

Tugas Chapter 3

Pemrograman II



Dyning Aida Batrishya

1184030

D4 Teknik Informatika 2B

Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering

Politeknik Pos Indonesia

Bandung 2019

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

Acknowledgements

Assalamualaikum warahmatullahi wabarakaatuh. Segala puji bagi Allah SWT yang telah memberikan kemudahan sehingga dapat menyelesaikan laporan Tugas Chapter 3 ini, tanpa bantuan-Nya maka penulis tidak dapat menyelesaikannya dengan baik dan tepat pada waktunya. Shalawat serta salam semoga terlimpahkan kepada junjungan Nabi Muhammad SAW yang akan kita nantikan syafaatnya di yaumul qimayah nanti.

Laporan ini disusun guna memenuhi kelulusan matakuliah Pemrograman II Program Studi DIV Teknik Informatika. Proses penyelesaian laporan ini tidak luput dari bantuan berbagai pihak. Oleh karenanya, penulis mengucapkan terima kasih kepada :

1. Allah SWT yang telah memberikan rahmat dan hidayah-Nya
2. Orang tua yang selalu memberikan dukungan dan motivasi dalam penyelesaian laporan
3. Bapak Rolly Awangga yang telah memberikan dukungan dan bantuan dalam penyelesaian laporan
4. Teman-teman yang saya sayangi yang selalu memberikan dukungan dan motivasinya kepada penulis

Penulis mengharapkan kritik dan saran dari pembaca jika terdapat kesalahan dalam penyusunan laporan ini sehingga penulis dapat memperbaiki penyelesaian tugas yang selanjutnya

Bandung, 10 November 2019

Penulis

Chapter 1

LAPORAN

1.1 TUGAS TEORI

- Fungsi

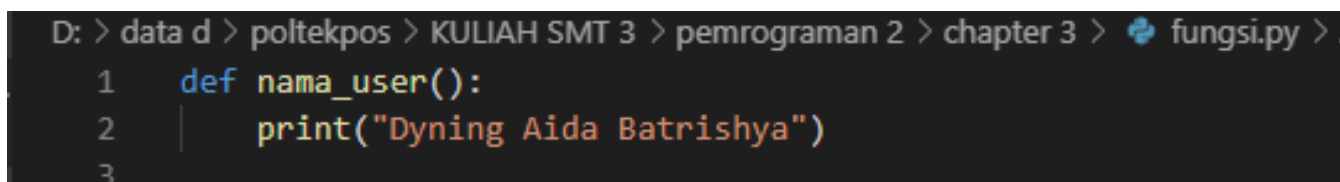
Fungsi merupakan sekumpulan kode blok yang disusun secara terorganisir dan dapat digunakan kembali di kemudian untuk membuat sebuah tindakan.

1. Inputan Fungsi

Untuk membuat inputan fungsi, terdapat beberapa aturan yang harus dipatuhi, di antaranya yaitu :

- (a) Penulisan fungsi dimulai dengan `def` sebagai kata kunci, kemudian diikuti oleh nama fungsi itu sendiri disertai dengan tanda kurung `()`
- (b) Setiap parameter dalam fungsi tersebut dituliskan di dalam kurung tersebut, di dalamnya juga dapat dilakukan penentuan atau pengaturan terhadap variabel di dalamnya.
- (c) Blok kode yang dituliskan pada fungsi selalu diikuti dengan titik dua `:` dan diikuti oleh indentasi setelahnya.

Berikut ini merupakan contoh pembuatan inputan fungsi yang benar

A screenshot of a code editor window. The title bar shows the file path: "D: > data d > poltekpos > KULIAH SMT 3 > pemrograman 2 > chapter 3 > fungsi.py". The code content is as follows:

```
1  def nama_user():
2      print("Dyning Aida Batrishya")
3
```

Figure 1.1: inputan fungsi

2. Berikut ini merupakan cara pemanggilan kode fungsi

```

1  def nama_user():
2      print("Dyning Aida Batrishya")
3
4  nama_user()

```

Figure 1.2: pemanggilan fungsi

Tata cara pemanggilan fungsi dapat dilakukan dengan menuliskan nama dari fungsi yang telah dibuat sebelumnya yang kemudian diikuti dengan tanda kutung (()) yang diikuti dengan menuliskan parameter yang terdapat di dalamnya.

3. Contoh kode program sederhana yang menggunakan fungsi :

```

D: > data d > poltekpos > KULIAH SMT 3 > pemrograman 2 > chapter 3 > fungsi.py > ...
1  def nama_user(str):
2      nama = input("Masukkan nama anda :")
3      print("Selamat datang", nama)
4
5  nama_user(str)

```

Figure 1.3: contoh program sederhana yang menggunakan fungsi

- Pengertian Paket Library dan Cara penggunaannya class digunakan untuk membuat struktur data pengguna yang berisikan tentang informasi terhadap suatu objek yang akan didefinisikan. contoh pemakaian class yaitu
- Pengertian kelas, objek, atribut, dan method
 1. Pengertian Kelas
Kelas adalah blue print atau prototipe dari suatu kelas yang mendefinisikan atribut dari suatu objek
 2. Pengertian Objek
Objek merupakan instansiasi atau perwujudan suatu class, bisa disebut juga bahwa objek merupakan hasil atau barang jadinya

3. Pengertian Atribut

Suatu data yang merupakan anggota dari sebuah variabel yang diakses dengan menggunakan notasi titik.

4. Pengertian Method

Method merupakan suatu pendefinisian fungsi dari suatu class.

- Cara pemanggilan library melalui instansiasi

Instansiasi adalah cara pemanggilan instance atau objek dari suatu kelas yang telah dibuat sebelumnya.

- Contoh pemakaian paket dengan perintah from kalkulator import

Berikut ini merupakan contoh pembuatan paket kalkulator

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sat Nov  2 15:08:38 2019
4
5 @author: User
6 """
7
8 def Penambahan(a,b):
9     r = a + b
10    return r
11
12 def Pengurangan(a, b):
13     r = a - b
14     return r
15 def Perkalian(a, b):
16     r = a * b
17     return r
18 def Pembagian(a, b):
19     r = a / b
20     return r
```

Cara pemanggilan perintah from kalkulator import

```
1 from kalkulator import Penambahan
2
3 x = Penambahan(13,29)
4 print(x)
```

- Kode ketika pemanggilan paket fungsi dalam sebuah library yang sama

1. Buat kelas yang berisikan kumpulan fungsi di dalamnya

2. Panggil dengan menggunakan kata kunci "import from (nama fungsi)".

Berikut ini merupakan contohnya :

Contoh pembuatan paket fungsi

```

1 def salam():
2     print("Haii , selamat datang!")
3     return salam
4
5 salam()
6
7 def nama():
8     print("Nama saya adalah :")
9
10 nama()
11
12 def umur():
13     print("Umur saya adalah:")
14
15 umur()

```

Contoh pemanggilan paket fungsi

```

1 from kelasfungsi import salam
2
3 x = salam()
4 print(x)

```

- Kode ketika pemanggilan paket fungsi dalam sebuah library yang berbeda
Untuk pemanggilan paket fungsi yang berbeda folder, maka harus dituliskan nama folder yang menyimpan paket fungsi tersebut dahulu sebelum mengimpor fungsinya

1.2 KETERAMPILAN

1.

```

1 def npm1():
2     print(" +++ +++   +++++   +++  +++   ++++++   +++++   ++++++ ")
3     print(" +++ +++ +++   +++   +++  +++   +++   +++   +++   +++ ")
4     print(" +++ +++   +++++   ++++++   +++   +++   +++   +++   +++ ")
5     print(" +++ +++ +++   +++   +++   +++   +++   +++   +++   +++ ")
6     print(" +++ +++   +++++   +++   ++++++   +++++   ++++++ ")

```

2.

```

1 def npm2(npm):
2     npm=int(npm)
3     duadigit=abs(npm)%100
4     for i in range(duadigit):
5         print("Halo, ", npm, " bagaimana kabarmu ?")

```

3.

```

1 def npm3(npm):
2     for i in range(int(str(npm)[4])+int(str(npm)[5])+int(str(npm)[6])):

```

```

3         print("Haiiii , "+str(npm)[4]+str(npm)[5]+str(npm)[6]+" apa kabar
        ?")
4
5     i=0
6     npm=input("Masukan NPM : ")
7     while i<1:
8         if len(npm) < 7:
9             print("NPM Kurang dari 7 digit")
10            npm=input("Masukan NPM : ")
11        elif len(npm) > 7:
12            print("NPM lebih dari 7 digit")
13            npm=input("Masukan NPM : ")
14        else :
15            i=1
16    npm3(npm)

```

4.

```

1 def npm4(npm):
2     x=npm%1000
3     str_key=str(x)
4     print("Halo, "+str_key[0]+" apa kabar ?")
5
6     i=0
7     npm=input("Masukan NPM : ")
8     while i<1:
9         if len(npm) < 7:
10            print("NPM Kurang dari 7 digit")
11            npm=input("Masukan NPM : ")
12        elif len(npm) > 7:
13            print("NPM lebih dari 7 digit")
14            npm=input("Masukan NPM : ")
15        else :
16            i=1
17    npm4(npm)

```

5.

```

1 def npm5(npm):
2     a=npm[0]
3     b=npm[1]
4     c=npm[2]
5     d=npm[3]
6     e=npm[4]
7     f=npm[5]
8     g=npm[6]
9
10    for x in a,b,c,d,e,f,g:
11        print(x)
12
13    i=0
14    npm=input("Masukan NPM : ")
15    while i<1:
16        if len(npm) < 7:
17            print("NPM Kurang dari 7 digit")

```



```

18     npm=input("Masukan NPM : ")
19     elif len(npm) > 7:
20         print("NPM lebih dari 7 digit")
21         npm=input("Masukan NPM : ")
22     else :
23         i=1
24 npm5(npm)

```

6.

```

1 def npm6(npm) :
2
3     a=npm[0]
4     b=npm[1]
5     c=npm[2]
6     d=npm[3]
7     e=npm[4]
8     f=npm[5]
9     g=npm[6]
10    y=0
11
12    for x in a,b,c,d,e,f,g:
13        y+=int(x)
14    print(y)
15
16    i=0
17    npm=input("Masukan NPM : ")
18    while i<1:
19        if len(npm) < 7:
20            print("NPM Kurang dari 7 digit")
21            npm=input("Masukan NPM : ")
22        elif len(npm) > 7:
23            print("NPM lebih dari 7 digit")
24            npm=input("Masukan NPM : ")
25        else :
26            i=1
27    npm6(npm)

```

7.

```

1 def npm7(npm) :
2
3     a=npm[0]
4     b=npm[1]
5     c=npm[2]
6     d=npm[3]
7     e=npm[4]
8     f=npm[5]
9     g=npm[6]
10    conv=1
11
12    for x in a,b,c,d,e,f,g:
13        conv*=int(x)
14    print(conv)
15

```

```

16 i=0
17 npm=input("Masukan NPM : ")
18 while i<1:
19     if len(npm) < 7:
20         print("NPM Kurang dari 7 digit")
21         npm=input("Masukan NPM : ")
22     elif len(npm) > 7:
23         print("NPM lebih dari 7 digit")
24         npm=input("Masukan NPM : ")
25     else :
26         i=1
27 npm7(npm)

```

8.

```

1 def npm8(npm) :
2     a=npm[0]
3     b=npm[1]
4     c=npm[2]
5     d=npm[3]
6     e=npm[4]
7     f=npm[5]
8     g=npm[6]
9     for x in a,b,c,d,e,f,g:
10         if int(x)%2==0:
11             if int(x)==0:
12                 x=""
13             print(x,end =" ")
14
15
16 i=0
17 npm=input("Masukan NPM : ")
18 while i<1:
19     if len(npm) < 7:
20         print("NPM Kurang dari 7 digit")
21         npm=input("Masukan NPM : ")
22     elif len(npm) > 7:
23         print("NPM lebih dari 7 digit")
24         npm=input("Masukan NPM : ")
25     else :
26         i=1
27 npm8(npm)

```

9.

```

1 def npm9(npm) :
2     a=npm[0]
3     b=npm[1]
4     c=npm[2]
5     d=npm[3]
6     e=npm[4]
7     f=npm[5]
8     g=npm[6]
9     for x in a,b,c,d,e,f,g:
10

```

```

11         if int(x)%2==1:
12             print(x,end =" ")
13
14     i=0
15     npm=input("Masukan NPM : ")
16     while i<1:
17         if len(npm) < 7:
18             print("NPM Kurang dari 7 digit")
19             npm=input("Masukan NPM : ")
20         elif len(npm) > 7:
21             print("NPM lebih dari 7 digit")
22             npm=input("Masukan NPM : ")
23         else :
24             i=1
25     npm9(npm)

```

10.

```

1 def npm10(npm):
2     a=npm[0]
3     b=npm[1]
4     c=npm[2]
5     d=npm[3]
6     e=npm[4]
7     f=npm[5]
8     g=npm[6]
9     for x in a,b,c,d,e,f,g:
10         if int(x) > 1:
11             for i in range(2,int(x)):
12                 if (int(x) % i) == 0:
13                     break
14             else :
15                 print(int(x),end =" ") ,
16
17     i=0
18     npm=input("Masukan NPM : ")
19     while i<1:
20         if len(npm) < 7:
21             print("NPM Kurang dari 7 digit")
22             npm=input("Masukan NPM : ")
23         elif len(npm) > 7:
24             print("NPM lebih dari 7 digit")
25             npm=input("Masukan NPM : ")
26         else :
27             i=1
28     npm10(npm)

```

11.

```

1 def npm1():
2     print(" +++ +++   ++++   +++ ++   ++++++   +++++   ++++++ ")
3     print(" +++ +++ +++   +++   +++ ++   +++ ++   +++   +++ ++ ")
4     print(" +++ +++   ++++   ++++++++   +++ ++   +++   +++ ++ ")
5     print(" +++ +++ +++   +++   +++   +++ ++   +++   +++ ++ ")
6     print(" +++ +++   ++++   +++   ++++++   +++++   ++++++ ")

```

```

7 def npm2(npm):
8     NPM=int(input("berapa npm kamu? :"))
9     tigadigit=NPM%100
10    for NPM in range(tigadigit):
11        print("Haiii ", NPM, " apa kabar ?")
12 def npm3(npm):
13     for i in range(int(str(npm)[4])+int(str(npm)[5])+int(str(npm)[6])):
14         print("Hallo, "+str(npm)[4]+str(npm)[5]+str(npm)[6]+" bagaimana
15         kabarmu ?")
16     return None
17 def npm4(npm):
18     key=npm%1000
19     str_key=str(key)
20     print("Halo, "+str_key[0]+" apa kabar ?")
21 def npm5(npm):
22     a=npm[0]
23     b=npm[1]
24     c=npm[2]
25     d=npm[3]
26     e=npm[4]
27     f=npm[5]
28     g=npm[6]
29
30     for x in a,b,c,d,e,f,g:
31         print(x)
32 def npm6(npm):
33
34     a=npm[0]
35     b=npm[1]
36     c=npm[2]
37     d=npm[3]
38     e=npm[4]
39     f=npm[5]
40     g=npm[6]
41     y=0
42
43     for x in a,b,c,d,e,f,g:
44         y+=int(x)
45     print(y)
46 def npm7(npm):
47
48     a=npm[0]
49     b=npm[1]
50     c=npm[2]
51     d=npm[3]
52     e=npm[4]
53     f=npm[5]
54     g=npm[6]
55     conv=1
56
57     for x in a,b,c,d,e,f,g:
58         conv*=int(x)
59     print(conv)

```

```

60 def npm8(npm):
61     a=npm[0]
62     b=npm[1]
63     c=npm[2]
64     d=npm[3]
65     e=npm[4]
66     f=npm[5]
67     g=npm[6]
68     for x in a,b,c,d,e,f,g:
69         if int(x)%2==0:
70             if int(x)==0:
71                 x=""
72             print(x,end = " ")
73 def npm9(npm):
74     a=npm[0]
75     b=npm[1]
76     c=npm[2]
77     d=npm[3]
78     e=npm[4]
79     f=npm[5]
80     g=npm[6]
81     for x in a,b,c,d,e,f,g:
82
83         if int(x)%2==1:
84             print(x,end = " ")
85 def npm10(npm):
86     a=npm[0]
87     b=npm[1]
88     c=npm[2]
89     d=npm[3]
90     e=npm[4]
91     f=npm[5]
92     g=npm[6]
93     for x in a,b,c,d,e,f,g:
94         if int(x) > 1:
95             for i in range(2,int(x)):
96                 if (int(x) % i) == 0:
97                     break
98             else:
99                 print(int(x),end = " "),

```

12.

```

1 import kelas3lib
2 import lib3
3
4 try:
5     2/0
6 except Exception:
7     print(Exception)
8
9
10
11 npm=input("Masukan NPM kalian : ")
12 i=0

```

```

13 while i<1:
14     if len(npm) < 7:
15         print("NPM Kurang dari 7 digit")
16         npm=input("Masukan NPM kalian : ")
17     elif len(npm) > 7:
18         print("NPM lebih dari 7 digit")
19         npm=input("Masukan NPM kalian : ")
20     else :
21         i=1
22
23 #Contoh pemanggilan fungsi pada class
24 cobakelas=kelas3lib.Kelas3ngitung(npm)
25 hasilkelas=cobakelas.npm1()
26
27
28 print("")
29
30 #Contoh pemanggilan fungsi pada library
31 lib3.npm3(npm)

```

```

1 import lib3
2 class Kelas3ngitung:
3     def __init__(self,npm):
4         self.npm = npm
5     def npm1(self):
6         return lib3.npm1()
7     def npm2(self):
8         return lib3.npm2(self.npm)
9     def npm3(self):
10        return lib3.npm3(self.npm)
11    def npm4(self):
12        return lib3.npm4(self.npm)
13    def npm5(self):
14        return lib3.npm5(self.npm)
15    def npm6(self):
16        return lib3.npm6(self.npm)
17    def npm7(self):
18        return lib3.npm7(self.npm)
19    def npm8(self):
20        return lib3.npm8(self.npm)
21    def npm9(self):
22        return lib3.npm9(self.npm)
23    def npm10(self):
24        return lib3.npm10(self.npm)

```

1.3 KETERAMPILAN PENANGANAN ERROR

1. peringatan error pada praktek ketiga dan penjelasan penanganan error tersebut.

Peringatan error yang muncul salah satunya yaitu seperti gambar di bawah ini :

```
1 def Pembagian(a,b):  
2     r = a / b  
3     return r  
4  
5 try:  
6     Pembagian(17,0)  
7 except Exception as e:  
8     pass  
9 print("angka yang anda masukkan tidak tepat")
```

Cara penanganan yang digunakan untuk mengatasi error tersebut yaitu dengan memberikan try untuk menghasilkan/mencetak fungsi sebelumnya, kemudian jika parameter yang ditangkap tidak sesuai maka, program akan mengeksekusi exception dan mencetak peringatannya.

1.4 SCAN PLAGIARISME

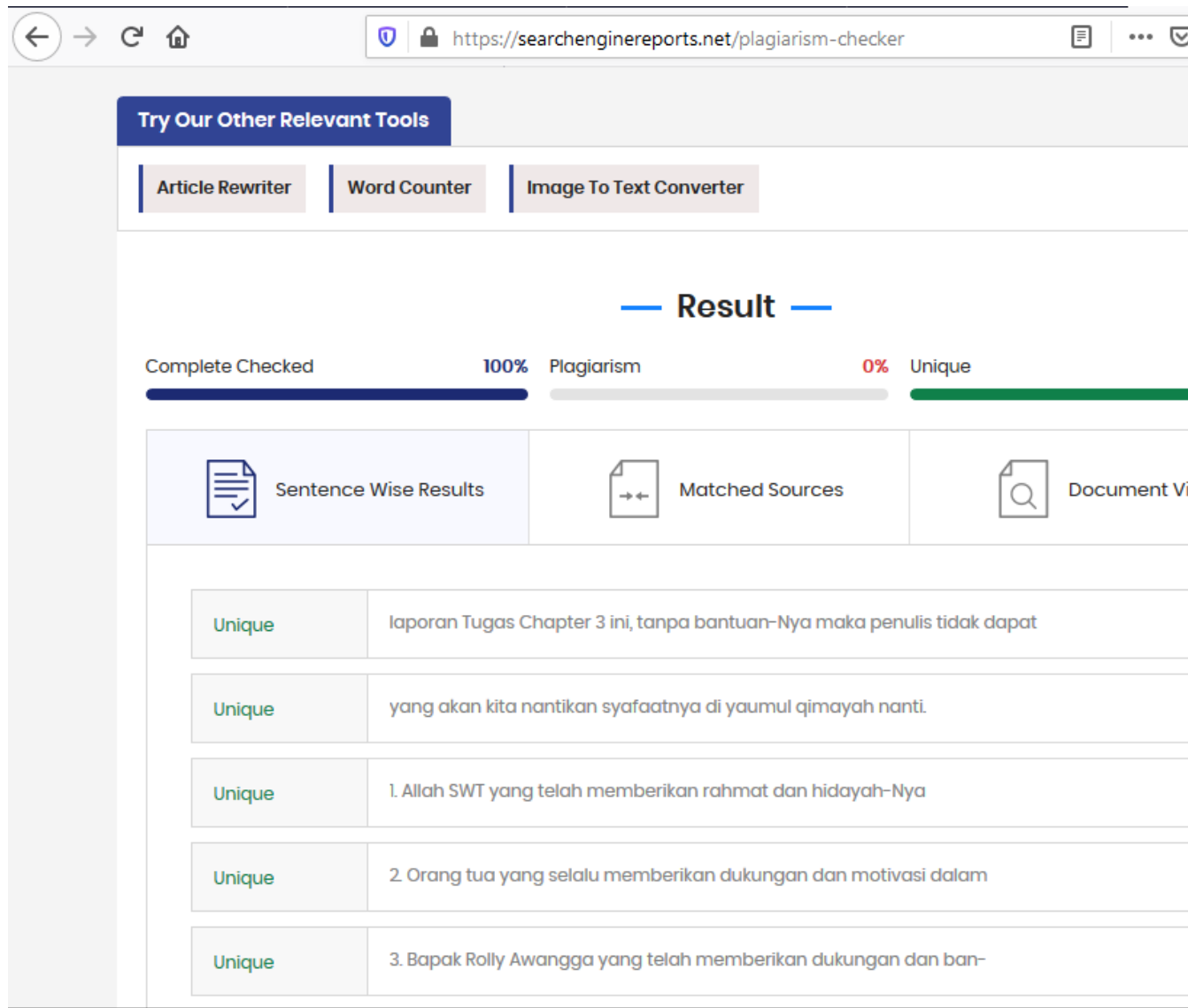


Figure 1.4: hasil scan plagiarisme