



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230978
Nama Lengkap	Jonathan Satriani Gracio Andrianto
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%)

Fungsi, Argument dan Parameter

Secara umum fungsi adalah blok kode yang digunakan untuk menjalankan tugas tertentu. Fungsi memungkinkan kita untuk mengelompokkan kode ke dalam unit yang lebih kecil, sehingga lebih mudah dibaca, dipahami, dan dikelola. Fungsi juga memungkinkan Anda menggunakan potongan kode yang sama berkali-kali tanpa harus menuliskannya ulang. Berdasarkan asalnya, fungsi dibagi dua jenis yaitu :

- Fungsi bawaan (built-in function). Daftar fungsi bawaan Python 3 dapat dilihat di <https://docs.python.org/3/library/functions.html>
- Fungsi yang dibuat oleh programmer .
Sebagai contoh perhatikan fungsi kurang() berikut ini yang dapat digunakan untuk mengurangi bilangan a dan bilangan b yang diberikan:

```
def kurang(a, b) :  
    hasil = a - b  
    return hasil
```

Fungsi kurang tersebut terdiri dari beberapa hal yang perlu diperhatikan:

- Keyword **def** digunakan untuk mendefinisikan sebuah fungsi.
- Nama fungsi yang dibuat adalah **kurang()**.
- Isi dari fungsi harus anda tuliskan menjorok ke dalam 1 tab. Perhatikan bagian tambah(a,b): sebagai penanda blok.
- Fungsi kurang() membutuhkan dua argument, yang nantinya akan dikenali sebagai parameter **a** dan **b**.
- Fungsi tersebut akan menghasilkan hasil pengurangan yang dapat ditampung di sebuah variabel. Keyword **return** digunakan untuk mengembalikan/mengeluarkan nilai dari suatu fungsi.

Penggunaan fungsi tersebut dapat dilihat pada program dibawah ini :

```
def kurang(a, b) :  
    hasil = a - b  
    return hasil  
  
c = kurang(7, 5)  
print(c)
```

Penggunaan fungsi dalam program

*variable c akan diisi dari hasil pemanggilan fungsi kurang(). Fungsi kurang() dipanggil dengan argument 7 dan 5 (sesuai dengan urutan).

2

Hasil dari pemanggilan fungsi kurang()

Return Value

Berdasarkan hasil yang dikeluarkan, fungsi terbagi menjadi dua jenis yaitu : fungsi yang mengembalikan nilai, dan fungsi yang tidak mengembalikan nilai atau **void function**. Sebagai contoh, fungsi `print_twice()` berikut ini adalah fungsi yang tidak mengembalikan nilai :

```
def print_twice(message):  
    print(message)  
    print(message)  
  
print_twice("Hello World!")
```

Contoh void function

Fungsi tersebut tidak mengembalikan nilai dikarenakan tidak ada perintah untuk menampilkan nilainya. Fungsi `print_twice()` sebenarnya mengembalikan nilai `None` jika kita coba panggil dengan cara berikut:

```
def print_twice(message):  
    print(message)  
    print(message)  
  
print(print_twice("Hello World!"))
```

Fungsi `print_twice()` yang mengembalikan nilai `none`

```
Hello World!  
Hello World!  
None
```

Output

Berbeda halnya dengan fungsi `tambah()` berikut ini:

```
def tambah(a, b, c):  
    hasil = a + b + c  
    return hasil
```

Fungsi `tambah()`

Perbedaannya dengan fungsi `print_twice()` adalah keyword **return**. Keyword `return` digunakan untuk :

- Mengeluarkan nilai yang merupakan hasil dari fungsi.
- Mengakhiri fungsi.

Optional Argument dan Named Argument

Fungsi dapat memiliki optional parameter, yaitu memiliki nilai bawaan (default) yang sudah didefinisikan sebelumnya. Untuk mendefinisikan optional parameter, anda harus mendefinisikan nilai bawaannya terlebih dahulu seperti pada contoh berikut ini :

```
def hitung_belanja(belanja, diskon=0):  
    bayar = belanja - (belanja * diskon)/100  
    return bayar
```

Fungsi yang memiliki nilai bawaan

Pada fungsi tersebut parameter default yaitu 0 (yaitu 0%). Untuk memanggil fungsi hitung_belanja() tersebut, kita dapat melakukan cara dibawah :

```
def hitung_belanja(belanja, diskon=0):  
    bayar = belanja - (belanja * diskon)/100  
    return bayar  
  
print(hitung_belanja(500000))  
print(hitung_belanja(500000, 10))  
print(hitung_belanja(500000, 50))
```

Pemanggilan fungsi hitung_belanja()

Pemanggilan pertama menggunakan pertama hanya menggunakan satu argument, sedangkan yang kedua dan ketiga menggunakan dua argument.

Output yang dihasilkan adalah :

```
500000.0  
450000.0  
250000.0
```

Output

Setiap parameter pada fungsi memiliki nama. Karena itu pada pemanggilan fungsi juga dapat disertakan nama parameternya dan tidak perlu sesuai dengan urutan yang diberikan. Perhatikan contoh berikut ini :

```
def angka (a, b, c):  
    print("Nilai a: ",a)  
    print("Nilai b: ",b)  
    print("Nilai c: ",c)  
  
angka(b=20, a=30, c=40)
```

Fungsi angka()

```
500000.0
450000.0
250000.0
```

Output

Anonymous Function (Lambda)

Sesuai dengan namanya, anonymous function adalah fungsi tanpa nama. Anonymous function pada Python adalah fitur tambahan, bukan merupakan fitur utama. Sebagai contoh, kita akan menerapkan lambda pada fungsi tambah() :

```
tambah = lambda a, b: a + b
print(tambah(10,20))
```

Penerapan lambda pada fungsi tambah()

Setiap anonymous function pada Python terdiri dari beberapa bagian berikut ini :

- Keyword: lambda
- Bound variable: argument pada lambda function
- Body: bagian utama lambda, berisi ekspresi atau statement yang menghasilkan suatu nilai

Anonymous/Lambda Function

Perhatikan fungsi yang dapat menentukan apakah suatu bilangan merupakan kelipatan 5 atau tidak berikut ini :

```
def kelipatan_lima(angka):
    if angka % 5 == 0:
        return True
    else:
        return False
```

Fungsi kelipatan_lima()

Bentuk lambdanya :

```
kelipatan_lima = lambda angka: angka % 5 == 0
```

Penerapan lambda pada fungsi kelipatan_lima()

Contoh penggunaan lambda function :

```
kelipatan_lima = lambda angka: angka % 5 == 0  
print(kelipatan_sembilan(25))  
print(kelipatan_sembilan(54))
```

Contoh penggunaan lambda function

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

Source Code

```
def cek_angka(a, b, c) :  
    if (a != b != c) and (a + b == c or a + c == b or b + c == a) :  
        return "True"  
    else :  
        return "False"  
  
print(cek_angka(5, 3, 8))  
print(cek_angka(6, 5, 9))
```

Output

```
True  
False
```

Penjelasan

def cek_angka(a, b, c) : ➔ mendefinisikan fungsi

if (a != b != c) ➔ kondisi Ketiga parameter tersebut nilainya berbeda semua.

and ➔ operasi logika

(a + b == c or a + c == b or b + c == a) : ➔ Ada kemungkinan jika diambil dua parameter dan dijumlahkan hasilnya sama dengan parameter lainnya (yang tersisa). Fungsi tersebut akan menghasilkan nilai

return "True" ➔ nilai jika semua kondisi terpenuhi

else :

return "False" ➔ nilai jika selain kondisi tidak terpenuhi

print(cek_angka(5, 3, 8)) ➔ menampilkan output dari test case fungsi, yaitu a = 5, b = 3 dan c = 8

print(cek_angka(6, 5, 9)) ➔ menampilkan output dari test case fungsi, yaitu a = 6, b = 5 dan c = 9

SOAL 2

Source Code

```
def cek_digit_belakang(a, b, c) :  
    if (a % 10 == b % 10) or (a % 10 == c % 10) or (b % 10 == c % 10) :  
        return "True"  
    else :  
        return "False"  
  
a = int(input("Masukkan angka pertama = "))  
b = int(input("Masukkan angka kedua = "))  
c = int(input("Masukkan angka ketiga = "))  
  
print(cek_digit_belakang(a, b, c))
```

Output

```
Masukkan angka pertama = 20  
Masukkan angka kedua = 15  
Masukkan angka ketiga = 5  
True
```

Penjelasan

def cek_digit_belakang(a, b, c) : → mendefinisikan fungsi

if (a % 10 == b % 10) or (a % 10 == c % 10) or (b % 10 == c % 10) : → kondisi jika minimal dua dari tiga parameter yang diberikan memiliki digit paling kanan yang sama. Kita gunakan logical operation "or" karena kondisinya "minimal dua dari tiga", bukan semuanya.

return "True" → nilai jika kondisi terpenuhi

else :

return "False" → nilai jika kondisi tidak terpenuhi

a = int(input("Masukkan angka pertama = ")) → meminta input user untuk parameter a

b = int(input("Masukkan angka kedua = ")) → meminta input user untuk parameter b

c = int(input("Masukkan angka ketiga = ")) → meminta input user untuk parameter c

print(cek_digit_belakang(a, b, c)) → menampilkan nilai dari fungsi cek_digit_belakang()

SOAL 3

Soruce Code

```
celcius_to_fahrenheit = lambda c : (9/5) * c + 32
celcius_to_reamur = lambda c : (0.8) * c

hasil_1 = int(celcius_to_fahrenheit(100))
print("F = ", hasil_1)

hasil_2 = int(celcius_to_reamur(80))
print("R = ", hasil_2)
```

Output

```
F = 212
R = 64
```

Penjelasan

`celcius_to_fahrenheit = lambda c : (9/5) * c + 32` → fungsi lambda dari konversi celcius ke fahrenheit

`celcius_to_reamur = lambda c : (0.8) * c` → fungsi lambda dari konversi celcius ke reamur

`hasil_1 = int(celcius_to_fahrenheit(100))` → hasil dari test case 1 dengan parameter celcius = 100

`print("F = ", hasil_1)` → menampilkan hasil dari test case 1

`hasil_2 = int(celcius_to_reamur(80))` → hasil dari test case 2 dengan parameter celcius = 80

`print("R = ", hasil_2)` → menampilkan hasil dari test case 2