



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71200609
Nama Lengkap	Airell Aristo Subagia
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

Modular Programming

Fungsi, Argument dan Parameter

- Fungsi

Adalah sebuah blok kode yang dapat. Fungsi biasanya digunakan untuk menata agar kode lebih mudah dimengerti, digunakan kembali, dan diubah. Fungsi dapat memiliki nama, parameter, dan nilai kembalian.

Contohnya :

```
def luasPersegi(sisi) :  
    rumus = sisi**2  
    return rumus  
  
print(luasPersegi(4))
```

Outputnya :

```
16
```

- Argument

Adalah nilai yang diberikan kepada sebuah fungsi saat dipanggil. Argumen digunakan untuk memberikan nilai kepada parameter fungsi tersebut.

Contohnya :

```
luasPersegi(4)
```

- Parameter

Adalah variabel yang didefinisikan di dalam tanda kurung () pada saat mendefinisikan sebuah fungsi. Pada contoh dibawah, parameternya adalah yang diberi nama sisi.

Contohnya :

```
def luasPersegi(sisi) :  
    rumus = sisi**2  
    return rumus  
  
print(luasPersegi(4))
```

Return Value

Fungsi secara umum memiliki dua jenis, yaitu :

1. Fungsi yang mengembalikan nilai

Pada fungsi ini, terdapat pernyataan "return" yang digunakan untuk mengembalikan nilai tertentu ketika fungsi dipanggil.

Contoh :

```
def luasSegitiga(alas,tinggi) :  
    rumus = 1/2 * alas * tinggi  
    return rumus  
  
print(luasSegitiga(4,2))
```

Outputnya :

```
4.0
```

2. Fungsi yang tidak mengembalikan nilai

Pada fungsi ini, tidak terdapat pernyataan "return" yang digunakan untuk mengembalikan nilai tertentu ketika fungsi dipanggil.

Contoh :

```
def ganjilGenap(angka) :  
    if angka % 2 == 0 :  
        print(angka, "adalah angka genap")  
    else :  
        print(angka, "adalah angka ganjil")  
  
ganjilGenap(7)
```

Outputnya :

```
7 adalah angka ganjil
```

Optional Argument dan Named Argument

1. Optional Argument

Adalah parameter dalam definisi fungsi yang memiliki nilai bawaan (default). Optional Argument tidak wajib diberikan saat memanggil fungsinya. Jika value tidak diberikan saat pemanggilan fungsi, maka value default pada parameter akan yang akan digunakan.

Contohnya :

```
def sapa(name="Anon"):  
    print("Hello, " + name)  
  
sapa()
```

Outputnya :

```
Hello, Anon
```

Atau jika parameternya diisi akan menjadi seperti contoh dibawah :

```
def sapa(name="Anon"):  
    print("Hello, " + name)  
  
sapa("Airell")
```

Outputnya :

```
Hello, Airell
```

2. Named Argument

Adalah argument yang pada saat pemanggilan fungsi, parameternya akan dituliskan. Pada saat menggunakan Named Argument, pemanggilan argumen dapat digunakan dalam urutan berbeda.

Contohnya :

```
def sapa(sapaan="Hello", name="Anon"):  
    print(sapaan, name)  
  
sapa(name="Airell", sapaan="Hi")
```

Outputnya :

Hi Airell

Anonymus/Lambda Function

Merupakan cara singkat untuk mendefinisikan sebuah fungsi anonim. Fungsi Lambda biasanya digunakan ketika, ingin membuat fungsi sederhana.

Contohnya :

```
def ganjilGenap(angka) :  
    if angka % 2 == 0 :  
        print(angka, "adalah angka genap")  
    else :  
        print(angka, "adalah angka ganjil")  
  
ganjilGenap(7)
```

Jika kita ubah menjadi bentuk lambda, maka bentuknya akan menjadi :

lambda argument: expression

```
ganjilGenapLambda = (lambda angka: print(angka, "adalah angka genap")  
                      if (angka % 2 == 0)  
                      else print(angka, "adalah angka ganjil"))  
  
ganjilGenapLambda(2)
```

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

#Latihan 4.1

```
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  
        else :  
            return False  
    else :  
        return False  
  
print(cek_angka(1,4,5))  
print(cek_angka(1,1,5))  
print(cek_angka(1,4,1))  
print(cek_angka(2,3,7))
```

Outputnya :

```
True  
False  
False  
False
```

Penjelasannya :

1. Pertama-tama, akan dibuat function bernama **cek_angka** dengan 3 parameter bernama **a,b,c**. 3 parameter tersebut akan digunakan untuk menyimpan input dari user.
2. Setelah itu, dibuat logika yang pertama dimana ketiga angka tidak boleh ada yang sama sehingga menjadi **a != b and a != c and b != c**. Yang dimana disana kita menggunakan tanda **!=** tidak sama dengan, dan digunakan ekspresi **and** agar jika ada satu kemungkinan yang true akan tetapi yang lain false maka akan dikembalikan false.
3. Kemudian, dibuat logika kedua yang dimana penjumlahan dari dua variable itu sama dengan variable ketiga, sehingga didapat **a+b == c or a+c == b or b+c == a**. Pada logika ini digunakan **or** karena jika salah satu dari kemungkinan itu True maka akan mengembalikan nilai True.

SOAL 2

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

Outputnya :

```
True  
True  
False  
True  
False
```

Penjelasannya :

1. Pertama-tama, akan dibuat function bernama **cek_digit_belakang** dengan 3 parameter bernama **a,b,c**. 3 parameter tersebut akan digunakan untuk menyimpan input dari user.
2. Kemudian, variable yang telah disimpan di parameter **a,b,c** akan di ubah/convert menjadi string dimana disana digunakan variable **toStringA, toStringB, toStringC**.
3. Setelah itu, dibuat logika untuk pengecekan digit belakang dari angka menggunakan indexing pada string ditambah dengan pengecekan dengan angka yang lain. Sehingga menjadi salah satu fungsi **toStringA[len(toStringA)-1] == toStringB[len(toStringB)-1]**. Cara pembacaan indexing tersebut adalah, dengan pengambilan panjang dari string kemudian dikurangi dengan 1 untuk mendapatkna angka terluar.
4. Pada logika ini digunakan or karena jika salah satu dari kemungkinan itu True maka akan mengembalikan nilai True.

SOAL 3

#Latihan 4.3

```
suhu= int(input("C = "))

konversi_suhu = [lambda suhu : "F = " + str(int((9/5) * suhu +32)),
                  lambda suhu : "R = " + str(int(0.8 * suhu))]
print("Input C =", suhu, "Output",konversi_suhu[0](suhu))
print("Input C =", suhu, "Output",konversi_suhu[1](suhu))
```

Outputnya :

```
C = 30
Input C = 30 Output F = 86
Input C = 30 Output R = 24
```

Penjelasannya :

1. Pertama-tama, dibuat variable **suhu** untuk mengambil inputan dari user.
2. Setelah itu, dibuat fungsi lambda yang disimpan pada variable **konversi_suhu**. Disana saya menggunakan list untuk memisahkan antara konversi suhu fahrenheit dengan celcius.
3. Kemudian, isi pada fungsi lambda tersebut adalah argument yang digunakan ialah parameter **suhu**. Setelah itu parameter suhu akan dimasukkan pada rumus-rumus tiap suhu. Pada pembuatan rumus disana dapat dilihat bahwa terjadi konversi angka dari integer ke string, hal ini terjadi karena didepannya saya menambahkannya dengan sebuah string maka harus di konversi.
4. Kemudian, untuk menggunakan fungsi lambda tersebut akan memanggil indeks dari list tersebut jika ingin menggunakan fahrenheit, akan digunakan indeks pertama **konversi_suhu[0](suhu)** , tetapi jika ingin menggunakan reamur maka akan digunakan indeks kedua **konversi_suhu[1](suhu)**.

Link Github :

https://github.com/AirellAristo/TugasPrakAlpro/tree/main/71200609_Pertemuan4Guided