COMPUTER
PROGRAMING 2

PENGENALAN & KONSEP OOP

MODUL 7

2023

DAFTAR ISI

PENGENALAN & KONSEP OOP

Dalam proses penulisan program, kita biasanya akan terlibat sebuah “diskusi” tentang kapan struktur sebuah kode dikatakan bagus. Apakah yang menganut konsep **Object Oriented Programming (OOP)**, atau yang mengikuti konsep prosedural, atau bahkan konsep fungsional?

Mungkin bagi pemula, kita belum pernah menulis kode program yang cukup besar sampai ribuan baris kode dan puluhan bahkan ratusan file. Kalau kita sudah pernah melakukan hal itu entah projek akhir sekolah atau projek akhir kuliah, kita baru akan sadar bahwa ternyata semua hal bisa menjadi sangat rumit jika kita tidak mengadopsi pendekatan yang baik.

Pada tulisan kali ini dan beberapa tulisan kedepan, kita akan membahas tentang **Object Oriented Programming (OOP)** atau yang biasa dialihbahasakan menjadi **Pemrograman Berorientasi Objek (PBO)** [1]. Tentunya tetap dalam bingkai bahasa pemrograman Python.

A. Pengertian Paradigma Pemrograman

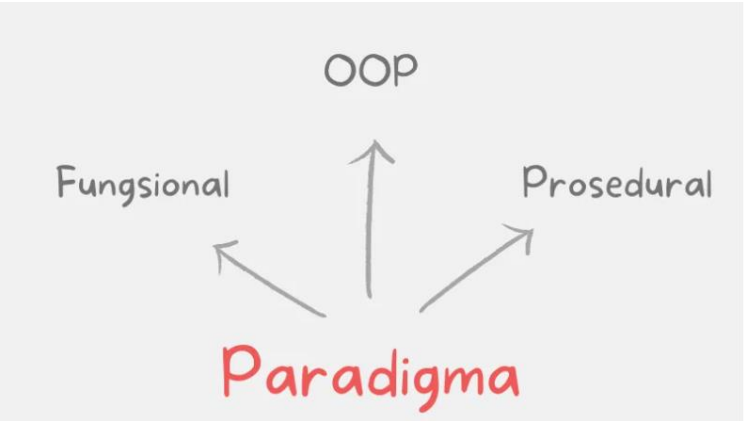
Apa yang dimaksud dengan paradigma pemrograman?

Kata “**paradigma**” sebenarnya tidak merujuk pada suatu bahasa pemrograman tertentu. Ia lebih merujuk kepada sebuah pendekatan atau konsep dalam menstrukturisasi suatu kode program [2]. Atau, kita bisa juga menyebut paradigma sebagai “**cara berfikir**” untuk menyusun pola penulisan program.

Sebagian bahasa pemrograman dibuat mudah untuk mengadopsi suatu paradigma tertentu, dan sebagian bahasa yang lain juga dibuat lebih mudah untuk mengadopsi paradigma yang lain.

3 Paradigma Pemrograman yang Cukup Populer

Di antara berbagai macam paradigma pemrograman, terdapat 3 paradigma yang setidaknya cukup populer (bahkan pertanyaan tersebut mendapatkan lebih dari 191k views di stackoverflow [3]).



Ketiga paradigma tersebut adalah:

- Paradigma prosedural – melakukan pengelompokan satu tugas tertentu yang bisa digunakan berkali-kali sebagai pendekatan dalam pemecahan suatu masalah.
- Paradigma fungsional – hampir sama dengan prosedural, hanya saja paradigma ini lebih berorientasi terhadap “input-output” dari pada mengubah data secara langsung seperti pada paradigma prosedural.
- Paradigma PBO (Pemrograman Berorientasi Objek) – yaitu paradigma yang menjadikan semua komponen program sebagai objek. Yang mana setiap objek memiliki identitas dan tugasnya masing-masing.

Apa Perbedaan Paradigma Prosedural dan Fungsional?
Untuk prosedural dan fungsional, sebenarnya tidak jauh berbeda. Ketika dulu saya mengambil mata kuliah Algoritma Pemrograman, salah seorang dosen kala itu memang membedakan antara fungsi dan prosedur:

Fungsi adalah fungsi yang mengembalikan nilai.

Sedangkan **prosedur** adalah fungsi yang tidak mengembalikan nilai.

Ada pun pada pelajaran-pelajaran bahasa pemrograman yang lebih modern, kebanyakan orang tidak lagi membedakan antara fungsi yang mengembalikan nilai dan fungsi yang tidak mengembalikan nilai, baik secara istilah mau pun secara penulisan sintaks (berbeda dengan beberapa bahasa pemrograman low level yang mana sintaks pembuatan fungsi dan prosedur memang dibuat berbeda).

Mari kita lihat perbedaannya secara langsung
Untuk lebih jelasnya, kita coba buat contoh aplikasi yang menggunakan pendekatan prosedural dan juga pendekatan fungsional.

Aplikasi yang kita buat adalah aplikasi untuk menghitung luas segitiga.

Pendekatan Prosedural

```
def hitung_luas ():
  alas = float(input('Masukkan alas: '))
  tinggi = float(input('Masukkan tinggi: '))
  print('Luas =', 0.5 * alas * tinggi)

# kita bisa panggil berkali-kali
hitung_luas()
# panggil lagi
hitung_luas()
```

Pendekatan Fungsional

```
def input_alas_dan_tinggi ():
  alas = float(input('Masukkan alas: '))
  tinggi = float((input('Masukkan tinggi: ')))

  return alas, tinggi

def hitung_luas (alas, tinggi):
  return 0.5 * alas * tinggi

"""kalau fungsional, kita sendiri yang mengelola
hasil kembaliannya"""

# satu fungsi bisa dipanggil secara independen
print(hitung_luas(5, 10))

# contoh dengan inputan alas dan tinggi
alas, tinggi = input_alas_dan_tinggi()
print(hitung_luas(alas, tinggi))
```

Contoh output dari 2 kode program di atas:

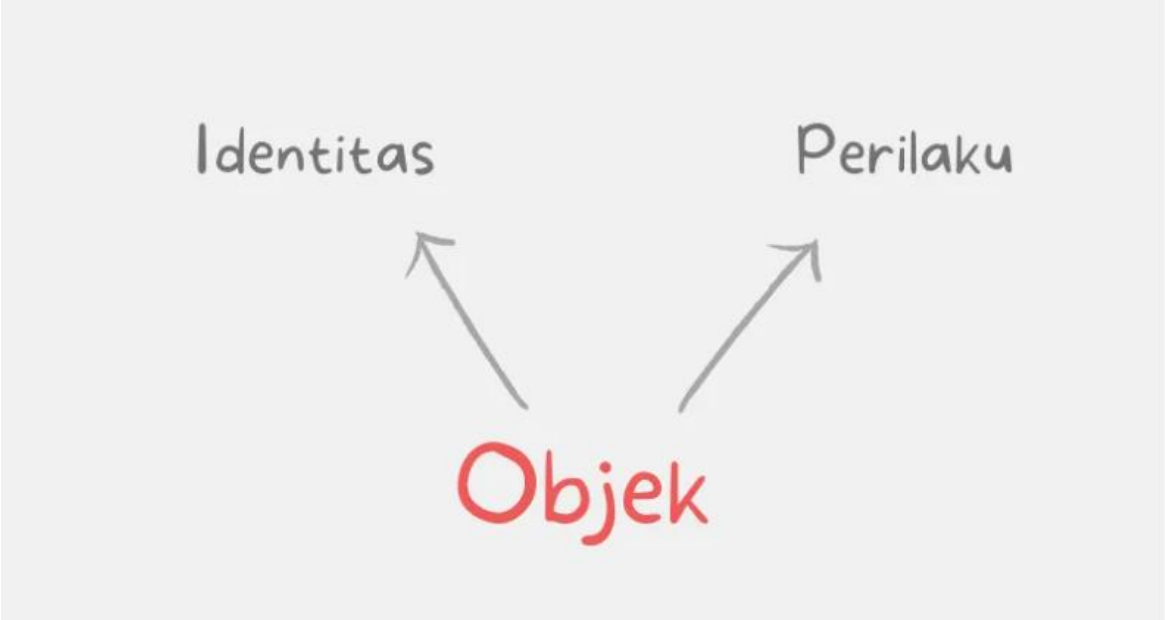
```
Masukkan alas: 5
Masukkan tinggi: 10
Luas = 25.0
Masukkan alas: 20
Masukkan tinggi: 5
Luas = 50.0
```

B. Konsep Dasar Pemrograman Berorientasi Objek (PBO)

Paradigma prosedural dan fungsional memang sekilas mirip dan memiliki banyak persamaan (baik dari sintaks mau pun dari penggunaan fungsi).

Tapi berbeda dengan konsep paradigma PBO yang dari cara berpikir saja sudah berbeda.

Pada konsep PBO, kita akan membuat semua komponen program seolah-olah adalah sebuah objek. Sebuah objek selalu memiliki identitas dan juga perilaku atau kemampuan untuk melakukan tugas tertentu.

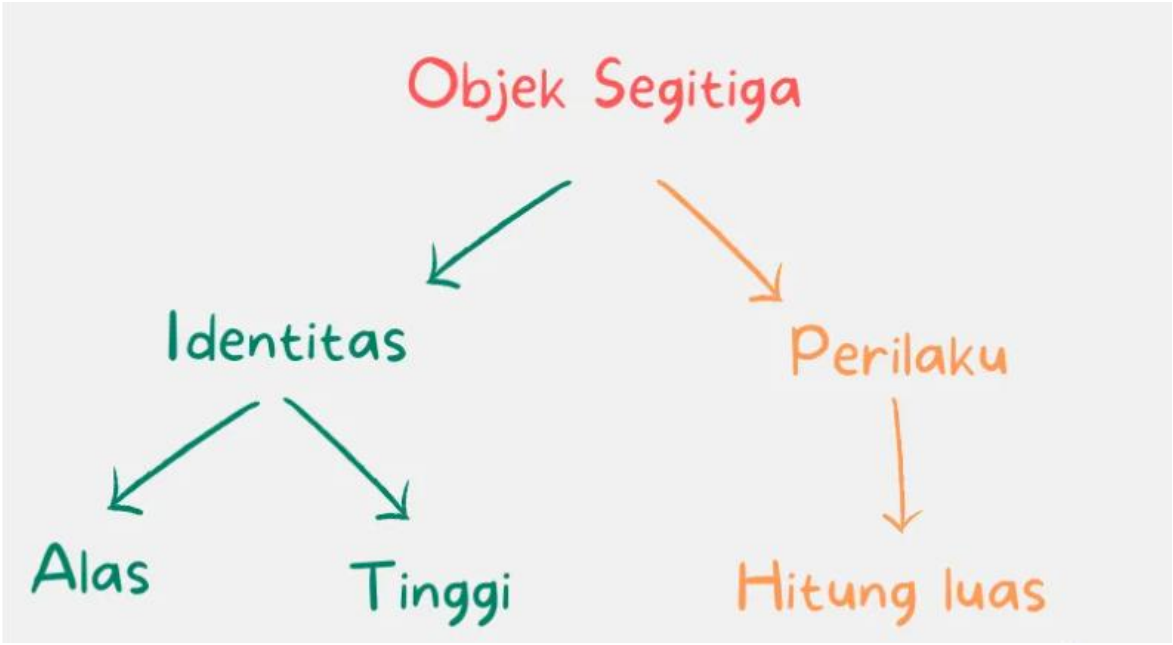


Kita ambil contoh saja program yang ada di atas: cara menghitung segitiga.

Maka cara berpikir kita akan seperti ini:

- Segitiga adalah sebuah objek.
- Objek segitiga memiliki 2 identitas berupa alas dan tinggi.
- Objek segitiga memiliki “kemampuan” untuk menghitung luasnya sendiri.

Kalau kita ilustrasikan akan terlihat seperti ini:



C. Pemrograman Beriorientasi Objek Pada Python

Kita akan membahas lebih dalam pada pertemuan berikutnya tentang bagaimana menerapkan PBO pada Python.

Akan tetapi sebagai gambaran awal, berikut ini kode program untuk menyelesaikan perhitungan luas segitiga dengan paradigma OOP / PBO:

```
class Segitiga:
  def __init__(self, alas, tinggi):
    self.alas = alas
    self.tinggi = tinggi

  def get_luas(self):
    return 0.5 * self.alas * self.tinggi

segitiga1 = Segitiga(5, 10)
segitiga2 = Segitiga(10, 10)

print('luas segitiga1:', segitiga1.get_luas())
print('luas segitiga2:', segitiga2.get_luas())
```

D. Kesimpulan

Dari pertemuan kali ini, ada beberapa poin yang bisa kita simpulkan:

1. Sebuah paradigma pemrograman sangat penting untuk dipertimbangkan dengan baik ketika kita mulai menulis sebuah program yang besar.
2. Terdapat beberapa paradigma pemrograman, di antara yang paling populer adalah paradigma fungsional, prosedural, dan OOP.
3. Paradigma fungsional dan prosedural sangat bergantung pada sebuah fungsi. Bedanya, paradigma fungsional menggunakan fungsi yang mengembalikan data, sedangkan prosedural menggunakan fungsi yang tidak mengembalikan data (meskipun kita bisa melakukan penggabungan mix and match).
4. Paradigma OOP memaksa cara berpikir kita untuk menganggap setiap komponen dari aplikasi sebagai objek.

=== To be Continue ===

REFERENSI

[1] N. Huda, "Jago Ngoding," 7 March 2021. [Online]. Available: https://jagongoding.com/python/menengah/oop/konsep/. [Accessed 28 November 2023].

[2] Lili, "Medium," 31 Ocktober 2017. [Online]. Available: https://medium.com/@LiliOuakninFelsen/functional-vs-object-oriented-vs-procedural-programming-a3d4585557f3. [Accessed 28 November 2023].

[3] Admin, "Stack Overflow," 30 Juni 2026. [Online]. Available: https://stackoverflow.com/questions/552336/oop-vs-functional-programming-vs-procedural. [Accessed 28 November 2023].