**Dhevan Abi Armadani**

**BE Developer**

**Pemrograman Berorientasi Objek**

1. **Berikan penjelasan tentang Pemrograman Berorientasi Objek atau OOP!**

* OOP adalah metode pemrograman yang berorientasi pada object, maksud dari object yaitu suatu case dalam program.

1. **Berikan alasan kenapa perlu memahami tentang konsep Pemrograman Berorientasi Objek atau PBO?**

* Alasannya yaitu, missal ketika mengerjakan suatu project secara tim, setiap programmer bisa membuat class-class secara individual, kemudian komponen-komponen tersebut bisa digabung menjadi satu, (mempercepat waktu pengerjaan project). Mudah digunakan untuk project-project lainnya dengan langsung pakai ataupun sedikit memodifikasi. Data-data pada OOP mudah untuk diakses ketika membutuhkan.

1. **Apa saja fitur utama yang ada dalam Pemrograman Berorientasi Objek atau OOP? Berikan penjelasan detail dari masing - masing komponen!**

* Inheritance

Bisa membentuk class baru yang mewarisi dari class yang sudah ada, konsepnya seperti menu drop-down yang satu menu bisa menampung banyak submenu.

* Encapsulation

Digunakan untuk menyatukan(dikapsulkan) data dari kumpulan data-data menjadi satu unit data. Bisanaya juga digunakan untuk *information-hiding-mechanism* atau menghilangkan akses publik ke atribut-atribut yang terdapat pada kapsul tersebut.

* Abstract

Adalah class yang yang memiliki informasi abstract dan metode dari sekumpulan data. Class abstract tidak bisa diubah dan berlaku sebagai kerangka. Suatu class abstract memiliki informasi yang dapat diturunkan ke subclassnya.

* Polymorphism

Adalah konsep dimana suatu object berbeda-beda bisa diakses melalui satu interface. Fitur Polymorphism memiliki 2 jenis yaitu static dan dynamic.

Static Polymorphism yang umum digunakan adalah method Overloading.

Dynamic Polymorphism adalah subclass yang bisa menimpa method dari superclassnya.

1. **Apa keuntungan menggunakan Pemrograman Berorientasi Objek atau OOP?**

* Modularitas

Memudahkan problem solving, karena sudah menjadi dalam satu kapsul sehingga tidak perlu mencari kesalahan ke banyak class.

* Reuseable

Tidak perlu menulis code berulang-ulang.

* Fleksible

Class nya mudah diadaptasikan untuk class-class lain.