

Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



NIM	71200645
Nama Lengkap	Edith Felicia Putri
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
2023

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.

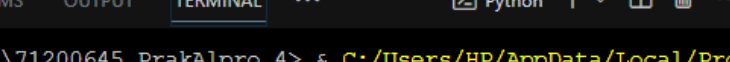
MATERI 1

FUNGSI, ARGUMENT, DAN PARAMETER

```
nama = input("Masukkan nama: ")
hobi = input("Masukkan hobi: ")
print("Hallo! Saya", nama, "\nHobi saya ", hobi, "\nSelamat pagi!")
```

Program tersebut meminta pengguna untuk memasukkan nama dan hobi, kemudian program akan menyapa nama yang diinputkan oleh pengguna. Fungsi yang digunakan adalah bawaan dari Python (built-in function), yaitu `input()` dan `print()`. Fungsi `input()` digunakan untuk membaca input yang diberikan oleh pengguna, sedangkan fungsi `print()` digunakan untuk menampilkan tulisan di layar.

OUTPUT:



```
PS D:\71200645_PrakAlpro_4> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_4/modul.py
Masukkan nama: Mona
Masukkan hobi: menyanyi
Hallo! Saya Mona
Hobi saya menyanyi
Selamat pagi!
PS D:\71200645_PrakAlpro_4>
```

Fungsi

- ⇒ Kumpulan perintah-perintah yang dijadikan satu, memiliki suatu tujuan dan kegunaan khusus serta dapat digunakan ulang.

Modular

- ⇒ Program yang membutuhkan langkah yang banyak, dimana perlu mengelompokkan beberapa kode program menjadi bagian-bagian (block) dari suatu program yang besar.
- ⇒ Program terdiri dari beberapa bagian modular yang memiliki kegunaan khusus dan dapat digunakan ulang.

Berdasarkan asalnya, fungsi dibagi menjadi dua jenis, yaitu:

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

SOURCE CODE:

```
def kurang(a, b):  
    hasil = a - b  
    return hasil  
  
c = kurang(20, 15)  
print(c)
```

Fungsi kurang terdiri dari beberapa hal:

- Keyword **def** (mendefinisikan sebuah fungsi).
- Nama fungsi, yaitu **kurang()**.
- Isi dari fungsi harus menjorok ke dalam 1 tab.
- Fungsi **kurang()** membutuhkan dua argument, yang dikenal sebagai parameter a dan b.
- Fungsi tersebut akan menghasilkan hasil pengurangan yang dapat ditampung di sebuah variabel **c**. Keyword **return** digunakan untuk mengembalikan/mengeluarkan nilai dari suatu fungsi.

Return Value

Berdasarkan hasil yang dikeluarkan oleh fungsi, secara umum ada dua jenis yaitu: (1) fungsi yang tidak mengembalikan nilai dan (2) fungsi yang mengembalikan nilai. Fungsi yang tidak mengembalikan nilai sering disebut sebagai void function. Sebagai contoh, fungsi `print_twice()` berikut ini adalah fungsi yang tidak mengembalikan nilai:

SOURCE CODE:

```
def print_twice(message):  
    print(message)  
    print(message)  
    return message  
  
print(print_twice("Sin is unbelief in Jesus Christ"))
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_4> & C:/Users/HP/AppData/Local/Programs  
/Python/Python311/python.exe d:/71200645_PrakAlpro_4/modul.py  
Sin is unbelief in Jesus Christ  
Sin is unbelief in Jesus Christ  
Sin is unbelief in Jesus Christ  
PS D:\71200645_PrakAlpro_4> □
```

Fungsi `print_twice()` membutuhkan 1 parameter yaitu `message`. Kemudian fungsi `print_twice()` akan menampilkan nilai dari variabel `message` sebanyak 2 kali. Lalu jika ditambah dengan `return message`, maka akan menghasilkan `message` sekali lagi.

MATERI 2

OPTIONAL ARGUMENT DAN NAMED ARGUMENT

Fungsi dapat memiliki *optional parameter*, yaitu parameter yang bersifat opsional dan memiliki nilai bawaan (*default*) yang sudah didefinisikan sebelumnya.

```
def hitung_belanja(belanja, diskon=0):  
    bayar = int(belanja - (belanja * diskon)/100)  
    return bayar  
  
print(hitung_belanja(100000))  
print(hitung_belanja(100000, 10))  
print(hitung_belanja(100000, 50))
```

Fungsi `hitung_belanja()` memiliki dua parameter yaitu `belanja` dan `diskon`. Parameter `diskon` secara default memiliki nilai 0 (yang artinya 0%).

OUTPUT:

```
PS D:\71200645_PrakAlpro_4> & C:/Users/HP/AppData/Local/Programs/  
Python/Python311/python.exe d:/71200645_PrakAlpro_4/modul.py  
100000  
90000  
50000  
PS D:\71200645_PrakAlpro_4> █
```

Pemanggilan pertama hanya menggunakan satu *argument*, sedangkan pemanggilan kedua dan ketiga menggunakan dua *argument*. Untuk mengubah nilai `diskon`, selain dari nilai default. Setiap parameter pada fungsi memiliki nama. Karena itu pada pemanggilan fungsi juga dapat disertakan nama parameternya dan tidak perlu sesuai dengan urutan yang diberikan.

SOURCE CODE:

```
print(hitung_belanja(diskon = 20, belanja = 120000))  
print(hitung_belanja(120000, 20))
```

OUTPUT:

```
96000  
96000  
PS D:\71200645_PrakAlpro_4>
```

MATERI 3

ANONYMOUS FUNCTION (LAMBDA)

Anonymous function adalah fungsi tanpa nama (*anonymous*). Berupa fitur tambahan pada Python dan bukan fitur utama. Berbeda dengan bahasa-bahasa pemrograman seperti Haskell, Lisp dan Erlang yang merupakan bahasa pemrograman fungsional. Digunakan keyword *lambda* untuk mendefinisikan *anonymous function*. Sebagai contoh, perhatikan fungsi kurang() berikut ini:

SOURCE CODE:

```
kurang = lambda a, b: a - b
print(kurang(15, 12))
```

Fungsi di atas menghasilkan nilai 3 dari hasil pengurangan 15 dan 12.

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.

Mendefinisikan Fungsi.

Menghitung tagihan listrik (pasca bayar).

```
def tagihan_listrik(pemakaian, golongan=3):
    bayar = 0
    pemakaian_100 = 100 if pemakaian > 100 else pemakaian
    pemakaian_100_lebih = pemakaian - pemakaian_100
    if golongan == 1:
        bayar = pemakaian_100 * 1500 + pemakaian_100_lebih * 2000
    elif golongan == 2:
        bayar = pemakaian_100 * 2500 + pemakaian_100_lebih * 3000
    elif golongan == 3:
        bayar = pemakaian_100 * 4000 + pemakaian_100_lebih * 5000
    elif golongan == 4:
        bayar = pemakaian_100 * 5000 + pemakaian_100_lebih * 7000
    return bayar

print(tagihan_listrik(130))
print(tagihan_listrik(80,4))
print(tagihan_listrik(golongan=1, pemakaian=175))
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_4> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_4/modul.py
550000
400000
300000
PS D:\71200645_PrakAlpro_4>
```

Argument dan Parameter.

Pengiriman parameter pada Python menggunakan prinsip yang disebut sebagai *Call-by-Object*. Perlakuan terhadap parameter tergantung dari apakah argument yang diberikan *immutable* atau tidak. Jika argument yang dikirimkan bersifat *immutable* artinya argument tersebut tidak bisa diubah oleh fungsi. Tipe data seperti integer, string dan tuple bersifat *immutable*.

```
def abc(a, b, c):
    a = b * c
    b = c // a
    c = a - b

angka_1 = 60
angka_2 = 100
angka_3 = 85
abc(angka_1, angka_2, angka_3)
print(angka_1)
print(angka_2)
print(angka_3)
```

Pada fungsi abc() dikirimkan 3 argument integer, yaitu angka_1, angka_2, angka_3. Karena itu dapat dikatakan argument yang dikirimkan akan bersifat immutable, sehingga tidak akan terpengaruh nilainya oleh perubahan nilai parameter dalam fungsi abc()

OUTPUT:

```
PS D:\71200645_PrakAlpro_4> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_4/modul.py
60
100
85
PS D:\71200645_PrakAlpro_4>
```

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

Latihan 4.1 Buatlah sebuah fungsi yang dapat menentukan apakah ketiga parameter memenuhi semua ketentuan berikut ini:

- Ketiga parameter tersebut nilainya berbeda semua.
- Ada kemungkinan jika diambil dua parameter dan dijumlahkan hasilnya sama dengan parameter lainnya (yang tersisa).

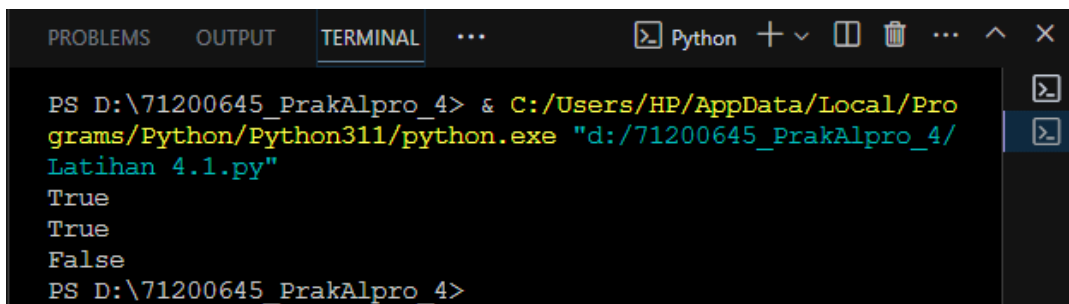
Fungsi tersebut akan menghasilkan nilai *True* jika semua ketentuan tersebut dipenuhi. Jika tidak terpenuhi maka fungsi akan menghasilkan nilai *False*. Fungsi anda harus diberi nama *cek_angka()*.

Pengerjaan soal menggunakan fungsi dengan *if-else* untuk mengecek ketiga angka dalam parameter a, b, dan c berbeda semua, lalu mengecek dengan if untuk menjumlahkan dua parameter yang kemungkinan menghasilkan nilai parameter yang tersisa dengan return *True*, selain itu return *False*.

SOURCE CODE:

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

OUTPUT:



```
PROBLEMS OUTPUT TERMINAL ... Python + - [ ] [X] ... ^ X  
PS D:\71200645_PrakAlpro_4> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_4/Latihan 4.1.py"  
True  
True  
False  
PS D:\71200645_PrakAlpro_4>
```

SOAL 2

Latihan 4.2 Buatlah sebuah fungsi yang dapat menentukan apakah **minimal dua** dari tiga parameter yang diberikan memiliki digit paling kanan yang sama. Fungsi tersebut menghasilkan nilai True jika memenuhi dan False jika tidak memenuhi. Gunakan fungsi tersebut untuk mengecek beberapa test-case berikut ini:

- Input = 30, 20, 18. Output yang diharapkan = True
- Input = 145, 5, 100. Output yang diharapkan = True
- Input = 71, 187, 18. Output yang diharapkan = False
- Input = 1024, 14, 94. Output yang diharapkan = True
- Input = 53, 8900, 658. Output yang diharapkan = False

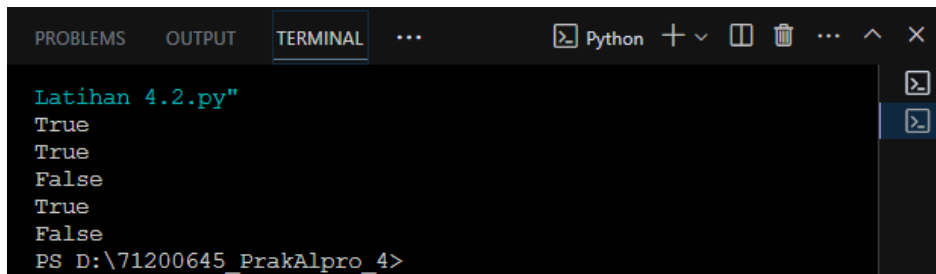
Ketiga bilangan tersebut diinputkan oleh pengguna, sehingga anda perlu membaca input dari pengguna. Fungsi anda harus diberi nama `cek_digit_belakang()`.

Pengerjaan soal untuk menentukan minimal dua dari tiga parameter (a, b, c) yang memiliki digit paling kanan yang sama dengan mengubah angka menjadi string untuk mengambil digit paling kanan menggunakan metode *slicing*.

SOURCE CODE:

```
def cek_digit_belakang2(a, b, c):  
    a = str(a)  
    b = str(b)  
    c = str(c)  
    if a[-1:] == b[-1:] or a[-1:] == c[-1:] or b[-1:] == c[-1]:  
        return True  
    else:  
        return False  
  
print(cek_digit_belakang2(30, 20, 18)) # True  
print(cek_digit_belakang2(145, 5, 100)) # True  
print(cek_digit_belakang2(71, 187, 18)) # False  
print(cek_digit_belakang2(1024, 14, 94)) # True  
print(cek_digit_belakang2(53, 8900, 658)) # False
```

OUTPUT:



```
PROBLEMS OUTPUT TERMINAL ... Python + - [ ] [ ] ... ^ x  
Latihan 4.2.py"  
True  
True  
False  
True  
False  
PS D:\71200645_PrakAlpro_4>
```


SOAL 3

Latihan 4.3 Buatlah fungsi-fungsi konversi suhu menggunakan lambda function. Fungsi-fungsi yang harus anda implementasikan:

- Celcius to Fahrenheit. $F = (9/5) * C + 32$
- Celcius to Reamur. $R = 0.8 * C$

Berikan contoh penggunaannya untuk test-case berikut ini:

- Input C = 100. Output F = 212.
- Input C = 80. Output R = 64.
- Input = 0. Output F = 32.

Pengerjaan soal dengan membuat dua fungsi terpisah untuk Celcius ke Fahrenheit dan Celcius ke Reamur dalam bentuk *lambda function*.

SOURCE CODE:

```
fahrenheit = lambda celcius: int((9/5) * celcius + 32)
reamur = lambda celcius: int(0.8 * celcius)

celcius1 = 100
fahrenheit1 = fahrenheit(celcius1)
print("Input C =", celcius1, " Output F =", fahrenheit1)

celcius2 = 80
reamur2 = reamur(celcius2)
print("Input C =", celcius2, " Output R =", reamur2)

celcius3 = 0
fahrenheit3 = fahrenheit(celcius3)
print("Input C =", celcius3, " Output F =", fahrenheit3)
```

OUTPUT:

```
Input C = 100  Output F = 212
Input C = 80  Output R = 64
Input C = 0   Output F = 32
PS D:\71200645_PrakAlpro_4> 
```

Link Github:

https://github.com/EdithFelicia/71200645_Guided/tree/main/71200645_PrakAlpro_4