

Modul 4

Fungsi (Function)

Fungsi adalah bagian dari program yang dapat digunakan ulang. 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 pengenalan fungsi diikuti dengan parameter yang diapit oleh tanda kurung dan diakhiri dengan tanda titik dua **:**. Baris berikutnya berupa blok fungsi yang akan dijalankan jika fungsi dipanggil.

```
# lat15.py

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

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

Parameter Fungsi

Fungsi dapat membaca parameter, parameter adalah nilai yang disediakan kepada fungsi, dimana nilai ini akan menentukan output yang akan dihasilkan fungsi.

Parameter dikirim dalam tanda kurung saat pemanggilan fungsi. Nilai parameter saat pemanggilan fungsi dinamakan *argument*.

```
# lat16.py

def halo(nama):
    print('Halo %s!' % nama)

def cetak_maksimal(a, b):
    if a > b:
        print('%s merupakan nilai maksimal' % a)
    elif a == b:
        print('%s sama dengan %s' % (a, b))
    else:
        print('%s merupakan nilai maksimal' % b)

halo('Dunia') # memanggil fungsi halo dengan argumen 'Dunia'
halo('Indonesia') # memanggil fungsi halo dengan argumen 'Indonesia'

cetak_maksimal(10, 100)

x = 9
y = 3

cetak_maksimal(x, y)
```

Variabel Lokal

Jika ada variabel yang dideklarasikan didalam blok fungsi, variabel ini tidak ada kaitannya dengan variabel lain dengan nama yang sama diluar fungsi, dengan kata lain nama variabel hanya lokal untuk fungsi. Hal ini disebut juga scope variabel.

```
# lat17.py

x = 50

def fungsi(x):
    print 'x = ', x
    x = 2
    print('merubah lokal variabel x = ', x)

fungsi(100)

print('nilai x masih %s' % x)
```

Penggunaan Statemen Global

Dalam blok fungsi kita dapat mengakses variabel diluar fungsi, akses ini terbatas hanya akses baca. Jika blok fungsi ingin menulis variabel diluar fungsi anda dapat menggunakan statemen global.

```
# lat17.py

x = 50

def fungsi():
    print('x = ', x)

def fungsi2():
    x = 100 # menulis ke lokal variabel
    print('x = ', x)

def fungsi3():
    global x
    x = 100
    print('x = ', x)

fungsi()
print('nilai x = ', x)

fungsi2()
print('nilai x = ', x)

fungsi3()
print('nilai x = ', x)
```

Nilai Argumen Default

Untuk beberapa fungsi yang ingin menyediakan paramater opsional dan menggunakan nilai default jika pengguna tidak menyediakan argumen saat fungsi dipanggil. Anda bisa menspesifikasikan nilai default dengan tanda sama dengan = setelah nama parameter.

```
# lat18.py
```

```
def katakan(pesan, jumlah=1):  
    print(pesan * jumlah)  
  
katakan('Halo ')  
katakan('Halo ', 3)
```

Keyword Argumen

Jika anda membuat fungsi dengan banyak parameter dan hanya ingin menspesifikasikan sebagian, anda dapat menggunakan keyword argumen. Kita menggunakan nama (keyword) melainkan posisi (argumen posisi, normal pemanggilan).

```
# lat19.py  
  
def fungsi(a, b=5, c=10):  
    print 'a = ', a  
    print 'b = ', b  
    print 'c = ', c  
  
fungsi(3, 7)  
fungsi(25, c=24)  
fungsi(c=50, a=100)
```

Parameter VarArgs

Terkadang anda ingin membuat fungsi yang dapat menerima jumlah argumen yang tidak tentu, hal ini dapat dilakukan menggunakan tanda bintang *.

```
# lat20.py  
  
def total(*bilangan, **keywords):  
    hitung = 0  
    for bil in bilangan:  
        hitung += bil  
    for key in keywords:  
        hitung += keywords[key]  
    return hitung  
  
print(total(1, 2, 3, 4, 5))  
print(total(daging=2, sayur=10, buah=3))  
print(total(7, 8, 5, daging=2, sayur=10, buah=3))
```

Statemen Return

Statemen return digunakan untuk keluar dari fungsi. Kita juga dapat menspesifikasikan nilai kembalian. Seperti pada latihan 20 melainkan mencetak hasil jumlah dalam blok fungsi, fungsi total mengembalikan nilai jumlah ke pemanggil.

Doc String

Python memiliki fitur *documentation string*, seringnya disebut dengan nama *docstring*. Docstring berguna untuk mendokumentasikan program agar mudah untuk dipahami dan digunakan.

```
# lat21.py

def katakan(pesan, jumlah=1):
    "mencetak pesan <pesan> dengan jumlah <jumlah>"
    print(pesan * jumlah)

print(katakan.__doc__)
```

Secara interaktif anda dapat mengakses docstring dengan fungsi `help`.

```
>>> import lat21
>>> help(lat21.katakan)
```

Latihan :

1. Membuat menu sederhana untuk Input, Edit, Hapus dan Menampilkan Data.
Ketikan program berikut dan amati hasilnya:

```
# Variabel global untuk menyimpan data Buku
buku=[]

# fungsi untuk menampilkan semua data
def show_data():
    if len(buku) <= 0:
        print("BELUM ADA DATA")
    else:
        for indeks in range(len(buku)):
            print("[%d] %s" % (indeks, buku[indeks]))

# fungsi untuk menambah data
def insert_data():
    buku_baru = input("Judul Buku: ")
    buku.append(buku_baru)

# fungsi untuk edit data
def edit_data():
    show_data()
    indeks = int(input("Inputkan ID buku: "))
    if(indeks > len(buku)):
        print("ID salah")
    else:
        judul_baru = input("Judul baru: ")
        buku[indeks] = judul_baru

# fungsi untuk menghapus data
def delete_data():
    show_data()
    indeks = int(input("Inputkan ID buku: "))
    if(indeks > len(buku)):
        print("ID salah")
    else:
        buku.remove(buku[indeks])

# fungsi untuk menampilkan menu
def show_menu():
    print("\n")
```

```

print("----- MENU -----")
print("[1] Show Data")
print("[2] Insert Data")
print("[3] Edit Data")
print("[4] Delete Data")
print("[5] Exit")

menu = input("PILIH MENU> ")
print("\n")

if menu == "1":
    show_data()
elif menu == "2":
    insert_data()
elif menu == "3":
    edit_data()
elif menu == "4":
    delete_data()
elif menu == "5":
    exit()
else:
    print("Salah pilih!")

if __name__ == "__main__":

    while(True):
        show_menu()

```

2. Buatlah sebuah function/method yang menerima nilai panjang dan lebar sebuah persegi panjang kemudian menghitung nilai luasnya. Function/method kemudian mengembalikan (return) nilai luas tersebut. Nilai panjang dan lebar diinputkan melalui keyboard.
3. Buatlah sebuah function/method yang menerima sebuah list/array dengan elemen bertipe double kemudian mengembalikan nilai elemen paling kecil yang ada pada list/array tersebut tersebut.
4. Buatlah sebuah function/method yang menerima sebuah list dengan elemen bertipe double kemudian mengembalikan list baru dengan elemen yang sama namun nilai yang sudah terurut dari besar ke kecil / kecil ke besar.