



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71230976</b>
<b>Nama Lengkap</b>	<b>Galih Pramana Chandra Prasetya</b>
<b>Minggu ke / Materi</b>	<b>04 / Modular Programming</b>

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.

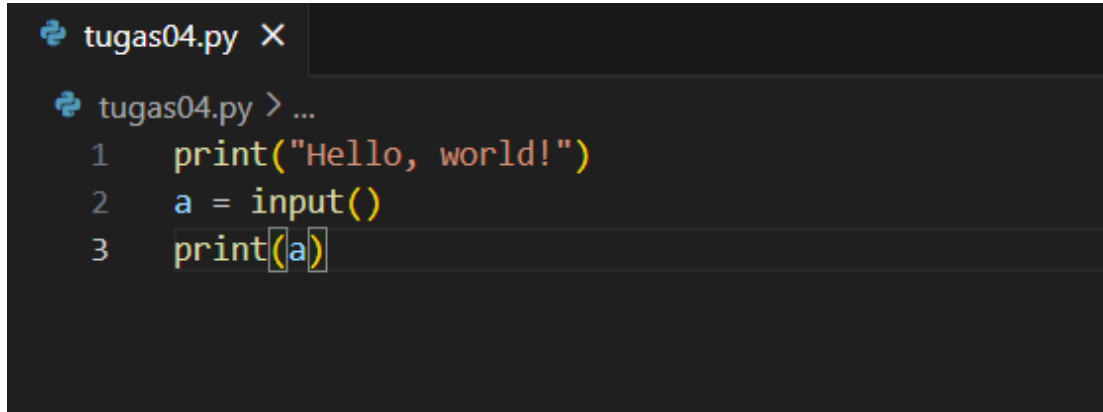
SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
YOGYAKARTA  
2024

# Modular Programming

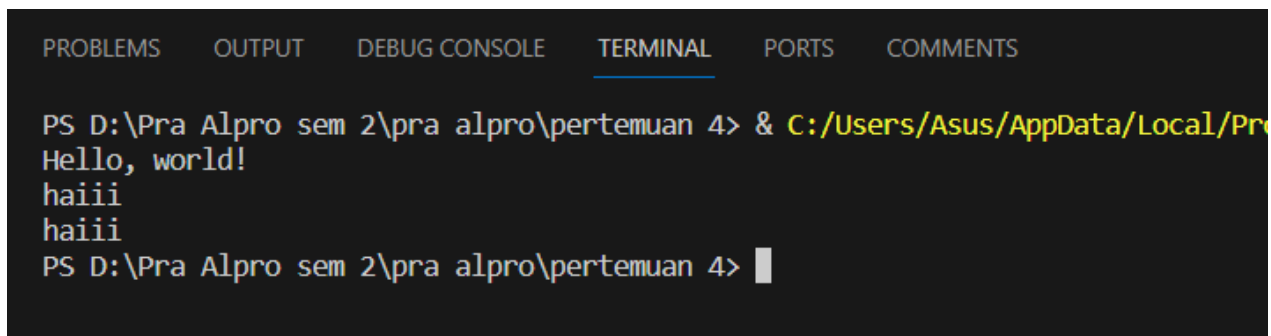
## Fungsi, Argument dan Parameter

Fungsi adalah kumpulan perintah-perintah yang dijadikan satu, memiliki suatu tujuan dan kegunaan khusus serta dapat digunakan ulang. Fungsi merupakan fitur bawaan dari Python itu sendiri. Setiap fungsi memiliki kegunaan masing-masing. Berikut ini adalah contoh penggunaan fungsi **Print** dan fungsi **Input**.



```
tugas04.py X
tugas04.py > ...
1  print("Hello, world!")
2  a = input()
3  print(a)
```

Gambar 1. Fungsi input & Fungsi print



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS D:\Pra Alpro sem 2\pra alpro\pertemuan 4> & C:/Users/Asus/AppData/Local/Pr
Hello, world!
haiii
haiii
PS D:\Pra Alpro sem 2\pra alpro\pertemuan 4> █
```

Gambar 2. Outputl fungsi input & fungsi print

Fungsi dibagi menjadi dua jenis yaitu:

- Fungsi bawaan (built-in function).
- Fungsi yang dibuat sendiri oleh programmer.

Beberapa hal yang perlu diperhatikan saat membuat fungsi:

- Keyword **def** digunakan untuk mendefinisikan sebuah fungsi.
- Memberi nama pada fungsi.
- Isi dari fungsi harus anda tuliskan menjorok ke dalam 1 tab.

- Untuk menampilkan output dapat dilakukan return, print(), maupun memanggil dan print fungsi.

Berikut ini adalah contoh pembuatan fungsi,

```

9  def lingkaran(jarijari, pi= 3.14):
10     luas = 0.5 * pi * jarijari**2
11     return luas
12     print(lingkaran(5,3.14))

```

Gambar 3. Contoh Fungsi Menghitung luas lingkaran

## Return Value

Berdasarkan hasil yang dikeluarkan oleh fungsi, ada dua jenis yaitu: (1) fungsi yang tidak mengembalikan nilai dan (2) fungsi yang mengembalikan nilai.

1. Fungsi yang tidak mengembalikan nilai sering disebut sebagai **void function**.

```

5  def print_twice(message):
6     print(message)
7     print(message)
8
9  print(print_twice("Hello World!"))

```

Gambar 4. Contoh Fungsi Tidak Mengembalikan Nilai (Void Function) .

```

PS D:\Pra Alpro sem 2\pra alpro\pertemuan 4> & C:/Users/Asus/AppData/Local/Prog
Hello World!
Hello World!
None

```

Gambar 3. Output Fungsi Tidak Mengembalikan Nilai (Void Function)

Fungsi print\_twice() tidak menghasilkan suatu nilai yang dapat digunakan untuk proses berikutnya. Fungsi print\_twice() akan mengembalikan nilai **None**.

2. Fungsi yang mengembalikan nilai (void function)

Untuk membuat fungsi yang dapat mengembalikan nilai perlu menambahkan perintah **return**. Keyword **return** digunakan untuk mengeluarkan nilai yang merupakan hasil dari fungsi mengakhiri fungsi.

```

26  def luas( sisi1,sisi2):
27      luas = sisi1 * sisi2
28      return luas
29
30  print(luas(5,5))
31

```

```

PS D:\Pra Alpro sem 2\pra alpro\pertemuan 4> & C:/Users/Asus/AppData/Local/Pro
25

```

Gambar 6. Contoh Fungsi Mengembalikan Nilai

## Optional Argument dan Named Argument

Optional parameter, adalah parameter yang bersifat opsional dan memiliki nilai bawaan (default) yang sudah didefinisikan sebelumnya. Untuk mendefinisikan optional parameter, nilai perlu didefinisikan bawaannya terlebih dahulu .

```

11  def hitung_belanja(belanja, diskon=0):
12      bayar = belanja- (belanja * diskon)/100
13      return bayar
14
15  print(hitung_belanja(100000))
16  print(hitung_belanja(100000, 10))
17  print(hitung_belanja(100000, 50))

```

Gambar 7. Contoh Optional Argument

Komputer akan menghitung secara berurutan sesuai program yang tertulis.

Named argument dapat membuat computer membaca nilai-nilai yang ada secara tidak berurutan. Kita dapat memasukkan nilai-nilai parameter yang dibutuhkan sebuah fungsi dengan menggunakan nama parameternya. Dengan menggunakan named argumen, kita dapat memasukkan nilai yang dibutuhkan dengan urutan bebas, aturannya adalah :

- nilai-nilai harus diberi nama sesuai parameter-parameter yang ada.
- menggunakan tanda koma ( , ) sebagai pemisah.

```

9  def lingkaran(jarijari, pi= 3.14):
10      luas = 0.5 * pi * jarijari**2
11      return luas
12
13  print(lingkaran(5,3.14))
14  print(lingkaran(pi = 3.14, jarijari= 5))

```

Gambar 8. Contoh Named Argument

*Gambar 9. Output Named Argument*

## Anonymous Function (Lambda)

1. Keyword: lambda
2. Bound variable: variabel berisikan fungsi lambda
3. Body: berisikan ekspresi atau statement untuk menjalankan fungsi lambda
4. Input: untuk membaca nilai-nilai dari pengguna (jika diperlukan)
5. Print: untuk menampilkan hasil dari fungsi lambda

```

PS D:\Pra Alpro sem 2\pra alpro\pertemuan 4> & C:/Users/Asus/AppData/Local/Programs/Python/Py
input x: 5
input y: 6
pilih operator: *
30.0

```

Gambar 10. Program kalkulator tanpa menggunakan lambda

Diatas adalah contoh program kalkulator jika tidak menggunakan lambda.

```

19  calculator = {
20      "+": lambda x, y: x + y,
21      "-": lambda x, y: x - y,
22      "/": lambda x, y: x / y,
23      "*": lambda x, y: x * y
24  }
25
26  x = float(input("masukkan x: "))
27  y = float(input("masukkan y: "))
28  op = input("masukkan op: ")
29
30  print(calculator[op](x,y))

```

```

PS D:\Pra Alpro sem 2\pra alpro\pertemuan 4> & C:/Users/Asus/AppData/Local/Programs/Python/Pyth
masukkan x:5
masukkan y:6
masukkan op:*
30.0

```

Gambar 11. Program kalkulator menggunakan lambda

Di atas adalah contoh penggunaan lambda pada program kalkulator. Dalam implementasi kalkulator ini, fungsi lambda digunakan tanpa memerlukan percabangan. Sebagai gantinya, kita dapat mendefinisikan operator diikuti oleh lambda, nilai-nilai, dan rumus dalam bentuk array. Pengguna dapat memasukkan nilai-nilai dan operator yang diinginkan melalui fungsi input. Akhirnya, fungsi dipanggil dengan perintah print, menampilkan nama fungsi lambda beserta parameter-parameternya..

## 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:

```
#Latihan 4.1
def cek_angka(a,b,c):
    if a != b and a != c and b != c and ( a + b == c or a + c == b or b + c == a):
        return True
    else:
        return False

print(cek_angka(5,3,2))
print(cek_angka(1,2,3))
print(cek_angka(4,5,4))
```

Output:

```
True
True
False
```

Penjelasan:

- Fungsi cek\_angka didefinisikan dengan tiga parameter: a, b, dan c.
- Pada blok if, ada beberapa kondisi yang harus dipenuhi agar fungsi mengembalikan True:
  - a tidak boleh sama dengan b, dan
  - a tidak boleh sama dengan c, dan
  - b tidak boleh sama dengan c, dan
  - Salah satu dari tiga kemungkinan penjumlahan harus benar, yaitu:  $a + b == c$ , atau  $a + c == b$ , atau  $b + c == a$ .
- Jika semua dari kondisi-kondisi tersebut terpenuhi, maka fungsi mengembalikan "True", yang berarti tiga angka tersebut memenuhi kriteria yang ditetapkan. Jika tidak, fungsi mengembalikan "False".

## SOAL 2

Source code:

```
#Latihan 4.2

def cek_digit_belakang(a,b,c):
    a = a%10
    b = b%10
    c = c%10
    if (a == b or a == c or b == c):
        return True
    else:
        return False

inputA =int(input("Masukkan bilangan a: "))
inputB =int(input("Masukkan bilangan b: "))
inputC =int(input("Masukkan bilangan c: "))

print("Hasil:", cek_digit_belakang(inputA,inputB,inputC))
```

Output:

```
➞ Masukkan bilangan a: 30
    Masukkan bilangan b: 20
    Masukkan bilangan c: 18
    Hasil: True
```


Penjelasan: Program ini memiliki tujuan untuk memeriksa apakah setidaknya dua dari tiga bilangan yang dimasukkan memiliki digit belakang yang sama.

- Fungsi `cek_digit_belakang(a, b, c)` mengambil tiga bilangan. Dengan menggunakan operasi modulo ( $\% 10$ ) akan memperoleh digit belakang masing-masing. Kemudian, fungsi memeriksa apakah setidaknya dua dari tiga digit belakang yang diperoleh sama. Jika ya, fungsi mengembalikan `True`, jika tidak, mengembalikan `False`.
- Program kemudian meminta pengguna untuk memasukkan tiga bilangan (a, b, c) menggunakan fungsi `input`.
- Hasil dari pemanggilan fungsi `cek_digit_belakang(inputA, inputB, inputC)` dicetak dengan pesan "Hasil:", yang menunjukkan apakah setidaknya dua digit belakang dari tiga bilangan tersebut sama atau tidak.




### SOAL 3

Source code:

```
 #Latihan 4.3
celcius_to_fahrenheit = lambda C: (9/5*C)+32
celcius_to_reamur = lambda C:0.8*C

print(celcius_to_fahrenheit(100))
print(celcius_to_reamur(80))
print(celcius_to_fahrenheit(0))
```

Output:

```
 212.0
  64.0
  32.0
```

Penjelasan: Program ini menggunakan fungsi lambda untuk mengonversi suhu dari Celsius ke Fahrenheit dan Reamur

Dua fungsi lambda didefinisikan:

- celcius\_to\_fahrenheit: Mengonversi suhu dari Celsius ke Fahrenheit menggunakan rumus :  
 **$(9/5 * C) + 32$ .**
- celcius\_to\_reamur : Mengonversi suhu dari Celsius ke Reamur menggunakan rumus:  
 **$0.8 * C$ .**