MODUL IV
FUNGSI (Function)

Tujuan :

1. Dapat mengetahui dan memahami fungsi dalam Python
2. Dapat mengetahui bentuk umum dari fungsi
3. Dapat menggunakan dan mendeklarasikan fungsi

Tugas Pendahuluan

1. Sebutkan dan jelaskan macam-macam perintah fungsi ?
2. Buatlah contoh soal dan program sederhana dengan menggunakan fungsi ?
3. Jelaskan perbedaan antara Def dan Lamda ? dengan menggunakan tabel
4. Buatlah programdengan menggunakan fungsi `len(), max(), min(),abs()!`

1. Dasar Teori

1.1. Pengertian Fungsi

Fungsi (Function) adalah suatu program terpisah dalam blok sendiri yang berfungsi sebagai sub-program (modul program) yang merupakan sebuah program kecil untuk memproses sebagian dari pekerjaan program utama. Fungsi digunakan untuk mengumpulkan beberapa perintah yang sering dipakai dalam sebuah program.

Fungsi juga bisa diartikan sebagai bagian dari program yang dapat digunakan kembali. Hal ini bisa dicapai dengan memberi nama pada blok statemen, kemudian nama ini dapat dipanggil di manapun dalam program. Kita telah menggunakan beberapa fungsi builtin seperti *range*.

Fungsi dalam Python didefinisikan menggunakan kata kunci *def*. Setelah *def* ada nama pengenal fungsi diikut dengan parameter yang diapit oleh tanda kurung dan diakhir dingan tanda titik dua :. Baris berikutnya berupa blok fungsi yang akan dijalankan jika fungsi dipanggil.

```
def halo_dunia():
    print 'Halo Dunia!'

halo_dunia()   # memanggil fungsi halo_dunia
halo_dunia()   # fungsi halo_dunia dipanggil lagi
```

Keuntungan menggunakan fungsi :

- Program besar dapat di pisah-pisah menjadi program-program kecil melalui function.
- Kemudahan dalam mencari kesalahan-kesalahan karena alur logika jelas dan kesalahan dapat dilokalisasi dalam suatu modul tertentu.
- Memperbaiki atau memodifikasi program dapat dilakukan pada suatu modul tertentu saja tanpa menggangu keseluruhan program.
- Dapat digunakan kembali (Reusability) oleh program atau fungsi lain.
- Meminimalkan penulisan perintah yang sama.

Kategori Fungsi

1. Standard Library Function
   fungsi-fungsi yang telah disediakan oleh Interpreter Python dalam file-file atau librarinya.
   Misalnya: `raw_input()`, `input()`, `print()`, `open()`, `len()`, `max()`, `min()`, `abs()` dll.
2. Programme-Defined Function
   function yang dibuat oleh programmer sendiri. Function ini memiliki nama tertentu yang unik dalam program, letaknya terpisah dari program utama, dan bisa dijadikan satu ke dalam suatu library buatan programmer itu sendiri.

1.2 Mendeklarasikan dan Memakai Fungsi

Dalam python terdapat dua perintah yang dapat digunakan untuk membuat sebuah fungsi, yaitu

a) Statemen *Def*

Statemen *def* adalah perintah standar dalam python untuk mendefinisikan sebuah fungsi. *def* dalam python merupakan perintah yang executable, artinya function tidak akan aktif sampai python me*running* perintah *def* tersebut

Statemen *def* digunakan untuk mendeklarasikan fungsi. Sedangkan statemen *return* digunakan untuk mengembalikan suatu nilai kepada bagian program yang memanggil fungsi. Bentuk umum untuk mendeklarasikan fungsi adalah sebagai berikut :

```
def <nama_fungsi>(arg1, arg2, arg3, …,argN) :
     <statemen-statemen>
```

Sebuah fungsi diawali dengan statemen *def* kemudian diikuti oleh sebuah *nama_fungsi* nya. Sebuah fungsi dapat memiliki daftar argumen (parameter) ataupun tidak. Tanda titik dua (:) menandakan awal pendefinisian tubuh dari fungsi yang terdiri dari statemen-statemen.

Tubuh fungsi yang memiliki statemen *return* :

```
def <nama_fungsi>(arg1, arg2, arg3, …,argN) :
    <statemen-statemen>
    ...
    return <value>
```

Statemen *return* dapat diletakkan di bagian mana saja dalam tubuh fungsi. Statemen *return* menandakan akhir dar pemanggilan fungsi dan akan mengirimkan suatu nilai balik kepada program yang memanggil fungsi tersebut. Statemen *return* bersifat opsional, artinya jika sebuah fungsi tidak memiliki statemen *return*, maka sebuah fungsi tidak akan mengembalikan suatu nilai apapun.

b) **Statemen *Lambda***

Selain statemen *def*, Python juga menyediakan suatu bentuk ekspresi yang menghasilkan objek fungsi. Karena kesamaannya dengan tools dalam bahasa Lisp, ini disebut *lambda* .Seperti *def*, ekspresi ini menciptakan sebuah fungsi yang akan dipanggil nanti, tapi mengembalikan fungsi dan bukan untuk menetapkan nama.

lambda dalam python lebih dikenal dengan nama Anonymous Function (Fungsi yang tidak disebutkan bukanlah sebuah perintah (statemen) namun lebih namanya). *Lambda* kepada ekspresi (expression). Dalam prakteknya, mereka sering digunakan sebagai cara untuk inline definisi fungsi, atau untuk menunda pelaksanaan sepotong kode.

Bentuk umum lambda adalah kata kunci *lambda*, diikuti oleh satu atau lebih argument (persis seperti daftar argumen dalam tanda kurung di *def* header), diikuti oleh ekspresi setelah tandatitik dua:

```
lambda argument1, argument2,... argumentN :expression
using arguments
```

lambda memiliki perbedaan dengan *def* antara lain :

1. *Lambda* adalah sebuah ekspresi, bukan pernyataan. Karena ini, sebuah *lambda* dapat muncul di tempat-tempat *def* tidak diperbolehkan oleh sintaks Python-di dalam daftar harfiah atau pemanggilan fungsi argumen, misalnya. Sebagai ekspresi, *lambda* mengembalikan nilai (fungsi baru) yang opsional dapat diberi nama. Sebaliknya, pernyataan *def* selalu memberikan fungsi baru ke nama di header, bukannya kembali sebagai hasilnya.
2. Tubuh *lambda* adalah ekspresi tunggal, bukan satu blok statemen. Tubuh lambda sama dengan apa yang akan dimasukkan ke dalam statemen *return* dalam tubuh *def*.

Fungsi Rekursif

Fungsi Rekursif merupakan suatu fungsi yang memanggil dirinya sendiri. Artinya, fungsi tersebut dipanggil di dalam tubuh fungsi itu sendiri. Tujuan dilakukan rekursif adalah untuk menyederhanakan penulisan program dan menggantikan bentuk iterasi. Dengan rekursi, program akan lebih mudah dilihat.

Mencari nilai faktorial dari suatu bilangan bulat positif adalah salah satu pokok bahasan yang memudahkan pemahaman mengenai fungsi rekursif.

Scope Variabel

Scope variabel atau cakupan variabel merupakan suatu keadaan dimana pendeklarasian sebuah variabel di tentukan. Dalam scope variabel dikenal dua istilah yaitu local dan global .

- Variabel local **:** ketika variabel tersebut didefinisikan didalam sebuah fungsi (*def*). Artinya, variabel tersebut hanya dapat di gunakan dalam cakupan fungsi tersebut saja
- Variabel global **:** didefinisikan diluar fungsi . Artinya, variabel tersebut dapat digunakan oleh fungsi lain atau pun program utamanya.

2. Praktikum

Latihan

Contoh penggunaan fungsi :

```
def ucapan():
    print ("selamat malam")
ucapan()
```

```
def masukan_data():
    nama=str(input("masukan nama : "))
    NRP =int(input("masukan NIM : "))
def cetak_string():
    print ("ini adalah fungsi yang mencetak string")
    print ("silakan masukan data")
cetak_string()
masukan_data()
```

Contoh program dengan melibatkan nilai balik (return):

```
def perkalian(a,b):
    c = a*b
    return c
print(perkalian(5,10))
```

Contoh penggunaan lambda :

```
f = lambda x, y, z: x + y + z
print (f(10,20,30))
```

Contoh:

```
z = (lambda a = "tic", b = "tac", c = "toe" : a + b + c)
print (z("ZOO"))
```

Contoh penggunaan scope variabel :

```
def contohScope(X):
    X = 10
    print ("Nilai X di dalam fungsi, x = ", X)
X = 30
print ("Nilai x di luar fungsi, x = ", X)
contohScope(X)
```

3. Tugas Praktikum

1. Buatlah program Konversi Suhu dengan menggunakan fungsi
2. Buatlah program untuk menghitung bilangan Fibonacci dengan menggunakan fungsi
3. Buatlah program menampilkan segitiga bintang menggunakan fungsi yang menerima 2 parameter. Parameter pertama untuk menentukan tinggi segitiga, dan parameter kedua untuk menentukan jumlah segitiga yang akan ditampilkan.

` Contoh Output :