

WHATSAPP TANPA DERITA

WHATSAPP TANPA DERITA

Dengan Bot

**Dinda Majesty, Tri Angga Dio Simamora, Rolly M. Awangga, Syafrial
Fachri Pane**
Politeknik Pos Indonesia



Kreatif Industri Nusantara

Penulis:

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:

M. Yusril Helmi Setyawan

Penyunting:

Syafrial Fachrie Pane

Khaera Tunnisia

Diana Asri Wijayanti

Desain sampul dan Tata letak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasisih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara
apapun tanpa ijin tertulis dari penerbit

*'Jika Kamu tidak dapat
menahan lelahnya
belajar, Maka kamu harus
sanggup menahan
perihnya Kebodohan.'*

Imam Syafi'i

CONTRIBUTORS

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

SYAFRIAL FACHRI PANE, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

DINDA MAJESTY, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

TRI ANGGA DIO SIMAMORA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

CONTENTS IN BRIEF

1 Mengenal Python	1
2 Instalasi Anaconda	19
3 Instalasi Pip dan Perintah CLI	33
4 Fungsi, Method, dan Kelas	37
5 Pengelolaan File Csv	45

DAFTAR ISI

Daftar Gambar	xiii
Daftar Tabel	xv
Foreword	xix
Kata Pengantar	xxi
Acknowledgments	xxiii
Acronyms	xxv
Glossary	xxvii
List of Symbols	xxix
Introduction	xxxi
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
1 Mengenal Python	1
1.1 Sejarah Python	1
1.1.1 Perbedaan Python 2 dan 3	2
1.1.2 Implementasi dan Penggunaan Python di Perusahaan Dunia	2
	ix

1.1.3	Jenis-Jenis Variabel	3
1.1.4	Input dan Output	4
1.1.5	Operator Aritmatika dan Konversi Tipe Data	4
1.1.6	Perulangan	5
1.1.7	IF Statement	6
1.2	Praktek Pertama Python	7
1.2.1	Modulus	7
1.2.2	Hello NPM	8
1.2.3	Hello NPM (3 Digit Belakang)	8
1.2.4	Hello NPM (Digit ke-3)	8
1.2.5	Variabel Alfabet	8
1.2.6	Penjumlahan NPM	9
1.2.7	Perkalian NPM	9
1.2.8	Print Vertical	10
1.2.9	Digit Genap NPM	10
1.2.10	Digit Ganjil NPM	10
1.2.11	Bilangan Prima NPM	11
1.3	Peringatan Error dan Cara Mengatasinya	11
1.3.1	Except	12
1.3.2	Run Script Hello World di Spyder	13
1.3.3	Pemakaian Variable Explorer	13
1.4	Indentasi	15
1.4.1	Penjelasan Indentasi	15
1.4.2	Jenis-Jenis Error Indentasi	15
1.4.3	Cara Membaca Error	15
1.4.4	Cara Menangani Error	15
1.4.5	Quiz: 1	18
2	Instalasi Anaconda	19
2.1	Instalasi	19
2.1.1	Instalasi Anaconda 3 Windows 10 x64	19
2.1.2	Update Anaconda dan Spyder	26
2.1.3	Instalasi Anaconda Ubuntu 19.04	26
2.1.4	Konfigurasi <i>Python</i>	30
2.1.5	Quiz 2	31
3	Instalasi Pip dan Perintah CLI	33
3.0.1	Instalasi Pip	33

3.0.2	Command Line Interface/Interpreter	35
4	Fungsi, Method, dan Kelas	37
4.1	Fungsi	37
4.2	Package	38
4.3	Kelas, Objek, Atribut, dan Method	38
4.3.1	Pemanggilan Library Kelas	39
4.3.2	Pemakaian Package Fungsi Apabila File Didalam Folder	39
4.3.3	Pemakaian Package Kelas Apabila File didalam Folder	39
4.4	Praktek Membuat Method dan Pemanggilan Method	39
4.4.1	Modulus	39
4.4.2	Hello NPM	40
4.4.3	Hello NPM (3 Digit Belakang)	40
4.4.4	Hello NPM (Digit ke-3)	41
4.4.5	Variabel Alfabet	41
4.4.6	Penjumlahan NPM	41
4.4.7	Perkalian NPM	41
4.4.8	Print Vertical	42
4.4.9	Digit Genap NPM	42
4.4.10	Digit Ganjil NPM	42
4.4.11	Bilangan Prima NPM	42
4.4.12	Pemanggilan Fungsi pada Main.py	43
5	Pengelolaan File Csv	45
5.1	Pengelolaan File CSV, Sejarah, dan Contoh	45
5.2	Aplikasi yang bisa Menciptakan File CSV	46
5.3	Menulis dan Membaca File CSV pada Ms.Excel	47
5.3.1	Sejarah Library CSV	48
5.3.2	Sejarah Library Pandas	48
5.3.3	Fungsi - fungsi yang terdapat di library CSV	48
5.4	Praktek CSV	49
5.4.1	CSV List	49
5.4.2	CSV Dictionary	50
5.4.3	Hello NPM (Pandas List)	50
5.4.4	Hello NPM (Pandas Dictionary)	50
5.4.5	Pandas Date	50
5.4.6	Pandas Ubah Index	51
5.4.7	Pandas Ubah Column	51

5.4.8	Main CSV	51
5.4.9	Main Pandas	51

DAFTAR GAMBAR

1.1	Perbedaan Python 2 dan Python 3	2
1.2	<i>Launch Spider</i>	13
1.3	<i>Print Hello World</i>	14
1.4	<i>Hello World</i>	14
1.5	<i>Variable Explorer</i>	14
1.6	<i>Indentasi</i>	16
1.7	<i>Error Indentasi</i>	16
1.8	<i>Syntax Error</i>	17
1.9	<i>Syntax yang Telah Diperbaiki</i>	17
2.1	Run Setup Anaconda	20
2.2	Setup Loading	20
2.3	Welcome to Anaconda Setup	21
2.4	<i>License Agreement</i>	21

2.5	<i>Just Me(recomended)</i>	22
2.6	<i>Pilih lokasi</i>	23
2.7	<i>Centang Anaconda to my PATH</i>	23
2.8	<i>Waiting Installation Complete</i>	24
2.9	<i>Installation Complete</i>	24
2.10	<i>Anaconda+JetBrains</i>	25
2.11	<i>Thanks for install Anaconda</i>	26
2.12	Gambar halaman download	27
2.13	Gambar install anaconda	27
2.14	Gambar eksekusi anaconda	28
2.15	Gambar anaconda license agreement	28
2.16	Gambar perintah yes or no	29
2.17	Gambar path anaconda	29
2.18	Gambar perintah install spyder3	30
2.19	Gambar setpath	31
3.1	<i>Install pip</i>	34
3.2	<i>Install pip Selesai</i>	34
3.3	<i>Melihat Versi pip</i>	35
3.4	<i>CLI in Command Prompt</i>	36
5.1	Contoh Penulisan CSV pada Excel	47

DAFTAR TABEL

Listings

1.1	Input dan Output	4
1.2	While Loop	5
1.3	For Loop	5
1.4	Nested Loop	5
1.5	if Statement	6
1.6	Elif	6
1.7	Else	6
1.8	Nested If	7
1.9	Modulus	7
1.10	Hello NPM	8
1.11	3 Digit Belakang	8
1.12	Digit ke-3	8
1.13	Variabel Alfabet	9
1.14	Penjumlahan NPM	9
1.15	Perkalian NPM	9
1.16	Print Vertical	10
1.17	Digit Genap NPM	10
1.18	Digit Ganjil NPM	10

1.19 Bilangan Prima NPM	11
1.20 Try Except	12
4.1 Modulus	40
4.2 Hello NPM	40
4.3 3 Digit Belakang	40
4.4 Digit ke-3	41
4.5 Variabel Alfabet	41
4.6 Penjumlahan NPM	41
4.7 Perkalian NPM	41
4.8 Print Vertical	42
4.9 Digit Genap NPM	42
4.10 Digit Ganjil NPM	42
4.11 Bilangan Prima NPM	42
4.12 Bilangan Prima NPM	43
5.1 Contoh CSV	45
5.2 Contoh Kode Python to Read CSV	46
5.3 Contoh Kode Python to Read CSV	46
5.4 Contoh CSV	47
src/fcsv.py	48
src/fcsv.py	48
src/fcsv.py	49
src/fcsv.py	49
5.5 CSV List	50
5.6 CSV Dictionary	50
5.7 Pandas List	50
5.8 Pandas Dictionary	50
5.9 Pandas Date	51
5.10 Pandas Ubah Index	51
5.11 Pandas Ubah Column	51
5.12 Main CSV	51
5.13 Digit Genap NPM	51

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini berisi pengantar serta tata cara yang harus dilakukan dalam pembuatan whatsapp chatbot. Buku ini diharapkan dapat membantu orang-orang memahami cara kerja chatbot, hal yang dibutuhkan dalam pembuatan aplikasi chatbot serta cara membuat chatbot.

TIM PENULIS

Bandung, Jawa Barat

Januari, 2020

ACKNOWLEDGMENTS

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

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

R. M. A.

ACRONYMS

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

GLOSSARY

chatbot	Merupakan sebuah teknologi layanan obrolan yang ditangani oleh sistem atau robot.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald

SYMBOLS

A Amplitude

$\&$ Propositional logic symbol

a Filter Coefficient

B Number of Beats

INTRODUCTION

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

Informatics Research Center
Bandung, Jawa Barat, Indonesia

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

BAB 1

MENGENAL PYTHON

Dalam praktek membuat whatsapp chatbot ini, hal pertama yang harus dilakukan adalah mengenal bahasa pemrograman Python dan mengetahui tentang software yang digunakan yaitu Anaconda3.

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.

1.1.1 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.

Gambar 1.1 Perbedaan Python 2 dan Python 3

1.1.2 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 0MQ. 0MQ 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 sajak 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 pemprograman 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 queuenya atau fitur dimana setiap pengguna dapat berbagi foto atau video ke beberapa social network lainnya seperti facebook, twitter, dan lain-lainnya.

1.1.3 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.

1.1.4 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 1.1 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.

1.1.5 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.

1.1.6 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 1.2 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 minum", minuman)

```

Listing 1.3 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 1.4 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.

1.1.7 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 > a:
5     print("a lebih besar dari b")

```

Listing 1.5 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 1.6 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 1.7 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 1.8 Nested If

1.2 Praktek Pertama Python

1.2.1 Modulus

Praktek kali ini kita akan mencoba praktek menggunakan modulus atau sisa bagi, kita membuat inputan terlebih dahulu menggunakan perintah input. Kemudian buatlah variabel untuk menampung hasil sisa bagi dari jumlah yang diinputkan. Misalnya, teman-teman menginputkan nilai yaitu 1184011, maka hasil modulus 1184011 mod 3 adalah 1 maka print 1184011 menggunakan tanda pagar, jika hasil modulus adalah 0 maka print 1184011 menggunakan tanda bintang.

```

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 1.9 Modulus

1.2.2 Hello NPM

Setelah selesai praktek modulus kita akan menampilkan output berupa kalimat "Hello 1184011 Apa Kabar" sebanyak 2 digit belakang angka, yaitu angka 11, maka akan berulang sebanyak 11 kali.

```

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 1.10 Hello NPM

1.2.3 Hello NPM (3 Digit Belakang)

Jika telah selesai, maka selanjutnya kita akan melakukan praktek untuk menampilkan output berupa kalimat "Hallo 011 Apa Kabar?" sebanyak angka keenam ditambah angka ketujuh atau 1 ditambah 1, sehingga kalimat tersebut akan berulang sebanyak 2 kali.

```

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 1.11 3 Digit Belakang

1.2.4 Hello NPM (Digit ke-3)

Kemudian kita akan melakukan praktek untuk menampilkan output berupa kalimat "Hallo 0 Apa Kabar?".

```

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

```

Listing 1.12 Digit ke-3

1.2.5 Variabel Alfabet

Teman-teman, sekarang mari kita coba menambahkan variabel pada angka 1184011, tambahkan abcdefg kedalam sebuah variabel dan tambahkan variabel index dengan

nilai 0, buatlah perulangan agar variabel huruf menyesuaikan dengan variabel angka sehingga menjadi a=1, b=1, c=8, d=4, e=0, f=1 ,g=1.

```

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 1.13 Variabel Alfabet

1.2.6 Penjumlahan NPM

Praktek selanjutnya adalah menjumlahkan angka 1184011 dengan menerapkan perulangan dan penambahan. Apabila nilai 1 telah didapatkan maka akan ditambahkan dengan nilai 1 sehingga menjadi 2, kemudian nilai 2 ditambahkan lagi dengan nilai selanjutnya yaitu 8 sehingga menjadi 10, begitu seterusnya.

```

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 1.14 Penjumlahan NPM

1.2.7 Perkalian NPM

Praktek selanjutnya adalah mengalikan angka 1184011 dengan menerapkan perulangan dan perkalian. Apabila nilai 1 telah didapatkan maka akan dikalikan dengan nilai 1 sehingga menjadi 1, kemudian nilai 1 dikalikan lagi dengan nilai selanjutnya yaitu 8 sehingga menjadi 8, begitu seterusnya.

```

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 1.15 Perkalian NPM

1.2.8 Print Vertical

Melakukan print secara vertikal hanya perlu menambahkan perulangan terhadap nilai 1184011.

```

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

```

Listing 1.16 Print Vertical

1.2.9 Digit Genap NPM

Selanjutnya, mari kita lakukan print hanya terhadap digit genap pada angka 1184011 dengan memanfaatkan perulangan, if statement dan operator logika. Logika yang akan diterapkan adalah masing-masing angka 1184011 akan di cek terlebih dahulu, apakah angka tersebut memiliki angka yang apabila dibagi 2 akan menghasilkan sisa bagi sama dengan 0 dan angka tersebut tidak boleh sama dengan 0, karena angka 0 bukan merupakan angka ganjil maupun genap.

```

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 1.17 Digit Genap NPM

1.2.10 Digit Ganjil NPM

Selanjutnya, mari kita lakukan print hanya terhadap digit ganjil pada angka 1184011 dengan memanfaatkan perulangan, if statement dan operator logika. Logika yang akan diterapkan adalah masing-masing angka 1184011 akan di cek terlebih dahulu, apakah angka tersebut memiliki angka yang apabila dibagi 2 akan menghasilkan sisa bagi tidak sama dengan 0 dan angka tersebut tidak boleh sama dengan 0, karena angka 0 bukan merupakan angka ganjil maupun genap.

```

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

```

```

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

```

Listing 1.18 Digit Ganjil NPM

1.2.11 Bilangan Prima NPM

Praktek terakhir adalah menampilkan hasil dari bilangan prima angka 1184011 dengan cara apabila angka 1184011 memiliki angka yang merupakan angka prima maka akan ditampilkan, logika yang akan diterapkan adalah apabila angka kecil sama dengan angka 1 maka angka tersebut bukan bilangan prima. Jika angka 1184011 dibagi 2 setelah itu dibagi dengan angka itu sendiri dan menghasilkan sisa bagi sama dengan 0, maka angka tersebut bukan bilangan prima

```

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 1.19 Bilangan Prima NPM

1.3 Peringatan Error dan Cara Mengatasinya

1. NameError, terjadi apabila kode mengeksekusi nama yang tidak terdefenisikan. 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 eror 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. eror ini dapat diatasi dengan mengubah tipe data string menjadi integer.

```
a = "10"
b = 5

print(a + b)
```

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

4. IndentationError, terjadi apabila kode perulangan atau pengkondisian tidak menjorok kedalam (tidak menggunakan identasi), 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 eror IndentationError: expected an indented block

1.3.1 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")
```

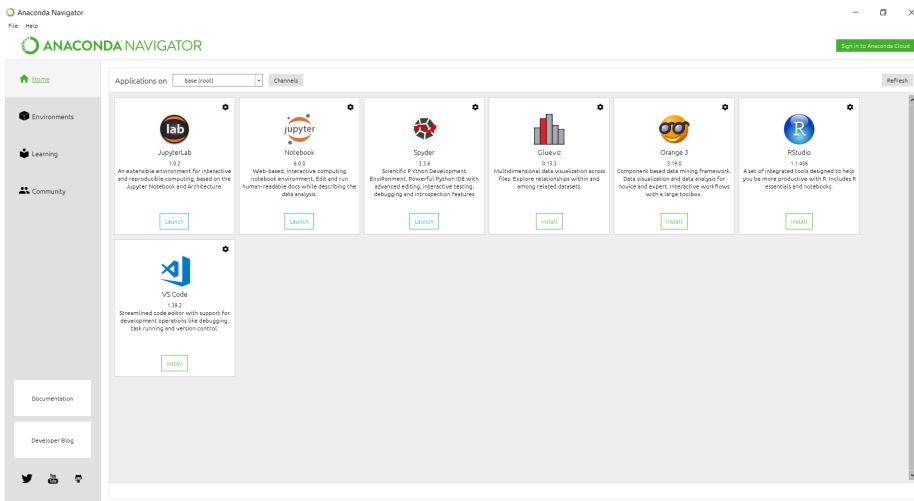
Listing 1.20 Try Except

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.

1.3.2 Run Script Hello World di Spyder

Jika teman-teman telah berhasil melakukan instalasi anaconda3 maka teman-teman akan memiliki sofware spyder, pada software ini teman-teman akan mengetikkan kode program yang akan teman-teman buat nantinya. Sekarang ayo kita coba menjalankan program untuk mencetak kata *Hello world*.

1. Buka anaconda navigator, lalu klik launch seperti pada gambar 1.2.

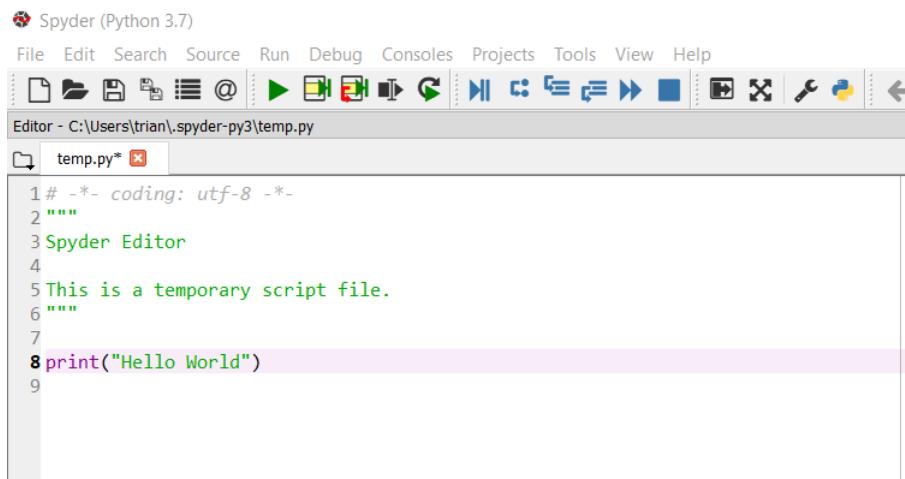


Gambar 1.2 Launch Spider

2. ketikkan `print("Hello World")` dan run spyder dengan cara mengklik tombol berwarna hijau yang terletak ditengah toolbar, untuk lebih jelas dapat dilihat pada gambar 1.3.
3. hasilnya akan tampak seperti pada gambar 1.4

1.3.3 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. Contoh penggunaan variabel explorer dapat teman-teman lihat pada gambar 1.5.



Gambar 1.3 Print Hello World

```

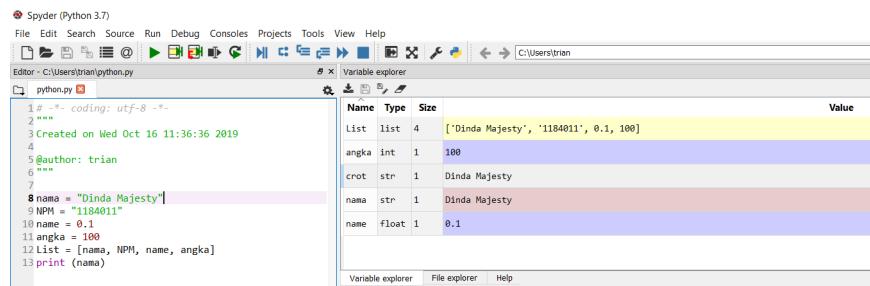
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

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

```

Gambar 1.4 Hello World



Gambar 1.5 Variable Explorer

1.4 Indentasi

1.4.1 Penjelasan Indentasi

Identasi adalah bagian paragraf yang menjorok ke dalam pada baris-baris paragraf. Mengatur indentasi dapat menggunakan tab atau spasi. Identasi 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
```

1.4.2 Jenis-Jenis Error Indentasi

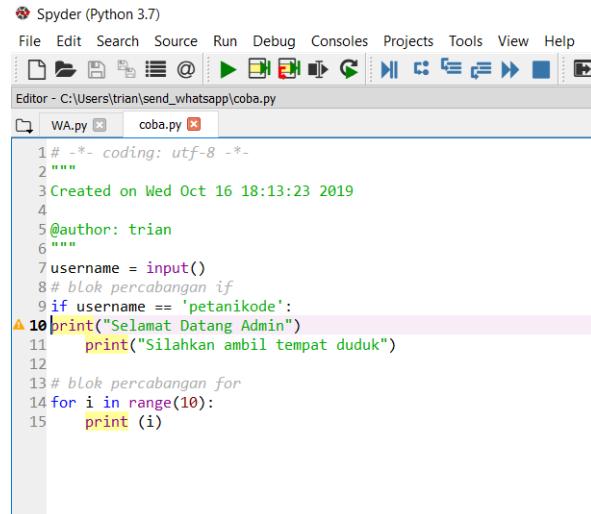
IndentationError: unexpected indent. Error diatas terjadi apabila syntax kekurangan tab atau spasi. Contoh error indentasi dapat dilihat pada gambar 1.6. Apabila di running akan memunculkan error seperti pada gambar 1.7.

1.4.3 Cara Membaca Error

Jika terjadi error maka cari di line berapa error terjadi, pada gambar 1.7 terdapat error indentasi pada line 10.

1.4.4 Cara Menangani Error

Menangani error indentasi dapat dilakukan dengan cara menambahkan tab atau space pada line yang error. Untuk lebih jelasnya dapat teman-teman lihat pada gambar 1.8.



Gambar 1.6 Indentasi

```

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-6-0a8945c5cd46>", line 1, in <module>
    runfile('C:/Users/trian/send_whatsapp/coba.py', wdir='C:/Users/trian/
send_whatsapp')

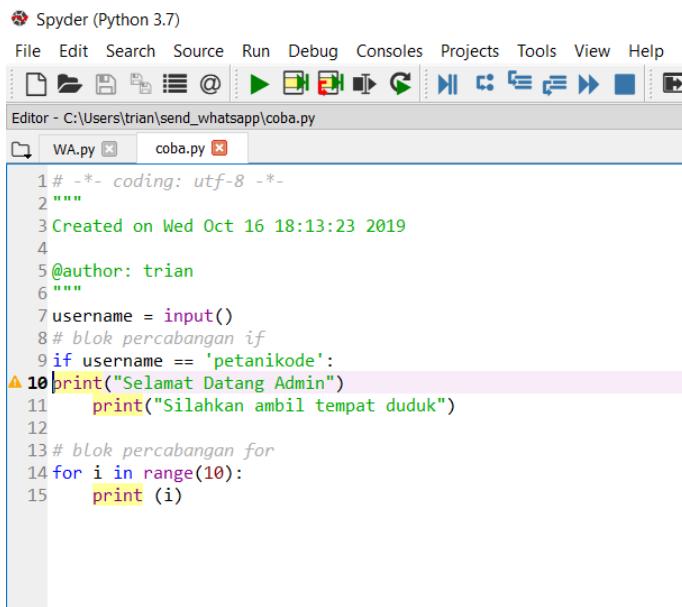
  File "C:/Users/trian/Anaconda3/lib/site-packages/spyder_kernels/customize
.spydercustomize.py", line 827, in runfile
    execfile(filename, namespace)

  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

```

Gambar 1.7 Error Indentasi



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. The title bar says "Editor - C:\Users\trian\send_whatsapp\coba.py". There are two tabs open: "WA.py" and "coba.py". The "coba.py" tab contains the following Python code:

```

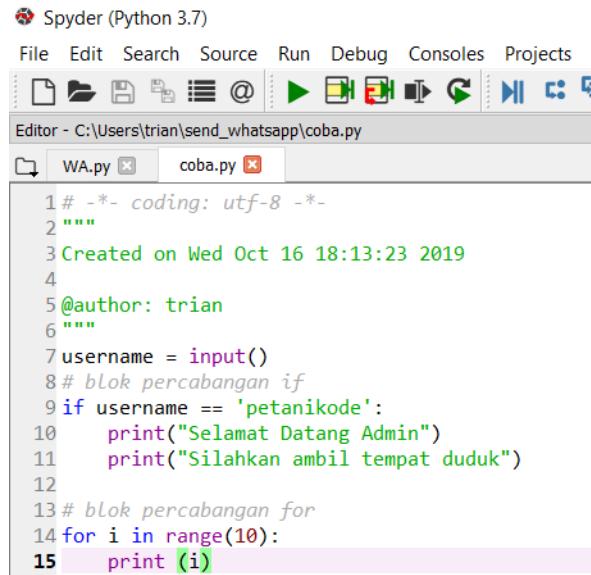
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 16 18:13:23 2019
4
5 @author: trian
6 """
7 username = input()
8 # blok percabangan if
9 if username == 'petanikode':
10    print("Selamat Datang Admin")
11    print("Silahkan ambil tempat duduk")
12
13 # blok percabangan for
14 for i in range(10):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

```

The line "print (i)" is highlighted in yellow, indicating a syntax error.

Gambar 1.8 *Syntax Error*

Penulisan syntax identasi yang benar dapat dilihat pada gambar 1.9.



The screenshot shows the Spyder Python 3.7 IDE interface. The menu bar, toolbar, and title bar are identical to the previous screenshot. The "coba.py" tab now contains the following corrected Python code:

```

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)

```

The line "print (i)" is no longer highlighted in yellow, indicating that the syntax error has been fixed.

Gambar 1.9 *Syntax yang Telah Diperbaiki*

1.4.5 Quiz: 1

1. Siapakah pencipta Python?
 - a) Guido van Rossum
 - b) Guido van Persie
 - c) Guido van Linguini
 - d) Guido van Laptop
2. Pada tahun berapa python dikembangkan?
 - a) 1999
 - b) 1995
 - c) 1945
 - d) 1990
3. Apa itu indentasi?
 - a) Bagian paragraf yang menjorok ke dalam
 - b) Bagian paragraf yang menjorok ke sungai
 - c) Bagian paragraf yang menjorok ke laut
 - d) Bagian paragraf yang menjorok ke hati
4. Apa itu try except?
 - a) Perintah untuk penanganan masalah hidup
 - b) Perintah untuk penanganan masalah error
 - c) Perintah untuk penanganan masalah galau
 - d) Perintah untuk penanganan masalah sakit perut
5. Apa itu if statement?
 - a) Kondisi yang digunakan untuk percabangan logika
 - b) Kondisi yang digunakan untuk percabangan pohon
 - c) Kondisi yang digunakan untuk percabangan tunas
 - d) Kondisi yang digunakan untuk percabangan akar

BAB 2

INSTALASI ANACONDA

2.1 Instalasi

Instalasi pada kali ini menggunakan 2 sistem operasi yaitu Windows 10 x64 dan Ubuntu 19.04. Hal yang dibutuhkan adalah laptop dengan os windows atau ubuntu dan internet.

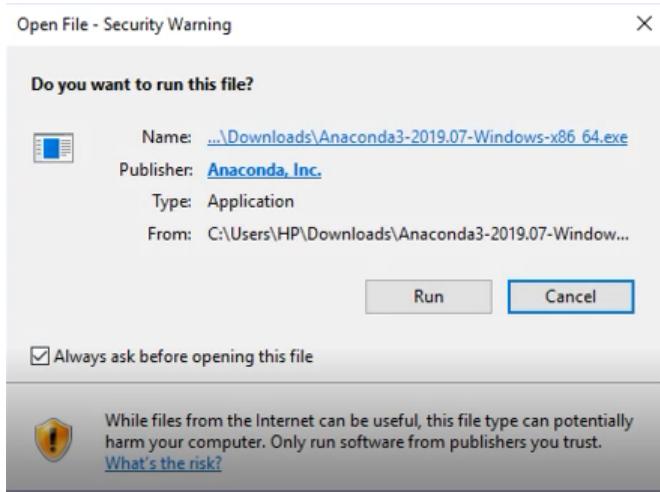
2.1.1 Instalasi Anaconda 3 Windows 10 x64

Hal yang harus diperhatikan sebelum melakukan instalasi *Anaconda Python*

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

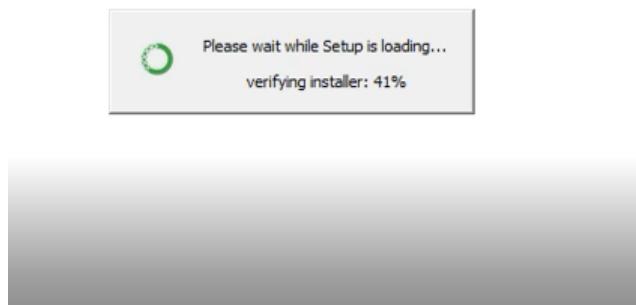
Berikut langkah-langkah instalasi anaconda.

1. Buka aplikasi *installer Anaconda* yang telah didownload lalu akan muncul gambar 2.1, lalu pilih run untuk menjalankan proses instalasi.



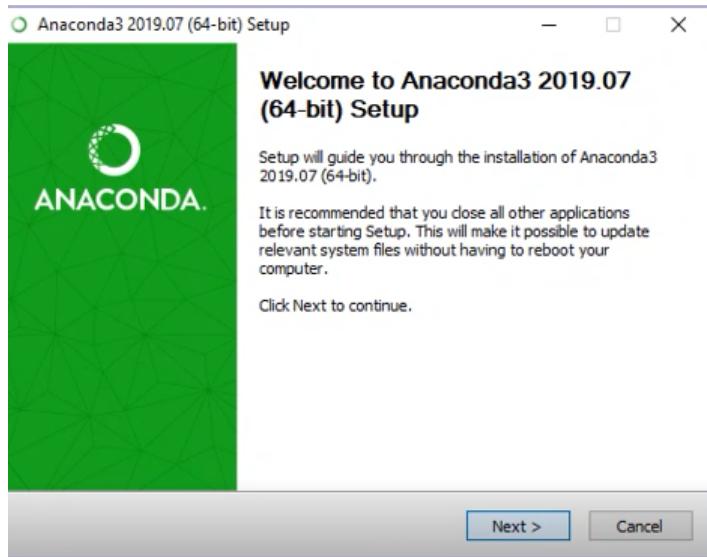
Gambar 2.1 Run Setup Anaconda

2. Tunggu hingga *setup loading* selesai seperti pada gambar 2.2.



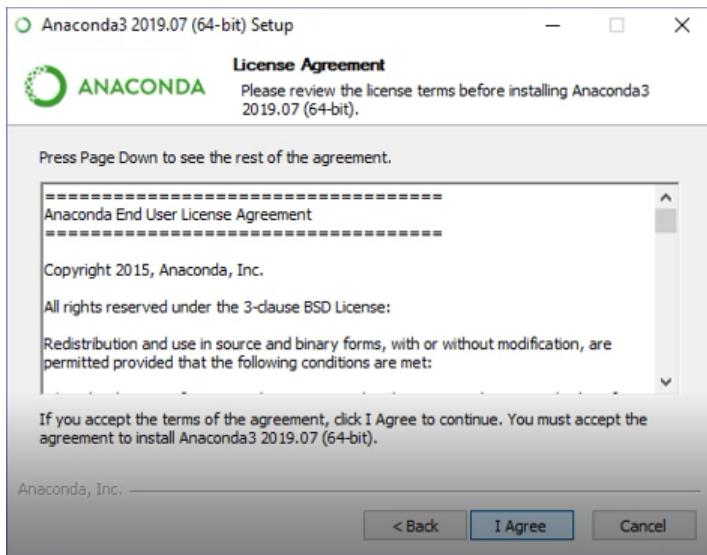
Gambar 2.2 Setup Loading

3. Jika *setup loading* telah selesai, maka klik *next* seperti pada gambar 2.3.



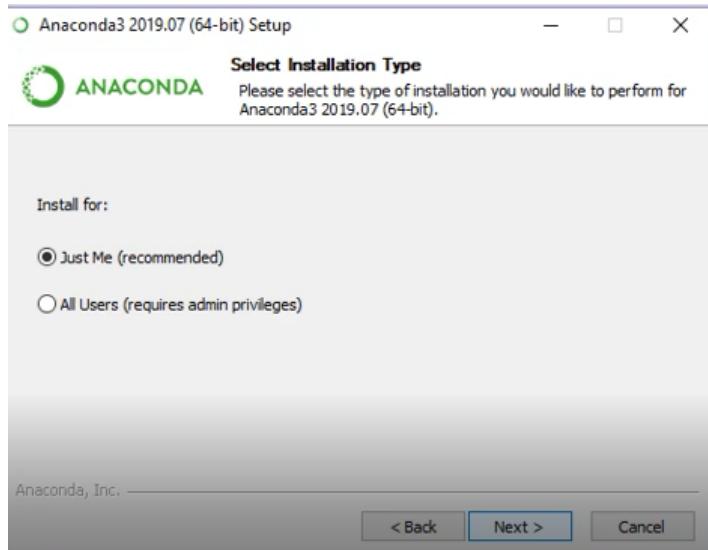
Gambar 2.3 Welcome to Anaconda Setup

4. Pada *License Agreement* klik *I Agree* karena jika teman-teman tidak menyetujui lisensi anaconda maka teman-teman tidak akan bisa melanjutkan proses instalasi. lakukan langkah ini seperti pada gambar 2.4



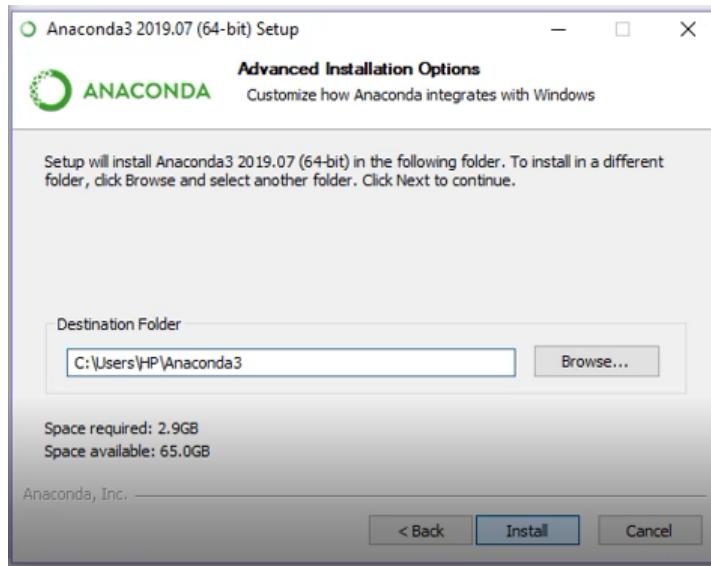
Gambar 2.4 License Agreement

5. Kemudian pilih *Just Me(Recomended)* agar sesuai dengan komputer yang digunakan, kemudian klik *next* seperti pada gambar 2.5.



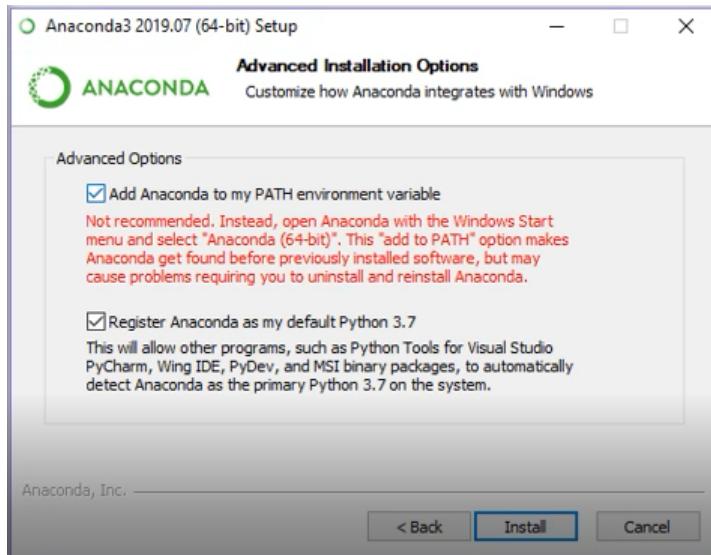
Gambar 2.5 *Just Me(recomended)*

6. Kemudian pilih direktori tempat kita akan *menginstall anaconda* seperti pada gambar 2.6



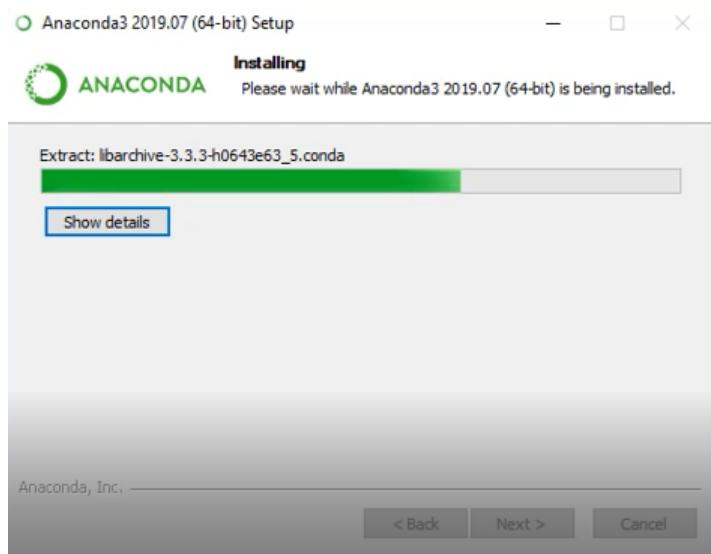
Gambar 2.6 Pilih lokasi

7. Kemudian centang *Add Anaconda to my Path environment variable*, agar saat melakukan instalasi package anaconda, package tersebut akan langsung tertuju ke path anaconda tidak ke aplikasi yang lain. kemudian Klik *install* seperti pada gambar 2.7.



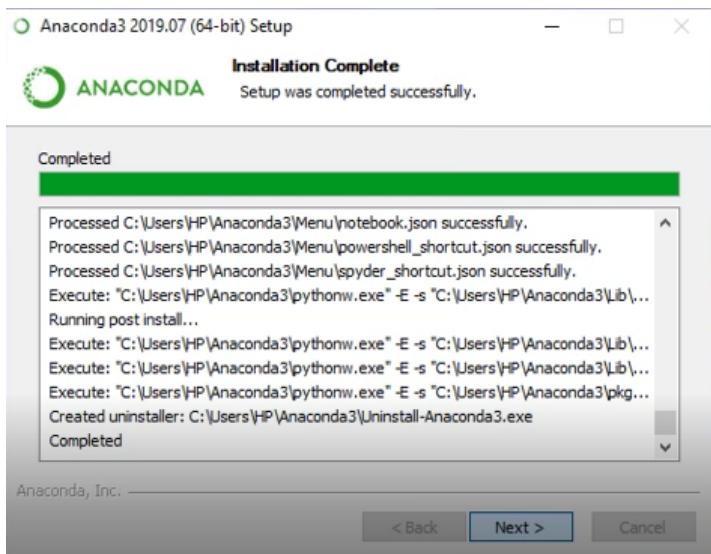
Gambar 2.7 Centang Anaconda to my PATH

8. Tunggu sampai proses *installasi* selesai seperti pada gambar 2.8.



Gambar 2.8 Waiting Installation Complete

9. Apabila instalasi telah selesai maka akan terlihat seperti gambar 2.9, kemudian klik *next*



Gambar 2.9 Installation Complete

10. apabila muncul gambar 2.10, maka klik *next*



Gambar 2.10 *Anaconda+JetBrains*

11. Jika instalasi telah selesai maka akan ada ucapan terima kasih telah menginstall anaconda 3 seperti pada gambar 2.11, hal ini menandakan bahwa teman-teman telah selesai dan berhasil melakukan instalasi anaconda. Kemudian klik *finish* untuk mengakhiri instalasi.



Gambar 2.11 *Thanks for install Anaconda*

2.1.2 Update Anaconda dan Spyder

Kenapa kita harus melakukan update anaconda dan spyder? melakukan update diperlukan agar software yang kita gunakan merupakan software yang terbaru, karena versi lama dan versi baru akan memiliki banyak perbedaan dan akan menjadi masalah nantinya ketika kita membuat program atau mengimportkan modul-modul python yang digunakan. Berikut 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

Berikut 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

2.1.3 Instalasi Anaconda Ubuntu 19.04

Untuk instalasi python pada Ubuntu 19.04 dibutuhkan sebagai berikut:

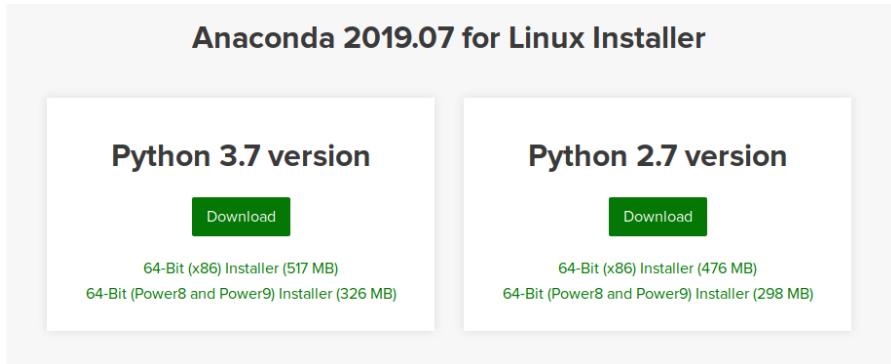
1. Internet

2. Anaconda installer (64bit or 32bit)

3. enter, dan yes atau no

Ikuti langkah berikut:

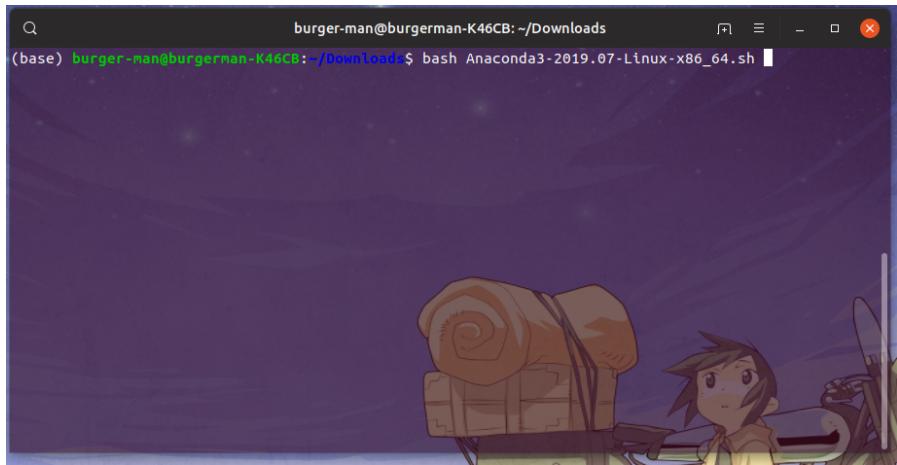
- Pertama kita kunjungi situs <https://www.anaconda.com/distribution/#download-section> seperti gambar 2.12 dan pilih **64-Bit (x86) Installer (517 MB)**



Gambar 2.12 Gambar halaman download

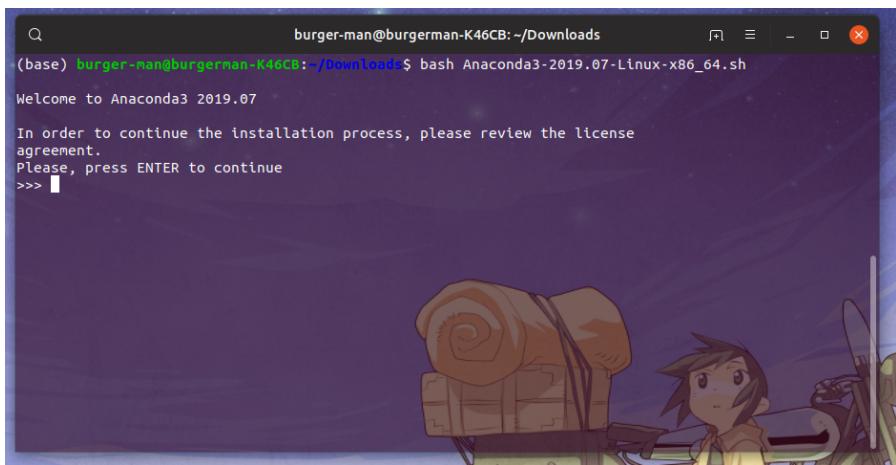
2. Kedua kita buka **terminal** kita lalu arahkan ke direktori kita menyimpan file download anaconda

3. Ketiga kita ketikkan sebagai berikut **bash namafileanaconda.sh** lalu enter, contoh seberti gambar 2.13



Gambar 2.13 Gambar install anaconda

- Setelah itu, tekan **ENTER** saja seperti gambar 2.14



Gambar 2.14 Gambar eksekusi anaconda

- Lalu akan muncul sebuah tulisan **End User License Agreement** seperti gambar 2.15, tekan **ENTER** dan tahan hingga seperti gambar



Gambar 2.15 Gambar anaconda license agreement



Gambar 2.16 Gambar perintah yes or no

6. Lalu setelah muncul perintah '**yes**' or '**no**' ketik **yes** lalu enter
 7. Setelah itu muncul path direktori instalasi anaconda kita seperti gambar 2.17 lalu tekan enter

```
Q                                     burger-man@burgerman-K46CB: ~/Downloads

Please answer 'yes' or 'no':'
>>>
Please answer 'yes' or 'no':'
>>> yes

Anaconda3 will now be installed into this location:
/home/burger-man/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

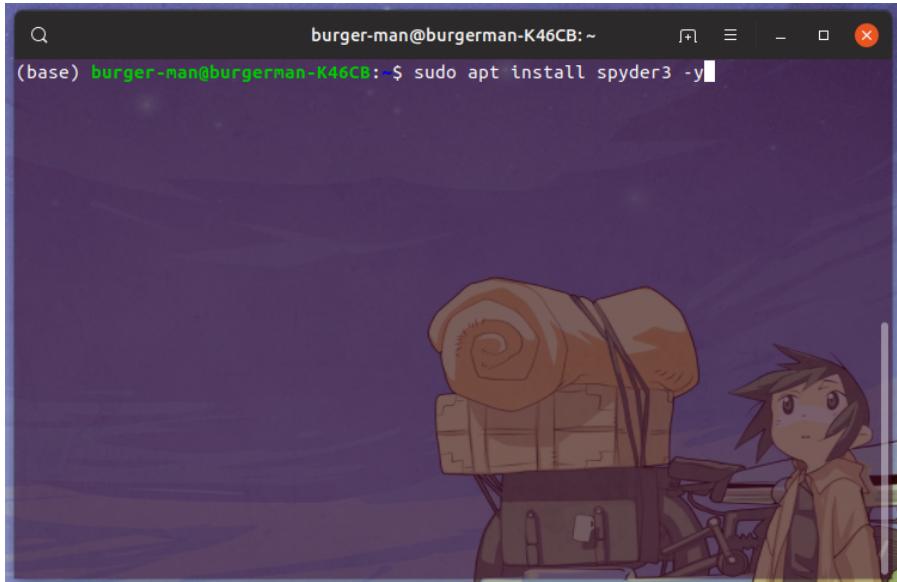
[~/home/burger-man/anaconda3] >>> █
```



Gambar 2.17 Gambar path anaconda

Setelah kita selesai instalasi anaconda jangan lupa juga untuk menginstal spyder ide, caranya seperti berikut:

- (a) ketikkan perintah `sudo apt install spyder3 -y` seperti gambar 2.18



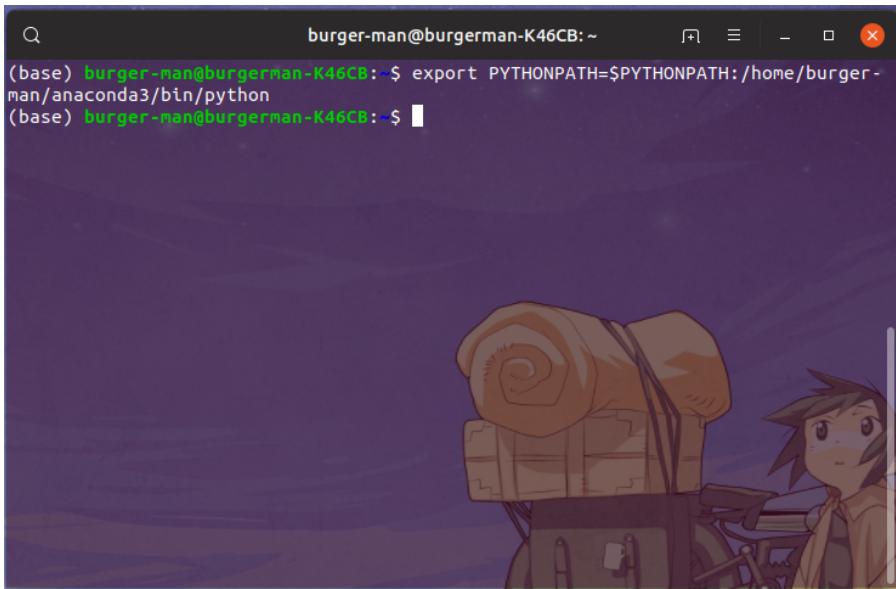
Gambar 2.18 Gambar perintah install spyder3

(b) lalu jalankan dengan perintah **spyder** atau **spyder3**

2.1.4 Konfigurasi *Python*

Setelah kita selesai instal Anaconda dan Spyder, selanjutnya kita akan mempelajari bagaimana cara setting environments python kita? caranya sebagai berikut

1. pertama kita buka terminal kita lalu ketikkan perintah **export PYTHONPATH=\$PYTHONPATH:\$PWD** contoh seperti gambar 2.19, lalu enter



Gambar 2.19 Gambar setpath

2.1.5 Quiz 2

1. Sistem operasi apa yang digunakan untuk instalasi pada buku ini?
 - a) Ubuntu dan Windows
 - b) Usus buntu dan Jendela
 - c) Penguin dan Kaca
 - d) Hewan dan Benda
2. Apa yang harus diperhatikan ketika ingin download Anaconda?
 - a) Kamu
 - b) Dia
 - c) Aku
 - d) Versi Sistem Operasi
3. Ubuntu versi berapa pada instalasi buku ini??
 - a) 19.04
 - b) 19.10
 - c) 19.99
 - d) 19.90
4. Apakah berbeda instalasi Anaconda pada windows dan linux? a) Ya
b) Tidak

5. Apa yang kita memerlukan internet untuk instal Anaconda?
 - a) Ya
 - b) Tidak

BAB 3

INSTALASI PIP DAN PERINTAH CLI

3.0.1 Instalasi Pip

Pip merupakan modul atau paket yang harus kita miliki apabila kita menggunakan bahasa pemrograman python. Pip digunakan untuk menginstal package-package python yang akan kita gunakan dalam pembuatan kode program. Cara melakukan instalasi pip pada anaconda CLI:

1. buka anaconda prompt (Start -> Anaconda Prompt)

2. ketikkan conda install -c anaconda pip seperti pada gambar 3.1.

```
(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
ca-certificates-2019.5.15 |             0    166 KB  anaconda
certifi-2019.6.16        | py37_0      155 KB  anaconda
```

Gambar 3.1 *Install pip*

3. ketik y, lalu enter. Tunggu hingga proses instalasi selesai seperti pada gambar 3.2.

```
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

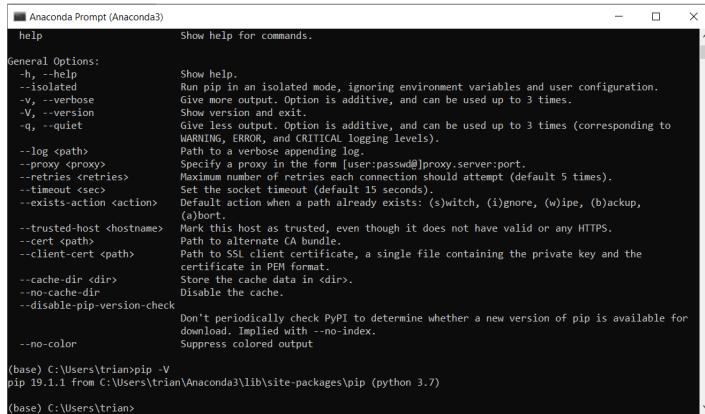
Proceed ([y]/n)? y

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>
```

Gambar 3.2 *Install pip Selesai*

4. jika telah selesai, lakukan pengecekan versi pip dengan mengetikkan pip -V seperti pada gambar 3.3.



```

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   Shows version and exits.
-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 appendable log.
--proxy <proxy> Specify a proxy in the form [user:password@]proxy.server:port.
--retries <retries> Maximum number of retries each connection should attempt (default 5 times).
--timeout <sec> Set the socket timeout (default 5 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 alternative 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>

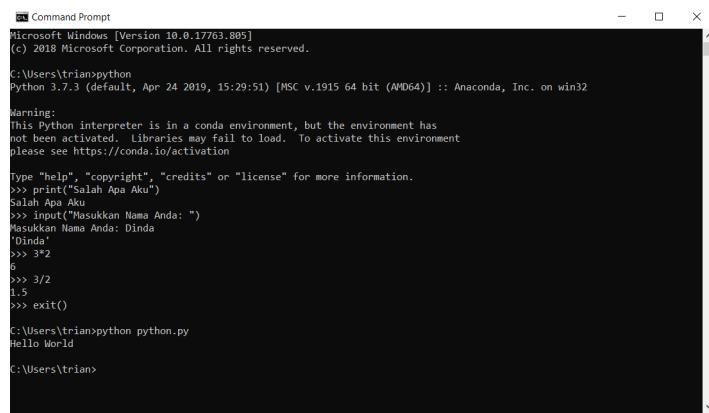
```

Gambar 3.3 Melihat Versi pip

3.0.2 Command Line Interface/Interpreter

Command line interpreter atau yang biasa teman-teman ketahui sebagai command prompt memungkinkan kita untuk menuliskan baris perintah yang akan dijalankan oleh komputer. Command line interpreter hanya berupa script atau tulisan kode program, berbeda dengan graphic user interface atau GUI yang memungkinkan kita memerintahkan sesuatu kepada komputer dengan hanya menggunakan tombol-tombol dan tampilan yang mudah dipahami. Cara menjalankan program python pada CLI:

1. Buka command prompt lalu ketikkan python seperti pada gambar 3.4.
2. Buatlah perintah print, input, perkalian, dan pembagian seperti pada gambar 3.4.
3. Bisa juga menjalankan file .py yang telah dibuat di IDE dengan cara python namafile.py, lalu klik enter seperti pada gambar 3.4.



The screenshot shows a Microsoft Windows Command Prompt window titled "Command Prompt". The window displays the following text:

```
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
9
>>> 3/2
1.5
>>> exit()

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

C:\Users\trian>
```

Gambar 3.4 *CLI in Command Prompt*

BAB 4

FUNGSI, METHOD, DAN KELAS

4.1 Fungsi

Fungsi adalah sebuah blok kode yang memiliki nama fungsi dan kode program di dalamnya. Fungsi dapat dipanggil berkali-kali sesuai dengan nama fungsi yang telah didefinisikan. Fungsi memiliki nilai kembalian (return).

contoh fungsi:

```
def my_biodata(nama, umur) :  
    bio = "nama saya " + nama + " umur saya " + umur  
    return bio
```

Inputan pada fungsi berada di dalam (). Contoh (str), ini merupakan inputan yang terdapat pada fungsi. Return merupakan kembalian dari fungsi. Misalnya return nama, maka program akan mengembalikan string yang terdapat dalam variabel nama yaitu "Nama Saya Dinda Majesty".

4.2 Package

Package merupakan sekumpulan modul yang dikemas oleh programmer dengan tujuan agar mempermudah dalam pembuatan kode program. Kita dapat membuat sebuah kode program atau fungsi didalamnya dan dapat secara mudah menggunakan kode program itu dengan cara memanggilnya pada kode program lainnya atau import package.

Contoh package:

```
def my_biodata(nama, umur):
    bio = "nama saya " + nama + " umur saya " + umur
    return bio
def my_study(kampus, prodi):
    study = "saya berkuliahan di " + kampus + " program studi " + prodi
    return study
```

Kode diatas merupakan isi dari file fungsi.py, sedangkan saya ingin menjalankan program fungsi.py pada main.py sehingga kode program pada file main.py akan dituliskan seperti berikut:

```
import fungsi

nama = "Dinda Majesty"
umur = "19 Tahun"
biodata = my_biodata(nama, umur)
print(biodata)

kampus = "Politeknik Pos Indonesia"
prodi = "D4-Teknik Informatika"
kuliah = my_study(kampus, prodi)
print(kuliah)
```

Kode program pada file main.py akan mengimport kode program yang ada pada file fungsi.py, sehingga dengan adanya fungsi dan package kita dapat dengan mudah melakukan pemanggilan fungsi yang telah kita deskripsikan sebelumnya, walaupun berada pada file python yang berbeda.

4.3 Kelas, Objek, Atribut, dan Method

Kelas merupakan blueprint dari sebuah objek atau kode program yang berisi fungsi dan dibuat untuk mendefenisikan objek dengan atribut yang sesuai dengan kelas yang telah dibuat.

Objek merupakan wujud dari kelas. Sebuah kelas harus memiliki objek yang nantinya akan di kodekan sesuai dengan fungsi yang telah dibuat pada kelas, tanpa adanya objek sebuah kelas tidak akan bisa menjalankan fungsi-fungsi didalamnya.

Atribut berisi variabel yang memiliki tipe data dan dapat kita berikan pada objek.

Method merupakan kode program yang berisi tindakan atau perintah untuk menjalankan objek.

4.3.1 Pemanggilan Library Kelas

Pemanggilan library kelas dapat dilakukan dengan cara import dan membuat objek dari kelas tersebut.

Contohnya, kita memiliki file python yang diberi nama ngitung dan didalamnya terdapat class Ngitung yang memiliki banyak fungsi didalamnya. Untuk melakukan pemanggilan class maka kita bisa mengetikkan kode seperti berikut.

```
import ngitung  
  
hitung = ngitung.Ngitung
```

4.3.2 Pemakaian Package Fungsi Apabila File Didalam Folder

Pemakaian Package fungsi apabila file terdapat didalam sebuah folder maka kita bisa menggunakan from folder import file dan from file import fungsi. Contohnya, kita memiliki folder src yang didalamnya terdapat file fungsi.py dan didalam fungsi.py terdapat fungsi Berhitung, untuk mengimportkan fungsi maka kita dapat mengetikkan kode seperti berikut.

```
from src import fungsi  
from fungsi import Berhitung
```

4.3.3 Pemakaian Package Kelas Apabila File didalam Folder

Pemakaian package kelas apabila file terdapat didalam sebuah folder maka kita bisa menggunakan from folder import file dan from file import kelas. Contohnya, kita memiliki folder src yang didalamnya terdapat file fungsi.py dan didalam fungsi.py terdapat kelas Ngitung, maka untuk melakukan import kelas kita dapat mengetikkan kode sebagai berikut.

```
from src import fungsi  
Kelas = fungsi.Ngitung(a,b)
```

4.4 Pretek Membuat Method dan Pemanggilan Method

Cara membuat method hanya dengan menambahkan def Nama Method, self atau variabel inputan, dan indentation pada source code yang telah dipraktekkan sebelumnya. Pembuatan method merupakan hal yang wajib saat kita membuat sebuah program yang berorientasi objek (OOP).

4.4.1 Modulus

```

1 def Modulus( self ):
2     #Modulus
3     print("Soal no 1")
4
5     npm = int(self.NPM) % 3
6     print(npm)
7
8     print("###    ###    #####    ###    ###    #####    ###")
9     print("###    ###    #####    ###    ###    #####    ###")
10    print("###    ###    ###    ###    ###    ###    ###    ###")
11    print("###    ###    ###    ###    ###    ###    ###    ###")
12    print("###    ###    #####    #####    ###    ###    ###")
13    print("###    ###    ###    ###    ###    ###    ###    ###")
14    print("###    ###    ###    ###    ###    ###    ###    ###")
15    print("###    ###    #####    #####    ###    #####    ###")
16    print("###    ###    #####    #####    ###    #####    ###")

```

Listing 4.1 Modulus

4.4.2 Hello NPM

```

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

```

Listing 4.2 Hello NPM

4.4.3 Hello NPM (3 Digit Belakang)

```

1 def NPM_DigitBelakang( self ):
2     #3 digit belakang NPM
3     print("Soal No 3")
4
5     Loop = self.NPM[4:7]
6
7     total = int(self.NPM[5]) + int(self.NPM[6])
8
9     for x in range(total):

```

```
10     print("Hallo " + Loop + " Apa Kabar?")
```

Listing 4.3 3 Digit Belakang

4.4.4 Hello NPM (Digit ke-3)

```
1 def NPM_DigitKetiga(self):
2     #digit ke 3
3     print("Soal No 4")
4
5     Loop = self.NPM[4]
6     print("Hello " + Loop + " Apa Kabar?")
```

Listing 4.4 Digit ke-3

4.4.5 Variabel Alfabet

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

Listing 4.5 Variabel Alfabet

4.4.6 Penjumlahan NPM

```
1 def Penjumlahan_NPM(self):
2     #penjumlahan NPM
3     print("Soal no 6")
4
5     index = 0
6     angka = 0
7
8     for i in self.NPM:
9         jumlah = int(self.NPM[index]) + int(angka)
10        angka = jumlah
11        index += 1
12
13     print(jumlah)
```

Listing 4.6 Penjumlahan NPM

4.4.7 Perkalian NPM

```
1 def Perkalian_NPM(self):
2     #perkalian NPM
3     print("Soal no 7")
4
5     index = 0
```

```

6     angka = 0
7
8     for i in self.NPM:
9         jumlah = int(self.NPM[index]) * int(angka)
10        angka = jumlah
11        index += 1
12
13     print(jumlah)

```

Listing 4.7 Perkalian NPM

4.4.8 Print Vertical

```

1 def Print_Vertical(self):
2     #print vertical
3     print("Soal no 8")
4
5     for i in self.NPM:
6         print(i)

```

Listing 4.8 Print Vertical

4.4.9 Digit Genap NPM

```

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

```

Listing 4.9 Digit Genap NPM

4.4.10 Digit Ganjil NPM

```

1 def DigitGanjil(self):
2     #digit ganjil NPM
3     print("Soal no 10")
4
5     index = 0
6     for i in self.NPM:
7         if (int(self.NPM[index])%2 != 0) & (int(self.NPM[index])
!= 0):
8             print(self.NPM[index])
9             index += 1

```

Listing 4.10 Digit Ganjil NPM

4.4.11 Bilangan Prima NPM

```

1 def PrimaNPM(self):

```

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

Listing 4.11 Bilangan Prima NPM

4.4.12 Pemanggilan Fungsi pada Main.py

```
1 import fungsi
2
3 print("Masukkan NPM anda: ")
4 NPM = input()
5 crot = fungsi.Kelas(NPM)
6
7 soal1 = crot.Modulus()
8 soal2 = crot.Hello_NPM()
9 soal3 = crot.NPM_DigitBelakang()
10 soal4 = crot.NPM_DigitKetiga()
11 soal5 = crot.Variabel_Alfabet()
12 soal6 = crot.Penjumlahan_NPM()
13 soal7 = crot.Perkalian_NPM()
14 soal8 = crot.Print_Vertical()
15 soal9 = crot.DigitGenap()
16 soal10 = crot.DigitGanjil()
17 soal11 = crot.PrimaNPM()
```

Listing 4.12 Bilangan Prima NPM

BAB 5

PENGELOLAAN FILE CSV

5.1 Pengelolaan File CSV, Sejarah, dan Contoh

File csv (comma separated value) sering digunakan dalam dunia pemrograman untuk menampilkan data. file csv memiliki format yang sangat sederhana, setiap baris dipisahkan oleh enter dan setiap kolom dipisahkan oleh tanda koma.

Kegunaan file csv dibandingkan file dengan format lainnya adalah dari segi kompatibilitas karena file csv dapat diolah, dimodifikasi, digunakan, import/export menggunakan berbagai software dan bahasa pemrograman, salah satunya python.

Contoh file csv:

```
1 nomor , nama klub , jumlah main , poin , tanggal  
2 1 , Manchester City ,8 ,19 ,1/2/1999  
3 2 , Arsenal ,8 ,18 ,1/3/1999  
4 3 , Tottenham Hotspurs ,8 ,18 ,1/4/1999  
5 4 , Liverpool ,8 ,17 ,1/5/1999  
6 5 , Chelsea ,8 ,16 ,1/6/1999  
7 6 , Everton ,8 ,15 ,1/7/1999  
8 7 , Manchester United ,8 ,14 ,1/8/1999  
9 8 , Southampton ,8 ,12 ,1/9/1999  
10 9 , AFC Bournemouth ,8 ,12 ,1/10/1999
```

11 10, Crystal Palace ,8,11,1/11/1999

Listing 5.1 Contoh CSV

Contoh kode program python untuk membaca file csv.

```
1 #CSV Reader
2 import csv
3
4 with open('FCSV.csv') as csvfile:
5     readCSV = csv.reader(csvfile, delimiter=',')
6     for row in readCSV:
```

Listing 5.2 Contoh Kode Python to Read CSV

Contoh kode program python untuk membaca file csv menggunakan library pandas.

```
1 import pandas as pd
2 df1=pd.read_csv("FCSV.csv")
3 print(df1)
```

Listing 5.3 Contoh Kode Python to Read CSV

Sejarah CSV.

CSV sudah digunakan sejak tahun 1972, CSV dapat dikompilasi pada bahasa pemrograman IBM Fortran. Data yang dipisahkan oleh koma apabila terdapat spasi di dalamnya maka harus diberi tanda petik di awal dan akhir isi dari data tersebut. Nama CSV digunakan pada tahun 1983. Pada panduan dari Osborne Executive Computer terdapat kutipan yang membolehkan isi karakter memiliki koma. Pada tahun 2005 dengan RFC4180, CSV didefinisikan sebagai MIME Content Type. Lalu pada tahun 2013, defisiensi dari RFC4180 dipecahkan oleh rekomendasi dari W3C. Pada tahun 2014, IETF mempublikasi RFC7111 yang mendeskripsikan pecahan Uniform Resource Identifier(URI) ke dokumen CSV. RFC7111 menjelaskan tentang bagaimana baris, kolom dapat digunakan dalam dokumen CSV menggunakan indeks posisi. Pada Tahun 2015, W3C mempublikasikan draft rekomendasi untuk CSV-metadata standard yang dimulai pada bulan Desember 2015.

5.2 Aplikasi yang bisa Menciptakan File CSV

Aplikasi yang dapat kita gunakan untuk membuat file csv ada banyak, diantaranya:

1. Spreadsheet

Spreadsheet merupakan aplikasi pembuatan file CSV dengan cara memasukan data sesuai baris dan kolom yang diinginkan. Contoh spreadsheet seperti Google Spreadsheet, Microsoft Excel, dan aplikasi lainnya.

2. Bahasa

Bahasa pemrograman merupakan media untuk membuat aplikasi yang dapat

Pemrograman

digunakan untuk membuat file CSV khusus dengan bahasa pemrograman tertentu yang support dengan pembuatan file CSV. Seperti Python, C Sharp, dan lain sebagainya.

3. Notepad atau Text Editor
 Text editor juga dapat membuat file CSV, cukup dengan membuat file sesuai format CSV dan save file tersebut dengan ekstensi .csv.

5.3 Menulis dan Membaca File CSV pada Ms.Excel

Membuat file csv melalui ms.excel sangatlah mudah, isikan nomor pada kolom, kemudian isikan variabel sebagai judul pada baris, lalu isikan data sesuai dengan variabel yang telah ditentukan, contoh:

	nomor	nama klub	jumlah main	poin
1	1	Manchester	8	19
2	2	Arsenal	8	18
3	3	Tottenham	8	18
4	4	Liverpool	8	17
5	5	Chelsea	8	16
6	6	Everton	8	15
7	7	Manchest	8	14
8	8	Southham	8	13
9	9	AFC Bourn	8	12
10	10	Crystal Pa	8	11

Gambar 5.1 Contoh Penulisan CSV pada Excel

Setelah mengetikkan data pada excel, lalu pilih menu file, pilih menu export, pilih menu change file type, pilih csv, lalu klik tombol save as. Maka file excel akan menjadi file csv.

```

1 nomor , nama klub , jumlah main , poin , tanggal
2 1 , Manchester City ,8 ,19 ,1/2/1999
3 2 , Arsenal ,8 ,18 ,1/3/1999
4 3 , Tottenham Hotspurs ,8 ,18 ,1/4/1999
5 4 , Liverpool ,8 ,17 ,1/5/1999
6 5 , Chelsea ,8 ,16 ,1/6/1999
7 6 , Everton ,8 ,15 ,1/7/1999
8 7 , Manchester United ,8 ,14 ,1/8/1999
9 8 , Southampton ,8 ,12 ,1/9/1999

```

```
10 9 , AFC Bournemouth ,8 ,12 ,1/10/1999
11 10, Crystal Palace ,8 ,11 ,1/11/1999
```

Listing 5.4 Contoh CSV

5.3.1 Sejarah Library CSV

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

5.3.2 Sejarah Library Pandas

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

5.3.3 Fungsi - fungsi yang terdapat di library CSV

Berikut fungsi-fungsi yang terdapat pada library csv.

1. `csv.reader(csvfile, dialect='excel', **fmparms)`

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

```
1 #CSV Reader
2 import csv
3
4 with open('FCSV.csv') as csvfile:
5     readCSV = csv.reader(csvfile, delimiter=',')
6     for row in readCSV:
7         print(row)
```

2. `csv.writer(csvfile, dialect='excel', **fmparms)`

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

```
1 #CSV Writer
2 import csv
3
4 with open('asal.csv', mode='w') as csvfile:
5     fieldnames = ['nomor', 'nama klub', 'jumlah main', 'poin']
6     csv_writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
7
8     csv_writer.writeheader()
```

```
9     csv_writer.writerow({ 'nomor': '1', 'nama klub': 'Asal-Asalan' }  
10    , 'jumlah main': '8', 'poin': '10'})  
11    csv_writer.writerow({ 'nomor': '2', 'nama klub': 'Asal-Asalan2' }  
12    , 'jumlah main': '8', 'poin': '12'})
```

3. `csv.register_dialect(name[, dialect[, **fmtparams]])`
Mengasosiasikan dialek dengan nama, nama yang dimasukkan harus berupa karakter.
4. `csv.unregister_dialect(name)`
Menghapus asosiasi dialek dengan nama pada registry dialek.
5. `csv.get_dialect(name)`
Mengambil dialek yang telah diasosiasikan dengan nama.
6. `csv.list_dialects()`
Mengembalikan dialek yang telah diregistrasi.
7. `csv.field_size_limit([new_limit])`
Mengembalikan maksimal kolom data yang diperbolehkan oleh pembaca.

5.3.3.1 Fungsi - fungsi yang terdapat di library Pandas

1. `pandas.read_csv(filepath_or_buffer[, sep, ...])`
Untuk membaca file CSV dan menyimpannya ke DataFrame. Contohnya:

```
1 import pandas as pd  
2 df1=pd.read_csv("FCSV.csv")  
3 print(df1)
```
2. `pandas.read_excel(io[, sheet_name, header, names, ...])`
Membaca file excel dan menyimpannya ke DataFrame. Contohnya:

```
1 import pandas as pds  
2 df2=pds.read_excel("FCSV.xlsx", index_col=None, header=None)  
3 print(df2)
```
3. `to_csv([path, index, sep, na_rep, ...])`
Untuk membuat file CSV dari data yang ada

5.4 Praktek CSV

5.4.1 CSV List

```

1 import csv
2
3 def CsvList():
4     with open('FCSV.csv') as csv_file:
5         csv_reader = csv.reader(csv_file, delimiter=',')
6         line_count = 0
7         for row in csv_reader:
8             if line_count == 0:
9                 print(f'Nama Kolom adalah {" ".join(row)}')
10                line_count += 1
11            else:
12                print(f'\t Nomor : {row[0]}, Nama Klub : {row[1]},'
13 Jumlah Main : {row[2]}, Poin : {row[3]}')
14                line_count += 1
15                print(f'Jumlah Total {line_count} lines.')

```

Listing 5.5 CSV List

5.4.2 CSV Dictionary

```

1 def CsvDict():
2     with open('dictionary.csv', mode='r') as csv_file:
3         csv_reader = csv.DictReader(csv_file)
4         line_count = 0
5         for row in csv_reader:
6             if line_count == 0:
7                 print(f'Nama Kolom adalah {" ".join(row)}')
8                 line_count += 1
9             else:
10                print(f'\t First Name : {row["first_name"]} Last Name'
11 : {row["last_name"]}')
12                line_count += 1
13                print(f'Jumlah Total {line_count} lines.')

```

Listing 5.6 CSV Dictionary

5.4.3 Hello NPM (Pandas List)

```

1 import pandas
2
3 def pandasList():
4     pd = pandas.read_csv("FCSV.csv")
5     print(pd)

```

Listing 5.7 Pandas List

5.4.4 Hello NPM (Pandas Dictionary)

```

1 def pandasDict():
2     pds = pandas.read_csv("dictionary.csv")
3     crot = pandas.DataFrame.from_dict(pds)
4     print(crot)

```

Listing 5.8 Pandas Dictionary

5.4.5 Pandas Date

```

1 def Tanggal():
2     tanggal = pandas.read_csv("FCSV.csv")
3     tanggal['tanggal']=pandas.to_datetime(tanggal['tanggal'])
4     print(tanggal)

```

Listing 5.9 Pandas Date

5.4.6 Pandas Ubah Index

```

1 def ubahIndex():
2     baris = pandas.read_csv("FCSV.csv")
3     barisindex = [ '1', '2', '0', '3', '5', '4', '6', '7', '9', '8' ]
4     oke = baris.reindex(barisindex)
5     print(oke)

```

Listing 5.10 Pandas Ubah Index

5.4.7 Pandas Ubah Column

```

1 def ubahColumn():
2     ubah = pandas.read_csv("FCSV.csv")
3     okey = ubah.rename(columns={"nomor" : "number"})
4     print(okey)

```

Listing 5.11 Pandas Ubah Column

5.4.8 Main CSV

```

1 import D1184011_csv
2
3 csvList = D1184011_csv.CsvList()
4 csvDict = D1184011_csv.CsvDict()

```

Listing 5.12 Main CSV

5.4.9 Main Pandas

```

1 import D1184011_pandas
2
3 pandasList = D1184011_pandas.pandasList()
4 pandasDict = D1184011_pandas.pandasDict()
5 pandasDate = D1184011_pandas.Tanggal()
6 pandasUbah = D1184011_pandas.ubahIndex()
7 pandasUbahCol = D1184011_pandas.ubahColumn()

```

Listing 5.13 Digit Genap NPM

