

Laporan Tugas

Instalasi dan Penggunaan

Anaconda



Dinda Majesty
NPM : 1.18.4.011

Informatics Research Center
Applied Bachelor Program of Informatics Engineering
Bandung 2019

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

Acknowledgements

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

Abstract

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

Contents

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

2.1.6	Error	25
2.1.7	Try Except	27
2.2	Keterampilan Pemrograman	27
2.2.1	Modulus	27
2.2.2	Hello NPM	27
2.2.3	Hello NPM (3 Digit Belakang)	28
2.2.4	Hello NPM (Digit ke-3)	28
2.2.5	Variabel Alfabet	28
2.2.6	Penjumlahan NPM	28
2.2.7	Perkalian NPM	29
2.2.8	Print Vertical	29
2.2.9	Digit Genap NPM	29
2.2.10	Digit Ganjil NPM	29
2.2.11	Bilangan Prima NPM	30
2.3	Keterampilan Penanganan Error	30
2.3.1	Peringatan Error dan Cara Mengatasinya	30
2.3.2	Percobaan Try Except	31
3	Fungsi dan Kelas	32
3.1	Teori	32
3.2	Fungsi	32
3.3	Package	32
3.4	Kelas, Objek, Atribut, dan Method	33
3.4.1	Pemanggilan Library Kelas	34
3.4.2	From kalkulator import Penambahan	34
3.4.3	Pemakaian Package Fungsi Apabila File Didalam Folder	34
3.4.4	Pemakaian Package Kelas Apabila File didalam Folder	35
3.5	Keterampilan Pemrograman	35
3.5.1	Modulus	35
3.5.2	Hello NPM	35
3.5.3	Hello NPM (3 Digit Belakang)	36
3.5.4	Hello NPM (Digit ke-3)	36
3.5.5	Variabel Alfabet	36
3.5.6	Penjumlahan NPM	36
3.5.7	Perkalian NPM	37
3.5.8	Print Vertical	37

3.5.9	Digit Genap NPM	37
3.5.10	Digit Ganjil NPM	38
3.5.11	Bilangan Prima NPM	38
3.5.12	Pemanggilan Fungsi pada Main.py	38
3.6	Keterampilan Penanganan Error	39
3.6.1	Peringatan Error dan Cara Mengatasinya	39
4	Pengelolaan File CSV	40
4.1	Teori	40
4.2	Fungsi File CSV, Sejarah, dan Contoh	40
4.3	Aplikasi yang bisa Menciptakan File CSV	41
4.4	Menulis dan Membaca File CSV pada Ms.Excel	42
4.4.1	Sejarah Library CSV	43
4.4.2	Sejarah Library Pandas	43
4.4.3	Fungsi - fungsi yang terdapat di library CSV	43
4.4.3.1	Fungsi - fungsi yang terdapat di library Pandas	44
4.5	Keterampilan Pemrograman	45
4.5.1	CSV List	45
4.5.2	CSV Dictionary	45
4.5.3	Hello NPM (Pandas List)	45
4.5.4	Hello NPM (Pandas Dictionary)	45
4.5.5	Pandas Date	46
4.5.6	Pandas Ubah Index	46
4.5.7	Pandas Ubah Column	46
4.5.8	Main CSV	46
4.5.9	Main Pandas	46

List of Figures

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

1.29	<i>Variable Explorer</i>	16
1.30	<i>Indentasi</i>	18
1.31	<i>Error Indentasi</i>	18
1.32	<i>Error</i>	19
1.33	<i>Syntax Error</i>	19
1.34	<i>Syntax yang Telah Diperbaiki</i>	20
2.1	<i>Try Except</i>	31
4.1	Contoh Penulisan CSV pada Excel	42

Listings

2.1	Input dan Output	22
2.2	While Loop	23
2.3	For Loop	23
2.4	Nested Loop	23
2.5	if Statement	24
2.6	Elif	24
2.7	Else	25
2.8	Nested If	25
2.9	Try Except	27
2.10	Modulus	27
2.11	Hello NPM	28
2.12	3 Digit Belakang	28
2.13	Digit ke-3	28
2.14	Variabel Alfabet	28
2.15	Penjumlahan NPM	28
2.16	Perkalian NPM	29
2.17	Print Vertical	29
2.18	Digit Genap NPM	29
2.19	Digit Ganjil NPM	29
2.20	Bilangan Prima NPM	30
2.21	Bilangan Prima NPM	31
3.1	Modulus	35
3.2	Hello NPM	36
3.3	3 Digit Belakang	36
3.4	Digit ke-3	36
3.5	Variabel Alfabet	36
3.6	Penjumlahan NPM	37
3.7	Perkalian NPM	37

3.8	Print Vertical	37
3.9	Digit Genap NPM	37
3.10	Digit Ganjil NPM	38
3.11	Bilangan Prima NPM	38
3.12	Bilangan Prima NPM	38
4.1	Contoh CSV	40
4.2	Contoh Kode Python to Read CSV	40
4.3	Contoh Kode Python to Read CSV	41
4.4	Contoh CSV	42
	src/fcsv.py	43
	src/fcsv.py	43
	src/fcsv.py	44
	src/fcsv.py	44
4.5	CSV List	45
4.6	CSV Dictionary	45
4.7	Pandas List	45
4.8	Pandas Dictionary	45
4.9	Pandas Date	46
4.10	Pandas Ubah Index	46
4.11	Pandas Ubah Column	46
4.12	Main CSV	46
4.13	Digit Genap NPM	46

Chapter 1

Fungsi dan Kelas

1.1 Teori

1.2 Fungsi

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

contoh fungsi:

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

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

1.3 Package

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

Contoh package:

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

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

```
import fungsi

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

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

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

1.4 Kelas, Objek, Atribut, dan Method

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

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

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

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

1.4.1 Pemanggilan Library Kelas

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

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

```
import ngitung
```

```
hitung = ngitung.Ngitung
```

1.4.2 From kalkulator import Penambahan

Kode ini berarti dari module kalkulator kita ingin mengimportkan fungsi penambahannya saja. contohnya, kita memiliki sebuah module bernama `kalkuator.py` yang didalamnya terdapat fungsi `Penambahan`, untuk bisa menggunakan fungsi `Penambahan` yang ada pada module kalkulator maka kita bisa mengetikkan kode seperti berikut.

```
from kalkulator import Penambahan  
print(Penambahan(5,7))
```

berikut kode yang terdapat pada module `kalkulator.py`

```
def Penambahan(a ,b) :  
    r = a + b  
    return r
```

1.4.3 Pemakaian Package Fungsi Apabila File Didalam Folder

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

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

1.4.4 Pemakaian Package Kelas Apabila File didalam Folder

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

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

1.5 Keterampilan Pemrograman

1.5.1 Modulus

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

Listing 1.1: Modulus

1.5.2 Hello NPM

```

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

```

Listing 1.2: Hello NPM

1.5.3 Hello NPM (3 Digit Belakang)

```

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

```

Listing 1.3: 3 Digit Belakang

1.5.4 Hello NPM (Digit ke-3)

```

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

```

Listing 1.4: Digit ke-3

1.5.5 Variabel Alfabet

```

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

```

Listing 1.5: Variabel Alfabet

1.5.6 Penjumlahan NPM


```

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

```

Listing 1.6: Penjumlahan NPM

1.5.7 Perkalian NPM

```

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

```

Listing 1.7: Perkalian NPM

1.5.8 Print Vertical

```

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

```

Listing 1.8: Print Vertical

1.5.9 Digit Genap NPM

```

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

```

```

8         print(self.NPM[index])
9         index += 1

```

Listing 1.9: Digit Genap NPM

1.5.10 Digit Ganjil NPM

```

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

```

Listing 1.10: Digit Ganjil NPM

1.5.11 Bilangan Prima NPM

```

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

```

Listing 1.11: Bilangan Prima NPM

1.5.12 Pemanggilan Fungsi pada Main.py

```

1     import fungsi
2
3     print("Masukkan NPM anda: ")
4     NPM = input()
5     crot = fungsi.Kelas(NPM)
6
7     soal1 = crot.Modulus()
8     soal2 = crot.Hello_NPM()
9     soal3 = crot.NPM_DigitBelakang()
10    soal4 = crot.NPM_DigitKetiga()

```

```

11 soal5 = crot.Variabel_Alfabet()
12 soal6 = crot.Penjumlahan_NPM()
13 soal7 = crot.Perkalian_NPM()
14 soal8 = crot.Print_Vertical()
15 soal9 = crot.DigitGenap()
16 soal10 = crot.DigitGanjil()
17 soal11 = crot.PrimaNPM()

```

Listing 1.12: Bilangan Prima NPM

1.6 Keterampilan Penanganan Error

1.6.1 Peringatan Error dan Cara Mengatasinya

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