



## MODUL 7

### FUNGSI – FUNGSI

#### A. Tujuan

1. Mampu memahami konsep dari fungsi dalam pemrograman
2. Mampu menerapkan penggunaan dari fungsi dalam pemrograman

#### B. Pengertian dan Jenis Fungsi

Fungsi adalah suatu blok kode yang digunakan untuk menjalankan tugas pemrograman. Dalam pemrograman, fungsi sering disebut sebagai subprogram. Salah satu keuntungan fungsi adalah dapat digunakan berulang. Ini berarti bahwa fungsi hanya perlu didefinisikan satu kali namun dapat digunakan berulang kali.

Bab ini akan membahas teknik pendefinisian fungsi dalam python, pengiriman parameter kedalam fungsi, penggunaan fungsi dan hal hal lain yang berkaitan dengan fungsi. Fungsi-fungsi ini dikelompokkan berdasarkan kategorinya. Tujuannya adalah untuk memperkaya wawasan dan meingkatkan kemampuan anda dalam menggunakan bahasa pemrograman Python untuk menangani kasus kasus pemrograman yang sedang anda hadapi.

Dalam pemrograman, fungsi dibedakan menjadi dua jenis, yaitu

##### 1. Fungsi tanpa nilai balik

Fungsi tanpa nilai balik adalah fungsi yang tidak menghasilkan nilai. Fungsi tersebut hanya untuk melakukan proses tertentu. Contohnya seperti fungsi `print()`, Fungsi ini hanya digunakan untuk melakukan pencetakan teks atau bilangan ke layar. Ini berarti bahwa tidak ada nilai yang dihasilkan ketika anda mengeksekusi fungsi tersebut.

##### 2. Fungsi dengan nilai balik

Fungsi dengan nilai balik adalah fungsi yang ketika dipanggil akan menghasilkan nilai. Contohnya seperti fungsi `sqrt()`, Fungsi ini akan menghitung nilai akar kuadrat dari suatu bilangan yang dilewatkan sebagai parameternya.



### C. Memanggil Fungsi

Ketika suatu fungsi dipanggil, eksekusi program akan meloncat atau mengarah ke bagian blok fungsi bersangkutan. Selanjutnya, program akan mengeksekusi satu per satu perintah yang terdapat di dalam fungsi secara beruntun.

Contoh pemanggilan fungsi, yaitu

1. Untuk Fungsi tanpa Nilai Balik

Contoh 7.1

```
cetak = 35
print("Umur= ", cetak)
#output: Umur= 35
```

2. Untuk Fungsi dengan Nilai Balik

Contoh 7.2

```
x = 10
y = 1
print("x = ", x)
print("y = ", y)
print("hasil (x+y)= ", x+y)

#output:
#x = 10
#y = 1
#hasil (x+y)= 11
```

### D. Variabel Global dan Variabel Lokal

Variabel global adalah variabel yang didefinisikan di luar fungsi sedangkan variabel lokal adalah variabel yang didefinisikan di dalam fungsi. Program python mulai mencari variabel lokal terlebih dahulu, kalau ada maka itu yang digunakan. Namun jika tidak ada, pencarian terus ke Global. Jika tidak ada, lalu ke Built-in.

Contoh 7.3



```
#Variabel Global
nama = "POSI"
tahun = "2019"

def help():
    #Variabel Lokal
    nama = "Lab POSI"
    tahun = "19"
    print("nama= %s" % nama)
    print("tahun= %s" % tahun)

print("nama= %s" % nama)
print("tahun= %s" % tahun)
help()
✓ 0.1s

nama= POSI
tahun= 2019
nama= Lab POSI
tahun= 19
```

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()`. Kenapa tidak tampil yang global? Karena Python mulai mencari dari lokal, ke global, dan build-in.

### E. Parameter di Dalam Fungsi

Parameter adalah suatu nilai yang dikirimkan ke dalam fungsi, kemudian akan ikut diproses di dalam badan fungsi. Dengan menggunakan parameter, suatu fungsi dapat memberikan hasil yang dinamis atau berubah ubah setiap fungsi tersebut dipanggil.

Contoh 7.4

```
import math
def main():
    x = math.sqrt(4)
    y = math.sqrt(81)
    print("x= ", x)
    print("y= ", y)
main()
✓ 0.5s

x= 2.0
y= 9.0
```

Fungsi `sqrt()` akan memberikan hasil berbeda sesuai dengan nilai parameter yang dilewatkan ke dalam fungsi. Fungsi `sqrt()` tersimpan di dalam modul `math` dan digunakan untuk menghitung akar kuadrat dari suatu bilangan yang dilewatkan sebagai parameternya.



## F. Mendefinisikan Fungsi di Dalam Fungsi

Pada kasus tertentu di dalam pemrograman, ada kalanya kita ingin membuat fungsi yang berada di dalam fungsi lain.

### Contoh 7.5

```
import time

def countdownTimer(start_minute, start_second):
    total_second = start_minute * 60 + start_second
    while total_second:
        mins, secs = divmod(total_second, 60)
        print(f'{mins:02d}:{secs:02d}', end='\r')
        time.sleep(1)
        total_second -= 1
    def main():
        countdownTimer(0, 10)
        print("Done!")

    main()

✓ 10.2s
Done!
```

Pada kode diatas, fungsi `countdownTimer()` memiliki dua paramter. Parameter pertama untuk menit awal dan parameter kedua untuk detik awal. Fungsi `divmod()` akan membagi total detik (`total_second`) dengan 60, kemudian hasilnya akan disimpan ke dalam variabel `mins` dan sisa baginya akan disimpan ke dalam variabel `secs`. Selanjutnya, dilakukan `print()` isi kedua variabel. Simbol `02d` artinya kita ingin mencetak angka dalam dua digit. Lalu `\r` artinya teks yang dicetak akan ditindih atau ditulis ulang. Berikutnya kita menunda waktu eksekusi selama satu detik dengan fungsi `sleep()` dari modul `time`.

## G. Fungsi yang Mengembalikan Fungsi

Dalam Python, fungsi adalah objek. Dengan demikian, fungsi dapat digunakan sebagai parameter maupun nilai balik dari fungsi lain.

### Contoh 7.6

```
def panggil(func):
    return func
def helloworld():
    return "HELLO WORLD"
def main():
    s = panggil(helloworld())
    print(s)
if __name__ == '__main__':
    main()

✓ 0.8s
HELLO WORLD
```



Contoh diatas menunjukkan fungsi yang diperankan sebagai parameter dan nilai kembali.

## H. Membuat Fungsi Tanpa Nama dengan Operator Lambda

Untuk fungsi sederehana yang mengembalikan nilai, anda dapat mendefinisikan menggunakan lambda. Fungsi seperti ini sering disebut lambda function atau fungsi tanpa nama.

Contoh 7.7

```
def main():
    daftarnama = ["Adi", "Cahyo", "budi", "Dedi"]
    print("Keadaan awal")
    print(daftarnama)

    print("\nMenggunakan sorted():")
    print(sorted(daftarnama))

    daftarnama.sort(key=lambda n: n.lower())

    print("\nKeadaan akhir:")
    print(daftarnama)
if __name__ == '__main__':
    main()
```

✓ 0.8s

Keadaan awal  
['Adi', 'Cahyo', 'budi', 'Dedi']

Menggunakan sorted():  
['Adi', 'Cahyo', 'Dedi', 'budi']

Keadaan akhir:  
['Adi', 'budi', 'Cahyo', 'Dedi']

Pada contoh diatas, kita melewati lambda function ke dalam fungsi sort(). Cara ini digunakan agar elemen elemen di dalam list diurutkan berdasarkan abjad tanpa mempedulikan huruf kecil atau besar.

## I. Rekursi

Suatu fungsi dikatakan sebagai fungsi rekursi jika di dalam badan fungsi tersebut terdapat pemanggilan terhadap dirinya.

Contoh 7.8



```
07_Fungsi.ipynb 5
07_Fungsi.ipynb > Masukkan bilangan= (Press 'Enter' to confirm or 'Escape' to cancel)
+ Code + Markdown | ▶ Run All | Clear Outputs of All Cells | Go To Running Cell | Restart | Interrupt | Variables

import sys

def faktorial(n):
    if n==0:
        return 1
    else :
        return n * faktorial(n-1)
def main():
    bil = int(input("Masukkan bilangan= "))

    if bil < 0:
        print("ERROR")
        sys.exit(1)

    print("%d! = %d" % (bil, faktorial(bil)))
if __name__ == '__main__':
    main()
```

Note : inputkan bilangan pada kotak di atas yang muncul

Output :

```
5! = 120
```

Contoh kasus yang paling sering digunakan untuk menerapkan konsep rekursi adalah pendefinisian fungsi untuk menghitung nilai faktorial dari suatu bilangan bulat non negatif.