

MODUL 4

Fungsi



CAPAIAN PEMBELAJARAN

1. Mampu membuat Fungsi menggunakan python
2. Mampu memanggil Fungsi menggunakan python



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Python
2. VS Code



DASAR TEORI

Fungsi (function) merupakan blok / unit dari program yang digunakan untuk mengerjakan suatu pekerjaan tertentu dan mengembalikan hasilnya ke pemanggil. Fungsi dimaksudkan utk membuat reusable code. artinya hanya di definisikan sekali saja, dan kemudian bisa digunakan berulang kali dari tempat lain di dalam program. Fungsi memecah keseluruhan program menjadi bagian – bagian yang lebih kecil . Dengan semakin besarnya program, maka fungsi akan membuatnya menjadi lebih mudah diorganisir dan *dimanage*.

Bentuk umum fungsi adalah sebagai berikut :

```
def namaFungsi([arg1, arg2,...argN]):  
    statemen_isi_fungsi  
    [return ekspresi]
```

Keterangan :

- Kata kunci def diikuti oleh nama fungsi, tanda kurung dan tanda titik dua (:) menandai header (kepala) fungsi.
- Parameter / argumen (arg1,..argN) bersifat optional, adalah input dari luar yang akan diproses di dalam tubuh fungsi.

- Setelah itu diletakkan baris – baris pernyataan (statements). Jangan lupa indentasi untuk menandai blok fungsi.
- return bersifat opsional. Gunanya adalah untuk mengembalikan suatu nilai expression dari fungsi / Nilai kembalian.

#fungsi tanpa parameter tanpa nilai kembalian

```
def sapa():
    print("Hallo apa kabar")
```

#fungsi dengan 1 parameter tanpa nilai kembalian

```
def salam(nama):
    print("Hallo, "+nama+" Selamat Pagi ")
```

#fungsi dengan 2 parameter dengan nilai kembalian

```
def kurang(a, b):
    c = a - b
    return c
```

Memanggil fungsi

Bila fungsi sudah didefinisikan, maka ia sudah bisa dipanggil dari tempat lain di dalam program. Untuk memanggil fungsi, caranya adalah dengan mengetikkan nama fungsi dan jika diperlukan dilengkapi dengan paramaternya.

Untuk contoh fungsi2 diatas dapat dipanggil dengan cara berikut :

```
sapa()
salam("Cosmas")
hasil = kurang(5,3)
print("Hasil ",hasil)

atau

print("Hasil ",kurang(5,3))
```

Akan menghasilkan :

```
Hallo apa kabar
Hallo, Cosmas Selamat Pagi
Hasil 2
```

Pemanggilan yang salah :

```
sapa("Cosmas")
```

Akan menghasilkan output berikut dikarenakan fungsi sapa tidak meminta parameter:

Traceback (most recent call last):

```
File "fung1.py", line 17, in <module>
    sapa("Cosmas")
TypeError: sapa() takes 0 positional arguments but 1 was given
```

```
salam()
```

Akan menghasilkan output berikut dikarenakan fungsi salam butuh 1 parameter:

```
Traceback (most recent call last):
  File "fung1.py", line 18, in <module>
    salam()
TypeError: salam() missing 1 required positional argument: 'nama'
```

```
hasil = kurang(5)
```

Akan menghasilkan output berikut dikarenakan fungsi kurang butuh 2 parameter:

```
Traceback (most recent call last):
  File "fung1.py", line 18, in <module>
    hasil = kurang(5)
TypeError: kurang() missing 1 required positional argument: 'b'
```

Pengiriman parameter bisa dilakukan juga dengan menyebut nama variabelnya

```
hasil = kurang(b=10, a=7)
print(hasil)
```

akan menghasilkan output : -3 bukan 3 dikarenakan meskipun secara posisi 10 lebih dahulu dibanding 7, tetapi disitu dituliskan bahwa 10 adalah isi parameter b sedangkan 7 isi parameter a, sehingga $a - b$ menjadi $7 - 10 = -3$

Parameter default

Suatu fungsi dapat memiliki parameter default, sehingga apabila parameter tersebut tidak dikirimkan oleh pemanggil maka tidak akan terjadi error, dan nilai yang dipakai adalah nilai default.

```
def jumlah(a, b=3):
    return a + b
```

```
print(jumlah(6))
```

Meskipun fungsi jumlah butuh 2 parameter, tetapi parameter yg kedua, yaitu b sudah didefault angka 3, maka saat dipanggil hanya dengan 1 parameter → jumlah(6), otomatis parameter kedua (b) diisi angka 6, sehingga keluaran script di atas adalah **9**

Struktur Kontrol dalam fungsi

Fungsi adalah potongan program, sehingga penggunaan struktur kontrol di dalam fungsi sama dengan program biasa.



PRAKTIK

1. Buka VS Code
2. Buat file baru di dalam direktori c:\python dengan nama fungsi1.py
3. Ketikkan script berikut ini:

```
1  def tambah(a, b):
2      return a + b
3
4  print(tambah(4, 2))
5
6  str1 = "STMIK"
7  str2 = "AKAKOM"
8  print(str1, str2)
9
10 x = [1,2,3]
11 y = [4,5,6]
12 print(tambah(x,y))
```
4. Simpan script tersebut
5. Kembali ke command prompt, masuk ke c:\python
6. Untuk menjalankan ketikkan : py fungsi1.py atau python fungsi1.py, amati hasilnya!
Apakah muncul error karena parameter yang digunakan 1) int, 2) string 3) list ?
Jelaskan dampak penggunaan operator + (plus) untuk ketiga jenis tipe data tersebut!
7. Modifikasilah program fungsi1.py tersebut dengan ditambah script berikut ini:

```

14  nama = "andi"
15  nilai = 75
16  print(tambah(nama, nilai))

```

8. Jalankan kembali dan amati apa yang terjadi! Jelaskan penyebabnya!
9. Buat lagi file baru bernama fungsi2.py
10. Ketikkan script berikut ini :

```

1  def cekbilangan(bilangan):
2      if (bilangan % 2 == 0):
3          hasil = "GENAP"
4      else :
5          hasil = "GANJIL"
6      return hasil
7
8  angka = int(input("Masukkan sembarang bilangan :"))
9  hasilcek = cekbilangan(angka)
10 print("Hasil Pengecekan %d adalah bilangan %s " %(angka, hasilcek))

```

11. Jalankan script tersebut dan amati hasilnya!
12. Mengacu pada program fungsi2.py, buatlah program baru bernama fungsi2b.py, yang isinya dapat melakukan pengecekan apakah suatu bilangan disebut Positif, Negatif atau NOL.
13. Buat lagi file baru bernama fungsi3.py
14. Ketikkan script berikut ini :

```

1  def pembagian(pembilang, penyebut):
2      hasil = pembilang / penyebut
3
4  x = 75
5  y = 3
6  result = pembagian(x)
7  print("Hasil Bagi = ",result)

```

15. Untuk menjalankan ketikkan : py fungsi3.py, maka akan muncul error. Perbaikilah program tersebut sehingga saat dijalankan dapat menghasilkan keluaran :

```

Hasil Bagi = 25.0

```

16. Modifikasilah program tersebut pada baris 5, nilai X diisi 0, menjadi sbb :

```

4  x = 75
5  y = 0

```

Kemudian dijalankan, maka akan muncul error sbb :

```

Traceback (most recent call last):
  File "fungsi2.py", line 6, in <module>
    result = pembagian(x, y)
  File "fungsi2.py", line 2, in pembagian
    hasil = pembilang / penyebut
ZeroDivisionError: division by zero

```

17. Perbaiki program tersebut, sehingga saat dijalankan muncul pesan sebagai berikut :
Hasil Bagi = Penyebut dalam pembagian tidak boleh NOL



LATIHAN

Bagian berikut ini tidak wajib dikerjakan, tetapi akan bagus jika dipahami sebagai latihan.

Berikut ini contoh program yang memanfaatkan function yang memanggil function yang lain untuk membuat bingkai di tulisan :

```
1 def bingkai(str):
2     pjg = len(str) # panjang string
3     garis(pjg)
4     cetak(str)
5     garis(pjg)
6
7 def cetak(str):
8     print("| "+str+" |")
9
10 def garis(pjg):
11     print("+",end='')
12     for i in range(pjg+2):
13         print("-",end='')
14     print("+")
15
16 bingkai("Hallo Apa Kabar")
17 bingkai("Ini Adalah Tulisan Yang Otomatis Ada di dalam Kotak")
```



TUGAS

1. Buatlah program sesuai kasus berikut ini :
 - a. Buatlah function, yang apabila dipanggil dengan perintah :
`tulis(3, "STMIK AKAKOM")`
Akan menghasilkan tulisan STMIK AKAKOM sebanyak 3 kali, sbb:
STMIK AKAKOM

STMIK AKAKOM
STMIK AKAKOM

- b. Buatlah program yang berisi fungsi untuk mengkonversi nilai angka menjadi nilai huruf. Masukan berupa nilai UTS, Nilai UAS, Nilai Tugas, dengan bobot masing-masing : UTS=30%, UAS=30%, Tugas=40%. Kemudian otomatis akan dihasilkan nilai hurufnya. Gambaran tampilan sbb :

```
Nilai UTS : 80 [ENTER]
Nilai UAS : 60 [ENTER]
Nilai Tugas : 90 [ENTER]
Nilai Akhir anda = 78
Anda Mendapat Nilai = B
```

Aturan Penilaian nilai huruf adalah sbb :

```
Nilai 81 – 100 → A
Nilai 61 – 80 → B
Nilai 41 – 60 → C
Nilai 21 – 40 → D
Nilai <=20 → E
```



REFERENSI
