Lembar Praktikum

**Mata Kuliah** : Pemrograman Terstruktur  
**Semester** : 2  
**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



1. Jalankan script di atas, kemudian amatilah outputnya
2. Selanjutnya, tuliskan script berikut ini



1. Jalankan script di atas, lalu amatilah outputnya
2. Bandingkan output dari program nomor 2 dan 4! Samakah? Apa yang bisa disimpulkan dari kedua program tersebut?
3. Manfaat apa yang didapatkan ketika membuat function dilihat dari sisi efisiensi penulisan kode program?

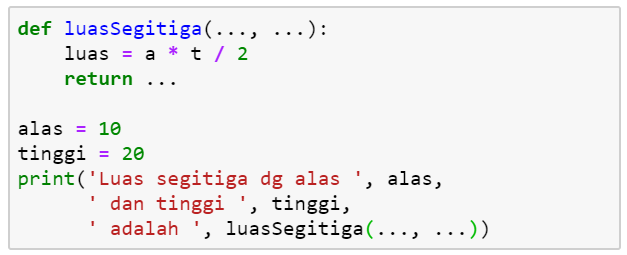
# Praktikum 2

**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



1. 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).
2. Selanjutnya buat program berikutnya seperti berikut



1. 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).
2. Amatilah perbedaan cara penulisan dan pemanggilan function dari program nomor 1 dan 3. Apakah kedua program menghasilkan output yang sama?
3. Modifikasilah kode program hasil nomor 1, tambahkan baris kode program untuk menghitung luas segitiga kedua dengan alas 15 satuan dan tinggi 45 satuan
4. Modifikasilah kode program hasil nomor 3, tambahkan baris kode program untuk menghitung luas segitiga kedua dengan alas 15 satuan dan tinggi 45 satuan
5. Apabila Anda ingin menghitung luas total kedua segitiga tersebut, maka bentuk function manakah yang paling mudah digunakan? Jelaskan alasannya! Lalu implementasikan ke dalam program.

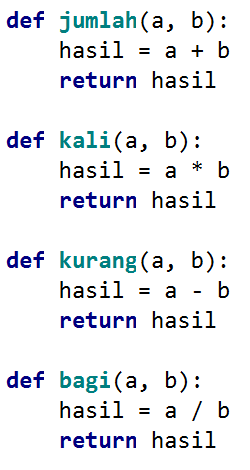
# Praktikum 3

**Tujuan Praktikum:**

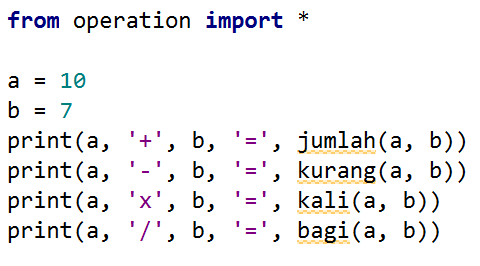
Mahasiswa dapat memahami cara mengimport function

**Langkah Kerja:**

1. Ketikkan kode program berikut ini menggunakan Spyder



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



1. Simpan program dari nomor 3 dengan nama file operasi.py pada direktori yang sama dengan operator.py
2. Modifikasilah kode program dari nomor 3 untuk menghitung dan menampilkan operasi berikut ini menggunakan function-function yang ada dalam operation.py
3. 2 + 4 \* 6 – 4
4. (4 + 7) \* (6 - 9)
5. (10 + 2) / (7 + 5) / (12 - 34)

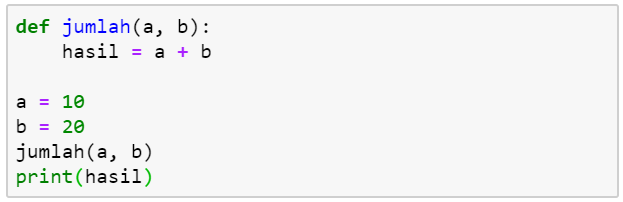
# Praktikum 4

**Tujuan Praktikum:**

Mahasiswa dapat memahami kegunaan variabel global dan lokal

**Langkah Kerja:**

1. Buatlah kode program berikut ini



1. Jalankan program di atas, dan amati outputnya! Apa yang terjadi?
2. Mengapa muncul pesan error yang menyatakan bahwa variabel ‘hasil’ tidak dikenal (not defined)?
3. Bagaimana solusi dari error tersebut supaya variabel ‘hasil’ bisa dikenali di luar function?
4. Ketikkan program berikut ini



1. Jalankan program di atas, dan amati outputnya!
2. Jelaskan mengapa dihasilkan output berikut ini?

20  
10

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

# 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

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

1. 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:

1. C(5, 3)
2. P(10, 7)
3. Buatlah sebuah file Python dengan nama **statistik.py** yang hanya berisi function-function berikut ini:
4. Buatlah function **sum(a, b, c, d, …)** dengan jumlah parameter tidak terbatas, yang digunakan untuk mencari jumlah total seluruh nilai parameternya yang berupa bilangan. Misalnya: sum(3, 5, 6, 7) akan menghasilkan nilai 21, 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

1. Buatlah function **average(a, b, c, d, …)** dengan jumlah parameter tidak terbatas, yang digunakan untuk mencari rata-rata dari seluruh nilai parameternya yang berupa bilangan. Manfaatkan function sum(a, b, c, d, …) untuk menghitung jumlah total data dalam proses perhitungan rata-ratanya.
2. Buatlah function **maks(a, b, c, d, …)** dengan jumlah parameter tidak terbatas, yang digunakan untuk mencari nilai maksimum dari seluruh nilai parameternya.
3. Buat pula function **min(a, b, c, d, …)** dengan jumlah parameter tidak terbatas, yang digunakan untuk mencari nilai minimum dari seluruh nilai parameternya.
4. Gunakan function-function yang ada di statistik.py untuk mencari rata-rata, nilai maks, dan minimum dari data-data berikut ini:
   1. 5, 10, 4, 9, 30, 16, 2, 11
   2. 81, 98, 12, 83, 45, 77, 69, 30, 56