



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231056
Nama Lengkap	TARA TIRTANATA
Minggu ke / Materi	04 / Modular Programming

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA

YOGYAKARTA

2024

BAGIAN 1: MATERI MINGGU INI (40%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

MATERI 1: Fungsi, Argument, dan Parameter

Perhatikan kode program di bawah ini.

```
age = input("enter your age (in numbers): ")
print("hello, i am", age, "years old")
```

input() dan **print()** merupakan beberapa contoh fungsi dalam bahasa pemrograman Python. Setiap fungsi tentunya memiliki kegunaan yang berbeda-beda. Untuk contoh yang sudah ada di atas, **input()** berfungsi untuk membaca input yang diberi pengguna, sedangkan fungsi **print()** digunakan untuk menampilkan tulisan yang diinginkan agar muncul ke layar.

Fungsi dan modular programming berkaitan karna jika kita mau membuat program dengan langkah yang banyak, maka kita harus mengelompokkannya menjadi bagian-bagian (block) dari program yang besar itu. Itu sebabnya ini disebut modular, yang dimana program kita terdiri dari beberapa bagian modular yang masing-masing punya kegunaan tertentu (dapat digunakan ulang).

Fungsi sendiri terbagi menjadi dua jenis, yaitu:

- built-in function (fungsi bawaan)
- function yang dibuat oleh programmer

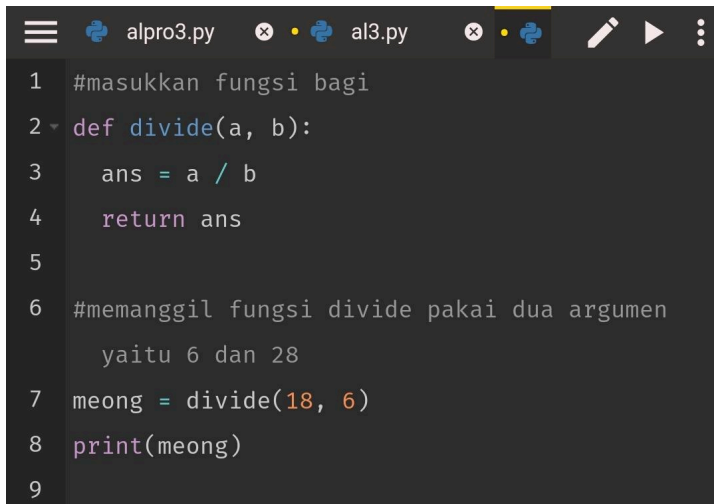
Berikut ini adalah contoh fungsi **divide()** untuk menghitung hasil bagi dua bilangan.

```
def divide(a, b):
    ans = a / b
    return ans
```

Nah ada beberapa hal yang perlu diperhatikan dan dipelajari dari fungsi bagi di atas.

- kata kunci **def** berfungsi untuk mendefinisikan suatu fungsi.
- nama fungsinya adalah **divide()**
- isi dari fungsi (yang ada di tanda kurung) membutuhkan dua argumen dan disebut parameter.
- tulisan isi dari fungsi harus menjorok ke dalam 1 tab.
- kata kunci **return** berfungsi untuk mengeluarkan nilai suatu fungsi (dalam case ini mengeluarkan fungsi bagi a dan b).

Berikut ini adalah penggunaannya yang lebih lengkap.



```

1 #masukkan fungsi bagi
2 def divide(a, b):
3     ans = a / b
4     return ans
5
6 #memanggil fungsi divide pakai dua argumen
   yaitu 6 dan 28
7 meong = divide(18, 6)
8 print(meong)
9

```

Penjelasan:

- Baris 1: **komentor**, diabaikan oleh interpreter Python
- Baris 2-4: def fungsi **divide()**, tidak akan dijalankan sampai fungsinya dipanggil terlebih dahulu.
- Baris 5: kosong, untuk spasi, diabaikan Python
- Baris 7: variabel **meong** yang diisi nilai dari hasil pemanggilan fungsi **divide()**. parameter a dan b sekarang bernilai 18 dan 6 (urut) dikarenakan pemanggilan argumen yaitu 18 dan 6.
- Baris 8: **print**, yang mana menampilkan output dari **meong**. Hasilnya adalah 3.
- Urutan jalannya program 1 -: 2 - 3 - 4 - 5 - 6 - 7 - 2 - 3 - 4 - 7 - 8

MATERI 2: Return Value

Dua jenis fungsi berdasarkan hasil yang dikeluarkan:

1. Fungsi yang tidak mengembalikan nilai (**void function**). Contoh:

```

def printDuaKali(pesan):
    print(pesan)
    print(pesan)

printDuaKali("Meong Rawrr AwoOoOoOo")

```

fungsi **printDuaKali()** punya satu parameter yaitu **pesan**, dan fungsi tersebut akan mengeprint nilai variabel **pesan** sebanyak dua kali. Fungsi tersebut tidak punya nilai untuk dipakai di proses selanjutnya. Jadi, sebenarnya fungsi tersebut return (mengembalikan) value **None** kalau dipanggil pakai cara di bawah ini.

```
1 #memanggil fungsi divide pakai dua argumen
   yaitu 6 dan 28
2 meong = divide(18, 6)
3 print(meong)
4
5 #masukkan fungsi bagi
6 def divide(a, b):
7     ans = a / b
8     return ans
```

line 2 memanggil fungsi divide()
padahal belum didefinisikan

line 6 baru mendefinisikan fungsi
divide()

← Python

```
>> Traceback (most recent call last):
      File "/lib/python3.10/asyncio/futures.py", line 201, in
        raise self._exception
      File "/lib/python3.10/asyncio/tasks.py", line 232, in _
        result = coro.send(None)
      File "/lib/python3.10/_pyodide/_base.py", line 532, in
        await CodeRunner(
      File "/lib/python3.10/_pyodide/_base.py", line 355, in
        coroutine = eval(self.code, globals, locals)
      File "<exec>", line 2, in <module>
      NameError: name 'divide' is not defined
<<
```

outputnya jadi not defined

Contoh lagi, fungsi ini:

```
1 def printDuaKali(pesan):
2     print(pesan)
3     print(pesan)
4 print(printDuaKali("Meong Rawrr Awoooo"))
```

Input

← Python

```
>> Meong Rawrr Awoooo
>> Meong Rawrr Awoooo
>> None
<<
```

Output

Berbeda dengan fungsi ini:

```
1 def divide(a, b, c):  
2     ans = a / b / c  
3     return ans
```

Fungsi **divide()** yang ini butuh 3 parameter, yaitu a, b, dan c. Di dalamnya ada variabel **ans** yang isinya hasil pembagian a / b / c. Lalu terakhir, variabel **ans** dikeluarkan dengan **return** (fungsi **return** adalah untuk mengeluarkan nilai hasil dr fungsi & mengakhiri fungsi).

MATERI 3: Optional Argument dan Named Argument

Suatu fungsi bisa punya parameter bersifat opsional yang sudah punya nilai default sebelumnya. Cara untuk mendefinisikannya adalah dengan mendefinisikan nilai bawaannya terlebih dahulu. Berikut adalah contohnya.

```
1 def buySausage(sausage, price=3000):  
2     totalPrice = sausage * price  
3     return totalPrice  
4  
5 print(buySausage(5))  
6 print(buySausage(18, 8000))  
7 print(buySausage(7193, 6500))
```

← Python

```
>> 15000  
>> 144000  
>> 46754500  
<<
```

MATERI 4: Anonymous Function (Lambda)

Anyonymus function (fungsi tanpa nama) adalah fitur tambahan di Python dengan keyword **Lambda**.

Berikut adalah penggunaan fungsi `def divide()` normal.

Input:

```
lambda.py > ...
1  def divide(a, b):
2      ans = a / b
3      return ans
4
5  print(divide(56, 2))
6
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
https://aka.ms/PowerShell-Release?tag=v7.4.1
Cannot load PSReadline module. Console is running without PSRe
PS C:\uni\VSC\week4> & 'c:\Users\asus\AppData\Local\Programs\Py
debugpy-2024.0.0-win32-x64\bundled\libs\debugpy\adapter\..\de
28.0
PS C:\uni\VSC\week4>
```

Berikut adalah penggunaan fungsi def divide() dengan lambda:

Input:

```
lambda.py > ...
1  divide = lambda a, b: a / b
2  print(divide(56,2))
3
```

Output:

```
PS C:\uni\VSC\week4> c:; cd 'c:\uni\VSC\wee
code\extensions\ms-python.debugpy-2024.0.0-w
\lambda.py'
28.0
PS C:\uni\VSC\week4> |
```

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

SOAL 1: Tiga Parameter

Fungsi untuk mengecek apakah ketiga parameter memenuhi ketentuan yang diberikan:

```
no1.py > ...
1 def numberCheck(a, b, c):
2     if a != b and b != c and a != c:
3         if a + b == c or a + c == b or b + c == a:
4             return True
5     return False
6
```

- 1) Jadi pertama buat fungsinya terlebih dahulu. Saya menamai fungsi tersebut **numberCheck()** untuk mengecek angka-angka nya.
- 2) Line 2, gunakan **if** untuk mengecek apakah nilai a, b, dan c beda semua.
- 3) Line 3, gunakan **if** lagi untuk mengecek apakah kemungkinan dua angka digabungkan bisa sama dengan angka ketiga nya.
- 4) Line 4 dan 5 menggunakan **return**, **True** jika dua kondisi **if** tersebut terpenuhi, dan **False** jika dua kondisi tersebut tidak terpenuhi.

Berikut contoh input dan outputnya:

```
no1.py > ...
1 def numberCheck(a, b, c):
2     if a != b and b != c and a != c:
3         if a + b == c or a + c == b or b + c == a:
4             return True
5     return False
6
7 print(numberCheck(3,4,5))
8 print(numberCheck(6,22,28))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Microsoft Windows [Version 10.0.19041.264]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\uni\VSC\week4>no1.py

C:\uni\VSC\week4>no1.py
False
True

SOAL 2: Cek digit belakang

```
no2.py > ...
1 def cek_digit_belakang(a, b, c):
2     digitA = a % 10
3     digitB = b % 10
4     digitC = c % 10
5     return (digitA == digitB and digitA != digitC) or (digitA == digitC and digitA != digitB) or (digitB == digitC and digitB != digitA)
6
7 a = int(input("Insert the first number: "))
8 b = int(input("Insert the second number: "))
9 c = int(input("Insert the third number: "))
10
11 hasil = cek_digit_belakang(a, b, c)
12 print(hasil)
13
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Insert the first number: 123
Insert the second number: 235
Insert the third number: 435
True
PS C:\uni\VSC\week4> 
```

Penjelasan:

- 1) Line 1 membuat fungsi seperti yang disuruh yaitu cek_digit_belakang()
- 2) Line 2-4 menggunakan modulus untuk mendapatkan digit paling kanan dari setiap bilangan.
- 3) Line 5 untuk mengeluarkan nilai hasil fungsi.
- 4) Line 7-9 itu membuat input angka pertama, kedua, dan ketiga untuk pengguna agar bisa memasukkan angka-angka yang mereka ingin untuk di tes.
- 5) Line 11 kita memanggil fungsi dan line 12 kita mencetak hasilnya agar terlihat di layar.

Berikut adalah ss untuk test case nya:

```
no2.py > ...
1 def cek_digit_belakang(a, b, c):
2     digitA = a % 10
3     digitB = b % 10
4     digitC = c % 10
5     return (digitA == digitB or digitA == digitC or digitB == digitC)
6
7 a = int(input("Insert the first number: "))
8 b = int(input("Insert the second number: "))
9 c = int(input("Insert the third number: "))
10
11 hasil = cek_digit_belakang(a, b, c)
12 print(hasil)
13
14 #testcase
15 print("Test Case:")
16 print(cek_digit_belakang(30, 20, 18))
17 print(cek_digit_belakang(145, 5, 100))
18 print(cek_digit_belakang(71, 187, 18))
19 print(cek_digit_belakang(1024, 14, 94))
20 print(cek_digit_belakang(53, 8900, 658))
21
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

True
Test Case:
True
True
False
True
False
PS C:\uni\VSC\week4> 
```


SOAL 3: Konversi Suhu

```
no3.py > ...
1  celcius_to_fahrenheit = lambda c: (9/5) * c + 32
2  celcius_to_reamur = lambda c: 0.8 * c
3
4  print("Test Case:")
5  suhu1 = 100
6  jwb1 = celcius_to_fahrenheit(suhu1)
7  print(f"C = {suhu1}, F = {jwb1}")
8
9  suhu2 = 80
10 jwb2 = celcius_to_reamur(suhu2)
11 print(f"C = {suhu2}, R = {jwb2}")
12
13 suhu3 = 0
14 jwb3 = celcius_to_fahrenheit(suhu3)
15 print(f"C = {suhu3}, F = {jwb3}")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS C:\uni\VSC\week4> c:; cd 'c:\uni\VSC\week4'; & 'c:\Users\asus\AppData\Local\Microsoft\WindowsApps\Microsoft.Windows.Common-Stack-1.0.0.0-x64\50738' '--' 'C:\uni\VSC\week4\no3.py'
Test Case:
C = 100, F = 212.0
C = 80, R = 64.0
C = 0, F = 32.0
PS C:\uni\VSC\week4> 
```

Penjelasan:

- 1) Pada line 1 dan 2 saya memasukkan variabel celsius ke fahrenheit dan celcius ke reamur menggunakan fungsi lambda.
- 2) Line 4 - 15 berisi beberapa test case yang sudah sesuai dengan soal untuk membuktikan fungsinya.

Link GitHub: <https://github.com/taratirtanata/Praktikum-Alpro>