



Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2024/2025

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.

NIM	<71241079>
Nama Lengkap	<Natalie Neysa Jessica Soesanto>
Minggu ke / Materi	05 / Modular Programming

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2025

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

Input :

```
1  nama = input("Masukkan nama: ")
2  print("Halo ", nama, " selamat pagi!")
```

Output :

```
• Masukkan nama: Jeje
  Hallo Jeje selamat pagi!
• PS C:\Users\ASUS\OneDrive\Desk
```

Program tersebut merupakan program yang meminta input nama pengguna lalu menyapa nama yang diinputkan oleh pengguna. Pada program ini kita menggunakan 2 fungsi bawaan dari Python, yaitu **input()** untuk membaca input yang dimasukkan oleh pengguna sedangkan **print()** untuk menampilkan fungsi di layar.

Fungsi merupakan kumpulan perintah yang digabungkan menjadi satu yang memiliki tujuan dan kegunaan khusus yang bisa digunakan ulang. Jika kita ingin membuat program yang membutuhkan langkah yang banyak, maka perlu untuk mengelompokkan beberapa kode program menjadi bagian-bagian (block). Karena itu disebut sebagai modular.

Input :

```
1  def tambah(a, b):
2      |      hasil = a + b
3      |      return hasil
4  print (tambah(10, 5))
```

Output :

```
● PS C:\User
"
15
○ PS C:\User
```

Pada fungsi di atas, “def” didefinisikan sebuah fungsi. Nama fungsi yang dibuat adalah tambah(). Setelah menuliskan “def” dan nama fungsi, isi dituliskan menjorok ke dalam dengan jarak 1 tab. Ini agar tidak terjadi eror. Pada fungsi ini membutuhkan minimal 2 parameter yang nantinya bisa disebut dengan parameter a dan b. Kita bisa menggunakan huruf apapun untuk parameternya. Lalu ada keyword return yang berfungsi untuk mengembalikan/mengeluarkan nilai dari suatu fungsi.

Kita menggunakan angka untuk mendefinisikan fungsi. Jika kita belum mendefinisikan fungsi maka akan terjadi kesalahan. Untuk memanggil fungsi di atas, kita bisa menggunakan **print(nama fungsi(parameter1, parameter2))**. Selanjutnya, kita dapat melihat output yang ditampilkan program. Parameter harus dituliskan secara berurutan. Parameter akan mengikuti parameter mana yang ditulis pertama. Seperti itulah ilustrasi pertama pada modular programming ini.

MATERI 2

Return Value

Berdasarkan hasil yang dikeluarkan, secara umum fungsi ada 2 jenis yaitu fungsi yang mengembalikan nilai dan fungsi yang tidak mengembalikan nilai. Fungsi yang tidak mengembalikan nilai disebut void function.

Contoh fungsi yang tidak mengembalikan nilai :

Input :

```
1 def print_twice(message):
2     print(message)
3     print(message)
4 print_twice("Hello World!")
```

Output :

```
● PS C:\Users\ASUS
"
Hello World!
Hello World!
○ PS C:\Users\ASUS
```

Pada fungsi `print_twice()`, hanya membutuhkan 1 parameter yaitu `message`. Di dalam fungsi `print_twice()` tersebut akan menampilkan "Hello World!" sebanyak 2 kali. Kita cukup menulis nama fungsi untuk memanggil variable tersebut lalu memasukkan parameter yang kita inginkan sesuai pada code di atas (Misal : "Hello World!").

Contoh kasus lain untuk return value :

Input :

```
1  def tambah(a, b, c):
2      |      hasil = a + b + c
3      |      return hasil
4
5  nilai1 = 70
6  nilai2 = 85
7  nilai3 = 55
8
9  rata_rata = tambah(nilai1, nilai2, nilai3)/3
10 print(rata_rata)
```

Output :

```
● PS C:\Users
"
70.0
○ PS C:\Users
```

Pada fungsi `tambah()` membutuhkan 3 parameter yaitu `a`, `b`, dan `c`. Kemudian di dalam fungsi `tambah()` didefinisikan suatu variabel `hasil` yang diisi oleh `a + b + c`. Lalu menggunakan keyword "return" untuk mengeluarkan variabel tersebut dari fungsi sebagai hasil dari fungsi `tambah()`. Keyword `return` digunakan untuk mengeluarkan nilai yang merupakan hasil dari fungsi dan mengakhiri fungsi. Fungsi `tambah()` tersebut bisa digunakan untuk mencari hasil penjumlahan tiga bilangan jika kita ingin menjumlahkan lebih dari itu maka kita bisa menambahkan variabel lagi di source codenya.

MATERI 3

Optional Argument dan Named Argument

Fungsi dapat memiliki optional parameter, yaitu parameter yang bersifat opsional dan memiliki nilai bawaan (default) yang sudah didefinisikan sebelumnya. Untuk mendefinisikan optional parameter, kita harus mendefinisikan nilai bawaannya terlebih dahulu.

Contoh :

Input :

```
1 def hitung_belanja(belanja, diskon=0):
2     bayar = belanja - (belanja * diskon)/100
3     return bayar
4
5 print(hitung_belanja(100000))
6 print(hitung_belanja(100000, 10))
7 print(hitung_belanja(100000, 50))
```

Output :

```
PS C:\Users\AS
"
100000.0
90000.0
50000.0
PS C:\Users\AS
```

Fungsi `hitung_belanja()` ini menggunakan 2 parameter, yaitu “belanja” dan “diskon”. Fungsi ini berguna untuk menghitung total belanja. Pada pemanggilan pertama menggunakan satu argument dan pemanggilan kedua serta ketiga menggunakan dua argument. Untuk menjalankan fungsi `hitung_belanja()` kita dapat memasukkan rumus **`bayar = belanja - (belanja * diskon) / 100`**, lalu direturn.

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

Contoh :

Input :

```

1  def cetak(a, b, c):
2      print("Nilai a: ",a)
3      print("Nilai b: ",b)
4      print("Nilai c: ",c)
5  cetak(20, 30, 40)

```

Output :

```

• Nilai a:  20
  Nilai b:  30
  Nilai c:  40
○ PS C:\Users\ASUS

```

Kita harus memastikan urutan argument harus sama pada source di atas, nah agar urutan argument tidak harus sama dengan urutan parameter maka dapat menggunakan source code berikut :

Input :

```

1  def cetak(a, b, c):
2      print("Nilai a: ",a)
3      print("Nilai b: ",b)
4      print("Nilai c: ",c)
5  cetak(b=30, c=40, a=20)

```

Output :

```

• Nilai a:  20
  Nilai b:  30
  Nilai c:  40
○ PS C:\Users\ASUS\O

```

MATERI 4

Anonymous Function (Lambda)

Anonymous function adalah fungsi tanpa nama (anonymous). Anonymous function pada Python adalah fitur tambahan, bukan merupakan fitur utama. Berbeda dengan bahasa – bahasa pemrograman fungsional. Pada Python, digunakan keyword lambda untuk mendefinisikan anonymous function. Sebagai contoh, perhatikan fungsi tambah() berikut :

Tidak Menggunakan Lambda

Input :

```
1 def tambah(a, b):  
2     hasil = a + b  
3     return hasil  
4 print(tambah(10,20))
```

Output :

```
PS C:\User  
"  
● 30  
○ PS C:\User
```

Menggunakan Lambda

Input :

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

Output :

```
● PS C:\User  
"  
30  
○ PS C:\User
```

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

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

Input :

```
1 def cek_angka(a, b, c):
2     if a != b and a != c and b != c:
3         if a + b == c or a + c == b or b + c == a:
4             return True
5     return False
6
7 print(cek_angka(1, 2, 3))
8 print(cek_angka(5, 8, 9))
```

Output :

```
● True
  False
○ PS C:\Users\A
```

Penjelasan :

Nama fungsi yang digunakan yaitu `cek_angka()`, fungsi tersebut mempunyai 3 parameter yaitu `a`, `b`, dan `c`. Pada kondisi pertama, dinyatakan ketiga parameter tersebut berbeda semua, jika ada yang sama maka hasilnya **FALSE**. Lalu pada kondisi kedua jika dua angka dijumlahkan hasilnya adalah angka yang tersisa maka **TRUE**. Jika tidak ada yang sesuai hasil penjumlahannya maka akan menghasilkan **FALSE**. Untuk memanggil dapat menggunakan `print(nama fungsi(parameter))`.

SOAL 2

Input :


```

1  def cek_digit_belakang(a, b, c):
2      k1 = a % 10
3      k2 = b % 10
4      k3 = c % 10
5
6      if k1 == k2 or k1 == k3 or k2 == k3:
7          return True
8      else:
9          return False
10
11  try:
12      a = int(input("Masukkan a: "))
13      b = int(input("Masukkan b: "))
14      c = int(input("Masukkan c: "))
15  except:
16      print("Format yang anda masukkan salah!")
17
18  print(cek_digit_belakang(a, b, c))

```

Output:

```

• PS C:\Users\ASUS\OneDrive\Dokumen\Sl
  .py"
  Masukkan a: 30
  Masukkan b: 20
  Masukkan c: 18
  True
• PS C:\Users\ASUS\OneDrive\Dokumen\Sl
  .py"
  Masukkan a: 145
  Masukkan b: 5
  Masukkan c: 100
  True
• PS C:\Users\ASUS\OneDrive\Dokumen\Sl
  .py"
  Masukkan a: 71
  Masukkan b: 187
  Masukkan c: 18
  False
• PS C:\Users\ASUS\OneDrive\Dokumen\Sl
  .py"
  Masukkan a: 1024
  Masukkan b: 14
  Masukkan c: 94
  True
• PS C:\Users\ASUS\OneDrive\Dokumen\Sl
  .py"
  Masukkan a: 53
  Masukkan b: 8900
  Masukkan c: 658
  False
○ PS C:\Users\ASUS\OneDrive\Dokumen\Sl

```

Penjelasan :

Nama fungsi yang digunakan yaitu cek_digit_belakang(), fungsi tersebut memiliki 3 parameter yaitu a, b, dan c. Untuk menghasilkan angka paling kanan belakang menggunakan modulus 10 (% 10). Maka dibuatlah 3 variabel yaitu k1, k2, k3 untuk menghasilkan angka paling kanan belakang. Lalu membuat kondisi digit paling kanan belakang harus ada yang sama dengan digit kanan belakang parameter lainnya setidaknya 1 yang sama. Jika tidak ada digit paling kanan yang sama maka akan menghasilkan output **FALSE**. Terakhir membuat fungsi untuk input dari pengguna.

SOAL 3

Input :

```
1  ctof = lambda c: int((9/5) * c + 32)
2  ctor = lambda c: int(0.8 * c)
3
4  c1 = 100
5  fahrenheit1 = ctof(c1)
6  print(f"Input C = {c1}. Output F = {fahrenheit1}.")
7
8  c2 = 80
9  reamur = ctor(c2)
10 print(f"Input C = {c2}. Output F = {reamur}.")
11
12 c3 = 0
13 fahrenheit3 = ctof(c3)
14 print(f"Input C = {c3}. Output F = {fahrenheit3}.")
```

Output :

```
● PS C:\Users\ASUS\OneDrive\Dokume
.py"
Input C = 100. Output F = 212.
Input C = 80. Output F = 64.
Input C = 0. Output F = 32.
○ PS C:\Users\ASUS\OneDrive\Dokume
```

Penjelasan :

Pada lambda function kita tuliskan nama fungsinya masing – masing dahulu yaitu ctof (celcius_to_fahrenheit) lalu masukkan rumus sesuai soal dan ctor (celcius_to_reamur) lalu masukkan rumus sesuai soal. Untuk pemanggilan fungsi lambda tersebut kita membuat variabel baru yaitu: c1, c2, c3. Terakhir untuk melakukan pemanggilan kita lakukan print() dan pastikan output sama dengan pernyataan di soal.

Link Github : <https://github.com/babydoll-05/Laprak-5.git>