



PRAKTIKUM DASAR PEMROGRAMAN

INF1005

LAPORAN PRAKTIKUM 9 : Fungsi & Lambda

Oleh :

Akhmad Qasim

2211102441237

Teknik Informatika
Fakultas Sains & Teknologi
Universitas Muhammadiyah Kalimantan Timur

Samarinda, 2022

Laporan Praktikum 9:

Fungsi & Lambda

9.1. Fungsi

- a.
- b. Syntax

```
1 def rumus_kubus():  
2     sisi = 8  
3     hasil = sisi * sisi * sisi  
4  
5     print('Volume kubus: ', hasil)  
6  
7  
8 rumus_kubus()  
9
```

Output

```
Volume kubus: 512
```

Fungsi merupakan kumpulan perintah yang akan dieksekusi jika kita memanggil fungsi tersebut. Fungsi `rumus_kubus` akan mengeksekusi sejumlah kode, pertama akan dibuat variable `sisi` bernilai 8, kedua melakukan operasi aritmatika, terakhir menampilkan hasil dari operasi aritmatika pada variable `hasil`.

9.2. Parameter pada Fungsi

- a.
- b. Syntax

```
1 def volume_balok(p, l, t, kode_balok="TBxx"):  
2     volume = p * l * t  
3     print("Volume Balok dari balok %s adalah %d" % (kode_balok, volume))  
4  
5  
6 volume_balok(12, 7, 3)  
7
```

Output

Volume Balok dari balok TBxx adalah 252

Parameter digunakan untuk memasukkan data ke dalam fungsi. Parameter bersifat opsional dan dapat dibuat lebih dari dua dengan memisahnya dengan koma. Pada baris kode 1 telah dibuat parameter untuk menampung data panjang, lebar, tinggi, dan kode balok. Pada bagian parameter kode balok digunakan tanda sama dengan yang artinya jika tidak ada parameter yang masuk maka menggunakan default value yang telah di assign ke parameter.

9.3. Fungsi dengan Nilai Balik/Return Value

- a.
- b. Syntax

```
1 def ganjil_atau_genap(n):
2     if n % 2 == 0:
3         return "ini adalah bilangan genap"
4     else:
5         return "ini adalah bilangan ganjil"
6
7
8 result = ganjil_atau_genap(10)
9 print(result)
10
```

Output

```
ini adalah bilangan genap
```

Return digunakan untuk mengembalikan sebuah nilai dan memberhentikan sebuah fungsi atau perulangan. Hasil return nilai pada fungsi ganjil_atau_genap akan disimpan ke dalam variable result yang kemudian mengeluarkan output dari nilai balikkan sebuah fungsi.

9.4. Fungsi Rekursif

- a.
- b.

c. Syntax

```
1  # Membuat fungsi rekursif
2  def rekursif(angka):
3      # Jika angka lebih besar dari -5
4      if angka > -5:
5          # Maka cetak variable angka
6          print(angka)
7          # Angka dikurangi 1
8          angka = angka - 1
9          # Panggil fungsi rekursif dengan parameter angka
10         rekursif(angka)
11     # Jika angka lebih kecil dari -5
12     else:
13         # Cetak angka
14         print(angka)
15
16
17     # Masukkan angka dan diubah menjadi integer
18     masukkan = int(input("masukkan angka: "))
19     # Panggil fungsi rekursif dengan parameter masukkan
20     rekursif(masukkan)
21
```

Output

```
masukkan angka: 5
5
4
3
2
1
0
-1
-2
-3
-4
-5
```

9.5. Fungsi Lambda

a.

b. Syntax

```

1  # Fungsi lambda untuk mengalikan angka dengan dikali 2
2  gandakan_angka = lambda angka: (angka * 2)
3  # Fungsi lambda untuk melakukan pemangkatan angka dengan pangkat 2
4  pangkatkan_angka = lambda angka: (angka ** 2)
5  # Fungsi lambda untuk mengecek apakah angka lebih kecil sama dengan 5
6  # Nilai balikkan berupa boolean
7  cek_bilangan_genap = lambda angka: angka <= 5
8  # Fungsi lambda untuk mengembalikan nilai berupa string "Akhmad Qasim"
9  hello = lambda: "Akhmad Qasim"
10 # Mengeluarkan output dari fungsi lambda hello()
11 print(hello())
12 # Mengeluarkan output dari fungsi lambda gandakan_angka()
13 print(gandakan_angka(5))
14 # Mengeluarkan output dari fungsi lambda pangkatkan_angka()
15 print(cek_bilangan_genap(7))
16 # Membuat list baru dengan nama angka_ajaib
17 angka_ajaib = [1, 2, 3, 4, 5, 6, 7, 8]
18 # Membuat list baru dan mengalikan setiap angka pada list angka_ajaib
19 print(list(map(lambda angka: (angka * 2), angka_ajaib)))
20 # Membuat list baru dan memangkatkan setiap angka pada list angka_ajaib
21 print(list(map(lambda angka: (angka ** 2), angka_ajaib)))
22 # Membuat list baru dan menyaring angka yang lebih kecil sama dengan 5
23 print(list(filter(lambda angka: angka <= 5, angka_ajaib)))

```

Output

```

Akhmad Qasim
10
False
[2, 4, 6, 8, 10, 12, 14, 16]
[1, 4, 9, 16, 25, 36, 49, 64]
[True, True, False, False, False, False, False, False]

```

Fungsi lambda atau juga sering disebut dengan *anonymous function* merupakan fungsi yang lebih sederhana dibandingkan fungsi (def), lambda di peruntukkan untuk membuat perintah yang lebih sederhana. Pembuatan lambda dengan mengetik lambda diikuti dengan parameternya jika ada, apabila tidak memerlukan parameter maka tidak perlu mendeklarasikannya. Penulisan ekspresi dalam langsung ditulis setelah tanda titik dua. Lambda digunakan untuk penulisan fungsi yang sederhana apabila ingin membuat fungsi yang lebih kompleks disarankan menggunakan fungsi (def).

Hasil dan analisa dari Percobaan & Latihan 9.1 sampai dengan 9.5 :

- 9.1. Pada latihan praktikum pertama, kita dapat mengetahui sebuah fungsi digunakan untuk menjalankan satu atau lebih perintah dengan hanya mendeklarasikan satu perintah saja. Hal ini juga dapat disebut dengan nama enkapulasi.
- 9.2. Latihan kedua kita dapat mengetahui bahwa sebuah fungsi dapat memiliki parameter yang berfungsi untuk memasukkan data ke dalam sebuah fungsi. Pendeklarasian parameter akan membuat sebuah variable local.
- 9.3. Pada latihan ketiga kita dapat mengetahui jika sebuah nilai yang telah diolah oleh sebuah fungsi dapat dikembalikan nilai nya dan disimpan ke dalam sebuah variable.
- 9.4. Kita juga dapat menyelesaikan permasalahan tertentu dengan menggunakan fungsi rekursif. Fungsi rekursif merupakan sebuah metode yang memanggil fungsi itu sendiri.
- 9.5. Fungsi lambda merupakan fungsi yang lebih sederhana daripada fungsi (def).
Terkadang kita ingin membuat sebuah fungsi yang hanya melakukan perintah yang sederhana. Oleh karena itu lebih disarankan menggunakan fungsi lambda dibandingkan fungsi (def).

Kesimpulan :

Pada praktikum kali ini, dapat disimpulkan bahwa python mendukung sebuah fungsi yang digunakan untuk mempermudah pengembang dalam menuliskan sebuah kode atau lebih karena tidak perlu menyebutkan perintah itu berulang kali. Dengan menggunakan fungsi (def) kita dapat mengeksekusi satu atau lebih baris kode untuk dieksekusi hanya dengan menyebutkan nama fungsi tersebut. Kita juga dapat menggunakan parameter pada sebuah fungsi untuk memasukkan data kedalam fungsi dengan membuat parameter. Python juga mendukung adanya fungsi kembalikan (return) yang berfungsi untuk mengembalikan data yang telah diolah oleh sebuah fungsi. Fungsi rekursif merupakan sebuah metode untuk melakukan pemanggilan sebuah fungsi itu sendiri, ini diperlukan pada studi kasus tertentu, misalkan dalam mencari hasil dari faktorial. Python memiliki fungsi yang lebih sederhana disebut dengan nama lambda. Fungsi ini memungkinkan kita dalam membuat sebuah fungsi yang hanya memerlukan sedikit kode atau perintah.