

Lembar Kegiatan Peserta Didik (LKPD)

Mata Kuliah : Bahasa Pemrograman Terstruktur

Semester : 1

SKS : 2

Chapter 6 – Functions

Tujuan Umum Praktikum:

1. Mahasiswa memahami kegunaan function
 2. Mahasiswa dapat memahami cara membuat function dan menggunakannya dalam program
 3. Mahasiswa memahami cara mengimport function
 4. Mahasiswa dapat memahami kegunaan variabel global dan lokal serta dapat menggunakannya dalam program
-

Praktikum 1

Tujuan Praktikum:

Mahasiswa memahami kegunaan function

Langkah Kerja:

1. Buka Jupyter Notebook atau Spyder
2. Tuliskan script berikut ini

```
print('-----')
print(' ***** HELLO WORLD ***** ')
print('-----')
print('-----')
print(' ***** HELLO WORLD ***** ')
print('-----')
```

3. Jalankan script di atas, kemudian amatilah outputnya
4. Selanjutnya, tuliskan script berikut ini

```
def buatTitle():
    print('-----')
    print(' ***** HELLO WORLD ***** ')
    print('-----')

buatTitle()
buatTitle()
```

5. Jalankan script di atas, lalu amatilah outputnya
6. Bandingkan output dari program nomor 2 dan 4! Samakah? Apa yang bisa disimpulkan dari kedua program tersebut?

Dalam menghasilkan output, kedua program tersebut menghasilkan output yang sama. Akan tetapi, dalam menjalankan program, keduanya berbeda. Pada program pertama untuk

menghasilkan program seperti itu, harus dituliskan satu persatu. Sedangkan pada program kedua, function akan didefinisikan terlebih dahulu dengan menggunakan 'def'. 'Def' dapat membuat program menjadi lebih efisien dan struktur.

7. Manfaat apa yang didapatkan ketika membuat function dilihat dari sisi efisiensi penulisan kode program?

Penulisan program akan menjadi lebih efisien dalam proses koding dan terstruktur, serta terhadap konsisten dalam alur program. Setiap kali programmer membutuhkan sebuah function yang sudah dibuatnya untuk melakukan suatu proses, maka programmer cukup memanggil nama functionnya saja, tanpa harus koding lagi secara berulang-ulang dengan baris program yang sama.

Praktikum 2

Tujuan Praktikum:

Mahasiswa dapat memahami cara membuat function dan menggunakannya dalam program

Langkah Kerja:

1. Berikut ini adalah contoh program untuk menghitung luas segitiga menggunakan function. Ketikkan kode program berikut ini

```
def luasSegitiga(..., ...):  
    luas = a * t / 2  
    return ...  
  
alas = 10  
tinggi = 20  
print('Luas segitiga dg alas ', alas,  
      ' dan tinggi ', tinggi,  
      ' adalah ', luasSegitiga(..., ...))
```

2. Lengkapi titik-titik dari kode program di atas sehingga dapat dijalankan tanpa error dan dihasilkan output yang benar. (Program tersebut digunakan untuk menghitung luas segitiga dengan alas 10 satuan dan tinggi 20 satuan).

```
def luasSegitiga(alas ,tinggi):  
    luas = alas * tinggi / 2  
    return luas  
  
alas = 10  
tinggi = 20  
print('Luas segitiga dengan alas ', alas,  
      ' dan tinggi ', tinggi,  
      ' adalah ', luasSegitiga(alas,tinggi)  
    )
```

3. Selanjutnya buat program berikutnya seperti berikut

```
def luasSegitiga2(..., ...):
    luas = a * t / 2
    print('Luas segitiga dg alas ', ...,
          ' dan tinggi ', ...,
          ' adalah ', ...)

alas = 10
tinggi = 20
luasSegitiga2(..., ...)
```

4. Lengkapi titik-titik dari kode program di atas sehingga dapat dijalankan tanpa error dan dihasilkan output yang benar. (Program tersebut digunakan untuk menghitung luas segitiga dengan alas 10 satuan dan tinggi 20 satuan).

```
def luasSegitiga2(alas, tinggi):
    luas = alas * tinggi / 2
    print('Luas segitiga dengan alas ', alas,
          ' dan tinggi ', tinggi, ' adalah ', luas)
alas = 10
tinggi = 20
luasSegitiga2(alas, tinggi)
```

5. Amatilah perbedaan cara penulisan dan pemanggilan function dari program nomor 1 dan 3. Apakah kedua program menghasilkan output yang sama?

Ya, keduanya menghasilkan output yang sama

6. Modifikasilah kode program hasil nomor 1, tambahkan baris kode program untuk menghitung luas segitiga kedua dengan alas 15 satuan dan tinggi 45 satuan

```
def luasSegitiga(alas, tinggi):
    luas = alas * tinggi / 2
    return luas

alas = 10
tinggi = 20
print('Luas segitiga dengan alas ', alas,
      ' dan tinggi ', tinggi,
      ' adalah ', luasSegitiga(alas,tinggi)
    )
alas = 15
tinggi = 45
print('Luas segitiga dengan alas ', alas,
      ' dan tinggi ', tinggi,
      ' adalah ', luasSegitiga(alas,tinggi)
    )
```

7. Modifikasilah kode program hasil nomor 3, tambahkan baris kode program untuk menghitung luas segitiga kedua dengan alas 15 satuan dan tinggi 45 satuan

```
print('----Nomor 4----')
def luasSegitiga2(alas, tinggi):
    luas = alas * tinggi / 2
    print('Luas segitiga dengan alas ', alas,
          ' dan tinggi ', tinggi, ' adalah ', luas)
alas = 10
```

```
tinggi = 20
luasSegitiga2(alas, tinggi)
#Nomor 7
alas = 15
tinggi = 45
luasSegitiga2(alas, tinggi)
```

8. Apabila Anda ingin menghitung luas total kedua segitiga tersebut, maka bentuk function manakah yang paling mudah digunakan? Jelaskan alasannya! Lalu implementasikan ke dalam program.

Menurut saya program yang lebih mudah digunakan adalah program pada nomor 2. Pada program no 2, perintah 'print' atau perintah outputnya sudah dituliskan pada bagian 'def', sehingga ketika kita ingin mengeluarkan beberapa output kita hanya perlu menuliskan functionnya saja.

Praktikum 3

Tujuan Praktikum:

Mahasiswa dapat memahami cara mengimport function

Langkah Kerja:

1. Ketikkan kode program berikut ini menggunakan Spyder

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

def kali(a, b):
    hasil = a * b
    return hasil

def kurang(a, b):
    hasil = a - b
    return hasil

def bagi(a, b):
    hasil = a / b
    return hasil
```

2. Simpan kode program di atas dengan nama file **operation.py**
3. Buat kode program yang baru sebagai berikut

```
from operation import *

a = 10
b = 7
print(a, '+', b, '=', jumlah(a, b))
print(a, '-', b, '=', kurang(a, b))
print(a, 'x', b, '=', kali(a, b))
print(a, '/', b, '=', bagi(a, b))
```

4. Simpan program dari nomor 3 dengan nama file operasi.py pada direktori yang sama dengan operator.py
5. Modifikasilah kode program dari nomor 3 untuk menghitung dan menampilkan operasi berikut ini menggunakan function-function yang ada dalam operation.py

- a. $2 + 4 * 6 - 4$
- b. $(4 + 7) * (6 - 9)$
- c. $(10 + 2) / (7 + 5) / (12 - 34)$

Praktikum 4

Tujuan Praktikum:

Mahasiswa dapat memahami kegunaan variabel global dan lokal

Langkah Kerja:

1. Buatlah kode program berikut ini

```
def jumlah(a, b):  
    hasil = a + b  
  
a = 10  
b = 20  
jumlah(a, b)  
print(hasil)
```

2. Jalankan program di atas, dan amati outputnya! Apa yang terjadi?

Terjadi eror

3. Mengapa muncul pesan error yang menyatakan bahwa variabel 'hasil' tidak dikenal (not defined)?

Karena variabel didefinisikan pada bagian 'def'. Sedangkan pada program, 'hasil' di tuliskan di luar bagian 'def', sehingga 'hasil' tidak dikenal.

4. Bagaimana solusi dari error tersebut supaya variabel 'hasil' bisa dikenali di luar function?

'Hasil' harus didefinisikan terlebih dahulu. Bisa dengan menuliskan seperti ini sebelum 'hasil' dituliskan.

Hasil = a + b

5. Ketikkan program berikut ini

```
def myFunction():  
    a = 20  
    print(a)  
  
a = 10  
myFunction()  
print(a)
```

6. Jalankan program di atas, dan amati outputnya!

7. Jelaskan mengapa dihasilkan output berikut ini?

20

10

Karena pada program yang dituliskan terlebih dahulu adalah myFunction(). Dimana perintah ini akan menghasilkan output 20. Setelah itu barulah perintah 'print(a)' yang akan menghasilkan output 10 karena di luar bagian function myFunction variabel a berisi 10. Pada python, program akan dijalankan secara urut.

8. Apa perbedaan variabel a di dalam function myFunction dengan variabel a yang ada di luar function?

Variabel a pada function myFunction adalah kepemilikan dari function myFunction. Jadi untuk memanggil a pada function myFunction, kita hanya perlu memanggilnya functionnya

saja tanpa perlu mendeklarasikan a lagi. Semua nilai a yang dipanggil pada myFunction akan bernilai sama yaitu 20. Sedangkan a di luar function, merupakan deklarasi tersendiri. Untuk menghasilkan output a, kita bisa menghasilkan output yang berbeda-beda dengan cara membuat a lagi dengan nilai yang berbeda, kemudian menuliskan 'print(a)' lagi dibawahnya.

Latihan

1. Buatlah function **isPythagoras(a, b, c)** yang digunakan untuk menentukan apakah pasangan a, b, c yang merupakan sisi-sisi sebuah segitiga merupakan triple Pythagoras atau bukan. Function tersebut menghasilkan nilai boolean: True apabila a, b, c merupakan triple Pythagoras, dan False jika ketiganya bukan triple Pythagoras.

Keterangan: c merupakan sisi miring segitiga

Untuk mengecek benar tidaknya function yang Anda buat, ujilah dengan beberapa pasangan nilai a, b, c berikut ini

- a. a = 3, b = 4, c = 5 -> True
- b. a = 5, b = 9, c = 12 -> False
- c. a = 8, b = 6, c = 10 -> True
- d. a = 7, b = 8, c = 11 -> False

2. Buatlah function **starFormation1(n)** yang digunakan untuk mencetak output berupa formasi bintang berikut ini

```
*
* *
* * *
* * * *
```

(contoh output function jika n = 4)

Dan buat pula function **starFormation2(n)** yang digunakan untuk mencetak output berupa formasi bintang berikut ini

```
* * * *
* * *
* *
*
```

(contoh output function jika n = 4)

Berdasarkan kedua function tersebut, gunakan keduanya untuk membentuk formasi bintang sebagai berikut

```
*
* *
* * *
* * * *
* * *
* *
*
```

(contoh output program jika n = 7)

3. Buatlah function **faktorial(n)** yang digunakan untuk menghitung nilai n faktorial. Output dari function ini adalah sebuah bilangan yang merupakan nilai dari n faktorial tersebut.
Gunakan function tersebut untuk menghitung nilai dari:
 - a. $C(5, 3)$
 - b. $P(10, 7)$
4. Buatlah sebuah file Python dengan nama **statistik.py** yang hanya berisi function-function berikut ini:
 - a. Buatlah function **sum(a, b, c, d, ...)** dengan jumlah parameter tidak terbatas, yang digunakan untuk mencari jumlah total seluruh nilai parameter yang berupa bilangan. Misalnya: $\text{sum}(3, 5, 6, 7)$ akan menghasilkan nilai 21, $\text{sum}(1, 0, 0, 2, 4)$ akan menghasilkan nilai 7.

Petunjuk: pelajari modul teori bab 6 untuk membuat function dengan parameter dinamis seperti pada kasus ini
 - b. Buatlah function **average(a, b, c, d, ...)** dengan jumlah parameter tidak terbatas, yang digunakan untuk mencari rata-rata dari seluruh nilai parameter yang berupa bilangan. Manfaatkan function $\text{sum}(a, b, c, d, \dots)$ untuk menghitung jumlah total data dalam proses perhitungan rata-ratanya.
 - c. Buatlah function **maks(a, b, c, d, ...)** dengan jumlah parameter tidak terbatas, yang digunakan untuk mencari nilai maksimum dari seluruh nilai parameter.
 - d. Buat pula function **min(a, b, c, d, ...)** dengan jumlah parameter tidak terbatas, yang digunakan untuk mencari nilai minimum dari seluruh nilai parameter.
5. Gunakan function-function yang ada di statistik.py untuk mencari rata-rata, nilai maks, dan minimum dari data-data berikut ini:
 - a. 5, 10, 4, 9, 30, 16, 2, 11
 - b. 81, 98, 12, 83, 45, 77, 69, 30, 56

Link GitHub: https://github.com/NIHanifah/K3520058_NurIsnainiHanifah_Chapter6.git