Modul Praktikum

# Algoritma Pemrograman Dasar

9

# "Fungsi dan Prosedur"

## 9.1. Tujuan

- Mahasiswa mampu memahami apa itu fungsi dan prosedur
- Mahasiswa mampu memahami apa itu variabel global dan lokal
- Mahasiswa mampu menerapkan fungsi dan prosedur pada program yang dibuat

## 9.2. Penjelasan

Program yang kompleks dan memiliki banyak fitur, diharuskan menggunakan fungsi. Mengapa harus menggunakan fungsi? Karena jika tidak menggunakannya kita akan kerepotan menulis kode programnya, coding yang 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. Hal ini akan kita coba pada contoh program yang sudah disediakan di bawah. Teori dasar dan hal apa saja yang harus kita ketahui tentang fungsi di Python.

## 9.3. Cara Membuat Fungsi pada Python

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

```
def nama_fungsi(parameter):
    pernyataan
```

Kita dapat membuat nama apa pun yang kita inginkan untuk fungsi yang kita buat, kecuali nama yang merupakan kata kunci Python, dan nama-nama tersebut harus mengikuti aturan. Di dalam fungsi bisa ada sejumlah pernyataan, tetapi pernyataan tersebut harus diindentasi dari def. Daftar parameter bisa kosong, atau berisi sejumlah parameter yang dipisahkan oleh koma.

**Contoh fungsi tanpa parameter:**

```
def nama_fungsi():
    print ("ini adalah fungsi di Python")
```

Sama seperti blok kode yang lain, kita juga harus memberikan indentasi (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 ("Selamat Pagi, FT Muda")
def kali():
     x = 6*4
     print(x)

# Pemanggilan Fungsi
salam()
kali()

# Hasilnya:
# Selamat Pagi, FT Muda
# 24

# Pemanggilan Fungsi 3 kali
salam()
salam()
salam()
kali()
kali()
kali()

# Hasilnya:
# Selamat Pagi, FT Muda
# Selamat Pagi, FT Muda
# Selamat Pagi, FT Muda
# 24
# 24
# 24
```

## 9.4. Fungsi dengan Parameter

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

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

```
    print(parameter)
```

Cara pemanggilan fungsi yang memiliki parameter adalah seperti ini:

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

"Selamat siang" adalah nilai parameter yang kita berikan. Agar lebih jelas kalian dapat membayangkan ini sama seperti fungsi pada matematika.

```
F(x)  = x+2
F(22) = (22) + 2
      = 24
```

Lalu bagaimana kalau parameternya lebih dari satu. Kita bisa menggunakan tanda koma (,) untuk memisahnya.

**Contoh Program:**

```
# Membuat fungsi dengan parameter (panjang, lebar)
def luas_persegi_panjang(panjang, lebar):
    luas = panjang * lebar
    print ("Luas persegi panjang:", luas)

# Pemanggilan fungsi luas_persegi_panjang
luas_persegi_panjang(4, 6)

# Output:
# Luas persegi : 24
```

## 9.5. Fungsi Pengembalian 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 menggunkan kata kunci **return** lalu diikuti dengan nilai atau variabel yang akan dikembalikan.

**Contoh Program :**

```
# Contoh Program mengembalikan hasil fungsi
def luas_persegi(sisi):
    luas = sisi * sisi
    return luas
```

```
# pemanggilan fungsi luas persegi
print ("Luas persegi :", luas_persegi(8))

# output :
#  Luas persegi : 64
```

Apa bedanya dengan fungsi luas_persegi_panjang() yang tadi?. Pada fungsi luas_persegi_panjang() 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.

**Contoh penggunaan pengembalian nilai (Return) :**

```
# 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
     print ("Volume Persegi = ", volume)
# pemanggilan Fungsi
luas_persegi(4)
volume_persegi(6)

Output :
Volume Persegi = 216
```

Pada program diatas dapat dilihat pada hasil luaran def luas_persegi(sisi) dapat dipergunakan oleh def volume_persegi(sisi) menggunakan fungsi **return**.

## 9.6. Variabel Global dan Lokal

Variabel Global adalah variabel yang bisa diakses dari semua fungsi, sedangkan variabel lokal hanya bisa diakses di dalam fungsi tempat dia 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 = "Informatika"
Mata_Kuliah = "Algoritma dan Pemrograman Dasar"

# membuat variabel  lokal
def info():
     Nama = "Teknik Elektro"
     Mata_Kuliah = "Pengantar Teknik ELektro"
     # mengakses variabel lokal
     print("Prodi:", Nama)
     print("Mata Kuliah:", Mata_Kuliah)

# mengakses variabel global
print("Prodi:", Nama)
print("Mata Kuliah:", Mata_Kuliah)

# memanggil fungsi info
info()
```

**Output :**

```
Prodi: Informatika
Mata Kuliah: Algoritma dan Pemrograman Dasar
Prodi: Teknik Elektro
Mata Kuliah: Pengantar Teknik ELektro
```

Perhatikanlah variabel nama yang berada di dalam fungsi info() dan diluar fungsi info(). Variabel nama yang berada di dalam fungsi info() adalah variabel lokal. Jadi, saat program memanggil fungsi info() maka nilai yang akan tampil adalah nilai yang ada di dalam fungsi info(). Cara kerjanya 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.

## 9.7. Program Menggunakan Fungsi

Pertama kita buat program dengan 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.

Dimulai dari fungsi show_data():

```
# fungsi untuk menampilkan semua data
buku =[]
def show_data():
     if len(buku) <= 0:
          print ("Belum Ada data")
     else:
          print("ID", "Nama Buku")
          for indeks in range(len(buku)):
               print (indeks, buku[indeks])
```

Fungsi di atas akan mengecek isi dari list buku. Jika isinya kosong (len(buku) <= 0) maka tampilkan "Belum Ada Data". Namun, apabila ada isinya, maka tampilkan semua isinya dengan perulangan.

```
# fungsi untuk menambah data
def insert_data():
```

```
    buku_baru = input("Judul Buku : ")
    buku.append(buku_baru)
```

Fungsi di atas akan mengambil input dari user kemudian diisi ke dalam list buku dengan fungsi append(). Fungsi append() adalah fungsi untuk menambahkan item di akhir list. Selain append() ada juga prepend(). Namun, untuk kasus ini, kita pakai append() saja. Selanjutnya membuat fungsi edit_data():

```
# fungsi untuk edit data
def edit_data():
    show_data()
    indeks = int(input("Inputkan ID buku: "))
    if(indeks >= len(buku) or indeks < 0):
        print ("ID salah")
    else:
        judul_baru = input("Judul baru: ")
        buku[indeks] = judul_baru
```

Fungsi di atas akan menampilkan isi dari list buku dengan memanggil fungsi show_data() di dalamnya. Setelah itu, kita meminta user untuk menginputkan ID atau nomer indeks buku yang akan diedit. Lalu kita lakukan pengecekan, jika ID yang diinputkan melebihi dari isi list buku (indeks >= len(buku)), maka tampilkan pesan "ID salah". Namun, apabila tidak melebihi dari isi buku,maka ambil input untuk judul baru dan simpan sesuaiID-nya.

Selanjutnya membuat fungsi delete_data():

```
# fungsi untuk menghapus data
def delete_data():
    show_data()
    indeks = int(input("Inputkan ID buku: "))
    if(indeks >= len(buku) or indeks < 0):
        print ("ID salah")
    else:
        buku.remove(buku[indeks])
```

Hampir sama dengan fungsi edit_data(). Fungsi delete_data() juga harus menampilkan isi list buku dan mengambil ID yang akan dihapus. Kita dapat menghapus item pada list dengan fungsi remove().

Apakah sudah selesai?
Belum, masih ada dua fungsi lagi yang kita butuhkan:

- Fungsi untuk menampilkan menu
- Fungsi untuk keluar (sudah ada di python: exit())

Ok, mari kita buat:

```
# 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!")
```

Fungsi di atas akan menampilkan menu dari 1–5, lalu memanggil fungsi-fungsi yang sudah dibuat berdasarkan menu yang dipilih. Terakhir, kita harus membuat *main loop* programnya.

```
if __name__ == "__main__":
     while(True):
          show_menu()
```

Program akan mengulang terus-menerus sampai fungsi exit() dieksekusi. `if __name__ == "__main__":` adalah blok main di Python. Sebenarnya tanpa ini, programnya sudah bisa dijalankan. Sehingga kode lengkapnya akan seperti ini:

```
# fungsi untuk menampilkan semua data
```

```
buku =[]
def show_data():
     if len(buku) <= 0:
          print ("Belum Ada data")
     else:
          print("ID", "Nama Buku")
          for indeks in range(len(buku)):
               print (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) or indeks < 0):
          print ("ID salah")
     else:
          judul_baru = input("Judul baru: ")
          buku[indeks] = judul_baru

# fungsi untuk menhapus data
def delete_data():
     show_data()
     indeks = int(input("Inputkan ID buku: "))
     if(indeks >= len(buku) or indeks < 0):
          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!")
while (True):
     show_menu()
```

**Output :**

```
----------- MENU----------
[1] Show Data
[2] Insert Data
[3] Edit Data
[4] Delete Data
[5] Exit
PILIH MENU> 2


Judul Buku : Filosofi Teras


----------- MENU----------
[1] Show Data
[2] Insert Data
[3] Edit Data
[4] Delete Data
[5] Exit
```

```
PILIH MENU> 2


Judul Buku : Rich Dad Poor Dad


----------- MENU----------
[1] Show Data
[2] Insert Data
[3] Edit Data
[4] Delete Data
[5] Exit
PILIH MENU> 1


ID Nama Buku
0 Filosofi Teras
1 Rich Dad Poor Dad


----------- MENU----------
[1] Show Data
[2] Insert Data
[3] Edit Data
[4] Delete Data
[5] Exit
PILIH MENU>
```

***Selamat Anda Telah Selesai menjadi programmer Python Pemula.***