

PYTHON UNTUK BIG DATA

PYTHON UNTUK BIG DATA

Syafrial Fachri Pane
Yogi Aditya Saputra
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 Tunnisa

Diana Asri Wijayanti

Desain sampul dan Tata letak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

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

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

CONTRIBUTORS

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

YOGI ADITYA SAPUTRA, Informatics Research Center., Politeknik Pos Indonesia,
Bandung, Indonesia

CONTENTS IN BRIEF

1	Berkenalan dengan Python	1
2	Data Science	23
3	Jupyter Notebook	31
4	Numpy	35
5	Pandas	37
6	Scikit-Learn	47
7	Matplotlib	49

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 Berkenalan dengan Python	1
1.1 Instalasi Python	1
1.1.1 Windows	1
1.1.2 Linux	6
1.2 Sejarah Python	7
	ix

1.3	Pengenalan Python	8
1.4	Mengapa harus Python	9
1.5	Cara Penggunaan	9
1.5.1	Cara Menjalankan	9
1.5.2	Komentar	11
1.5.3	Tipe Data	12
1.5.4	Variabel	14
1.5.5	Looping	15
1.5.6	Fungsi	16
1.5.7	Modul	16
1.6	Instalasi Pip	19
1.6.1	Windows	19
1.6.2	Linux	19
1.7	Pip	20
1.8	Cara Penggunaan Pip	20
1.9	Ekstensi File	21
1.9.1	.py	21
1.9.2	.ipynb	21
1.9.3	Konversi File	22
1.10	Mengapa harus ipynb	22
2	Data Science	23
2.1	Apa itu Big Data	23
2.1.1	Pengertian	23
2.1.2	Contoh <i>Big Data</i>	24
2.1.3	Cara Kerja <i>Big Data</i>	25
2.1.4	Penggunaan <i>Big Data</i>	25
2.1.5	Jenis <i>Big Data</i>	25
2.1.6	Tantangan <i>Big Data</i>	27
2.1.7	Karakteristik <i>Big Data</i>	27
2.1.8	Keuntungan dan Kerugian <i>Big Data</i>	28
2.2	Mengapa menggunakan python	28
2.3	Tools	30
2.4	Library	30
3	Jupyter Notebook	31
3.1	Apakah itu Jupyter Notebook ?	31
3.1.1	Notebook documents	31

3.1.2	Jupyter Notebook	31
3.1.3	Kernel	32
3.1.4	Notebook Dashboad	32
3.2	Perbedaan Jupyter Notebook dan Google Collab	32
3.3	Cara Instalasi	33
3.3.1	Windows	33
3.3.2	Ubuntu	33
3.3.3	Editor	33
3.4	Cara Penggunaan	33
3.4.1	On Browser	33
3.4.2	Editor	33
3.4.3	Toolbar	33
4	Numpy	35
4.1	Numpy	35
5	Pandas	37
5.1	Pandas	37
5.2	Arsitektur DataFrame	38
5.3	DataFrame	38
6	Scikit-Learn	47
6.1	Scikit-Learn	47
7	Matplotlib	49
7.1	Matplotlib	49
	Daftar Pustaka	51

DAFTAR GAMBAR

1.1	Pilih Python	2
1.2	Download Python	2
1.3	Run Installer	3
1.4	Pilih User	3
1.5	Pilih Location	4
1.6	Add Environment	4
1.7	Proses Instalasi	5
1.8	Instalasi Selesai	5
1.9	Cek Instalasi	6
1.10	Logo Python	8
1.11	Terminal	9
1.12	Penggunaan Perintah Python	9
1.13	Penggunaan Perintah print	10

1.14	Penggunaan Perintah exit	10
1.15	Editor Nano	10
1.16	Save File	11
1.17	Exit Editor	11
1.18	Running File	11
1.19	String Data	12
1.20	Integer Data	12
1.21	Float Data	13
1.22	List Data	13
1.23	Tuple Data	13
1.24	Dictionary Data	14
1.25	Variabel	15
1.26	Import Module	18
1.27	Import Module Alias	18
1.28	Import Module Sebagian	18
1.29	Terminal	19
1.30	Pip	19
1.31	Pip 3	20
5.1	Arsitektur DataFrame	38
5.2	Create DataFrame	39
5.3	Selection Column	40
5.4	Selection Row loc	41
5.5	Selection Row iloc	41
5.6	Missing Value	42
5.7	Fix Missing Value	42
5.8	Drop Missing Value	42
5.9	Iterrow Data	43
5.10	Lowercase Data	44
5.11	Uppercase Data	44
5.12	Replacement Data	45

DAFTAR TABEL

Listings

1.1	Root terminal	6
1.2	Tambah Source	6
1.3	root	6
1.4	Penggunaan Komentar	11
1.5	Tipe Data String	12
1.6	Tipe Data Integer	12
1.7	Tipe Data Float	13
1.8	Tipe Data List	13
1.9	Tipe Data tuple	13
1.10	Tipe Data Dictionary	14
1.11	Penggunaan Variabel	14
1.12	Penggunaan While Loop	15
1.13	Penggunaan For Loop	15
1.14	Fungsi Python	16
1.15	Modul	17
1.16	Import modul	17
1.17	Import modul	18
1.18	Import modul	18

1.19	Import module Seaborn	18
1.20	install package	20
1.21	List package	20
1.22	Show package	21
1.23	uninstall package	21
5.1	Import-Module	38
5.2	Create DataFrame	39
5.3	Selection Column	40
5.4	Selection Row loc	40
5.5	Selection Row iloc	41
5.6	Missing Value	41
5.7	Drop Missing Value	42
5.8	Looping Iterrow	43
5.9	Convert Lowercase	43
5.10	Convert Uppercase	44
5.11	Replacement Data	45
5.12	Replacement Data	45

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat
Februari, 2019*

ACKNOWLEDGMENTS

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

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

R. M. A.

ACRONYMS

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

GLOSSARY

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

SYMBOLS

- A Amplitude
- $\&$ Propositional logic symbol
- a Filter Coefficient

- \mathcal{B} Number of Beats

INTRODUCTION

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

Informatics Research Center
Bandung, Jawa Barat, Indonesia

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

Rumusnya cuman sebagai contoh aja biar keren[?].

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

BAB 1

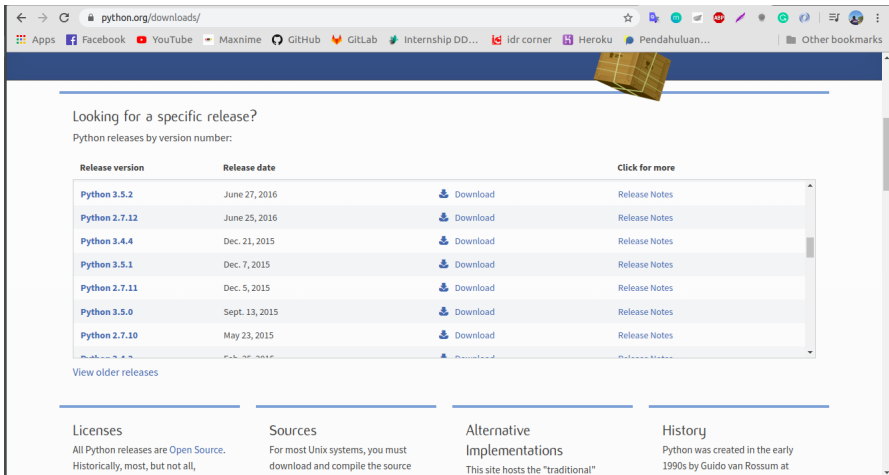
BERKENALAN DENGAN PYTHON

1.1 Instalasi Python

Tentunya jika ingin bisa menggunakan python, kita perlu memasang terlebih dahulu di komputer PC kita. Python sendiri juga bisa dipasang di berbagai macam sistem operasi seperti Linux, Windows dan Mac OS.

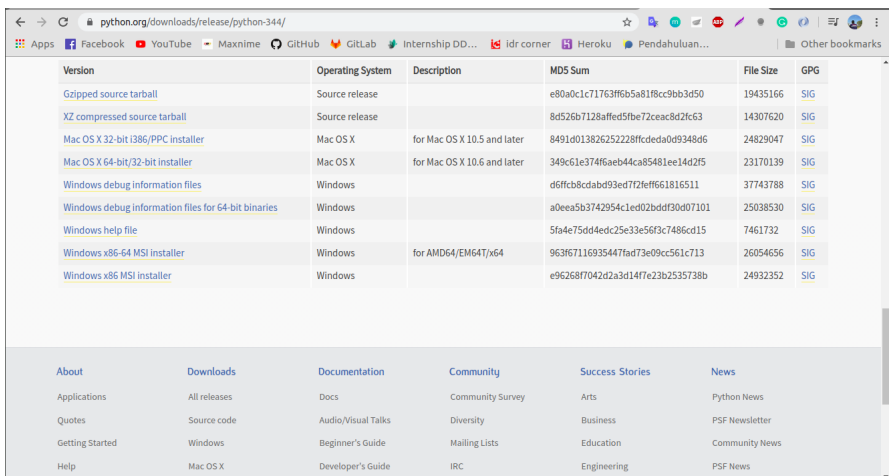
1.1.1 Windows

1. Download terlebih dahulu installer pythonnya di <https://www.python.org/downloads/>. Disini saya memilih python versi 3.4.



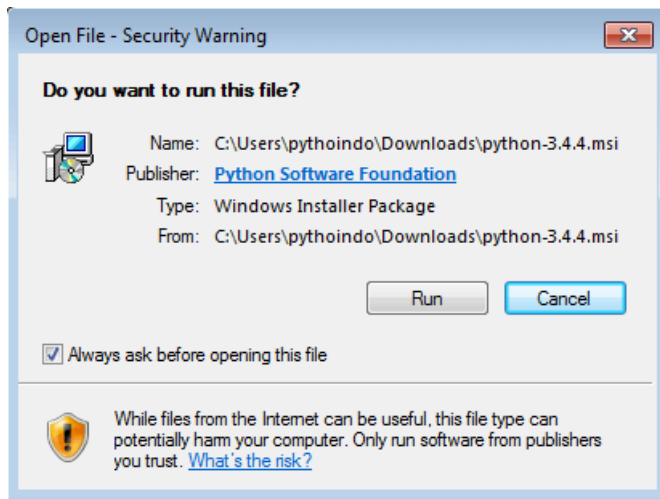
Gambar 1.1 Pilih Python

Setelah itu saya memilih versi x86 atau 32 bit. Tapi anda bebas mau memilih 32 bit atau 64 bit.



Gambar 1.2 Download Python

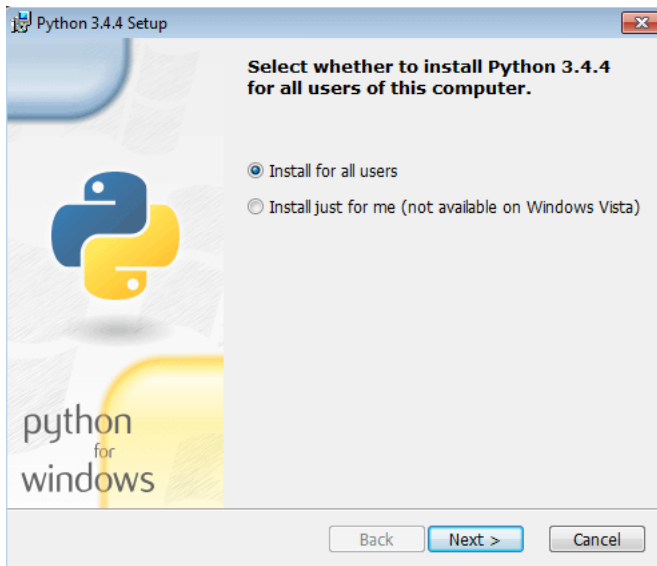
- Setelah selesai mendownload file installer tersebut, buka lokasi file installer tersebut dan tekan 2 kali untuk membukanya.



Gambar 1.3 Run Installer

Setelah muncul kotak dialog, tekan tombol run untuk menjalankan file tersebut.

3. Lalu tunggu beberapa saat, akan muncul kotak dialog untuk memilih user siapa saja yang bisa akses instalasi tersebut. Disini saya memilih install for all user.



Gambar 1.4 Pilih User

- Setelah itu, akan muncul kotak dialog tempat instalasi yang akan kita tem-
patkan.



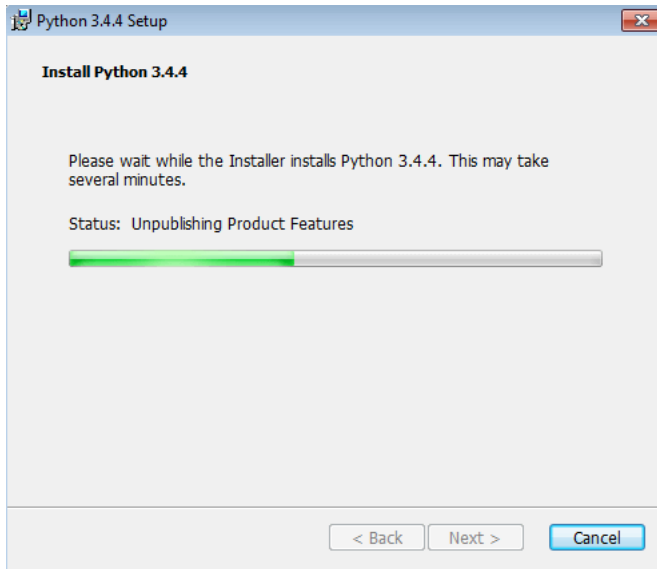
Gambar 1.5 Pilih Location

- Setelah itu, tambahkan environment python dengan menekan pilihan **Add python.exe to Path**.



Gambar 1.6 Add Environment

- Setelah itu, tunggu proses instalasi hingga selesai.



Gambar 1.7 Proses Instalasi

7. Proses instalasi selesai.



Gambar 1.8 Instalasi Selesai

1.1.2 Linux

1.1.2.1 via apt

1. Buka terminal di linux.
2. Masuk sebagai root di terminal.

```
1 sudo apt update
2 sudo apt install software-properties-common
3
```

Listing 1.1 Root terminal

Penjelasan pada 1.1, pada baris 1 menjelaskan untuk mengupdate package yang ada serta baris 2 untuk melakukan instalasi prerequisites.

3. Tambahkan deadsnakes PPA ke dalam source list sistem.

```
1 sudo add-apt-repository ppa:deadsnakes/ppa
2
```

Listing 1.2 Tambah Source

Ketika menjalankan perintah 1.2, pasti nanti diminta untuk tekan **enter** untuk melanjutkan instalasi. Maka tekan saja **enter**.

4. Setelah itu, ketik **sudo apt install python3.7** untuk melakukan instalasi python.
5. Setelah selesai melakukan instalasi, perlu dilakukan pengecekan apakah instalasi tersebut sukses atau tidak. Untuk mengecek ketik **python --version**.

```
(base) newbie@newbie:~$ python --version
Python 3.7.4
(base) newbie@newbie:~$
```

Gambar 1.9 Cek Instalasi

1.1.2.2 from source

1. Buka terminal.
2. Masuk sebagai root di terminal

```
1 sudo apt update
2 sudo apt install build-essential zlib1g-dev libncurses5-
  dev libgdbm-dev libnss3-dev libssl-dev libreadline-dev libffi
  -dev wget
3
```

Listing 1.3 root

Pada syntax 1.3, pada baris 1 untuk mengupdate list package yang ada, Sedangkan pada baris 2 untuk menginstal package yang dibutuhkan.

3. Download python versi 3.7.4 menggunakan wget.

```
1 wget https://www.python.org/ftp/python/3.7.4/Python-3.7.4.tgz
2
```

4. Ekstrak hasil download.

```
1 tar -xf Python-3.7.4.tgz
2
```

5. Masuk ke direktori hasil ekstraksi dan melakukan konfigurasi.

```
1 ./configure --enable-optimizations
2
```

Konfigurasi disini dijalankan untuk mengecek apakah semua dependency yang dibutuhkan sudah terdapat dalam sistem anda saat ini.

6. Memproses pembangunan python.

```
1 make -j 8
2
```

Disini, merupakan tahap untuk membuild python tersebut. Untuk mempercepat build, maka perlu sebuah modifikasi dengan mengetik **-j** yang relevan dengan jumlah core dari processor anda. Untuk dapat melihat jumlah core dari processor anda, cukup ketika **nproc**.

7. install binary python

```
1 sudo make altinstall
2
```

Setelah build pythonnya selesai, perlu menginstall binary python.

8. Install python sukses, dan untuk mengecek dengan cara berikut.

```
1 python --version
2
```

1.2 Sejarah Python

Pada tahun 1990, Guido van Rossum mengembangkan python di CWI, Amsterdam. Bahasa ini merupakan versi lanjut dari bahasa pemrograman ABC. Versi terakhir python yang dikeluarkan oleh CWI ialah versi 1.2.

Lalu tahun 1995, Guido berpindah dari CWI ke CNRI sambil melanjutkan proses pengembangan python. Versi python terakhir dikeluarkan adalah versi 1.6. Tahun

2000, Guido van Rossum dan tim inti pengembangan python berpindah dari CNRI ke BeOpen.com yang merupakan perusahaan komersial dan telah membentuk BeOpen PythonLabs. Dan BeOpen pun mengeluarkan versi python yang baru yaitu versi 2.0. Setelah Guido dan tim di BeOpen mengeluarkan python versi 2.0, mereka berpindah kembali ke DigitalCreations.

Hingga saat ini, pengembangan python masih terus dilakukan oleh sekumpulan pemrogram yang di koordinir oleh Guido van Rossum dan Python Software Foundation. Python Software Foundation adalah sebuah organisasi non-profit yang dibentuk sebagai hak cipta intelektual atas python sejak python versi 2.1 dan dengan untuk mencegah python dimiliki oleh perusahaan komersial. Saat ini, proses distribusi python sudah mencapai versi 2.6.1 dan versi 3.0.

Guido memilih nama python karena kecintaan Guido van Rossum terhadap sebuah acara televisi bernama Monty Python's Flying Circus.

1.3 Pengenalan Python

Python merupakan bahasa pemrograman yang dapat melakukan eksekusi sejumlah instruksi multiguna secara langsung (interpretatif) dengan berorientasi pada objek serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan kode atau syntax. Sebagian besar mengartikan python sebagai bahasa dengan tingkat kemampuan tinggi, menggabungkan kapabilitas, dan sintaks kode yang sangat jelas dan dilengkapi oleh fungsionalitas dari pustaka dasar yang sangat besar dan komprehensif. Walaupun python ini, digolongkan sebagai bahasa pemrograman tingkat tinggi, namun tetap Python dirancang sedemikian rupa supaya mudah dipahami serta dipelajari.

Python juga dapat berjalan di banyak platform seperti Mac, Linux dan Windows dll. Python bersifat *open source*, sehingga masih banyak orang yang berkontribusi untuk mengembangkan dimana yang hak kekayaan intelektual dipegang oleh PSF. Bahasa Python didukung oleh *library library* yang didalamnya menyediakan fungsi analisis data dan fungsi *machine learning*, *data preprocessing tools*, serta visualisasi data. Hal ini membuat Python menjadi bahasa pemrograman yang populer pada bidang *data science* dan analisis.



Gambar 1.10 Logo Python

1.4 Mengapa harus Python

Alasan mengapa python adalah salah satu bahasa pemrograman yang harus dipelajari adalah sebagai berikut :

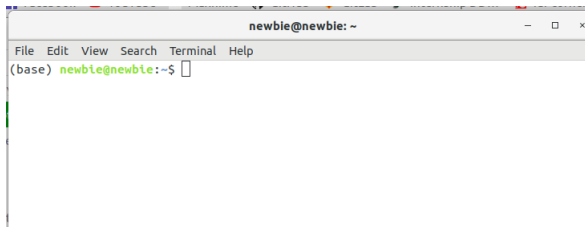
1. Python merupakan bahasa yang mudah dipelajari serta mudah digunakan.
2. Python juga merupakan bahasa yang lebih efisien dibandingkan dengan bahasa pemrograman lain. Contohnya jika dalam bahasa lain bisa sampai 5 baris, maka dengan python cukup 1 baris saja untuk menjalankan perintah tersebut.
3. Python merupakan bahasa multifungsi, dimana python bisa diterapkan dimana saja mulai dari pemrosesan data / teks, membuat website, membuat program jaringan, robotika, sampai dengan kecerdasan buatan.
4. Python juga memiliki dukungan pustaka yang cukup banyak.
5. Python juga bisa melakukan integrasi dengan bahasa pemrograman lainnya.

1.5 Cara Penggunaan

1.5.1 Cara Menjalankan

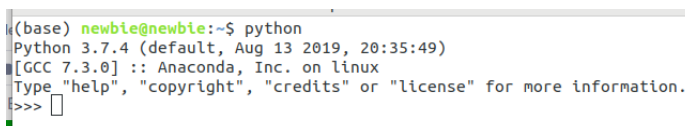
1.5.1.1 Linux

1. Buka terminal linux



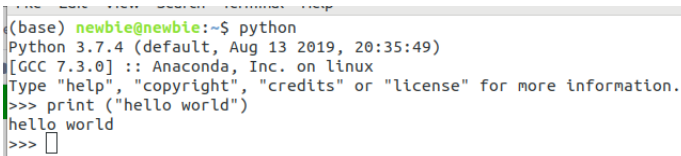
Gambar 1.11 Terminal

2. Ketik Python di terminal tersebut. Itu digunakan untuk masuk ke dalam sheel python.



Gambar 1.12 Penggunaan Perintah Python

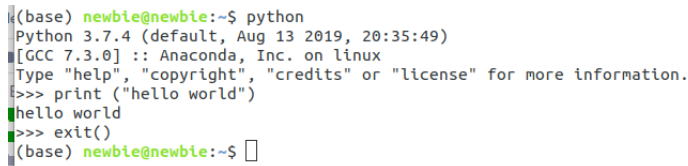
3. Lalu, tuliskan kode `print("hello world")`. Jika sudah tekan enter.



```
(base) newbie@newbie:~$ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("hello world")
hello world
>>> 
```

Gambar 1.13 Penggunaan Perintah `print`

4. Untuk keluar dari sheel python, ketik `exit()`.

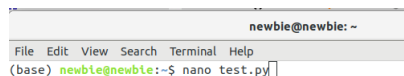


```
(base) newbie@newbie:~$ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("hello world")
hello world
>>> exit()
(base) newbie@newbie:~$ 
```

Gambar 1.14 Penggunaan Perintah `exit`

Atau bisa menggunakan cara ini, seperti berikut :

1. Menggunakan text editor, seperti nano. Untuk penjelasan tentang nano sendiri akan dijelaskan setelah point ini.



```
newbie@newbie: ~
File Edit View Search Terminal Help
(base) newbie@newbie:~$ nano test.py
```

Gambar 1.15 Editor Nano

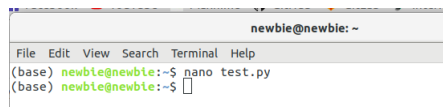
Pada gambar 1.15, digunakan untuk membuat file baru dengan nama `test.py`.

2. Lalu simpan file tersebut dengan menekan `Ctrl + O`, setelah itu akan muncul konfirmasi. Tekan enter saja.



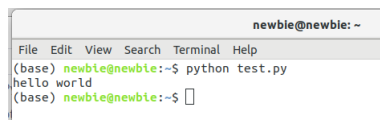
Gambar 1.16 Save File

3. Untuk keluar dari editor nano tersebut, tekan Ctrl + X.



Gambar 1.17 Exit Editor

4. Untuk menjalankan file py tersebut. ketik python namafile.py.



Gambar 1.18 Running File

1.5.2 Komentar

Komentar atau comment adalah kode yang berada dalam syntax python yang tidak dieksekusi atau tidak dieksekusi oleh mesin. Komentar biasanya digunakan untuk menandai atau memberikan suatu keterangan pada syntax python yang ada.

Komentar sering digunakan untuk memberikan penjelasan kepada orang lain terhadap syntax yang ada atau bisa digunakan untuk mengingatkan kepada seorang programmer jika ada yang ingin diubah dari syntax tersebut.

Untuk memberikan komentar, cukup dengan memberikan tanda (#) yang diikuti dengan isi komentar tersebut. Berikut Contoh penggunaan komentar di python.

```
1 #ini untuk menampilkan tulisan hello world
2 print("hello world")
```

Listing 1.4 Penggunaan Komentar

1.5.3 Tipe Data

Tipe data merupakan media atau memori pada komputer untuk menampung berbagai informasi sesuai jenisnya.

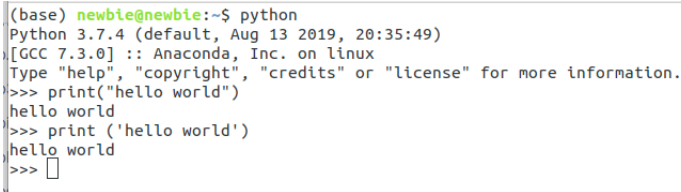
Python juga memiliki berbagai tipe data yang cukup unik jika dibandingkan dengan bahasa pemrograman lainnya.

Berikut beberapa tipe data dalam python.

1. Boolean, tipe data ini digunakan untuk menentukan dalam pengambilan keputusan. Jika benar atau True akan bernilai 1 dan jika salah atau False akan bernilai 0.
2. String, tipe data ini digunakan untuk menyatakan karakter / kalimat. Dan tipe data ini harus menggunakan tanda “atau ‘ untuk mengapit nilai String tersebut. Contoh implementasinya seperti berikut.

```
1 print("hello world")
2 print('hello world')
3
```

Listing 1.5 Tipe Data String



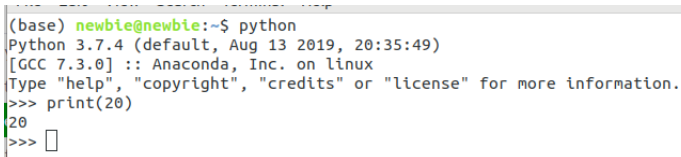
```
(base) newbie@newbie:~$ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello world")
hello world
>>> print('hello world')
hello world
>>> 
```

Gambar 1.19 String Data

3. Integer, tipe data ini untuk menyatakan bilangan bulat. Contoh implementasinya bisa dilihat seperti berikut.

```
1 print(20)
2
```

Listing 1.6 Tipe Data Integer



```
(base) newbie@newbie:~$ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print(20)
20
>>> 
```

Gambar 1.20 Integer Data

4. Float, tipe data ini untuk menyatakan bilangan yang memiliki koma. Contoh implementasinya bisa dilihat seperti berikut.

```

1 print(3.14)
2

```

Listing 1.7 Tipe Data Float

```

(base) newbie@newbie:~$ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print(3.14)
3.14
>>>

```

Gambar 1.21 Float Data

5. List, tipe data ini untuk menyimpan berbagai jenis tipe data dan isinya bisa diubah-ubah. Contoh implementasi bisa dilihat seperti berikut.

```

1 print([1,2,3,4,5])
2

```

Listing 1.8 Tipe Data List

```

(base) newbie@newbie:~$ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print([1,2,3,4,5])
[1, 2, 3, 4, 5]
>>>

```

Gambar 1.22 List Data

6. Tuple, tipe data ini untuk menyimpan berbagai jenis tipe data dan isinya tidak bisa diubah-ubah seperti list. Contoh implementasi bisa dilihat seperti berikut.

```

1 print((1,2,3,4,5))
2

```

Listing 1.9 Tipe Data tuple

```

(base) newbie@newbie:~$ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print([1,2,3,4,5])
[1, 2, 3, 4, 5]
>>> print((1,2,3,4,5))
(1, 2, 3, 4, 5)
>>>

```

Gambar 1.23 Tuple Data

7. Dictionary, tipe data ini untuk menyimpan berbagai tipe data berupa pasangan petunjuk dan nilainya. Contoh implementasi bisa dilihat seperti berikut.

```

1 print({"nama":"Budi", 'umur':20})
2

```

Listing 1.10 Tipe Data Dictionary

```

(base) newbie@newbie:~$ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print({"nama":"Budi", 'umur':20})
{'nama': 'Budi', 'umur': 20}
>>>

```

Gambar 1.24 Dictionary Data

1.5.4 Variabel

Variabel merupakan lokasi memori yang dicadangkan untuk menyimpan nilai. Variabel menyimpan data yang dilakukan selama program dieksekusi, yang nantinya isi dari variabel bisa dirubah-rubah suatu saat.

Variabel dalam pemrograman python, bersifat dinamis. Artinya tipe data dalam variabel tersebut tidak perlu di deklarasikan dan isi variabel tersebut bisa dirubah ketika menjalankan program.

Beberapa aturan dalam penulisan variabel di pemrograman python, sebagai berikut:

1. Karakter utama harus berupa huruf atau garis bawah
2. karakter selanjutnya boleh huruf, angka maupun garis bawah.
3. karakter pada nama variabel bersifat case-sensitif, artinya huruf kecil dan huruf besar memiliki makna yang berbeda. Contoh. variabel **contoh** dan **Contoh** merupakan variabel yang berbeda.

Untuk pembuatan variabel di pemrograman python sangat mudah, cukup ketik nama variabel dengan diikuti tanda `dan` diikuti dengan isi nilai variabel tersebut.

```

1 panjang = 10
2 lebar = 5
3 luas = panjang * lebar
4 print(luas)

```

Listing 1.11 Penggunaan Variabel

```
(base) newbie@newbie:~$ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> panjang = 10
>>> lebar = 5
>>> luas = panjang * lebar
>>> print(luas)
50
>>> □
```

Gambar 1.25 Variabel

1.5.5 Looping

Looping adalah sebuah kondisi dalam bahasa pemrograman yang dieksekusi secara berurutan. Jika pernyataan pertama dijalankan, maka akan diikuti oleh pernyataan yang kedua dan seterusnya. Tetapi terkadang ada dalam suatu kondisi tertentu, kita harus menulis banyak kode. Tentunya jika itu semua dilakukan secara manual akan tidak bisa memberikan performansi yang baik dalam pemrograman tersebut. Oleh karena itu, muncullah looping atau pengolahan.

Pengulangan dalam pemrograman python terbagi menjadi 3 bagian, seperti :

1. While loop

While loop disini akan dijalankan selama kondisi dalam pemrograman tersebut masih bernilai benar atau True. Contoh implementasi while loop.

```
1 count = 0
2 while (count < 9):
3     print 'The count is:', count
4     count = count + 1
5
6     print ("Good bye!")
```

Listing 1.12 Penggunaan While Loop

Pada syntax 1.12, dijelaskan pada baris pertama terdapat inisiasi variabel. Lalu baris kedua untuk melakukan looping berdasarkan variabel di baris pertama dengan kondisi jika variabel tersebut akan diloopng sampai dibawah 9. Lalu baris ketiga digunakan untuk menampilkan ini looping ke berapa. Lalu pada baris keempat untuk inisiasi variabel baris pertama dengan kondisi variabel tersebut ditambah 1.

2. For loop

Pengulangan dengan menggunakan for memiliki kemampuan untuk mengulang atau me looping item dari urutan yang ada seperti string ata list. Contoh implementasi for loop:

```
1 buah = ["nanas", "apel", "jeruk"]
2 for makanan in buah:
```



```
3 print "Saya suka makan", makanan
```

Listing 1.13 Penggunaan For Loop

Pada syntax 1.13, penjelasan untuk baris pertama ialah untuk inisiasi variabel dengan tipe data list. Lalu pada baris kedua digunakan untuk looping dengan variabel makanan yang isinya di ambil dari variabel di baris pertama. Lalu pada baris ketiga, untuk menampilkan hasil looping.

1.5.6 Fungsi

Fungsi dalam pemrograman python merupakan sebuah blok kode yang terorganisir dan dapat digunakan kembali untuk suatu action tertentu di suatu saat. Penggunaan fungsi dapat memberikan tingkat modularitas yang baik terhadap program tersebut serta tingkat penggunaan kode yang tinggi.

Dalam pendeklarasian fungsi dalam pemrograman python, terdapat beberapa aturan yang harus dilakukan, seperti berikut.

1. Pembuatan fungsi dimulai dengan kata **def** lalu diikuti dengan nama fungsi serta tanda kurung ().
2. Setiap parameter masukan harus dimasukkan kedalam tanda kurung (). Dan bisa di atur juga nilai dari parameter tersebut.
3. Setiap fungsi blok kode harus dimulai dengan tanda (:) dan indentasi.
4. Setiap fungsi blok kode harus memiliki pengembalian nilai.

Tentunya kita semua, jika hanya membaca teori mungkin masih kebingungan. Maka dari itu, langsung aja ke contoh implementasinya seperti berikut.

```
1 def printme( str ) :  
2     "This prints a passed string into this function"  
3     print ( str )  
4     return
```

Listing 1.14 Fungsi Python

Pada syntax 1.14, baris pertama menjelaskan tentang pendefinisian nama fungsi. Lalu baris kedua menjelaskan tentang isi string. Lalu pada baris ketiga menjelaskan untuk menampilkan isi string tersebut. Pada baris keempat menjelaskan untuk fungsi return ketika fungsi tersebut dipanggil.

1.5.7 Modul

Modul merupakan sebuah file py yang berisikan sekumpulan kode python. Sebuah file py bisa disebut modul.

Penerapan modul ini biasanya disebut konsep OOP (Object Orientied Programming) dalam pemrograman python. Karena pada dasarnya ini digunakan untuk

membagi file-file program yang besar menjadi lebih kecil supaya mudah dalam manage dan diorganisir. Sehingga nantinya bisa di reusable, artinya kode-kode tersebut bisa di gunakan kapan saja.

Contoh penerapan modul tersebut seperti berikut. Saya membuat file dengan nama test.py dengan isi file berikut.

```
1  def jumlah(a, b):  
2      """ Fungsi ini menambahkan dua bilangan  
3      dan mengembalikan hasilnya """  
4      result = a + b  
5      return result
```

Listing 1.15 Modul

Lalu kita ketik di command line python yang isi kodenya seperti berikut.

```
1  import test  
2  test.jumlah(5,6)
```

Listing 1.16 Import modul

Pada syntax 1.16, pada baris pertama digunakan untuk mengimport file test.py. Lalu pada baris kedua digunakan untuk memanggil nama fungsi di file test.py.

Dalam python juga, kita bisa menggunakan library-library yang telah disediakan oleh python itu sendiri. Untuk cara import library atau modul ada beberapa cara, seperti berikut.

1. cara import standard

```
1 import nama_module
2
```

Listing 1.17 Import modul

```
(base) newbie@newbie:~$ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> □
```

Gambar 1.26 Import Module

2. cara import dengan alias

```
1 import nama_module as alias
2
```

Listing 1.18 Import modul

```
(base) newbie@newbie:~$ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import os as s
>>> □
```

Gambar 1.27 Import Module Alias

3. cara import namun hanya mengambil sebagian dari library tersebut.

```
1 from nama_module import something
2
```

Listing 1.19 Import module Sebagian

```
(base) newbie@newbie:~$ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from sklearn.feature_extraction.text import TfidfVectorizer
>>> □
```

Gambar 1.28 Import Module Sebagian

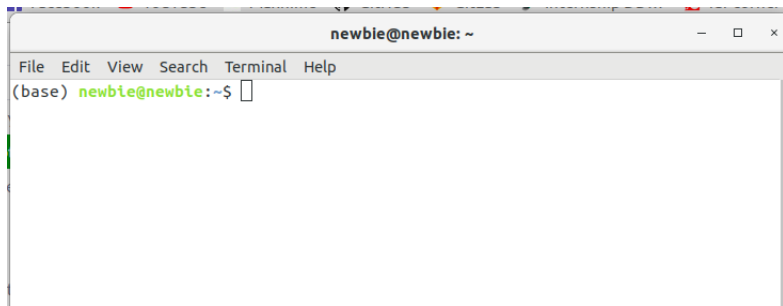
1.6 Instalasi Pip

1.6.1 Windows

1.6.2 Linux

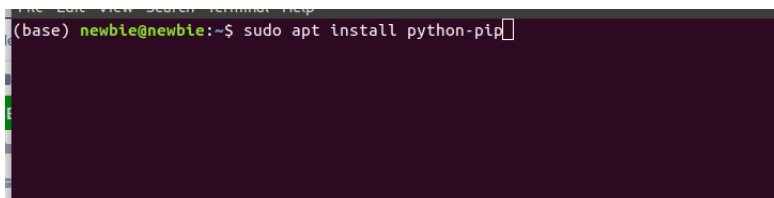
Dalam linux, cukup mudah untuk melakukan instalasi pip.

1. Buka Terminal linux.

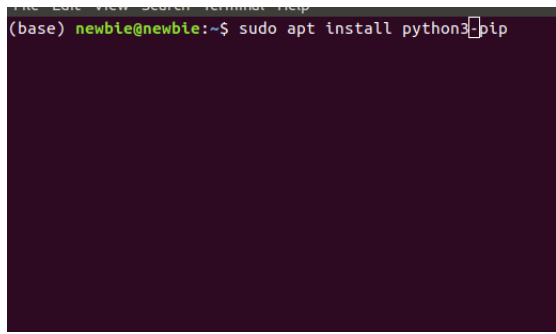


Gambar 1.29 Terminal

2. Ketik **sudo apt install python-pip** untuk python2
Sedangkan untuk python3, menggunakan perintah **sudo apt install python3-pip**



Gambar 1.30 Pip



```
(base) newbie@newbie:~$ sudo apt install python3-pip
```

Gambar 1.31 Pip 3

1.7 Pip

PIP adalah sebuah program di Python untuk mengelola, install, mencari dan uninstall package, jika diibaratkan Pip dalam Android itu seperti Google Play Store. Dimana kita mencari, menginstall dan uninstall apapun yang kita inginkan.

1.8 Cara Penggunaan Pip

Penerapan pip dalam python sangat memberikan kemudahan, dimana kita perlu mendownload source module dengan mencari di internet, menambahkan site-package dari module tersebut yang sudah disediakan oleh package tersebut. Untuk cara penggunaan dari pip itu sendiri menggunakan command line.

1.8.0.1 Install Package Untuk melakukan instalasi package, cukup ketik perintah.

```
1 pip install \textbf{nama_package}
2 pip install \textbf{nama_package}==version_of_package
```

Listing 1.20 install package

Pada baris 1, itu adalah perintah install package dimana mengambil version package terakhir. Sedangkan baris 2, perintah install package dengan menetapkan version package yang dibutuhkan.

1.8.0.2 Menampilkan Package Untuk menampilkan informasi package yang sudah terinstall, cukup menggunakan perintah.

```
1 pip list
```

Listing 1.21 List package

Sedangkan untuk menampilkan informasi package, version package, lokasi dan dependencies dari package tersebut, cukup menggunakan perintah.

```
1 pip show \textbf{nama_package}
```

Listing 1.22 Show package

1.8.0.3 Uninstall Package Untuk melakukan uninstall package, cukup menggunakan perintah.

```
1 pip uninstall \textbf{nama_package}
```

Listing 1.23 uninstall package

1.9 Ekstensi File

1.9.1 .py

File py adalah salah satu file program atau skrip yang ditulis menggunakan python, bahasa pemrograman berorientasi objek. File dapat dibuat dan diedit menggunakan teks editor, tetapi untuk menjalankannya membutuhkan bahasa Python. File PY sering digunakan untuk pemrograman server web dan sistem komputer administrasi lainnya.

1.9.2 .ipynb

File IPYNB adalah dokumen notebook yang digunakan oleh Jupyter Notebook, sebuah lingkungan komputasi interaktif yang dirancang untuk membantu para *scientists* bekerja menggunakan bahasa Python dan data mereka. File ini berisi semua konten dari sesi aplikasi web Notebook Jupyter, yang mencakup input dan output perhitungan, matematika, gambar, dan teks penjelasan. selain itu, file IPYNB dapat diekspor ke format .HTML, .PDF, reStructuredText, dan LaTeX. IPYN menggabungkan 2 komponen, yaitu:

1. Aplikasi web: alat berbasis browser untuk penulisan dokumen interaktif yang menggabungkan teks penjelas, matematika, perhitungan dan output media yang kaya.
2. Dokumen buku catatan: representasi semua konten yang terlihat dalam aplikasi web, termasuk input dan output dari perhitungan, teks penjelasan, matematika, gambar, dan representasi objek yang kaya media.

1.9.3 Konversi File

Sebelum melakukan konversi file, perlu ada beberapa hal yang harus dilakukan seperti menginstal library `ipython` dan `nbconvert`.

1.9.3.1 `.py to .ipynb` Cukup menjalankan kode `ipython nbconvert to script abc.py`

1.9.3.2 `.ipynb to .py` Cukup menjalankan ode `ipython nbconvert to script abc.ipynb`

1.10 Mengapa harus ipynb

IPYNB lebih unggul dalam pemrograman yang disebut *literate programming*. *Literate programming* merupakan gaya pengembangan perangkat lunak dipelopori oleh ilmuwan komputer Stanford, Donald Knuth. Jenis pemrograman ini menekankan pendekatan prosa pertama di mana eksposisi dengan teks *human-friendly* diselingi dengan blok kode.

BAB 2

DATA SCIENCE

2.1 Apa itu Big Data

Pembuatan data terjadi pada tingkat rekor. Pada 2010, dunia menghasilkan lebih dari 1ZB data; pada 2014, kita akan menghasilkan 7ZB setahun. Sebagian besar ledakan data ini adalah hasil dari peningkatan dramatis dalam perangkat yang terletak di pinggiran jaringan termasuk sensor yang tertanam, *smartphone*, dan komputer tablet. Semua data ini menciptakan peluang baru untuk "mengekstraksi lebih banyak nilai" dalam genomika manusia, perawatan kesehatan, minyak dan gas, pencarian, pengawasan, keuangan, dan banyak bidang lainnya. Kita memasuki zaman "*Big Data*." [1]

2.1.1 Pengertian

Sebelum memahami '*Big Data*', perlu terlebih dahulu tahu apa itu data. Data merupakan jumlah, karakter, atau simbol tempat operasi dilakukan oleh komputer, yang dapat disimpan dan dikirim dalam bentuk sinyal listrik dan direkam pada media perekaman magnetik, optik, atau mekanis. *Big Data* juga merupakan data tetapi dengan ukuran yang sangat besar. *Big Data* merupakan istilah yang digunakan un-

tuk mendeskripsikan kumpulan data yang berukuran sangat besar namun tumbuh secara eksponensial seiring waktu. Singkatnya, data tersebut sangat besar dan kompleks sehingga tidak ada alat manajemen data tradisional yang dapat menyimpan atau memprosesnya secara efisien.

Big Data adalah tentang tantangan yang semakin besar yang dihadapi organisasi ketika mereka berhadapan dengan sumber data atau informasi yang besar dan berkembang pesat yang juga menghadirkan berbagai analisis kompleks dan masalah penggunaan. Ini dapat mencakup [1]:

1. Memiliki infrastruktur komputasi yang dapat menelan, memvalidasi, dan menganalisis volume (ukuran dan / atau tingkat) data yang tinggi.
2. Menilai data campuran (terstruktur dan tidak terstruktur) dari berbagai sumber.
3. Berurusan dengan konten yang tidak dapat diprediksi tanpa skema atau struktur yang jelas.
4. Mengaktifkan pengumpulan, analisis, dan jawaban waktu-nyata-dekat-waktu-nyata.

2.1.2 Contoh *Big Data*

Contoh dari *Big Data* misalnya pada *New York Stock Exchange* yang menghasilkan sekitar satu *terabyte* data perdagangan baru per hari. Media sosial seperti Facebook, statistik menunjukkan bahwa setiap harinya lebih dari 500 *terabyte* data baru dapat dicerna ke dalam basis data situs. Data ini terutama dihasilkan dalam hal unggahan foto dan video, pertukaran pesan, memberi komentar, dll. *Big Data* bisa datang dari berbagai sumber, seperti sistem transaksi bisnis, database pelanggan, catatan medis, log clickstream internet, aplikasi mobile, jejaring sosial, repositori penelitian ilmiah, data yang dihasilkan mesin, dan sensor data real-time yang digunakan dalam internet benda (IOT) lingkungan.

Data dapat dibiarkan dalam bentuk mentah dalam sistem data besar atau diproses menggunakan alat penambahan data atau perangkat lunak persiapan data sehingga siap untuk penggunaan analitik tertentu. Menggunakan data pelanggan sebagai contoh, berbagai cabang analitik yang dapat dilakukan dengan informasi yang ditemukan dalam set data besar meliputi yang berikut ini:

1. ***Comparative analysis.*** Ini termasuk pemeriksaan metrik perilaku pengguna dan pengamatan keterlibatan pelanggan waktu nyata untuk membandingkan produk, layanan, dan otoritas merek perusahaan dengan pesaing.
2. ***Social media listening.*** Memuat informasi tentang apa yang orang katakan di media sosial tentang bisnis atau produk tertentu yang melampaui apa yang dapat disampaikan dalam jajak pendapat atau survei. Data ini dapat digunakan untuk membantu mengidentifikasi target audiens untuk kampanye pemasaran dengan mengamati aktivitas seputar topik spesifik di berbagai sumber.

3. **Marketing analysis.** Memuat informasi yang dapat digunakan untuk membuat promosi produk baru, layanan dan inisiatif lebih informatif dan inovatif.
4. **Customer satisfaction and sentiment analysis.** Semua informasi yang dikumpulkan dapat mengungkapkan bagaimana perasaan pelanggan tentang suatu perusahaan atau merek, jika ada masalah potensial yang mungkin timbul, bagaimana loyalitas merek dapat dipertahankan dan bagaimana upaya layanan pelanggan dapat ditingkatkan.

2.1.3 Cara Kerja *Big Data*

Big Data dapat dikategorikan sebagai tidak terstruktur atau terstruktur. Data terstruktur terdiri dari informasi yang sudah dikelola oleh organisasi dalam *database* dan *spreadsheet*; sering bersifat numerik. Data yang tidak terstruktur adalah informasi yang tidak terorganisir dan tidak termasuk dalam model atau format yang ditentukan sebelumnya. Termasuk juga data yang dikumpulkan dari sumber media sosial, yang membantu institusi mengumpulkan informasi tentang kebutuhan pelanggan.

Big Data dapat dikumpulkan dari komentar yang dibagikan secara publik di jejaring sosial dan situs *web*, dikumpulkan secara sukarela dari elektronik dan aplikasi pribadi, melalui kuesioner, pembelian produk, dan *check-in* elektronik. Kehadiran sensor dan input lain dalam perangkat pintar memungkinkan data dikumpulkan di berbagai situasi dan keadaan. *Big data* paling sering disimpan dalam *database* komputer dan dianalisis menggunakan perangkat lunak yang dirancang khusus untuk menangani set data yang besar dan kompleks. Banyak perusahaan perangkat lunak sebagai layanan (SaaS) mengkhususkan diri dalam mengelola jenis data yang kompleks ini.

2.1.4 Penggunaan *Big Data*

Analisis data melihat hubungan antara berbagai jenis data, seperti data demografis dan riwayat pembelian, untuk menentukan apakah ada korelasi. Penilaian semacam itu dapat dilakukan sendiri di dalam perusahaan atau secara eksternal oleh pihak ketiga yang berfokus pada pemrosesan data besar ke dalam format yang dapat dicerna. Bisnis sering menggunakan penilaian data besar oleh para ahli untuk mengubahnya menjadi informasi yang dapat ditindaklanjuti.

Hampir setiap departemen di perusahaan dapat memanfaatkan temuan dari analisis data, dari sumber daya manusia dan teknologi hingga pemasaran dan penjualan. Tujuan dari *big data* adalah untuk meningkatkan kecepatan di mana produk sampai ke pasar, untuk mengurangi jumlah waktu dan sumber daya yang dibutuhkan untuk mendapatkan adopsi pasar, target audiens, dan untuk memastikan bahwa pelanggan tetap puas.

2.1.5 Jenis *Big Data*

Big Data memiliki 3 jenis tipe data, yaitu :

1. *Structured*
2. *Unstructured*
3. *Semi-structured*

2.1.5.1 *Structured data*

Data yang disimpan dalam baris dan kolom, sebagian besar numerik, di mana makna setiap item data didefinisikan. Jenis data ini merupakan sekitar 10% dari total data saat ini dan dapat diakses melalui sistem manajemen basis data. Contoh sumber data terstruktur (atau tradisional) termasuk register resmi yang dibuat oleh lembaga pemerintah untuk menyimpan data pada individu, perusahaan dan real estat; dan sensor di industri yang mengumpulkan data tentang proses. Saat ini, data sensor adalah salah satu area yang tumbuh cepat, khususnya sensor yang dipasang di pabrik untuk memantau pergerakan, suhu, lokasi, cahaya, getaran, tekanan, cairan dan aliran.

Setiap data yang dapat disimpan, diakses dan diproses dalam bentuk format tetap disebut sebagai data 'terstruktur'. Selama periode waktu, bakat dalam ilmu komputer telah mencapai keberhasilan yang lebih besar dalam mengembangkan teknik untuk bekerja dengan data semacam itu (di mana formatnya sudah diketahui sebelumnya) dan terdapat nilai. Namun, saat ini, kami meramalkan masalah ketika ukuran data tersebut tumbuh sangat besar, ukuran tipikal sedang populer di banyak zettabytes.

2.1.5.2 *Unstructured data*

Berbagai bentuk data seperti mis. teks, gambar, video, dokumen, dll. Bisa juga dalam bentuk keluhan pelanggan, kontrak, atau *email* internal. Jenis data ini menyumbang sekitar 90% dari data yang dibuat pada abad ini. Faktanya, pertumbuhan vulkanik media sosial (mis. Facebook dan Twitter), sejak pertengahan dekade terakhir, bertanggung jawab atas bagian utama dari data tidak terstruktur yang kita miliki saat ini. Data yang tidak terstruktur tidak dapat disimpan menggunakan database relasional tradisional. Menyimpan data dengan variasi dan kompleksitas seperti itu membutuhkan penggunaan sistem penyimpanan yang memadai, yang biasa disebut sebagai basis data NoSQL, seperti mis. MongoDB dan CouchDB. Pentingnya data yang tidak terstruktur terletak pada hubungan timbal balik yang tertanam yang mungkin tidak ditemukan jika jenis data lain dipertimbangkan. Apa yang membuat data yang dihasilkan di media sosial berbeda dari tipe data lainnya adalah bahwa data di media sosial memiliki selera pribadi.

Setiap data dengan bentuk atau struktur yang tidak dikenal diklasifikasikan sebagai data yang tidak terstruktur. Selain ukurannya yang besar, data yang tidak terstruktur menimbulkan banyak tantangan dalam hal pemrosesan untuk mendapatkan nilai darinya. Contoh khas data tidak terstruktur adalah sumber data heterogen yang berisi kombinasi file teks sederhana, gambar, video dll. Sekarang organisasi saat ini memiliki banyak data yang tersedia dengan mereka tetapi sayangnya, mereka tidak tahu bagaimana mendapatkan nilai dari itu karena data ini dalam bentuk mentah atau format tidak terstruktur.

2.1.5.3 *Semi-structured*

Data semi-terstruktur dapat berisi kedua bentuk data. Kita dapat melihat data semi-terstruktur sebagai formulir terstruktur tetapi sebenarnya tidak didefinisikan dengan mis. definisi tabel dalam DBMS relasional. Contoh data semi-terstruktur adalah data yang direpresentasikan dalam file XML.

2.1.6 Tantangan *Big Data*

Selain kapasitas pemrosesan dan masalah biaya, merancang arsitektur data besar adalah tantangan umum lainnya bagi pengguna. Sistem big data harus disesuaikan dengan kebutuhan khusus organisasi, sebuah usaha DIY yang mengharuskan tim TI dan pengembang aplikasi untuk mengumpulkan satu set alat dari semua teknologi yang tersedia. Menyebarkan dan mengelola sistem big data juga membutuhkan keterampilan baru dibandingkan dengan yang dimiliki oleh *database administrator (DBA)* dan pengembang yang berfokus pada perangkat lunak relasional.

Kedua masalah tersebut dapat diatasi dengan menggunakan layanan *cloud* yang dikelola, tetapi manajer TI perlu mengawasi penggunaan *cloud* untuk memastikan biaya tidak keluar dari kendali. Selain itu, memigrasikan kumpulan data di tempat dan memproses beban kerja ke *cloud* sering kali merupakan proses yang rumit bagi organisasi.

Membuat data dalam sistem data besar dapat diakses oleh *data scientists* dan analis lain juga merupakan tantangan, terutama di lingkungan terdistribusi yang mencakup campuran berbagai *platform* dan penyimpanan data. Untuk membantu analis menemukan data yang relevan, tim TI dan analitik semakin berupaya untuk membuat katalog data yang menggabungkan fungsi manajemen metadata dan aliran data. Kualitas data dan tata kelola data juga perlu menjadi prioritas untuk memastikan bahwa kumpulan data besar bersih, konsisten dan digunakan dengan benar.

2.1.7 Karakteristik *Big Data*

1. *Volume*

Nama *Big Data* sendiri terkait dengan ukuran yang sangat besar. Ukuran data memainkan peran yang sangat penting dalam menentukan nilai dari data. Juga, apakah data tertentu benar-benar dapat dianggap sebagai Data Besar atau tidak, tergantung pada volume data. Oleh karena itu, '*Volume*' adalah salah satu karakteristik yang perlu dipertimbangkan saat berurusan dengan *Big Data*.

2. *Variety*

Aspek *Big Data* selanjutnya adalah keanekaragamannya. Varietas mengacu pada sumber-sumber yang heterogen dan sifat data, baik terstruktur dan tidak terstruktur. Selama hari-hari sebelumnya, *spreadsheet* dan basis data adalah satu-satunya sumber data yang dipertimbangkan oleh sebagian besar aplikasi. Saat ini, data dalam bentuk email, foto, video, perangkat pemantauan, PDF, audio, dll. Juga sedang dipertimbangkan dalam aplikasi analisis. Keragaman data

yang tidak terstruktur ini menimbulkan masalah tertentu untuk penyimpanan, penambahan, dan analisis data.

3. *Velocity*

Istilah '*velocity*' mengacu pada kecepatan pembuatan data. Seberapa cepat data dihasilkan dan diproses untuk memenuhi permintaan, menentukan potensi nyata dalam data. *Big Data Velocity* berkaitan dengan kecepatan di mana data mengalir dari sumber-sumber seperti proses bisnis, log aplikasi, jaringan, dan situs media sosial, sensor, perangkat *Mobile*, dll. Aliran data sangat besar dan berkelanjutan.

4. *Variability*

Mengacu pada ketidakkonsistenan yang dapat ditunjukkan oleh data pada waktu tertentu, sehingga menghambat proses untuk dapat menangani dan mengelola data secara efektif.

2.1.8 Keuntungan dan Kerugian *Big Data*

Peningkatan jumlah data yang tersedia menghadirkan peluang dan masalah. Secara umum, memiliki lebih banyak data tentang pelanggan seseorang (dan pelanggan potensial) harus memungkinkan perusahaan untuk menyesuaikan produk dan upaya pemasaran mereka dengan lebih baik untuk menciptakan tingkat kepuasan tertinggi dan mengulangi bisnis. Perusahaan yang mampu mengumpulkan data dalam jumlah besar diberikan kesempatan untuk melakukan analisis yang lebih dalam dan lebih kaya.

Sementara analisis yang lebih baik adalah positif, data besar juga dapat membuat kelebihan dan kebisingan. Perusahaan harus dapat menangani volume data yang lebih besar, sambil menentukan data mana yang mewakili sinyal dibandingkan dengan noise. Menentukan apa yang membuat data relevan menjadi faktor utama.

Selanjutnya, sifat dan format data dapat memerlukan penanganan khusus sebelum ditindaklanjuti. Data terstruktur, yang terdiri dari nilai numerik, dapat dengan mudah disimpan dan disortir. Data yang tidak terstruktur, seperti email, video, dan dokumen teks, mungkin memerlukan teknik yang lebih canggih untuk diterapkan sebelum menjadi berguna.

2.2 Mengapa menggunakan python

Memilih bahasa pemrograman dalam bidang *Big Data* merupakan hal spesifik dan tergantung pada tujuan proyek. Namun, apa pun yang menjadi tujuannya, *Python* dan *Big Data* adalah kombinasi yang tidak terpisahkan ketika kita mempertimbangkan bahasa pemrograman untuk fase pengembangan *Big Data*. Ini adalah keputusan penting karena sekali Anda mulai mengembangkan proyek, maka akan sulit untuk bermigrasi dalam bahasa lain. Selain itu, tidak semua proyek *big data* memiliki

tujuan yang sama. Misalnya, dalam proyek *big data*, tujuannya mungkin hanya memanipulasi data atau membangun analitik sedangkan yang lain bisa untuk *Internet of Things (IoT)*.

Python adalah bahasa pemrograman serba guna yang memungkinkan programmer menulis lebih sedikit baris kode dan membuatnya lebih mudah dibaca. Python memiliki fitur *scripting* dan selain itu menggunakan banyak perpustakaan canggih seperti NumPy, Matplotlib, dan SciPy yang membuatnya berguna untuk komputasi ilmiah. Python adalah alat yang sangat baik dan sangat cocok sebagai kombinasi *big data* python untuk analisis data karena alasan di bawah ini:

1. Sumber Terbuka (*Open source*)

Python adalah bahasa pemrograman *open source* yang dikembangkan menggunakan model berbasis komunitas. Itu dapat dijalankan di lingkungan Windows dan Linux. Selain itu, Anda dapat *platformporting* ke *platform* lain karena mendukung banyak *platform*.

2. Dukungan Perpustakaan (*Library Support*)

Python banyak digunakan untuk komputasi ilmiah di bidang akademik dan beberapa industri. Python terdiri dari sejumlah besar pustaka analitik yang teruji dengan baik yang mencakup paket-paket seperti :

- (a) Numerical computing
- (b) Data analysis
- (c) Statistical analysis
- (d) Visualization
- (e) Machine learning

3. Kecepatan (*Speed*)

Karena Python adalah bahasa tingkat tinggi, Python memiliki banyak manfaat yang mempercepat pengembangan kode. Ini memungkinkan ide prototyping yang membuat pengkodean cepat sambil menjaga transparansi yang besar antara kode dan pelaksanaannya. Sebagai hasil dari transparansi kode, pemeliharaan kode dan proses menambahkannya ke basis kode dalam lingkungan pengembangan multi-pengguna menjadi mudah.

4. Jangkauan (*Scope*)

Python adalah bahasa pemrograman berorientasi objek yang juga mendukung struktur data tingkat lanjut seperti daftar, set, tuple, kamus dan banyak lagi. Ini mendukung banyak operasi komputasi ilmiah seperti operasi matriks, bingkai data, dll. Kemampuan ini dalam Python meningkatkan ruang lingkup untuk menyederhanakan dan mempercepat operasi data.

5. Dukungan Pemrosesan Data (*Data Processing Support*)

Python menyediakan dukungan canggih untuk data gambar dan suara karena fitur *inbuilt* untuk mendukung pemrosesan data untuk data tidak terstruktur dan

tidak konvensional yang merupakan kebutuhan umum dalam data besar ketika menganalisis data media sosial. Ini adalah alasan lain untuk membuat Python dan data besar bermanfaat satu sama lain.

2.3 Tools

2.4 Library

BAB 3

JUPYTER NOTEBOOK

3.1 Apakah itu Jupyter Notebook ?

3.1.1 Notebook documents

Notebook documents (atau "buku catatan", semuanya huruf kecil) adalah dokumen yang diproduksi oleh Jupyter Notebook App, yang berisi kode komputer (mis. Python) dan elemen teks kaya (paragraf, persamaan, angka, tautan, dll ...). Dokumen Notebook adalah dokumen yang dapat dibaca manusia yang berisi uraian analisis dan hasilnya (angka, tabel, dll.) Serta dokumen yang dapat dieksekusi yang dapat dijalankan untuk melakukan analisis data.

3.1.2 Jupyter Notebook

Jupyter Notebook adalah aplikasi *web open-source* yang memungkinkan untuk membuat dan berbagi dokumen yang berisi kode langsung, persamaan, visualisasi, dan teks naratif. Penggunaan meliputi: pembersihan dan transformasi data, simulasi numerik, pemodelan statistik, visualisasi data, pembelajaran mesin, dan banyak lagi.

Jupyter Notebook App adalah aplikasi server-klien yang memungkinkan pengeditan dan menjalankan dokumen notebook melalui browser web. Aplikasi Notebook Jupyter dapat dijalankan pada desktop lokal yang tidak memerlukan akses internet (seperti dijelaskan dalam dokumen ini) atau dapat diinstal pada server jarak jauh dan diakses melalui internet. Selain menampilkan / mengedit / menjalankan dokumen notebook, Aplikasi Notebook Jupyter memiliki "Dasbor" (Dasbor Notebook), "panel kontrol" yang memperlihatkan file-file lokal dan memungkinkan untuk membuka dokumen notebook atau mematikan kernel mereka.

3.1.3 Kernel

Kernel notebook adalah "mesin komputasi" yang mengeksekusi kode yang terkandung dalam dokumen Notebook. Kernel ipython, dirujuk dalam mengeksekusi kode python. Ketika membuka dokumen Notebook, kernel yang terkait diluncurkan secara otomatis. Ketika notebook dijalankan (baik sel demi sel atau dengan menu Cell -> Run All), kernel melakukan perhitungan dan menghasilkan hasilnya. Bergantung pada jenis perhitungan, kernel dapat mengkonsumsi CPU dan RAM yang signifikan. Perhatikan bahwa RAM tidak dirilis sampai kernel dimatikan.

Ketika Jupyter memulai kernel, ia mengirimkannya file koneksi. Ini menentukan cara mengatur komunikasi dengan *frontend*. Ada dua opsi untuk menulis kernel:

1. Pengguna dapat menggunakan kembali mesin kernel IPython untuk menangani komunikasi, dan cukup jelaskan bagaimana mengeksekusi kode. Hal ini jauh lebih sederhana jika bahasa target dapat didorong dari Python.
2. Pengguna dapat mengimplementasikan mesin kernel dalam bahasa target. Ini lebih banyak bekerja pada awalnya, tetapi orang-orang yang menggunakan kernel pengguna mungkin lebih mungkin untuk berkontribusi jika itu dalam bahasa yang mereka tahu.

3.1.4 Notebook Dashboard

Notebook Dashboard adalah komponen yang ditampilkan pertama kali ketika membuka Aplikasi Notebook Jupyter. Dasbor Notebook terutama digunakan untuk membuka dokumen notebook, dan untuk mengelola kernel yang berjalan (memvisualisasikan dan mematikan). *Notebook Dashboard* memiliki fitur lain yang mirip dengan manajer file, yaitu menavigasi folder dan mengganti nama / menghapus file.

3.2 Perbedaan Jupyter Notebook dan Google Collab

Google Colaboratory (Google Colab) merupakan tools baru dari Google Internal Research yang ditujukan membantu para *Researcher* dalam mengolah data, khususnya bidang *Machine Learning*. Google Colab hampir mirip penggunaannya seperti Jupyter Notebook namun tidak memerlukan pengaturan atau setup terlebih dahulu

sebelum digunakan dan berjalan sepenuhnya pada Cloud dengan memanfaatkan media penyimpanan Google Drive. *Researcher* dapat menulis dan mengeksekusi kode, menyimpan dan membagikan analisis, serta mengakses sumber daya komputasi yang kuat seperti layanan GPU secara gratis dari browser.

Jupyter Notebook dan Google Colab memiliki perbedaan sebagai berikut:

1. Infrastruktur

Google Colab berjalan di Google Cloud Platform (GCP). Karena itu kuat, fleksibel.

2. Perangkat keras

Google Colab baru-baru ini menambahkan dukungan untuk Tensor Processing Unit (TPU) selain GPU dan CPU yang ada. Jadi, ini masalah besar bagi semua orang yang belajar mendalam.

3. Harga

Meskipun begitu mahir dalam hal perangkat keras, layanan yang disediakan oleh Google Colab sepenuhnya gratis. Ini membuatnya lebih dahsyat.

4. Integrasi dengan Google Drive

Menarik dapat menggunakan drive google sebagai sistem file interaktif dengan Google Colab. Ini membuatnya mudah untuk menangani file yang lebih besar.

5. Hikmah bagi Komunitas Riset dan Startup

Mungkin ini adalah satu-satunya alat yang tersedia di pasar yang menyediakan PaaS yang begitu bagus secara gratis bagi pengguna. Ini sangat membantu bagi startup, komunitas riset dan siswa di ruang belajar yang mendalam.

3.3 Cara Instalasi

3.3.1 Windows

3.3.2 Ubuntu

3.3.3 Editor

3.4 Cara Penggunaan

3.4.1 On Browser

3.4.2 Editor

3.4.3 Toolbar

BAB 4

NUMPY

4.1 Numpy

BAB 5

PANDAS

5.1 Pandas

Pandas adalah sebuah *library* yang sering digunakan dalam data science atau big data. Dimana *library* ini digunakan untuk menyediakan struktur data dan analisis data yang sangat mudah digunakan dalam bahasa pemrograman python. Atau lebih tepatnya, pandas untuk analisis dan struktur data yang diperlukan untuk membersihkan data dan ditampilkan dalam bentuk tabel.

Pandas juga dapat melakukan beberapa hal seperti menggabungkan data, menghilangkan data yang hilang, dan lain-lain. Oleh karena itu, pandas merupakan *library* yang wajib digunakan dalam pengolahan data tingkat tinggi atau biasa disebut statistik.

Pandas memiliki struktur dasar yaitu DataFrame. DataFrame itu sendiri merupakan sebuah kumpulan data berurutan. Jika diibaratkan DataFrame seperti database, dimana memiliki kolom dan baris serta setiap baris mewakili *record* data. Pandas bisa membaca berbagai macam ekstensi file, seperti **.csv**, **.txt**, **.tsv** dan lainnya.

5.2 Arsitektur DataFrame

	color	director_name	num_critc_for_reviews	duration	actor_2_facebook_likes	imdb_score	aspect_ratio	movie_facebook_likes
0	Color	James Cameron	723.0	178.0	...	936.0	7.9	1.78
1	Color	Gore Verbinski	302.0	169.0	...	5000.0	7.1	2.35
2	Color	Sam Mendes	602.0	148.0	...	393.0	6.8	2.35
3	Color	Christopher Nolan	813.0	164.0	...	23000.0	8.5	2.35
4	NaN	Doug Walker	NaN	NaN	...	12.0	7.1	NaN

Gambar 5.1 Arsitektur DataFrame

Atribut :

1. Column
2. Column Name
3. More column to display
4. Index label
5. Index
6. Missing Value
7. Data

5.3 DataFrame

DataFrame itu sendiri merupakan struktur dasar dari pandas ini dan pengertian DataFrame juga sudah dijelaskan sebelumnya. Oleh karena itu disini akan menjelaskan DataFrame penggunaannya bagaimana dan bisa untuk apa saja.

Hal yang harus diperhatikan dalam menggunakan *library* ini adalah melakukan *import library* tersebut dengan cara :

```
1 import pandas as pd
```

Listing 5.1 Import-Module

1. Membuat DataFrame.

Pandas yang pertama bisa membuat dataframe, dimana bisa menampilkan data yang ke dalam bentuk tabel.

```

1 test = pd.read_csv('products.csv')
2 test_DF = pd.DataFrame(test)
3 test_DF
4

```

Listing 5.2 Create DataFrame

Pada kode 5.2, pada baris pertama untuk membaca data yang ada bentuk **.csv**, sedangkan untuk baris kedua untuk membuat data yang telah dibaca pada baris pertama menjadi tabel dan pada baris ketiga untuk menampilkan kode baris kedua. Untuk melihat hasil penerapan kode diatas bisa dilihat pada gambar berikut :

```

In [4]: test = pd.read_csv('products.csv')
        test_DF = pd.DataFrame(test)
        test_DF

```

Out[4]:

	productid	title
0	1	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...
1	2	TOSHIBA Full HD Smart LED TV 40" - 40L5650VJ - ...
2	3	Samsung 40 Inch Full HD Flat LED Digital TV 40...
3	4	Sharp HD LED TV 24" - LC-24LE175I - Hitam
4	5	Lenovo Ideapad 130-15AST LAPTOP MULTIMEDIA I A...
...
2495	109487	Flashdisk Hayabusa Toshiba 64GB/ Flash Disk /F...
2496	111362	Flashdisk Toshiba USB [8 GB]
2497	111759	Rimas COD Combo Multi Card Reader + 3 USB HUB ...
2498	112556	Disket/Diskette Floppy Disk Maxell MF2HD Forma...
2499	112852	Toshiba Canvio Basics 2TB - HDD / HD / Hardisk...

2500 rows × 2 columns

Gambar 5.2 Create DataFrame

2. Menampilkan Kolom yang Terpilih

Pandas juga bisa menampilkan kolom yang terpilih dari data tersedia.

```
1 test_DF[['title']]
2
```

Listing 5.3 Selection Column

Pada kode 5.3, menjelaskan dimana mengambil variabel diatas lalu untuk mengambil kolom apa yang ditampilkan dengan memasukkan nama kolom yang mau ditampilkan seperti kode tersebut. Untuk menampilkan hasil penerapannya bisa dilihat digambar berikut :

```
In [6]: test_DF[['title']]
```

```
Out[6]:
```

	title
0	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...
1	TOSHIBA Full HD Smart LED TV 40" - 40L5650VJ -...
2	Samsung 40 Inch Full HD Flat LED Digital TV 40...
3	Sharp HD LED TV 24" - LC-24LE175I - Hitam
4	Lenovo Ideapad 130-15AST LAPTOP MULTIMEDIA I A...
...	...
2495	Flashdisk Hayabusa Toshiba 64GB/ Flash Disk / F...
2496	Flashdisk Toshiba USB [8 GB]
2497	Rimas COD Combo Multi Card Reader + 3 USB HUB ...
2498	Disket/Diskette Floppy Disk Maxell MF2HD Forma...
2499	Toshiba Canvio Basics 2TB - HDD / HD / Hardisk...

Gambar 5.3 Selection Column

3. Menampilkan baris terpilih dengan loc

Pandas juga bisa menampilkan data hanya untuk baris tertentu berdasarkan karakter string.

```
1 test1 = pd.read_csv("products.csv", index_col="title")
2 y = test1.loc["Flashdisk Toshiba USB [8 GB]"]
3 yy = test1.loc["Flashdisk Toshiba USB [8 GB]"]
4 print(y, "\n\n", yy)
5
```

Listing 5.4 Selection Row loc

Pada kode 5.4, pada baris pertama untuk membaca data dengan parameter title untuk menampilkan datanya. Lalu pada baris kedua menggunakan perintah `.loc` dimana untuk mengambil data pada baris tersebut dengan memasukkan sesuatu yang unik dari kolom tersebut. Begitupun juga dengan baris 3 pun sama. Sedangkan pada baris ke-empat untuk menampilkan data baris 2 dan 3 dengan memanggil variabel tersebut. Untuk hasil penerapannya bisa dilihat digambar berikut :

```
In [25]: test1 = pd.read_csv("products.csv", index_col="title")
y = test1.loc["Flashdisk Toshiba USB [8 GB]"]
yy = test1.loc["Flashdisk Toshiba USB [8 GB]"]
print(y, "\n\n", yy)

productId    111362
Name: Flashdisk Toshiba USB [8 GB], dtype: int64

productId    111362
Name: Flashdisk Toshiba USB [8 GB], dtype: int64
```

Gambar 5.4 Selection Row loc

4. Menampilkan baris terpilih dengan iloc

Pandas juga bisa menampilkan baris tertentu berdasarkan karakter integer.

```
1 a = test1.iloc[3]
2 a
3
```

Listing 5.5 Selection Row iloc

Pada kode 5.5, pada baris pertama digunakan untuk mengambil nilai dari data yang ada pada baris urutan ke-3 dan pada baris kedua kode tersebut digunakan untuk menampilkan dari perintah di baris pertama. Untuk hasil penerapan bisa dilihat pada gambar berikut :

```
In [26]: a = test1.iloc[3]
a

Out[26]: productId    4
Name: Sharp HD LED TV 24" - LC-24LE175I - Hitam, dtype: int64
```

Gambar 5.5 Selection Row iloc

5. Mengisi Missing Value di DataFrame

Pandas juga mengisi *missing value* dalam sebuah data. Missing Value dalam data ditandai dengan kata NaN. Untuk mengatasi tersebut, maka perlu adanya *library* ini untuk mengatasi permasalahan tersebut.

```
1 final = pd.pivot_table(Rating_avg, values='adg_rating',
2 index='customerId', columns='productId')
3 final.head()
4
5 final_product = final.fillna(final.mean(axis=0))
```

Listing 5.6 Missing Value

Pada kode 5.6, menjelaskan bahwa pada baris kedua menampilkan data dari baris pertama yang terdapat *missing value*, lalu pada baris keempat digunakan untuk mengisi nilai *missing value* dengan nilai 0. Untuk hasil penerapan bisa dilihat pada gambar berikut :

productid	1	2	3	4	5	6	7	9	10	11
customerid										
316	-0.829457	NaN	NaN	NaN	NaN	NaN	-1.329457	NaN	-0.829457	NaN
320	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
359	1.314526	NaN	NaN	NaN	NaN	1.314526	NaN	NaN	0.314526	0.314526
370	0.705596	0.205596	NaN	NaN	NaN	1.205596	NaN	NaN	NaN	NaN
910	1.101920	0.101920	-0.39808	NaN	-0.39808	-0.398080	NaN	NaN	NaN	0.101920

Gambar 5.6 Missing Value

In [6]: final_product.head()

Out[6]:

productid	1	2	3	4	5	6	7	9	10	11
customerid										
316	-0.829457	-0.436518	-0.468109	-0.770223	-0.615331	0.320415	-1.329457	-0.690175	-0.829457	-0.094277
320	0.200220	-0.436518	-0.468109	-0.770223	-0.615331	0.320415	-0.203889	-0.690175	-0.150642	-0.094277
359	1.314526	-0.436518	-0.468109	-0.770223	-0.615331	1.314526	-0.203889	-0.690175	0.314526	0.314526
370	0.705596	0.205596	-0.468109	-0.770223	-0.615331	1.205596	-0.203889	-0.690175	-0.150642	-0.094277
910	1.101920	0.101920	-0.398080	-0.770223	-0.398080	-0.398080	-0.203889	-0.690175	-0.150642	0.101920

Gambar 5.7 Fix Missing Value

6. Menghapus nilai missing value

Pandas juga bisa menghapus nilai missing value dalam sebuah data, namun ini sangat tidak dianjurkan dalam pengolahan data. Karena bisa mempengaruhi proses yang lainnya. Pandas juga bisa menghapus nilai missing value tersebut

1 final.dropna()

2

Listing 5.7 Drop Missing Value

Pada kode diatas, digunakan untuk menghapus baris data yang memiliki nilai *missing value* dan untuk hasil penerapan bisa dilihat pada gambar berikut :

In [6]: final.dropna()

Out[6]:

productid	1	2	3	4	5	6	7	9	10	11
customerid										

Gambar 5.8 Drop Missing Value

7. Pengulangan Data dengan iterrow

Pandas juga bisa menampilkan semua data yang ada ke dalam bentuk list dengan menggunakan iterrow.

```
1         for i, j in test.iterrows():
2             print(i, j)
3             print()
4
```

Listing 5.8 Looping Iterrow

Pada kode diatas, untuk menampilkan data yang ada dalam bentuk list dan untuk hasil penerapan bisa dilihat pada gambar berikut :

```
In [30]: for i, j in test.iterrows():
          print(i, j)
          print()

1 title    TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...
Name: 1, dtype: object

2 title    TOSHIBA Full HD Smart LED TV 40" - 40L5650VJ -...
Name: 2, dtype: object

3 title    Samsung 40 Inch Full HD Flat LED Digital TV 40...
Name: 3, dtype: object

4 title    Sharp HD LED TV 24" - LC-24LE175I - Hitam
Name: 4, dtype: object
```

Gambar 5.9 Iterrow Data

8. Konversi teks

Pandas juga bisa melakukan konversi teks menjadi huruf kecil semua dan sebaliknya.

```
1     test_DF["title"] = test_DF["title"].str.lower()
2     test_DF
3
```

Listing 5.9 Convert Lowercase

Pada kode 5.9, pada baris pertama untuk mengambil data di kolom title lalu di konversi menjadi huruf kecil semua. Sedangkan pada baris kedua untuk menampilkan hasil konversi data tersebut. Untuk hasil penerapan kode diatas bisa dilihat pada gambar :

```
In [3]: test_DF["title"] = test_DF["title"].str.lower()
test_DF
```

Out[3]:

	productid	title
0	1	toshiba smart hd led tv 32" - 32l5650vj free b...
1	2	toshiba full hd smart led tv 40" - 40l5650vj -...
2	3	samsung 40 inch full hd flat led digital tv 40...
3	4	sharp hd led tv 24" - lc-24le175i - hitam
4	5	lenovo ideapad 130-15ast laptop multimedia i a...
...
2495	109487	flashdisk hayabusa toshiba 64gb/ flash disk /f...
2496	111362	flashdisk toshiba usb [8 gb]
2497	111759	rimas cod combo multi card reader + 3 usb hub ...
2498	112556	disket/diskette floppy disk maxell mf2hd forma...
2499	112852	toshiba canvio basics 2tb - hdd / hd / hardisk...

Gambar 5.10 Lowercase Data

Untuk mengkonversi data ke dalam huruf besar, kita hanya perlu merubah *lower* menjadi *upper*.

```
1 test_DF["title"] = test_DF["title"].str.upper()
2 test_DF
3
```

Listing 5.10 Convert Uppercase

Pada kode 5.10, pada baris pertama untuk mengambil data di kolom title lalu di konversi menjadi huruf besar semua. Sedangkan pada baris kedua untuk menampilkan hasil konversi data tersebut. Untuk hasil penerapan kode diatas bisa dilihat pada gambar :

```
In [4]: test_DF["title"] = test_DF["title"].str.upper()
test_DF
```

Out[4]:

	productid	title
0	1	TOSHIBA SMART HD LED TV 32" - 32L5650VJ FREE B...
1	2	TOSHIBA FULL HD SMART LED TV 40" - 40L5650VJ -...
2	3	SAMSUNG 40 INCH FULL HD FLAT LED DIGITAL TV 40...
3	4	SHARP HD LED TV 24" - LC-24LE175I - HITAM
4	5	LENOVO IDEAPAD 130-15AST LAPTOP MULTIMEDIA I A...
...
2495	109487	FLASHDISK HAYABUSA TOSHIBA 64GB/ FLASH DISK /F...
2496	111362	FLASHDISK TOSHIBA USB [8 GB]
2497	111759	RIMAS COD COMBO MULTI CARD READER + 3 USB HUB ...
2498	112556	DISKET/DISKETTE FLOPPY DISK MAXELL MF2HD FORMA...
2499	112852	TOSHIBA CANVIO BASICS 2TB - HDD / HD / HARDISK...

Gambar 5.11 Uppercase Data

9. Mengganti Nilai Data

Pandas juga bisa mengganti nilai yang ada menjadi nilai seperti yang kita inginkan.

```

1      test_DF["productId"] = test_DF["productId"].replace(1, "
      One")
2      test_DF
3

```

Listing 5.11 Replacement Data

Pada kode 5.11, pada baris pertama untuk mengganti nilai yang ada di data menjadi yang kita inginkan. Disini kita mencontohkan nilai 1 menjadi *One* dan baris kedua untuk menampilkannya. Untuk hasil penerapan bisa dilihat pada gambar berikut :

```

In [5]: test_DF["productId"] = test_DF["productId"].replace(1, "One")
        test_DF

```

Out[5]:

	productId	title
0	One	TOSHIBA SMART HD LED TV 32" - 32L5650VJ FREE B...
1	2	TOSHIBA FULL HD SMART LED TV 40" - 40L5650VJ -...
2	3	SAMSUNG 40 INCH FULL HD FLAT LED DIGITAL TV 40...
3	4	SHARP HD LED TV 24" - LC-24LE1751 - HITAM
4	5	LENOVO IDEAPAD 130-15AST LAPTOP MULTIMEDIA I A...
...
2495	109487	FLASHDISK HAYABUSA TOSHIBA 64GB/ FLASH DISK /F...
2496	111362	FLASHDISK TOSHIBA USB [8 GB]
2497	111759	RIMAS COD COMBO MULTI CARD READER + 3 USB HUB ...
2498	112556	DISKET/DISKETTE FLOPPY DISK MAXELL MF2HD FORMA...
2499	112852	TOSHIBA CANVIO BASICS 2TB - HDD / HD / HARDISK...

Gambar 5.12 Replacement Data

10. Ekstraksi Data

Pandas juga bisa melakukan ekstraksi data dimana mengekstrak data yang semula 1 kolom menjadi 2 kolom.

```

1      test_DF["productId"] = test_DF["productId"].replace(1, "
      One")
2      test_DF
3

```

Listing 5.12 Replacement Data

BAB 6

SCIKIT-LEARN

6.1 Scikit-Learn

BAB 7

MATPLOTLIB

7.1 Matplotlib

DAFTAR PUSTAKA

1. R. L. Villars, C. W. Olofson, and M. Eastwood, “Big data: What it is and why you should care,” *White Paper; IDC*, vol. 14, pp. 1–14, 2011.

