



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230985
Nama Lengkap	TOMAS BECKET
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%)

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, Argumen, Dan Parameter

```
nama = input("Masukkan nama: ")
```

Kode diatas merupakan program yang diminta untuk memasukan nama dan akan memunculkan output untuk menyapa user melalui hasil yang diinputkan

```
print("Hallo ", nama, " selamat pagi!")
```

Kedua fungsi yang digunakan yaitu input() dan Print() dimana kedua fungsi ini merupakan bawaan dari python. Secara umum fungsi adalah kumpulan-kumpulan perintah yang dijadikan satu

MATERI 2

Adapun beberapa contoh kode yang menggunakan Def function contohnya seperti source code dibawah yang menjelaskan hasil tambah, keyword def sendiri digunakan untuk mendefinisikan suatu fungsi dan fungsi yang dibuat yaitu tambah() Dan fungsi yang dibuat harus memiliki sebuah argumen untuk dikenali

Terdapat juga keyword return yang digunakan untuk mengembalikan atau mengeluarkan hasil output dari fungsi tersebut

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

Sebagai representasi tambahan contoh :

```
1  # definisikan fungsi tambah terlebih dahulu  
2  def tambah(a, b):  
3      hasil = a + b  
4      return hasil  
5  
6  # panggil fungsi tambah dengan dua arguments berupa nilai: 10 dan 5  
7  c = tambah(10, 5)  
8  print(c)
```

Source kode diatas menambahkan variabel `c = tambah(10, 5)` untuk digunakan sebagai pemunculan hasil output yang digunakan oleh `print()` sehingga dibuatlah `print(c)`.

Contoh Source code lainnya memanggil fungsi `hitung_belanja`, seperti source kode dibawah:

```
1 def hitung_belanja(hargaBarang, diskon):
2     bayar = hargaBarang - (hargaBarang * diskon)/100
3     return bayar
4
5 print(hitung_belanja(100000, 0.1))
```

Arti `print(hitung_belanja(100000, 0.1))` sendiri untuk memanggil fungsi `hitung_belanja` dimana angka 100000 merepresentasikan nilai `hargaBarang`, sedangkan 0.1 direpresentasikan sebagai nilai diskon yang kemudian kedua ini akan dilakukan perhitungan yang menghitung nilai harga bayar dengan adanya diskon yang dimiliki.

MATERI 3

Return Value

Secara umum ada 2 jenis yang dikeluarkan oleh sebuah fungsi, yaitu fungsi yang tidak mengembalikan nilai dan fungsi yang mengembalikan nilai. Fungsi yang mengembalikan nilai sering disebut dengan void function. Contohnya seperti source kode dibawah:

```
def print_twice(message) :
    print(message)
    print(message)

print_twice("Hello World!")
```

Fungsi dari `print_twice` yang dituliskan pada line terakhir hanya mengembalikan nilai `None` jika dipanggil fungsi tersebut. Sehingga output yang dimunculkan seperti berikut:

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

MATERI 4

Optional Argument dan Named Argument yaitu parameter yang bersifat opsional dan memiliki nilai bawaan atau disebut (default) yang sebelumnya sudah terdefiniskan.

Fungsi `hitung_belanja()` pada source kode dibawah memiliki dua parameter yaitu `belanja` dan `diskon`. Yang secara langsung `diskon` memiliki default yang mempunyai nilai 0 (yang artinya 0%).

Dan cara memanggil fungsi `hitung_belanja()` sendiri dengan melakukan `print(hitung_belanja(parameter, parameter))` yang akan dibuat dan dilakuakn operasi nantinya.

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

Contoh lainnya yang simple seperti source kode dibawah :

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

Output yang dihasilkan adalah:

```
Nilai a: 20  
Nilai b: 30  
Nilai c: 40
```

Pada fungsi `cetak()` memiliki 3 parameter yang dimana nilai ketiga parameter tersebut yaitu 20, 30, dan 40 yang direpresentasi menjadi `a`, `b`, `c` untuk dimasukan ke fungsi `cetak(a, b, c)` dan dengan melakukan sebuah `print("nilai a : ", a)` akan memunculkan value dari variabel `a` yang dimiliki, dan `print b` juga `c` sama halnya seperti `print("nilai a :", a)` hanya saja diubah variabelnya.

MATERI 5

Contoh Mendefinisikan sebuah fungsi yang ada pada source kode tersebut :

```
1 def tagihan_listrik(pemakaian, golongan=3):
2     bayar = 0
3     pemakaian_100 = 100 if pemakaian > 100 else pemakaian
4     pemakaian_100_lebih = pemakaian - pemakaian_100
5     if golongan == 1:
6         bayar = pemakaian_100 * 1500 + pemakaian_100_lebih * 2000
7     elif golongan == 2:
8         bayar = pemakaian_100 * 2500 + pemakaian_100_lebih * 3000
9     elif golongan == 3:
10        bayar = pemakaian_100 * 4000 + pemakaian_100_lebih * 5000
11    elif golongan == 4:
12        bayar = pemakaian_100 * 5000 + pemakaian_100_lebih * 7000
13    return bayar
14
15 print(tagihan_listrik(130))
16 print(tagihan_listrik(80, 4))
17 print(tagihan_listrik(golongan=1, pemakaian=175))
```

Fungsi tersebut diminta untuk menghitung jumlah pemakaian listrik yang memiliki tarif berbeda seperti 100 kwh pertama tidak dikenakan biaya tambahan, sedangkan jika sudah melebihi maka akan dikenakan tarif tambahan.

Sehingga dengan fungsi pembelian_100 = 100 if pemakaian > 100 else pemakaian (menggunakan ternary) akan memberitahu apakah tarif yang digunakan melebihi atau belum.

Jika melebihi dan merupakan golongan satu maka pembayaran tarif akan dikenakan sebanyak jumlah pemakaian * 1500 tarif pertama + pemakaian * 2000 tarif saat sudah melebihi:

Dan seterusnya hingga mengembalikan fungsi menggunakan return untuk mendapatkan hasil output nya

MATERI 6

Anonymous Function (Lambda), atau fungsi tanpa nama ini merupakan fitur tambahan yang dimiliki oleh python namun bukan merupakan fitur utamanya. Sehingga keyword lambda sendiri didefinisikan sebagai anonymous function. Sebagai contoh source kode dibawah merupakan source kode yang biasa dilihat :

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

print(tambah(10,20))
```

Namun dalam Representasi Lambda code tersebut hampir mirip seperti ternary operation

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

Contoh lainnya menentukan kedua sisi apakah persegi atau bukan :

```
hasil = lambda sisi1, sisi2: (sisi1*sisi2) if (sisi1 == sisi2) else ("Bukan persegi")  
print(hasil(5, 5))  
print(hasil(5,7))
```

Adapun contoh lain lambda Function yang menghitung angka menggunakan konverensi modulus ke lambda function

```
kelipatan Sembilan = lambda angka: angka % 9 == 0  
  
print(kelipatan Sembilan(81))  
print(kelipatan Sembilan(2000))
```

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

1.1 SOURCE CODE

```
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(9, 5, 4))  
print(cek_angka(3, 5, 2))  
print(cek_angka(2, 4, 2))  
print(cek_angka(4, 7, 11))  
print(cek_angka(2, 5, 4))
```

1.2 PENJELASAN

Kode disana menggunakan fungsi def untuk memanggil fungsi yang ada yaitu cek_angka(a, b, c) dan fungsi selanjutnya menggunakan percabangan untuk menentukan hasil jawaban dari ketiga angka tersebut, apakah semua angka berbeda dan memiliki hasil pertambahan dengan hasil sisa atau tidak.

Yaitu dengan menggunakan operator And dan Or untuk menentukan ketiga angka tersebut

a != b artinya a bukan sama dengan c dan

a != c artinya a bukan sama dengan c dan terakhir

b != c artinya b bukan sama dengan c

kemudian menentukan apakah kedua angka dari ketiga angka tersebut memiliki hasil tambah yang merupakan sisa angka dari pertambahan tersebut

a + b == c jika hasil a + b sama dengan c maka true

dan seterusnya....

Jika tidak memenuhi fungsi pertama maka akan lanjut ke else

Dengan memberikan output False

1.3 HASIL OUTPUT

```
True  
True  
False  
True  
False
```

SOAL 2

2.1 SOURCE CODE

```
def cek_digit_belakang(a, b, c):  
    angka1 = a % 10  
    angka2 = b % 10  
    angka3 = c % 10  
  
    if angka1 == angka2 or angka1 == angka3 or angka2 == angka3:  
        return True  
    else:  
        return False  
  
a = int(input("Angka 1 : "))  
b = int(input("Angka 2 : "))  
c = int(input("Angka 3 : "))  
  
result = cek_digit_belakang(a, b, c)  
print(result)
```

2.2 PENJELASAN

disoal dijelaskan untuk membuat suatu fungsi yang menentukan 1 angka digit belakang dari ketiga angka yang dimiliki memiliki angka yang sama. Sehingga disini bisa menggunakan modulus 10 untuk mendapatkan angka digit paling belakang, contoh:

jika angka tersebut merupakan 1746 dan di modulus 10 maka akan mendapatkan angka 6

contoh lain jika 34875 di modulus 10 maka akan mendapatkan angka 5

seperti fungsi diatas

angka1, angka2, dan angka3 hanya sebagai variabel tempat menyimpan nilai input yang sudah dimodulus

sehingga untuk menentukan apakah ada 2 angka yang hasil modulus dari digit belakang sama atau tidak

contoh check test :

jika a = 30 maka mod 10 = 0 (sama)

jika b = 20 maka mod 10 = 0 (sama)

jika c = 18 maka mod 10 = 8 (beda)

dengan menggunakan fungsi `angka1 == angka2 or angka1 == angka3 or angka2 == angka3` akan menentukan hasil input TRUE atau FALSE

sehingga hasil output yang akan muncul karena terdapat 2 angka yang memiliki digit angka terakhir yang sama yaitu TRUE

2.3 HASIL OUTPUT

```
Angka 1 : 30  
Angka 2 : 20  
Angka 3 : 18  
True
```

SOAL 3

3.1 SOURCE CODE

```
celcius_fahrenheit = lambda c: (9/5) * c + 32  
celcius_reamur = lambda c: 0.8 * c  
  
celcius = int(input("Input Celcius = "))  
  
print("Ouput Farenheit : ", int(celcius_fahrenheit(celcius)))  
print("Output Reamur : ", int(celcius_reamur(celcius)))
```

3.2 PENJELASAN

Dikarenakan fungsi yang digunakan menggunakan representasi lambda sehingga line code yang digunakan hampir mirip seperti ternary operator hanya saja lambda digunakan untuk melakukan perhitungan.

Sehingga code diatas menggunakan 3 variabel yaitu celcius_fahrenheit, celcius_reamur, dan celcius sebagai input hasil.

Perhitungan yang digunakan seperti rumus yang diberikan soal, hanya saja menggunakan fungsi lambda

Sehingga ketika memunculkan hasil Ouput akan memberikan hasil perubahan celcius menjadi Farenheit atau Reamur

3.3 HASIL OUTPUT

```
Input Celcius = 100  
Ouput Farenheit : 212  
Output Reamur : 80
```

```
Input Celcius = 80  
Ouput Farenheit : 176  
Output Reamur : 64
```