

CERDAS MENGUASAI PYTHON

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

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

CONTENTS IN BRIEF

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

Listings

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan flask 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

MENGENAL PYTHON DAN ANACONDA

1.1 D. Irga B. Naufal Fakhri D4 TI 2C

1.1.1 Sejarah Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.

Python diciptakan oleh Guido van Rossum di Scitchting Mathematisch Centrum (CWI) di Belanda pada tahun 1990-an. Bahasa python terinspirasi dari bahasa pemrograman ABC dan merupakan kelanjutan dari bahasa tersebut. Nama python sendiri bukan berasal dari nama ular python namun karena Guido adalah penggemar grup komedi Inggris bernama Monty Python. Guido masih menjadi penulis utama untuk python, walaupun python bersifat open source sehingga ribuan orang juga berkontribusi dalam mengembangkan python.

Di tahun 1995, Guido melanjutkan pembuatan python di Corporation for National Research Initiative (CNRI) di Virginia Amerika, dimana dia merilis beberapa versi

dari python. Pada Mei 2000, Guido dan tim Python pindah ke BeOpen.com dan membentuk tim BeOpen PythonLabs. Di bulan Oktober pada tahun yang sama, tim python pindah ke Digital Creation (sekarang menjadi Perusahaan Zope). Pada tahun 2001, dibentuklah Organisasi Python yaitu Python Software Foundation (PSF). PSF merupakan organisasi nirlaba yang dibuat khusus untuk semua hal yang berkaitan dengan hak intelektual Python. Perusahaan Zope menjadi anggota sponsor dari PSF.

1.1.2 Tanggal Rilis Python

Semua versi python yang dirilis bersifat open source. Dalam sejarahnya, hampir semua rilis python menggunakan lisensi GFL-compatible. Berikut adalah versi major dan minor python berikut tanggal rilisnya.

- Python 1.0 – Januari 1994
- Python 1.2 – 10 April 1995
- Python 1.3 – 12 Oktober 1995
- Python 1.4 – 25 Oktober 1996
- Python 1.5 – 31 Desember 1997
- Python 1.6 – 5 September 2000
- Python 2.0 – 16 Oktober 2000
- Python 2.1 – 17 April 2001
- Python 2.2 – 21 Desember 2001
- Python 2.3 – 29 Juli 2003
- Python 2.4 – 30 Nopember 2004
- Python 2.5 – 19 September 2006
- Python 2.6 – 1 Oktober 2008
- Python 2.7 – 3 Juli 2010
- Python 3.0 – 3 Desember 2008
- Python 3.1 – 27 Juni 2009
- Python 3.2 – 20 Februari 2011
- Python 3.3 – 29 September 2012
- Python 3.4 – 16 Maret 2014
- Python 3.5 – 13 September 2015
- Python 3.6 – 23 Desember 2016

1.1.3 Perbedaan Python 2 dengan Python 3

Pada Python 2 dan Python 3 memiliki kesamaan kapabilitas namun cara penggunaannya berbeda

- Print

Pada python2, print lebih seperti statement daripada fungsi

```
1 print "Saya Belajar Python"
```

sedangkan pada python3, print digunakan sebagai fungsi

```
1 print("Saya Belajar Python")
```

- Pembagian pada Integer

Pada Python 2, semua tipe data angka yang tidak mengandung desimal akan diperlakukan sebagai integer. Terlihat mudah pada awalnya, ketika mencoba untuk membagi kedua integer akan didapatkan tipe data float.

```
1 3 / 2 = 1.5
```

Python 2 menggunakan floor division atau dibulatkan ke nilai paling rendah misalnya 1.5 jadi 1, 2.6 jadi 2 dan seterusnya. Pada Python 2.7 akan menjadi seperti ini:

```
1 3
2 4
3 x = 3 / 2
4 print a
5 #Output
6 1
```

Untuk desimal maka tambahkan .0 setelah bilangan dan menjadi seperti ini 3.0 / 2.0 untuk mendapatkan hasil 1.5 Pada Python 3, pembagian pada bilangan integer lebih intuitif:

```
1 a = 3 / 2
2 print(a)
3 #Output
4 1.5
```

Kita juga masih bisa melakukan 3.0 / 2.0 untuk mendapatkan 1.5 namun untuk mendapatkan floor division maka pada Python 3 gunakan //:

```
1 b = 3 // 2
2 print(b)
3 #Output
4 1
```

Fitur pada Python 3 ini tidak bisa digunakan pada Python 2.7

- Dukungan Unicode

Ketika bahasa pemrograman menangani tipe data string (yang mana merupakan sekumpulan karakter), mereka bisa melakukan beberapa cara berbeda sehingga komputer dapat mengubah angka ke huruf dan simbol lain. Python 2 menggunakan alfabet ASCII secara default, sehingga ketika kita mengetik "Halo!" maka Python 2 menangani string sebagai ASCII. Terbatas pada beberapa ratus karakter, ASCII mungkin bukan pilihan yang fleksibel untuk menangani proses encoding terutama yang non English.

Untuk menggunakan unicode yang lebih luwes, mendukung lebih dari 128,000 karakter maka kita harus mengetik u"Halo!" , dengan tambahan u di depannya yang mana berarti Unicode.

Python 3 menggunakan Unicode secara default, yang mana menyelamatkan programmer dari tambahan kode lagi, lebih hemat waktu dan mudah untuk diisikan dan ditampilkan. Karena Unicode mendukung berbagai karakter linguistik yang beragam termasuk menampilkan emoji, penggunaan karakter secara default dengan encoding memastikan perangkat mobile didukung oleh program yang kita buat.

Jika kita ingin kode Python 3 kita mendukung Python 2, tambahkan u di depan string.

1.1.4 Penggunaan Python di perusahaan dunia

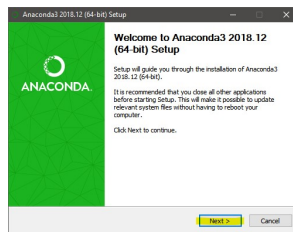
1. Google adalah perusahaan besar yang menggunakan banyak kode Python di dalam mesin pencarinya. Dan mesin pencari google adalah yang paling terkenal di dunia.
2. Youtube, situs video terbesar dan terpopuler di dunia, sebagian besar kodenya ditulis dalam bahasa Python.
3. Facebook, media sosial terbesar di dunia, menggunakan Tornado, sebuah framework Python untuk menampilkan timeline.
4. Instagram, siapa yang tidak kenal. Instagram menggunakan Django, framework python sebagai mesin pengolah sisi server dari aplikasinya.
5. Pinterest, banyak menggunakan python untuk membangun aplikasinya.
6. Dropbox, barangkali Anda adalah salah seorang pengguna layanan ini. Dropbox menggunakan python baik di sisi server maupun di sisi pengguna layanannya.
7. Quora, salah satu situs tanya jawab terbesar di dunia, dibangun menggunakan Python.
8. NASA, badan antariksa Amerika ini menggunakan Python untuk bidang sainsnya.
9. NSA, badan mata – mata Amerika banyak menggunakan Python untuk analisa kriptografi dan intelijen

10. Blender, Maya, software pembuat animasi 3D terkenal, menggunakan Python sebagai salah satu bahasa skrip pemrogramannya.
11. Raspberry Pi, komputer mini yang banyak digunakan sebagai mikrokontroller, menggunakan Python sebagai bahasa utamanya.
12. ESRI, produsen terkenal pembuat software pemetaan GIS banyak menggunakan Python di produknya.

Untuk lebih lengkapnya bisa mengunjungi www.python.org/about/success/

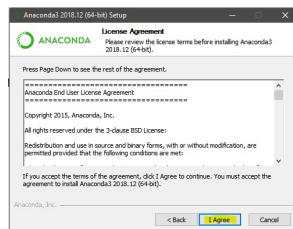
1.1.5 Cara menginstall Anaconda

1. Pastikan anda telah menginstall python dan anda mengetahui versi dari python yang telah anda install
2. Download Anaconda dari website www.anaconda.com/distribution
3. pilih sesuai dengan versi python anda, jika versi anda python3 maka pilih python3
4. Setelah itu buka file yang telah anda download
5. Setelah muncul gambar dibawah ini, tekan next



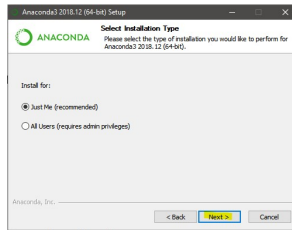
Gambar 1.1 Tampilan Instalasi 1

6. Baca license agreement lalu tekan 'I Agree'



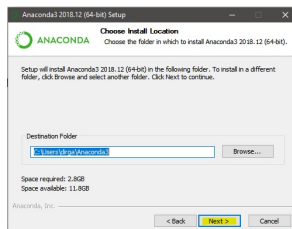
Gambar 1.2 Tampilan Instalasi 2

7. Setelah itu pilih mau diinstall pada user yang sedang anda pakai atau kesemua user, direkomendasikan untuk memilih just me yaitu hanya user yang sedang dipakai saja



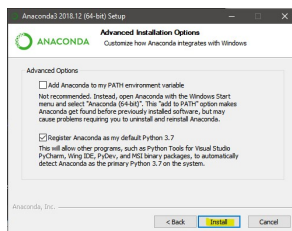
Gambar 1.3 Tampilan Instalasi 3

8. Catat tempat dimana anda akan menginstall anaconda, lalu tekan 'Next'



Gambar 1.4 Tampilan Instalasi 4

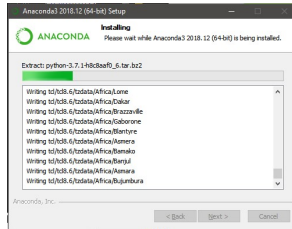
9. Setelah itu anda diberi pilihan, direkomendasikan untuk tidak mengubah pilihan tersebut, lalu tekan 'Install'



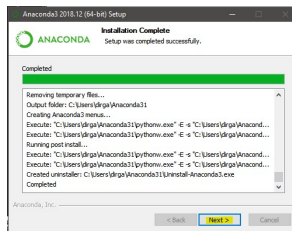
Gambar 1.5 Tampilan Instalasi 5

10. Tunggu sampai instalasi selesai

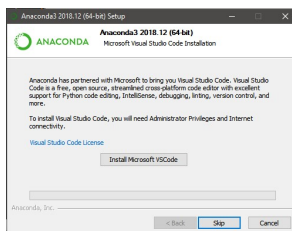
11. Setelah selesai tekan 'Next'



Gambar 1.6 Tampilan Instalasi 6



Gambar 1.7 Tampilan Instalasi 7



Gambar 1.8 Tampilan Instalasi 8

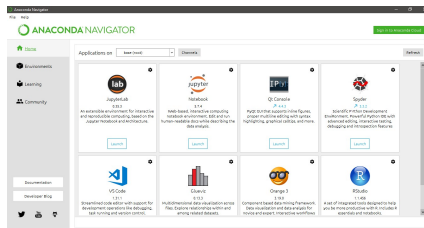
12. Setelah itu ada opsi untuk memilih untuk meinstall visual studio code, jika anda berminat klik 'Install VSCode' jika tidak tekan 'Skip'
13. Tekan 'Finish' untuk menyelesaikan instalasi



Gambar 1.9 Tampilan Instalasi 9

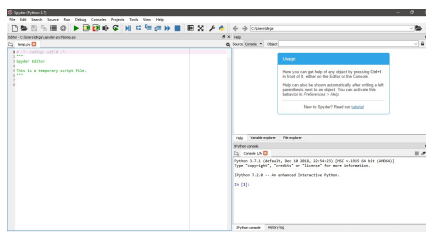
1.1.6 Cara menggunakan Spyder pada Anaconda

Pertama buka aplikasi Anaconda sampai muncul seperti ini

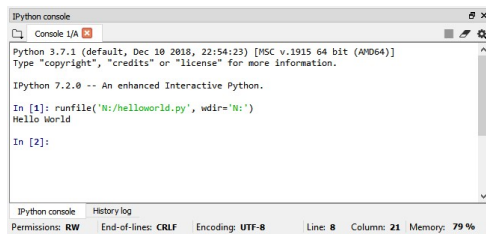


Gambar 1.10 Tampilan awal Anaconda

Setelah itu tekan Launch dibawah logo Spyder Tunggu sampai muncul seperti ini



Gambar 1.11 Tampilan spider



The image shows a screenshot of the IPython console window. The window title is "IPython console". The console output shows the following:

```
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 7.2.0 -- An enhanced Interactive Python.

In [1]: runfile('/N:/helloworld.py', wdir='N:')
Hello World

In [2]:
```

The bottom status bar of the window displays the following information:

IPython console	History log
Permissions: RW	End-of-lines: CRLF
Encoding: UTF-8	Line: 8
	Column: 21
	Memory: 79 %

Gambar 1.15 Hasil Program

BAB 2

PEMOGRAMAN DASAR

2.1 D. Irga B. Naufal Fakhri

2.1.1 Teori

1. Jenis-jenis variabel pada python dan cara penggunaannya:

(a) Boolean

```
1 #Boolean
2 varbool = True
3 print(varbool)
```

Listing 2.1 Contoh kode variable Boolean.

(b) String

```
1 #String
2 varstr = "D. Irga B. Naufal Fakhri"
3 print(varstr)
```

Listing 2.2 Contoh kode variable String.

(c) Integer


```
1 #Integer
2 varint = 1174066
3 print(varint)
```

Listing 2.3 Contoh kode variable Integer.

(d) Float

```
1 #Float
2 varflt = 1.1
3 print(varflt)
```

Listing 2.4 Contoh kode variable Float.

(e) Hexadecimal

```
1 #Hexadecimal
2 varhex = 0x1
3 print(varhex)
```

Listing 2.5 Contoh kode variable Hexadecimal.

(f) Complex

```
1 #Complex
2 varcmp = 7j
3 print(varcmp)
```

Listing 2.6 Contoh kode variable Complex.

(g) List

```
1 #List
2 varli = [1,2,3]
3 print(varli)
4 print(varli[2])
```

Listing 2.7 Contoh kode variable List.

(h) Tuple

```
1 #Tuple
2 vartu = (1,2,3)
3 print(vartu)
4 print(vartu[2])
```

Listing 2.8 Contoh kode variable Tuple.

(i) Set

```
1 #Set
2 varset = {1,2,3}
3 print(varset)
```

Listing 2.9 Contoh kode variable Set.

(j) Dictionary

```

1 #Dictionary
2 vardic = {1:'satu', 2:'dua', 'tiga':3}
3 print(vardic)
4 print(vardic[1])

```

Listing 2.10 Contoh kode variable Dictionary.

2. Permintaan Input dari user dan Outputnya

```

1 #Input dan Output
2 nama = input('Tolong masukan nama anda: ')
3 print('Hallo, '+nama)

```

Listing 2.11 Contoh kode input dan outputnya.

3. Operator dasar aritmatika dan perubahan tipe data variable

Operator dasar aritmatika

- (a) Perjumlahan (+) Operator ini berfungsi untuk melakukan operasi perjumlahan.

```

1 #Input dan Output
2 nama = input('Tolong masukan nama anda: ')
3 print('Hallo, '+nama)
4
5 #Operator Dasar Aritmatika
6 #Perjumlahan
7 var1 = 1
8 var2 = 2
9 hasil = var1 + var2
10 print(hasil)

```

Listing 2.12 Contoh kode operasi pertambahan.

- (b) Pengurangan (-) Operator ini berfungsi untuk melakukan operasi pengurangan.

```

1 #Pengurangan
2 var1 = 4
3 var2 = 3
4 hasil = var1 - var2
5 print(hasil)

```

Listing 2.13 Contoh kode operasi pengurangan.

- (c) Perkalian (*) Operator ini dipergunakan untuk melakukan operasi perkalian.

```

1 #Perkalian
2 var1 = 5
3 var2 = 6
4 hasil = var1 * var2
5 print(hasil)

```

Listing 2.14 Contoh kode operasi perkalian.

- (d) Pembagian (/) Operator ini dipergunakan untuk melakukan operasi pembagian.

```
1 #Pembagian
2 var1 = 8
3 var2 = 2
4 hasil = var1 / var2
5 print(hasil)
```

Listing 2.15 Contoh kode operasi pembagian.

- (e) Modulus (Operator ini dipergunakan untuk melakukan operasi modulus.

```
1 #Modulus
2 var1 = 7
3 var2 = 3
4 hasil = var1 % var2
5 print(hasil)
```

Listing 2.16 Contoh kode operasi modulus.

- (f) Perpangkatan (**) Operator ini dipergunakan untuk melakukan operasi perpangkatan.

```
1 #Perpangkatan
2 var1 = 10
3 var2 = 2
4 hasil = var1 ** var2
5 print(hasil)
```

Listing 2.17 Contoh kode operasi perpangkatan.

- (g) Pembulatan Kebawah Pada Hasil Pembagian (//) Operator ini dipergunakan untuk melakukan operasi pembulatan hasil bagi.

```
1 #Pembulatan Kebawah Pada Hasil Pembagian
2 var1 = 11
3 var2 = 3
4 hasil = var1 // var2
5 print(hasil)
```

Listing 2.18 Contoh kode operasi pembulatan hasil pembagian kebawah.

Perubahan tipe data variable

- (a) String menjadi Integer

```
1 #String menjadi Integer
2 varstr = '1'
3 varint = int(varstr)
4 print(varint)
```

Listing 2.19 Contoh kode variable string menjadi integer.

- (b) Integer menjadi String

```

1 #Integer menjadi String
2 varint = 2
3 varstr = str(varint)
4 print(varstr)

```

Listing 2.20 Contoh kode variable integer menjadi string.

4. Sintak perulangan (looping), jenis-jenisnya, dan penggunaannya.

- (a) While Loop While Loop adalah perulangan yang mengeksekusi statement terus menerus selama kondisi bernilai True.

```

1 #While Loop
2 hitung = 1
3 while (hitung < 6):
4     print (hitung)
5     hitung += 1

```

Listing 2.21 Contoh kode penggunaan while loop.

- (b) For Loop For Loop adalah pengulangan berdasarkan kondisi yang telah ditentukan biasanya kondisi pertambahan seperti 1 sampai 5

```

1 #For Loop
2 angka = [1,2,3,4,5]
3 for a in angka:
4     print(a)

```

Listing 2.22 Contoh kode penggunaan for loop.

- (c) Nested Loop Nested Loop merupakan pengulangan yang ada di dalam pengulangan

```

1 #Nested Loop
2 i = 1
3 while (i < 5):
4     j = 1
5     while (j <= 5):
6         print('TEST')
7         j += 1
8     i += 1

```

Listing 2.23 Contoh kode penggunaan nested loop.

5. Sintak kondisi dan penggunaannya.

- (a) If Kondisi ini digunakan untuk mengecek apabila kondisi tersebut dipenuhi akan mengeksekusi kode didalamnya.

```

1 #Kondisi
2 #If
3 var = 1
4 if var > 0:
5     print(var, "adalah angka lebih dari 0")

```

Listing 2.24 Contoh kode penggunaan if.

- (b) If Else Kondisi ini digunakan untuk mengecek apabila kondisi tersebut dipenuhi akan mengeksekusi kode didalamnya dan didalamnya memiliki dua kondisi.

```

1 #If Else
2 var = -1
3 if var > 0:
4     print(var, "adalah angka kurang dari 0")
5 else:
6     print(var, "adalah angka lebih dari 0")

```

Listing 2.25 Contoh kode penggunaan if else.

- (c) Elif Kondisi ini digunakan untuk mengecek apabila kondisi tersebut dipenuhi akan mengeksekusi kode didalamnya dan didalamnya memiliki dua kondisi atau lebih.

```

1 #Elif
2 var = 0
3 if var > 0:
4     print(var, "adalah angka lebih dari 0")
5 elif var < 0:
6     print(var, "adalah angka kurang dari 0")
7 else:
8     print(var, "adalah angka yang sama dengan 0")

```

Listing 2.26 Contoh kode penggunaan elif.

- (d) Kondisi di dalam kondisi Kondisi ini digunakan saat kondisi memerlukan kondisi lagi didalamnya

```

1 #Kondisi di dalam kondisi
2 if var > 0:
3     print(var, "adalah angka lebih dari 0")
4     if var > 10:
5         print(var, "adalah angka lebih dari 10.")
6     else:
7         print(var, "adalah angka lebih dari 10.")
8 elif var < 0:
9     print(var, "adalah angka kurang dari 0")
10 else:
11     print(var, "adalah angka yang sama dengan 0")

```

Listing 2.27 Contoh kode penggunaan kondisi di dalam kondisi.

6. Jenis-jenis error pada python dan cara mengatasinya.

- **Syntax Errors** Syntax Errors adalah kesalahan pada penulisan syntax atau kode. Solusinya adalah memperbaiki penulisan syntax atau kode
- **Zero Division Error** ZeroDivisonError adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0). Solusinya adalah tidak membagi suatu yang hasilnya nol.

- Name Error NameError adalah exception saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi atau tidak ada. Solusinya adalah memastikan variabel atau function yang akan dipanggil ada didalam program atau tidak salah mengetikannya.
- Type Error TypeError adalah exception saat melakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkonversi variabelnya sesuai dengan tipe data sesuai dengan yang akan digunakan.

7. Cara pemakaian Try Except.

```
1 #Try Except
2 try :
3     print(tidakada)
4 except :
5     print("Terjadi kesalahan penulisan kode")
6 finally :
7     print("Try except telah selesai")
```

Listing 2.28 Contoh kode penggunaan try except.

BAB 3

PRAKTEK PEMOGRAMAN DASAR

3.1 D. Irga B. Naufal Fakhri

3.1.1 Ketrampilan Pemrograman

1. Jawaban Soal 1

```
1 #Jawaban No. 1
2
3 print(1174066%3)
4 #mod3 = 1
5
6 print(" ## ## #####      ### #####      #####      #####")
7 print("#### #####      ##### ### ## ## ##      ###")
8 print(" ### ##      ## ## ## ## ## ## ##      ###")
9 print(" ### ##      ## ##### ## ## #####      #####")
10 print(" ### ##      ##      ## ## ## ## ## ##      ##")
11 print(" ### ##      ##      #####      #####      #####")
12 #Cara kedua Penulisan NPM tanda pagar menurut https://stackoverflow.com/questions/9632995/how-to-easily-print-ascii-art-text
13 text = "1174066"
14 from PIL import Image, ImageDraw, ImageFont
```



```

15 import numpy as np
16 myfont = ImageFont.truetype("verdanab.ttf", 12)
17 size = myfont.getsize(text)
18 img = Image.new("1", size, "black")
19 draw = ImageDraw.Draw(img)
20 draw.text((0, 0), text, "white", font=myfont)
21 pixels = np.array(img, dtype=np.uint8)
22 chars = np.array([' ', '#'], dtype="UI")[pixels]
23 strings = chars.view('U' + str(chars.shape[1])).flatten()
24 print( "\n".join(strings))

```

2. Jawaban Soal 2

```

1 #Jawaban No. 2
2
3 npm = input("Input : ")
4 i = 1
5 while(i <= int(npm[5:])):
6     print("Halo, "+str(npm)+" apa kabar?")
7     i += 1

```

3. Jawaban Soal 3

```

1 #Jawaban No. 3
2
3 npm = input("Input : ")
4 i = 1
5 akhir = int(npm[4:5]) + int(npm[5:6]) + int(npm[6:7])
6 while(i <= akhir):
7     print("Hallo, "+str(npm[4:7])+" apa kabar?")
8     i += 1

```

4. Jawaban Soal 4

```

1 #Jawaban No. 4
2
3 npm = input("Input : ")
4 print("Halo, "+str(npm[-3])+" apa kabar?")

```

5. Jawaban Soal 5

```

1 #Jawaban No. 5
2
3 a = 1
4 b = 1
5 c = 7
6 d = 4
7 e = 0
8 f = 6
9 g = 6
10
11 npm = [a,b,c,d,e,f,g]
12
13 for n in npm:
14     print(n, end = "")
15 print()

```

6. Jawaban Soal 6

```
1 #Jawaban No. 6
2
3 print(a+b+c+d+e+f+g)
```

7. Jawaban Soal 7

```
1 #Jawaban No. 7
2
3 print(a*b*c*d*e*f*g)
```

8. Jawaban Soal 8

```
1 #Jawaban No. 8
2
3 for n in npm:
4     print(n)
```

9. Jawaban Soal 9

```
1 #Jawaban No. 9
2
3 for n in npm:
4     if(n % 2 == 0):
5         if(n != 0):
6             print(n, end = "")
7
8 print()
```

10. Jawaban Soal 10

```
1 #Jawaban No. 10
2
3 for n in npm:
4     if(n % 2 != 0):
5         print(n, end = "")
6
7 print()
```

11. Jawaban Soal 11

```
1 #Jawaban No. 11
2
3 for n in npm:
4     for x in range(2,n):
5         if n%x != 0:
6
7             break
8     print(n, "Bilangan prima")
9     break
```

3.1.2 Ketrampilan Penanganan Error

1. Jawaban Soal No. 1

- **Syntax Errors** Syntax Errors adalah kesalahan pada penulisan syntax atau kode. Solusinya adalah memperbaiki penulisan syntax atau kode
- **Zero Division Error** ZeroDivisonError adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0). Solusinya adalah tidak membagi suatu yang hasilnya nol.
- **Name Error** NameError adalah exception saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi atau tidak ada. Solusinya adalah memastikan variabel atau function yang akan dipanggil ada didalam program atau tidak salah mengetikannya.
- **Type Error** TypeError adalah exception saat melakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkoversi varibelnya sesuai dengan tipe data sesuai dengan yang akan digunakan.

2. Jawaban Soal No. 2

```
1 a = "1"
2 b = 1
3
4 try :
5     print(a+b)
6 except TypeError:
7     print("Berbeda type datanya sehingga tidak bisa dijumlahkan")
```

BAB 4

FUNGSI DAN KELAS

4.1 D. Irga B. Naufal Fakhri

4.1.1 Pemahaman Teori

1. Fungsi

Fungsi adalah blok blok kode yang terorganisir yang dapat digunakan kembali didalam program yang digunakan untuk melakukan suatu perintah yang telah diberikan. untuk membuat fungsi kita harus menggunakan def kemudian nama fungsinya dan (variable)nya diakhiri oleh tanda :

```
1 #Contoh fungsi inputan
2 def printteks( str ):
3     print ( str )
4     return
5
6 printteks("Ini adalah Contoh fungsi")
```

Listing 4.1 Contoh kode fungsi inputan ke fungsi.

Fungsi juga berguna untuk melemparkan variable contohnya

```

1 #Contoh fungsi outputan
2 def output( str ):
3     full = "Ini adalah contoh dari " + str
4     return full
5
6 print( output("fungsi outputan"))

```

Listing 4.2 Contoh kode fungsi outputan ke fungsi.

2. Paket(Package) atau Library

Paket atau yang biasa disebut dengan library adalah kumpulan kode-kode fungsi atau method pada python yang dapat dipanggil kedalam program python yang kita buat. Package berada di file terpisah dari main program cara memanggil package: Pastikan file package ada didalam folder yang sama lalu ditambah import dengan nama filenya tanpa extensi (.py)

```

1 #Contoh import library atau package
2 import fungsi_1174066
3
4 input( fungsi_1174066.hello("Dirga"))

```

Listing 4.3 Contoh import package atau library.

3. Kelas (Class), Objek (Object), Atribut (Attribute), dan Method

Kelas(Class) adalah sebuah blueprint(cetakan) dari sebuah objek. Objek(Object) adalah hasil cetakan dari sebuah kelas(class). Atribut(Attribute) adalah nilai data yang ada didalam sebuah object. Method adalah sesuatu yang bisa dilakukan oleh object.

```

1 #Contoh Class
2 class Mahasiswa:
3     def __init__( self , nama ,npm, kelas ) :
4         self.nama = nama
5         self.npm = npm
6         self.kelas = kelas
7
8     def datadiri( self ):
9         print ( "Nama: ", self.nama, ", NPM: ", self.npm, ",
10             Kelas: ", self.kelas )
11
12 #Pembuatan Object
13 mhs1 = Mahasiswa("D. Irga",11174066,"D4TI2C")
14 mhs1.datadiri() ;

```

Listing 4.4 Contoh import package atau library.

4. Cara memanggil library dari instansiasi

Cara memanggilnya:

- Pertama kita import filenya
- kemudian buat variabelnya jika menggunakan variable untuk menampung data

- Kemudian panggil nama classnya(file) dan panggil fungsinya
- Kemudian menggunakan perintah print untuk menampilkan data

```

1 #Contoh package nomor 4
2 def Tambah(x,y):
3     z = x + y
4     return z

```

Listing 4.5 Contoh package atau library.

```

1 #Contoh cara memanggil library atau package
2 import fungsi_1174066
3
4 x = 3
5 y = 7
6 print(fungsi_1174066.Tambah(x,y))

```

Listing 4.6 Contoh import package atau library.

5. Contoh pemakaian paket dengan perintah from kalkulator import Penambahan
Pemakaian package(paket) dengan perintah from namafilenya import berfungsi untuk memanggil fungsi dari nama filenya

```

1 #Contoh cara memanggil library atau package dari from import
2 from fungsi_1174066 import Tambah
3
4 x = 3
5 y = 7
6 print(Tambah(x,y))

```

Listing 4.7 Contoh import package atau library.

6. Jelaskan dengan contoh kode, pemakaian paket fungsi didalam folder
Jika file paket ada didalam folder maka kita harus menambahkan lokasi filenya ada didalam folder apa dengan cara menggunakan namafolder.namafile

```

1 #Contoh cara memanggil library atau package dari folder
2 from folder_1174066.fungsi_1174066 import Tambah
3
4 x = 3
5 y = 7
6 print(Tambah(x,y))

```

Listing 4.8 Contoh import package atau library didalam folder.

7. Jelaskan dengan contoh kode, pemakaian paket fungsi didalam folder
Jika file paket ada didalam folder maka kita harus menambahkan lokasi filenya ada didalam folder apa dengan cara menggunakan namafolder.namafile

```

1 #Contoh cara memanggil library atau package dari folder
2 from folder_1174066.fungsi_1174066 import Tambah
3
4 x = 3

```

```
5 y = 7  
6 print(Tambah(x,y))
```

Listing 4.9 Contoh import package atau library didalam folder.

PRAKTEK FUNGSI DAN KELAS

5.1 D. Irga B. Naufal Fakhri

5.1.1 Keterampilan Pemograman

1. Jawaban nomor 1

```
1 #Jawaban nomor 1
2 def npm(npm):
3     text = str(npm)
4     from PIL import Image, ImageDraw, ImageFont
5     import numpy as np
6     myfont = ImageFont.truetype("verdanab.ttf", 12)
7     size = myfont.getsize(text)
8     img = Image.new("1", size, "black")
9     draw = ImageDraw.Draw(img)
10    draw.text((0, 0), text, "white", font=myfont)
11    pixels = np.array(img, dtype=np.uint8)
12    if int(text) % 3 == 0:
13        chars = np.array([' ', '*'], dtype="U1")[pixels]
14    elif int(text) % 3 == 1:
15        chars = np.array([' ', '#'], dtype="U1")[pixels]
16    elif int(text) % 3 == 2:
```



```

17     chars = np.array([' ', '+'], dtype="U1")[pixels]
18     strings = chars.view('U' + str(chars.shape[1])).flatten()
19     print( "\n".join(strings))

```

2. Jawaban nomor 2

```

1 #Jawaban nomor 2
2 def loop(npm):
3     i = 1
4     while (i <= int(npm[5:])):
5         print("Hallo , "+str(npm)+" apa kabar?")
6         i += 1
7
8 loop(input("Masukan NPM : "))

```

3. Jawaban nomor 3

```

1 #Jawaban nomor 3
2 def looptambah(npm):
3     i = 1
4     akhir = int(npm[4:5]) + int(npm[5:6]) + int(npm[6:7])
5     while (i <= akhir):
6         print("Hallo , "+str(npm[4:7])+" apa kabar?")
7         i += 1
8
9 looptambah(input("Input: "))

```

4. Jawaban nomor 4

```

1 #Jawaban nomor 4
2 def hallo(npm):
3     print("Halo , "+str(npm[-3])+" apa kabar?")
4
5 hallo(input("Input: "))

```

5. Jawaban nomor 5

```

1 #Jawaban nomor 5
2 def array(npm):
3     for x in npm:
4         print (x)

```

6. Jawaban nomor 6

```

1 #Jawaban nomor 6
2 def jumlah(npm):
3     npm = list(map(int , npm))
4     hasil = 0
5     for x in npm:
6         hasil += x
7     print("Hasil perjumlahan dari npm anda adalah: " + str(hasil)
8         )
9 jumlah(input("NPM: "))

```

7. Jawaban nomor 7

```
1 #Jawaban nomor 7
2 def kali(npm):
3     npm = list(map(int, npm))
4     hasil = 0
5     for x in npm:
6         hasil *= x
7     print("Hasil perkalian dari npm anda adalah: " + str(hasil))
8
9 kali(input("NPM: "))
```

8. Jawaban nomor 8

```
1 #Jawaban nomor 8
2 def printgenap(npm):
3     npm = list(map(int, npm))
4     for x in npm:
5         if(x % 2 == 0):
6             if(x != 0):
7                 print(x, end = "")
8
9 printgenap(input("NPM: "))
```

9. Jawaban nomor 9

```
1 #Jawaban nomor 9
2 def printganjil(npm):
3     npm = list(map(int, npm))
4     for x in npm:
5         if(x % 2 != 0):
6             print(x, end = "")
7
8 printganjil(input("NPM: "))
```

10. Jawaban nomor 10

```
1 #Jawaban nomor 10
2 def printprima(npm):
3     npm = list(map(int, npm))
4     prima = []
5     for n in npm:
6         bilPrima = True
7         if n == 0 or n == 1:
8             bilPrima = False
9         for x in range(2, n):
10             if n % x == 0:
11                 bilPrima = False
12         if bilPrima:
13             prima.append(n)
14
15     for p in prima:
16         print(p, end = "")
17
18 printprima(input("NPM: "))
```

11. Jawaban nomor 11

```
1 #Pemanggilan library 3lib
2 from lib_1174066 import *
3
4 npm(1174066)
5 loop(input("Masukan NPM : "))
6 looptambah(input("Input: "))
7 hallo(input("Input: "))
8 array(input("NPM: "))
9 jumlah(input("NPM: "))
10 kali(input("NPM: "))
11 printgenap(input("NPM: "))
12 printganjil(input("NPM: "))
13 printprima(input("NPM: "))
```

12. Jawaban nomor 12

```
1 #Pemanggilan kelas 3lib
2 from kelas3lib_1174066 import kelas3lib
3 npm = "1174066"
4 kelas = kelas3lib(npm)
5
6 kelas.npmprint()
7 kelas.loop()
8 kelas.looptambah()
9 kelas.hallo()
10 kelas.array()
11 kelas.jumlah()
12 kelas.kali()
13 kelas.printgenap()
14 kelas.printganjil()
15 kelas.printprima()
```

5.1.2 Ketrampilan Penanganan Error

▪ Syntax Errors

Syntax Errors adalah kesalahan pada penulisan syntax atau kode. Solusinya adalah memperbaiki penulisan syntax atau kode

▪ Zero Division Error

ZeroDivisonError adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0). Solusinya adalah tidak membagi suatu yang hasilnya nol.

▪ Name Error

NameError adalah exception saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi atau tidak ada. Solusinya adalah memasukan variabel atau function yang akan dipanggil ada didalam program atau tidak salah mengetikannya.

- Type Error

TypeError adalah exception saat melakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkonversi variabelnya sesuai dengan tipe data sesuai dengan yang akan digunakan.

```
1 #Pemanggilan kelas 3lib
2 from kelas3lib_1174066 import kelas3lib
3 npm = "1174066"
4 kelas = kelas3lib(npm)
5
6 kelas.npmprint()
7 kelas.loop()
8 kelas.looptambah()
9 kelas.hallo()
10 kelas.array()
11 kelas.jumlah()
12 kelas.kali()
13 kelas.printgenap()
14 kelas.printganjil()
15 kelas.printprima()
```


BAB 6

LIBRARY CSV DAN PANDAS

6.1 D. Irga B. Naufal Fakhri

6.1.1 Pemahaman Teori

1. CSV

CSV (Comma Separated Values file) adalah sebuah tipe file text biasa yang memiliki penataan khusus yang biasanya berfungsi untuk mengelola data. sesuai dengan namanya file csv memisahkan setiap data menggunakan koma (.).

Format data CSV pertama kali digunakan pada tahun 1978 pada complier FORTRAN 77, kemudian nama CSV baru muncul dan mulai digunakan pada tahun 1983

Contoh data pada csv:

```
1 npm,nama,kelas ,tanggalahir
2 11714066,D. Irga B.Naufal Fakhri ,D4 Teknik Informatika 2C
   ,11/07/98
```

2. Aplikasi yang bisa menciptakan CSV

Semua aplikasi teks editor seperti notepad++, vscode, sublime ataupun notepad dapat menciptakan CSV termasuk aplikasi spreadsheet seperti Microsoft Excel, Libre Office

3. Jelaskan bagaimana cara menulis dan membaca file csv di Excel atau spreadsheet

- Buka Microsoft Excel 2019-nya lalu buat dokumen baru
- Isikan data sesuai dengan kebutuhan, yang paling atas akan menjadi header dari file csv
- Setelah memasukkan data, klik file lalu klik Save As
- Pilih Browse dan pilih tempat menyimpannya akan dimana
- Masukkan nama file pada File Name
- Lalu pada Save As Type pilih CSV (comma delimited) (*.csv)
- Maka hasil file akan seperti ini

```
1 npm,nama,kelas,tanggallahir
2 11714066,D. Irga B.Naufal Fakhri,D4 Teknik Informatika 2C
   ,11/07/98
```

4. Jelaskan sejarah library csv

Module csv mengimplementasikan kelas untuk membaca dan menulis data kedalam format CSV. Hal ini memungkinkan programmer untuk "tuliskan data ini dalam format yang disukai oleh Excel," atau "baca data dari file yang dihasilkan oleh Excel," tanpa mengetahui detail yang tepat dari format CSV yang digunakan oleh Excel. Pemrogram juga dapat menggambarkan format CSV yang dipahami oleh aplikasi lain atau menentukan format CSV tujuan khusus untuk mereka sendiri.

5. Jelaskan sejarah library pandas

pandas adalah sebuah library open source dan berlisensi BSD yang menyediakan performa yang tinggi, mudah digunakan struktur data dan data analisis untuk python.

6. Jelaskan fungsi-fungsi yang terdapat di library csv

- csv.reader

Berfungsi untuk membaca dan mengembalikan data kedalam variable dari file csv. Fungsi reader dirancang untuk mengambil data pada setiap baris didalam file dan membuat daftar semua kolom. Kemudian, tinggal dipilih kolom mana yang diinginkan untuk data variabel.

```
1 def read(namafile):
2     with open(namafile) as csv_file:
3         csv_reader = csv.reader(csv_file, delimiter=',')
4         line_count = 0
5         for row in csv_reader:
```

```

6         if line_count == 0:
7             print(f'Column names are {", ".join(row)}')
8             line_count += 1
9         else:
10            print(f'\tNomor NPM: {row[0]} Nama: {row[1]}
11            Kelas: {row[2]}.' )
12            line_count += 1
13            print(f'Processed {line_count} lines.')
14 read('1174066.csv')

```

▪ csv.writer

Berfungsi untuk menuliskan data dari variable kedalam file csv. Fungsi writer akan membuat objek yang cocok untuk menulis. Untuk mengulang data yang ada di atas baris, gunakan fungsi writerow.

```

1 def write(namafile):
2     with open(namafile, mode='w') as employee_file:
3         employee_writer = csv.writer(employee_file, delimiter=
4             ',', quotechar='\"', quoting=csv.QUOTE_MINIMAL)
5
6         employee_writer.writerow(['npm', 'nama', 'kelas'])
7         employee_writer.writerow(['1174066', 'D. Irga 3', 'D4
8             Teknik Informatika 2C'])
9
10    write('1174066_baru.csv')

```

▪ csv.register_dialect

Mendaftarkan dialect pada csv

▪ csv.unregister_dialect

Menghapus dialect yang diasosiasi dengan nama dari registry dialect

▪ csv.list_dialects

Mengembalikan dialect yang diasosiasi dengan nama

▪ csv.field_size_limit

Mengembalikan ukuran field maksimum yang diizinkan oleh parser.

▪ csv.DictReader

Berfungsi untuk membaca dan mengembalikan data kedalam variable dictionary dari file csv.

```

1 def readdict(namafile):
2     with open(namafile, 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'Column names are {", ".join(row)}')
8                 line_count += 1
9                 print(f'\tNomor NPM: {row["npm"]} Nama: {row["nama
10                "]} Kelas: {row["kelas"]} ')
11                line_count += 1
12            print(f'Processed {line_count} lines.')

```



```

12
13 readdict('1174066.csv')

```

7. Jelaskan fungsi-fungsi yang terdapat di library pandas

▪ pandas.read_csv

Berfungsi untuk membaca dan mengembalikan data kedalam format DataFrame dari file csv.

```

1 def readpanda(namafile):
2     df = pandas.read_csv(namafile)
3     print(df)
4
5 readpanda('1174066.csv')

```

▪ to_dict

Berfungsi untuk membaca dan mengembalikan data kedalam format dictionary dari file csv.

```

1 def readpandadict(namafile):
2     df = pandas.read_csv(namafile).to_dict()
3     print(df)
4
5 readpandadict('1174066.csv')

```

▪ to_csv

Berfungsi untuk mengedit data didalam csv dan menulisnya kedalam file csv

```

1 def writepanda(namafile):
2     df = pandas.read_csv(namafile,
3         index_col='npm',
4         parse_dates=['tanggalahir'],
5         header=0,
6         names=['npm', 'nama', 'kelas', 'tanggalahir'])
7     df.to_csv('1174066_edit.csv')
8 writepanda('1174066.csv')

```

6.2 Fanny Shafira Damayanti | 1174069

6.2.1 Pemahaman Teori

1. Fungsi CSV, sejarah dan contoh

CSV (comma separated value) digunakan pada tahun 1983 merupakan suatu tipe file yang digunakan untuk pengolahan informasi yang dihasilkan spreadsheet yang di proses melalui mesin analitik. CSV juga digunakan sebagai file yang agnostik karena bisa digunakan oleh berbagai database untuk backup data.

Contoh :

```

1 import csv
2
3
4 siswa = [
5     ('fanny', 'A', 90),
6     ('shafira', 'B', 85),
7     ('dama', 'A', 80),
8     ('yanti', 'B', 90),
9
10 ]
11
12 # tentukan lokasi file, nama file, dan inisialisasi csv
13 f = open('siswa.csv', 'w')
14 w = csv.writer(f)
15 w.writerow(('Nama', 'Kelas', 'Nilai'))
16
17 # menulis file csv
18 for s in siswa:
19     w.writerow(s)
20
21 # menutup file csv
22 f.close()

```

2. Aplikasi untuk membuat file CSV

- Microsoft Excel
- Spyder
- Apple Number
- LibrareOffice
- Apple Office Calc
- Apache Open Office Calc

3. Cara menulis dan membaca file .csv di Excel atau Spreadsheet

Cara menulis :

Buka Ms. Excel, lalu buat file nya, ketika akan di save ganti jenis filenya menjadi .csv

- Download Template CSV.
- Buka file spreadsheet Anda di Excel.
- Buat dokumen baru di Excel.
- Tambahkan judul kolom, lalu ketikkan informasi dalam kolom tersebut
- Klik File , dan pilih Save As
- Masukkan nama file, lalu pilih CSV (Comma delimited) (* csv) dari drop-down Save as type .

Cara membaca :

- Buka Microsoft Excel.
 - Mulai / buka spreadsheet
 - Pilih tab Data
 - Pilih opsi Dari Teks. (Jika opsi berwarna abu-abu, Anda mungkin perlu membuka spreadsheet / workbook baru).
 - Temukan dan pilih file .csv yang telah Anda unduh dari Kotive. Klik pada file dan kemudian klik Impor.
 - Panduan impor Teks akan terbuka. Pastikan opsi Dibatasi dipilih. Klik tombol Berikutnya.
 - Pilih Koma di bawah Pembatas. Kualifikasi Teks harus menunjukkan “(tanda kutip ganda). Klik tombol Selesai. Anda mungkin ditanya Di mana Anda ingin meletakkan data? Klik pada sel kiri atas. Klik tombol OK.
 - Excel menampilkan data di buku kerja Anda
4. Sejarah Library CSV Inisiatif standarisasi utama - mentransformasikan “definisi fuzzy de facto” menjadi definisi yang lebih tepat dan de jure - adalah pada tahun 2005, dengan RFC4180, mendefinisikan CSV sebagai Tipe Konten MIME. Kemudian, pada 2013, beberapa kekurangan RFC4180 ditangani oleh rekomendasi W3C.
- Pada 2014 IETF menerbitkan RFC7111 yang menjelaskan aplikasi fragmen URI pada dokumen CSV. RFC7111 menentukan bagaimana rentang baris, kolom, dan sel dapat dipilih dari dokumen CSV menggunakan indeks posisi.
- Pada 2015 W3C, dalam upaya untuk meningkatkan CSV dengan semantik formal, mempublikasikan draft rekomendasi pertama untuk standar metadata CSV, yang dimulai sebagai rekomendasi pada bulan Desember tahun yang sama.
5. Sejarah Library Pandas
- Pandas muncul ketika ada bahasa pemrograman R dan Matlab. Pengembang Wes McKinney mulai mengerjakan pandas pada 2008 ketika di AQR Capital Management karena kebutuhan akan alat kinerja tinggi yang fleksibel untuk melakukan analisis kuantitatif pada data keuangan. Sebelum meninggalkan AQR, dia bisa meyakinkan manajemen untuk mengizinkannya membuka sumber perpustakaan.
- Pegawai AQR lainnya, Chang She, bergabung dengan upaya ini pada 2012 sebagai kontributor utama kedua ke perpustakaan.
- Pada 2015, panda menandatangani sebagai proyek NumFOCUS yang disponsori secara fiskal, sebuah badan amal nirlaba 501 (c) (3) di Amerika Serikat.
6. Fungsi yang terdapat pada Library CSV
- `csv.reader` berfungsi untuk membaca modul `csv`.

- `csv.writer` berfungsi untuk menulis modul `csv`.
- `csv.writerows` berfungsi untuk menambahkan baris baru.

7. Fungsi yang terdapat pada Library Pandas

Dengan `panda` kita dapat dengan mudah merubah data (`CSV`, `excel`, `JSON` atau `SQL`) menjadi sebuah object data yang terdiri dari baris dan kolom yang disebut dengan `DataFrame`. Fitur :

- `DataFrame` Object untuk manipulasi data dengan pengindeksan terintegrasi.
- Alat untuk membaca dan menulis data antara struktur data dalam memori dan berbagai format file.
- Penyelarasan data dan penanganan terpadu pada kehilangan data.
- Membentuk kembali dan memutar set data.
- Seleksi berbasis label, pengindeksan fantastis, dan melakukan subset kumpulan data besar.
- Penyisipan dan penghapusan kolom struktur data.
- Memungkinkan operasi `split-apply-combine` pada Data set.
- Menghubungkan dan menggabungkan Data set.
- Pengindeksan hierarki untuk bekerja dengan data dimensi tinggi dalam struktur data dimensi rendah.
- Fungsionalitas seri waktu: Pembuatan rentang tanggal dan konversi frekuensi.
- Menyediakan penyaringan data (`sorting` dan `filtering`).

BAB 7

PRAKTEK LIBRARY CSV DAN PANDAS

7.1 D. Irga B. Naufal Fakhri

7.1.1 Soal 1

Buatlah fungsi untuk membuka file csv dengan lib csv mode list

```
1 def read(namafile):
2     with open(namafile) as csv_file:
3         csv_reader = csv.reader(csv_file, delimiter=',')
4         line_count = 0
5         for row in csv_reader:
6             if line_count == 0:
7                 print(f'Column names are {", ".join(row)}')
8                 line_count += 1
9             else:
10                print(f'\tNomor NPM: {row[0]} Nama: {row[1]} Kelas: {
row[2]}'.')
11                line_count += 1
12                print(f'Processed {line_count} lines.')
```

7.1.2 Soal 2

Buatlah fungsi untuk membuka file csv dengan lib csv mode dictionary

```

1 def readdict(namafile):
2     with open(namafile, 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'Column names are {", ".join(row)}')
8                 line_count += 1
9                 print(f'\tNomor NPM: {row["npm"]} Nama: {row["nama"]}
10                Kelas: {row["kelas"]}')
11                line_count += 1
12            print(f'Processed {line_count} lines.')
```

7.1.3 Soal 3

Buatlah fungsi untuk membuka file csv dengan lib pandas mode list

```

1 def readpanda(namafile):
2     df = pandas.read_csv(namafile)
3     print(df)
```

7.1.4 Soal 4

Buatlah fungsi untuk membuka file csv dengan lib pandas mode dictionary

```

1 def readpandadict(namafile):
2     df = pandas.read_csv(namafile).to_dict()
3     print(df)
4
5 #readpandadict('1174066.csv')
```

7.1.5 Soal 5

Buat fungsi baru untuk mengubah format tanggal menjadi standar dataframe

```

1 def readrubahtanggal(namafile):
2     df = pandas.read_csv(namafile, index_col='npm', parse_dates=[
3         'tanggalahir'])
4     print(df)
```

7.1.6 Soal 6

Buat fungsi baru untuk mengubah index kolom

```

1 def changeindex(namafile):
2     df = pandas.read_csv(namafile, index_col='npm')
3     print(df)
```

7.1.7 Soal 7

Buat fungsi baru untuk mengubah atribut atau nama kolom

```
1 def renamekolom(namafile):
2     df = pandas.read_csv(namafile, header=0, names=['Nomor Induk
3         Mahasiswa', 'Nama Lengkap', 'Kelas dan Jurusan', 'Tanggal Lahir'])
4     print(df)
```

7.1.8 Soal 8

Buat program main yang menggunakan library NPM csv yang membuat dan membaca file csv

```
1 def write(namafile):
2     with open(namafile, mode='w') as employee_file:
3         tulis_file = csv.writer(employee_file, delimiter=',',
4             quotechar='"', quoting=csv.QUOTE_MINIMAL)
5
6         tulis_file.writerow(['npm', 'nama', 'kelas', 'tanggallahir'])
7         tulis_file.writerow(['1174066', 'D. Irga 3', 'D4 Teknik
8             Informatika 2C', '11/07/98'])
```

Listing 7.1 lib_1174066_csv.py

```
1 import lib_1174066_csv
2
3 # Membuat file csv baru yang isinya npm, nama, kelas dan tanggallahir
4 lib_1174066_csv.write('1174066_csv_baru.csv')
5
6 # Membaca file csv
7 # List
8 lib_1174066_csv.read('1174066.csv')
9 # Dictionary
10 lib_1174066_csv.readdict('1174066.csv')
```

Listing 7.2 main.py

7.1.9 Soal 9

Buat program main2.py yang menggunakan library NPM pandas.py yang membuat dan membaca file csv

```
1 def writepanda(namafile, filebaru):
2     df = pandas.read_csv(namafile,
3         index_col='npm',
4         parse_dates=['tanggallahir'],
5         header=0,
6         names=['npm', 'nama', 'kelas', 'tanggallahir'])
7     df.to_csv(filebaru)
```

Listing 7.3 main2.py


```

1 import lib_1174066_pandas
2
3 # Membuat file baru
4 lib_1174066_pandas.writepanda('1174066.csv', '1174066_baru_pandas.csv'
5 )
6 # Membaca file csv panda
7 # Dataframe
8 lib_1174066_pandas.readpanda('1174066.csv')
9 # Dictionary
10 lib_1174066_pandas.readpandadict('1174066.csv')

```

Listing 7.4 main2.py

7.1.10 Keterampilan Penanganan Error

Pada praktikum saat ini saya tidak mendapatkan error

7.2 Fanny Shafira Damayanti | 1174069

7.2.1 Keterampilan Pemrograman

1. NO 1

```

1 #No. 1
2 def bukaModeListCsv():
3     with open('teori.csv') as csv_file:
4         csv_reader = csv.reader(csv_file, delimiter=',')
5         for row in csv_reader:
6             print(row[0], row[1], row[2])

```

2. NO 2

```

1 #No. 2
2 def bukaModeDictCsv():
3     with open('teori.csv', mode='r') as csv_file:
4         csv_reader = csv.DictReader(csv_file)
5         for row in csv_reader:
6             print(row['npm'], row['nama'], row['kelas'])

```

3. NO 3

```

1 #No. 3
2 def bukaModeListPandas():
3     df = pandas.read_csv('teori.csv')
4     print(df)

```

4. NO 4

```

1 #No. 3
2 def bukaModeListPandas():
3     df = pandas.read_csv('teori.csv')
4     print(df)

```

5. NO 5

```

1 #No. 4
2 def bukaModeDictPandas():
3     df = pandas.read_csv('teori.csv')
4     dt = pandas.DataFrame.from_dict(df)
5     print(dt)

```

6. NO 6

```

1 #No. 5
2 def ubahFormatTanggal():
3     df = pandas.read_csv('teori.csv', parse_dates=['tanggal lahir'])
4     print(df)

```

7. NO 7

```

1 #No. 6
2 def ubahIndexKolom():
3     df = pandas.read_csv('teori.csv')
4     df.index = ['Row_1', 'Row_2']
5     print(df)

```

8. NO 8

```

1 lib = __import__('1174069_csv')
2
3 lib.bukaModeListCsv()
4 lib.bukaModeDictCsv()
5
6 lib.tulisCsv()

```

9. NO 9

```

1 lib = __import__('1174069_pandas')
2
3 lib.bukaModeListPandas()
4 lib.bukaModeDictPandas()
5
6 lib.tulisCsvPandas()

```

7.2.2 Penanganan Error

1. Peringatan error yang terdapat pada praktikum chapter 4 ini yaitu :

- Syntax Errors Syntax Errors terjadi ketika ada kesalahan dalam meuliskan kode. Solusinya adalah memperbaiki penulisan kode yang salah.
- Name Error NameError terjadi ketika salah mengetikan kode local name yang tidak terdefinisi. Solusinya adalah menuliskan kode dengan benar agar function nya dapat terpanggil.

- Type Error TypeError terjadi pada saat eksekusi terhadap fungsi dengan tipe objek tidak sesuai. Solusinya mengkonversi variabelnya harus sesuai dengan tipe datanya.

Contoh Penggunaan TryExcept

```
1 #Fungsi Try Except
2 def bacaCsvPandas():
3     try:
4         df = pandas.read_csv('teori.csv')
5         print(df)
6     except SyntaxError:
7         print("Kesalahan penulisan syntax")
8     except NameError:
9         print("Variable tersebut tidak ada")
10    except TypeError:
11        print("Tipe data salah")
12    except:
13        print("Terjadi sebuah kesalahan")
```