

5. Pada soal no 5 ini merupakan penggunaan perulangan dan kondisi. contoh penerapan dalam script nya adalah sebagai berikut :

```
# -*- coding: utf-8 -*-
"""
Created on Tue Oct 22 22:47:07 2019
@author: lenovo
"""
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]
for x in a,b,c,d,e,f,g:
    print(x,end = ""),

```

Maka akan keluar outputan :  
 Masukan NPM : 1184021  
 1184021

6. Setelah dilakukan perulangan maka seluruh variabel dijumlahkan . maka scrip-nya sebagai berikut :

```

# -*- coding: utf-8 -*-
"""
Created on Tue Oct 22 20:58:14 2019
@author: lenovo
"""
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)

```

Hasilnya adalah seperti ini :

```

Masukan NPM : 1184021
1
2
10
14
14
16
17

```

7. Sama hal-nya pada no 6, yang membedakannya pada no 7 ini adalah variabel tersebut dikalikan dengan aritmatikan perkalian, contohnya sricpt sebagai berikut :

```

# -*- coding: utf-8 -*-
"""
Created on Tue Oct 22 23:02:17 2019
@author: lenovo
"""
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:
    conv*=int(x)
    print(conv)

```

Maka Hasilnya adalah

Masukan NPM : 1184021

```

1
1
8
32
0
0
0

```

8. Dilakukannya Proses yang outputannya vertikal, script nya berupa seperti ini :

```

# -*- coding: utf-8 -*-
"""
Created on Tue Oct 22 21:13:37 2019
@author: lenovo
"""
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]
for x in a,b,c,d,e,f,g:
    print(x)

```

Maka Hasilnya adalah :

```

Masukan NPM : 1184021
1
1
8
4
0
2
1

```

9. Menampilkan digit genap script nya adalah :

```

# -*- coding: utf-8 -*-
"""
Created on Tue Oct 22 23:10:26 2019
@author: lenovo
"""
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==0:
        if int(x)==0:
            x=""
        print(x,end = "")

```

Maka Hasilnya adalah :

```

Masukan NPM : 1184021
8402

```

10. Dan hasil dari proses ganjil , script berupa :

```

# -*- coding: utf-8 -*-
"""
Created on Tue Oct 22 23:16:50 2019
@author: lenovo
"""
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 = "")

```

Hasilnya adalah

```

Masukan NPM : 1184021
111

```

11. Untuk menampilkan bilangan prima maka dilakukannya script dibawah ini :

```

# -*- coding: utf-8 -*-
"""
Created on Tue Oct 22 23:20:27 2019
@author: lenovo
"""
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) > 1:
        for i in range(2,int(x)):
            if (int(x) % i) == 0:
                break
        else:
            print(int(x),end =""),

```

Maka hasilnya adalah :

```

Masukan NPM : 1184021
tidak muncul outputan .

```

## 2.3 Ketrampilan Penanganan Error

Bagian Penanganan error dari script python.

1. Peringatan error yang terjadi adalah "SyntaxError: invalid syntax"
2. Contoh script yang di gunakan pada file 2err.py adalah

```

# -*- coding: utf-8 -*-
"""
Created on Thu Oct 24 10:32:47 2019
@author: lenovo
"""
q="2"

```

```
r=6
try:
    p+r
except:
    print("error, karena hanya bisa menggabungkan string dengan string")
```



# Bibliography