

# Memahami Fungsi dan Prosedur pada Python

Pada pembuatan program yang kompleks dan memiliki banyak fitur, kita diharuskan menggunakan fungsi.

Kenapa memangnya kalau tidak menggunakan fungsi?

Bisa jadi kita akan kerepotan menulis kode programnya, karena banyak yang harus ditulis dan kode akan menjadi sulit dibaca dan dirawat (*maintenance*).

Dengan fungsi, kita dapat memecah program besar menjadi sub program yang lebih sederhana.

Masing-masing fitur pada program dapat kita buat dalam satu fungsi.

Pada saat kita membutuhkan fitur tersebut, kita tinggal panggil fungsinya saja.

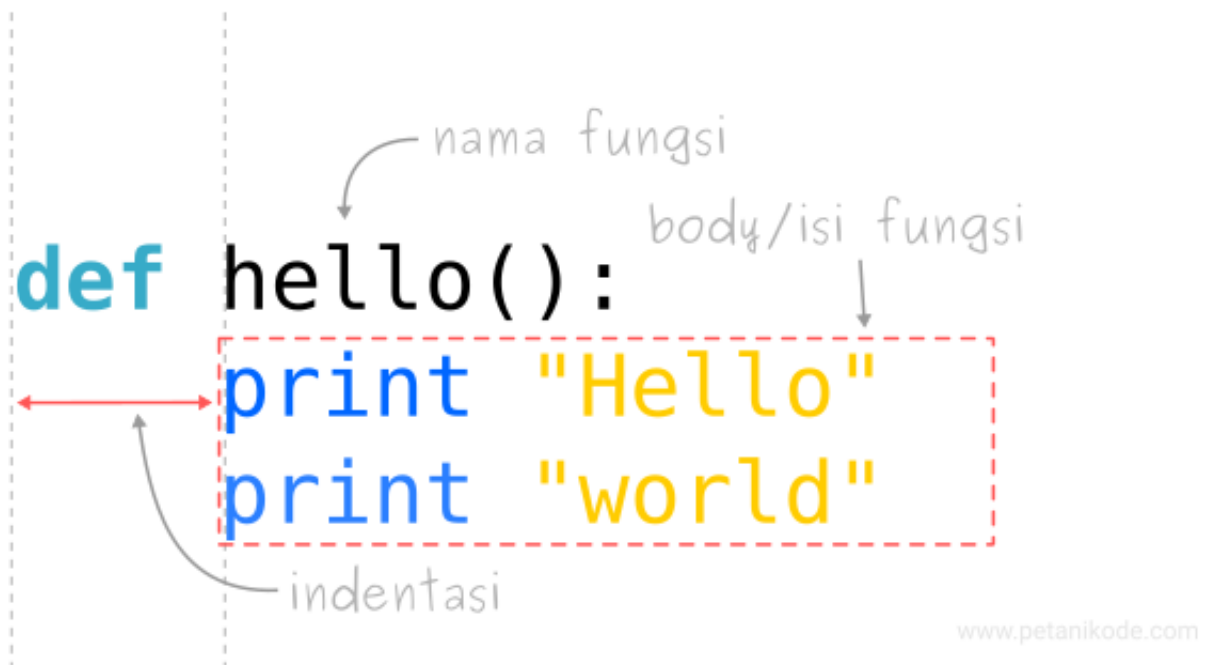
## Cara Membuat Fungsi pada Python

Fungsi pada Python, dibuat dengan kata kunci `def` kemudian diikuti dengan nama fungsinya.

Contoh:

```
def nama_fungsi():  
    print "Hello ini Fungsi"
```

Sama seperti blok kode yang lain, kita juga harus memberikan identasi (tab atau spasi 2x) untuk menuliskan isi fungsi.



Setelah kita buat, kita bisa memanggilnya seperti ini:

```
nama_fungsi()
```

Sebagai contoh, coba tulis kode program berikut:

```
# Membuat Fungsi
def salam():
    print "Hello, Selamat Pagi"

## Pemanggilan Fungsi
salam()
```

Hasilnya:

```
Hello, Selamat Pagi
```

Coba panggil sebanyak 3x:

```
# Membuat Fungsi
def salam():
    print "Hello, Selamat Pagi"

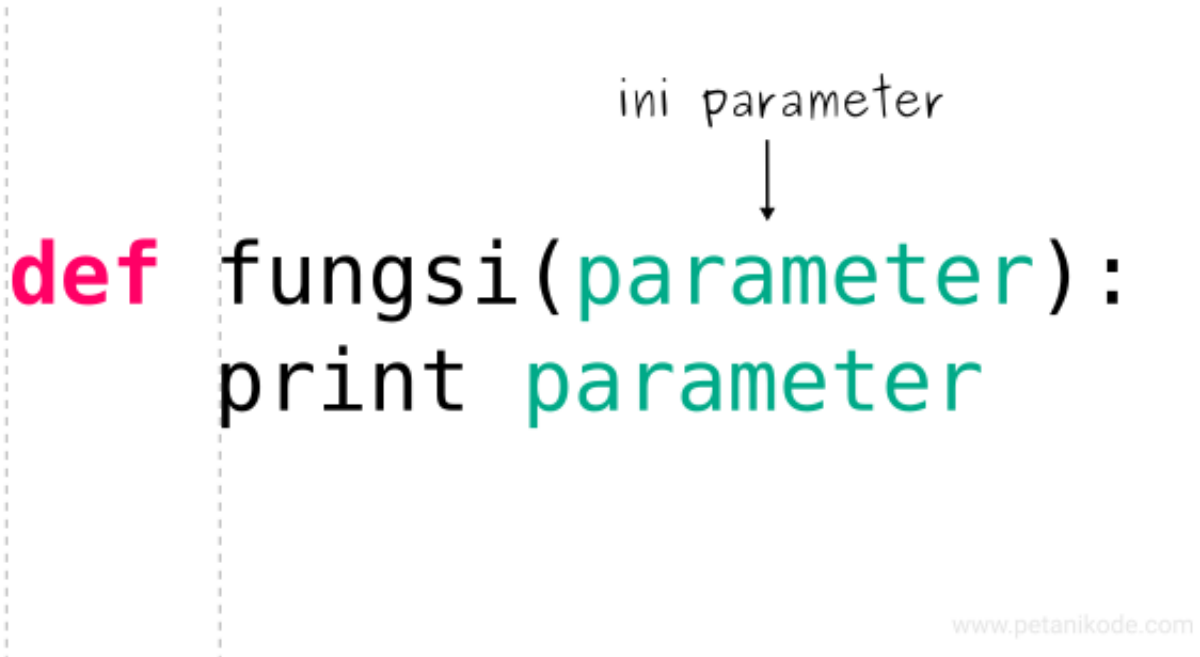
## Pemanggilan Fungsi
salam()
salam()
salam()
```

Hasilnya:

```
Hello, Selamat Pagi
Hello, Selamat Pagi
Hello, Selamat Pagi
```

# Fungsi dengan Parameter

Parameter adalah variabel yang menampung nilai untuk diproses di dalam fungsi.



```
def fungsi(parameter):  
    print parameter
```

ini parameter  
↓

www.petanikode.com

Contoh:

```
def salam(ucapan):  
    print(ucapan)
```

Pada contoh diatas, kita membuat fungsi dengan parameter `ucapan`.

Lalu bagaimana cara memanggilnya?

Cara pemanggilan fungsi yang memiliki parameter adalah seperti ini:

```
salam("Selamat siang")
```

"Selamat siang" adalah nilai parameter yang kita berikan.

Lalu bagaimana kalau parameternya lebih dari satu?

Kita bisa menggunakan tanda koma (,) untuk memisahkannya.

Contoh:

```
# Membuat fungsi dengan parameter  
def luas_segitiga(alas, tinggi):  
    luas = (alas * tinggi) / 2  
    print "Luas segitiga: %f" % luas;  
  
# Pemanggilan fungsi  
luas_segitiga(4, 6)
```

Hasilnya:

```
Luas segitiga: 12.000000
```

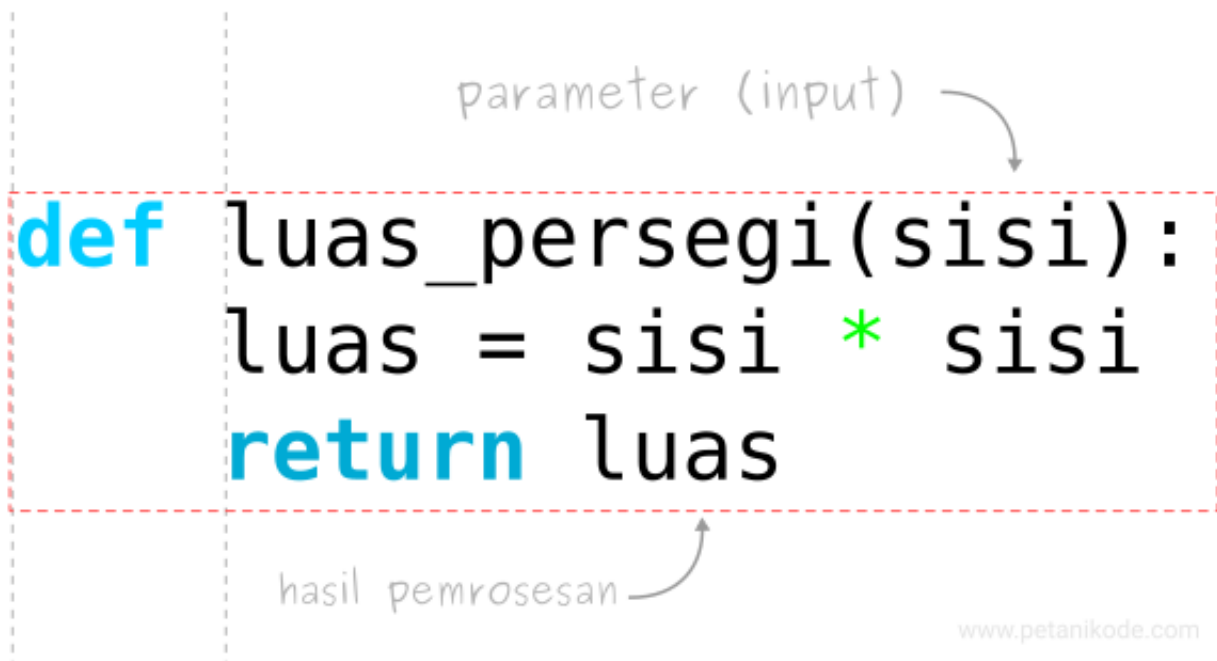
## Fungsi yang Mengembalikan Nilai

Fungsi yang tidak mengembalikan nilai biasanya disebut dengan prosedur.

Namun, kadang kita butuh hasil proses dari fungsi untuk digunakan pada proses berikutnya.

Maka fungsi harus mengembalikan nilai dari hasil pemrosesannya.

Cara mengembalikan nilai adalah menggunakan kata kunci `return` lalu diikuti dengan nilai atau variabel yang akan dikembalikan.



Contoh:

```
def luas_persegi(sisi):  
    luas = sisi * sisi  
    return luas  
  
# pemanggilan fungsi  
print "Luas persegi: %d" % luas_persegi(6)
```

Hasilnya:

```
Luas persegi: 36
```

Apa bedanya dengan fungsi `luas_segitiga()` yang tadi?

Pada fungsi `luas_segitiga()` kita melakukan `print` dari hasil pemrosesan secara langsung di dalam fungsinya.

Sedangkan fungsi `luas_persegi()`, kita melakukan `print` pada saat pemanggilannya.

Jadi, fungsi `luas_persegi()` akan bernilai sesuai dengan hasil yang dikembalikan.

Sehingga kita dapat memanfaatkannya untuk pemerosesan berikutnya.

Misalnya seperti ini:

```
# rumus: sisi x sisi
def luas_persegi(sisi):
    luas = sisi * sisi
    return luas

# rumus: sisi x sisi x sisi
def volume_persegi(sisi):
    volume = luas_persegi(sisi) * sisi
```

Pada contoh di atas, kita melakukan pemanggilan fungsi `luas_persegi()` untuk menghitung volume persegi.

## Variabel Global dan Lokal pada Python

Saat kita menggunakan fungsi, maka kita juga harus mengetahui yang namanya variabel Global dan Lokal.

Apa itu?

Variabel Global adalah variabel yang bisa diakses dari semua fungsi, sedangkan variabel lokal hanya bisa diakses di dalam fungsi tempat ia berada saja.

Pada Python, urutan pengaksesan variabel (*scope*) dikenal dengan sebutan LGB (Local, Global, dan Build-in).

Jadi program python mulai mencari vairabel lokal terlebih dahulu, kalau ada maka itu yang digunakan.

Tapi kalau tidak ada, pencarian terus ke Global, dan Build-in.

Variabel Build-in adalah variabel yang sudah ada di dalam Python.

Contoh program:

```
# membuat variabel global
nama = "unisbank"
versi = "1.0.0"

def help():
    # ini variabel lokal
    nama = "Programku"
    versi = "1.0.2"
    # mengakses variabel lokal
    print "Nama: %s" % nama
    print "Versi: %s" % versi

# mengakses variabel global
print "Nama: %s" % nama
print "Versi: %s" % versi

# memanggil fungsi help()
help()
```

Hasilnya:

```
Nama: Petanikode
Versi: 1.0.0
Nama: Programku
Versi: 1.0.2
```

Perhatikanlah variabel `nama` yang berada di dalam fungsi `help()` dan diluar fungsi `help()`.

Variabel `nama` yang berada di dalam fungsi `help()` adalah variabel lokal.

Jadi, saat kita memanggil fungsi `help()` maka nilai yang akan tampil adalah nilai yang ada di dalam fungsi `help()`.

Python mulai mencari dari lokal, ke global, dan build-in.

Kalau di tiga tempat itu tidak ditemukan, maka biasanya akan terjadi `NameError` atau variabel tidak ditemukan.

## Contoh Program dengan Fungsi

Pertama kita buat sebuah variabel global berupa list untuk menampung judul-judul buku.

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

Nanti program ini akan mampu melakukan operasi CRUD (*Create, Read, Update, dan Delete*). Maka kita membutuhkan fungsi-fungsi berikut:

- `show_data()` untuk menampilkan data dari list `buku`;
- `insert_data()` untuk menambahkan data ke list `buku`;
- `edit_data()` untuk mengedit data di list `buku`;
- `delete_data()` untuk untuk menghapus data dari list `buku`.

```
# Variabel global untuk menyimpan data Buku
```

```
buku = []
```

```
menu=0
```

```
# 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()
```

```
    indeksa = input("Inputkan ID buku: ")
```

```
    indeks=int(indeksa)
```

```
    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()
```

```
    indeksa = input("Inputkan ID buku: ")
```

```
    indeks=int(indeksa)
```

```
    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")
```

```
menua = input("PILIH MENU> ")
```

```
menu=int(menua)
```

```
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()
```

Tugas :

Buatlah program, dengan menu dan function menggunakan python untuk :

1. menghitung luas persegi panjang
2. menghitung keliling persegi panjang
3. menghitung luas lingkaran
4. menghitung keliling persegi panjang