



# **PRAKTIKUM DASAR PEMROGRAMAN**

**INF1005**

## **LAPORAN PRAKTIKUM 10 : Modularisasi**

Oleh :

*Akhmad Qasim*

*2211102441237*

Teknik Informatika  
Fakultas Sains & Teknologi  
Universitas Muhammadiyah Kalimantan Timur

Samarinda, 2022

# Laporan Praktikum 10: Modularisasi

## Latihan :

### 10.1. Modul

- a.
- b.
- c. Modul

```
1 def rumus_tabung(jari_jari, tinggi):  
2     hasil = 3.14 * jari_jari * jari_jari * tinggi  
3     print('Volume tabung: ', hasil)  
4
```

#### Syntax

```
1 import RumusTabung  
2  
3 RumusTabung.rumus_tabung(10, 20)  
4
```

#### Output

```
Akhmad  
Qasim  
2211102441237  
  
Jumlah index pada variable nama adalah 3
```

Modul merupakan kumpulan dari fungsi, kelas, dan variable yang memiliki fungsi yang telah ditata dan dikelompokkan berdasarkan penggunaannya. Apabila ingin digunakan akan lebih mudah jika menggunakan modul dalam memanggil fungsi yang diinginkan.

### 10.2. Lokasi Modul

- a.

b. Syntax

```
1  import RumusTabung
2  import sys
3  import site
4
5  # display RumusTabung
6  print(RumusTabung.__file__)
7
8  # display python path
9  print(sys.path)
10
11 # display default installation directory
12 print(site.getsitepackages())
13 |
```

Output

```
C:\Users\dhalia\iCloudDrive\Documents\Study\Pemrograman Dasar\Laporan 10\Source Code\RumusTabung.py
['C:\\Users\\dhalia\\iCloudDrive\\Documents\\Study\\Pemrograman Dasar\\Laporan 10\\Source Code', 'C:\\Use
['C:\\Users\\dhalia\\iCloudDrive\\Documents\\Study\\Pemrograman Dasar\\venv', 'C:\\Users\\dhalia\\iCloudD
```

Lokasi modul akan dicari oleh interpreter python yang membaca program dari atas kebawah secara berurut. Menampilkan lokasi modul dapat menggunakan `__file__` yang merupakan sebuah standard library dari python yang mencari lokasi modul. `sys.path` digunakan untuk menampilkan lokasi python. `Site.getsitepackages()` merupakan sebuah fungsi dari modul `site` yang digunakan untuk mencari lokasi interpreter python.

### 10.3. Import Modul

a.

b. Modul

```
1  def pertambahan(a, b):
2      return a + b
3
4
5  def pengurangan(a, b):
6      return a - b
7
8
9  def perkalian(a, b):
10     return a * b
11
12
13 def pembagian(a, b):
14     return a / b
15
16
17 def pangkat(a, b):
18     return a ** b
19
20
21 def akar(a):
22     return a ** 0.5
23
24
25 def modulus(a, b):
26     return a % b
27
```

Syntax

```
1  import Aritmatika
2
3  print(Aritmatika.pertambahan(1, 2))
4  print(Aritmatika.pengurangan(10, 2))
5  print(Aritmatika.perkalian(7, 2))
6  print(Aritmatika.pembagian(10, 2))
7  print(Aritmatika.pangkat(2, 2))
8  print(Aritmatika.akar(25))
9  print(Aritmatika.modulus(10, 3))
10
```

#### Output

```
3
8
14
5.0
4
5.0
1
```

Program mengimport modul Aritmatika untuk menggunakan operasi aritmatika yang terdapat pada modul Aritmatika. Fungsi pertambahan berfungsi untuk menjumlahkan 2 angka dan mengembalikan nilai. Fungsi pengurangan berfungsi untuk mengurangi nilai pertama dari nilai kedua dan mengembalikan nilainya. Fungsi perkalian berfungsi untuk mengalikan 2 nilai dan mengembalikan nilainya. Fungsi pembagian berfungsi untuk membagikan sebuah nilai pertama dari nilai kedua dan mengembalikan hasil pembagian. Fungsi pemangkatan berfungsi untuk memangkatkan angka pertama dengan angka kedua dan mengembalikan hasil dari perpangkatan. Fungsi akar berfungsi sebagai mengakarkan sebuah nilai dan mengembalikan nilai tersebut. Fungsi modulus berfungsi sebagai operasi modulus dan mengembalikan nilai hasil sisa bagi angka pertama dengan angka kedua.

#### **Hasil dan analisa dari Percobaan & Latihan 10.1 sampai dengan 10.3 :**

- 10.1. Modul merupakan suatu file yang berisi fungsi, variable, dan kelas yang saling terkait sehingga mudah dimengerti dalam penggunaannya.
- 10.2. Interpreter Python dapat melakukan pencarian nama dan lokasi sebuah modul.
- 10.3. Modul dapat di import dan digunakan dalam untuk mempermudah dalam pengerjaan tanpa harus membuat fungsi baru yang memerlukan waktu dan dapat terjadinya duplikasi kode yang membuat kode tidak efisien.

#### **Kesimpulan :**

Modularisasi merupakan teknik pengelompokkan fungsi, variable, dan kelas yang memiliki ketergantungan. Modularisasi juga tidak bergantung pada modul lain, yang artinya modul dapat berdiri sendiri. Tujuan dari modularisasi ini adalah untuk membuat kode yang lebih mudah diorganisir dan terstruktur.