**Funcional Programming**

Functional Programming atau biasa disingkat dengan FP, termasuk dalam turunan deklaratif programming, dimana membuat kita tidak lagi berpikir bagaimana (how) `step-by-step` langkah penyelesaian, yang penting apa (what) yang dihasilkan.

Beberapa prinsip dasar yang harus terpenuhi:

1. Harus menggunakan pure function, yang mana harus menerima parameter dan me-return suatu hasil.
2. Function yang dipanggil tidak memperbolehkan adanya efek samping (side effect) yang berakibat pada perubahan data di luar scope (pada kenyataannya terkadang efek samping tidak dapat dihindari, namun diminimalisir)

Tentu saja pada funcional programming ini memiliki kelebihan dan kekurangannya,

antara lain :

**Kelebihan**:

* Karena fungsi, bersifat Pure Functions, kode menjadi lebih bampang dibaca, karena selalu menerima parameter dan mereturn hasil yg sesuai ekspektasi.
* Function menjadi lebih reusable

**Kekurangan**:

* Di beberapa kasus, ketika sistem sudah kompleks, penulisan FP ini menjadikannya tidak readable.
* Untuk pemula, paradigma ini, agak sulit diimplementasikan karena benar-benar berkutat dengan fungsi, dan tidak bisa dinalar ke dunia nyata.

**Object Oriented Programming(OOP)**

**pemrograman berorientasi objek atau object oriented programming** adalah paradigma berdasarkan konsep objek yang berisi struktur data dalam bidang yang dikenal sebagai atribut dan kode dalam bentuk prosedur, juga dikenal sebagai metode. Bahasa pemrograman berorientasi objek yang paling populer adalah berbasis kelas, yang berarti objek disajikan sebagai turunan dari kelas. Fitur utama pemrograman berorientasi objek adalah abstraksi, pewarisan, polimorfisme, dan enkapsulasi. Pemrograman berorientasi objek dapat berarti hal yang berbeda bagi orang yang berbeda. Bahasa pemrograman digunakan dengan baik saat Anda memiliki serangkaian operasi tetap, dan saat Anda melakukan beberapa operasi dengan perilaku umum dan varian berbeda. Pada saat yang sama, OOP bagus ketika Anda memiliki serangkaian operasi tetap, dan Anda terutama menambahkan hal-hal baru. Di sisi lain, bahasa fungsional digunakan saat Anda memiliki sekumpulan hal yang tetap, dan Anda terutama menambahkan operasi baru ke hal yang sudah ada.

Pemrograman berorientasi objek adalah cara menjaga data dan perilaku terbungkus dalam satu objek. Ini berarti semua data dan fungsi digabungkan menjadi satu dan diprogram pada panjang gelombang yang sama. Ini berbeda dengan bahasa pemrograman fungsional yang memiliki seperangkat asumsi bahwa fungsi tidak digabungkan dengan data. Kebanyakan orang yang menulis Python hampir pasti lebih baik menulis kode berorientasi objek. Oleh karena itu, jika Anda baru dalam pemrograman, saya akan merekomendasikan memulai dengan bahasa berorientasi objek dengan menulis kode dan algoritme pembelajaran mendalam yang disediakan oleh Python. Namun, Anda dapat menggunakan salah satu paradigma gaya fungsi, seperti pemrograman fungsional, menggantikan program berorientasi objek setelah menguasai konsepnya. Misalnya, Anda dapat menggunakan pemrograman fungsi alih-alih OOP dalam konteks apa pun dengan Python, karena komputer tidak peduli dengan paradigma apa pun.

**PERBEDAAN**

Dari kedua Program pada Python tersebut pasti terdapat perdaan di dalamnya, sehingga kita dapat menentukan program mana yang akan kita gunakan untuk membuat suatu kode program. Berikut beberapa perbedaanya:

Fungsional :

Menggunakan fungsi PURE

Menghidari ststus bersama dan data yang dapat diubah

Deklaratif

Ringkas dan lebih mudah untuk diuji

Berorientasi pada Objek :

Mengenkapsulasi fungsionalitas menggunakan kelas

Status disimpan di properti kelas

Imperatif

Dirancang untuk memanfaatkan trik pemrograman agar dapat diuji