OOP

Menggabungkan kumpulan fungsi dan attribute, yang memiliki kesamaan dalam sebuah unit. Yang kita sebut sebagai objek.

Class

Merupakan blueprint, untuk mengembangkan sebuah objek. bisa disebut template yang didalamnya menjelaskan perilaku objek tersebut baik itu function maupun properties/attribute.

Penting dalam oop:

* Inheritance

Pewarisan, untuk meminimalisir penulisan berulang pada fungsi properti dan variable.

Inheritance memberi kesempatan kepada class child untuk menggunakan semua member yang dimiliki class parent.

Ini juga bisa disebut proses, memperoleh sifat dan prilaku yang sama dengan objek lainnya.

Ketika kita ingin memperoleh sifat dan prilaku yang sama dari objek lainnya, kita tidak perlu menulis ulang kode lagi pada class child tersebut.

Jenis inheritance :

Single inheritance (dari parent ke child) , multilevel inheritance (dari parent ke childe, dari child ke class lain) , multiple inheritance (dari satu parent ke 2 child), hierarchical inheritence

* Encapsulation

Pengisolasian data, dan tidak dapat diakses langsung dari luar.

Dapat lebih leluasa melakukan perubahan nilai tanpa harus mengakses attributenya. Cukup gunakan setter getter.

Encapsulation bukan menyembunyikan data, tetapi encapsulation yang menyebabkan data tersembunyi.

* Abstraction

Abstriksi memungkinkan developer untuk membuat class abstrak yang tidak dapat diinstansiasi

(myClass myObject = new MyClass(); // instantiate sebuah objek)

Tetapi dapat digunakan sebagai kerangka dasar untuk kelas turunan konkret.

Class abstract tidak dapat diinstansiasi secara langsung, tetapi dapat diwarisi oleh subclass. Class abstract mendefinisikan metode abstrak, yang harus diimplementasikan oleh subclassnya.

Subclass yang mewarisi class abstrak harus mengimplementasikan semua metode abstrak yang didefinisikan oleh superclassnya.

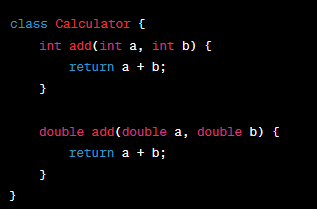
* Interface, kumpulan metode abstrak yang dapat diimplementasikan oleh class lain. Menyediakan kontrak atau spesifikasi tentang behaviour yang harus diikuti oleh class yang mengimplementasikannya.
* Polymorphism  
  kemampuan sebuah objek untuk mengambil banyak bentuk. Memungkinkan objek untuk merespon metode yang sama dengan cara yang berbeda.

Memungkinkan penggunaan metode yang sama untuk berbagai jenis objek yang berbeda.

Terjadi ketika suatu superclass direferensikan ke dalam subclass.

Overloading:

Method overloading terjadi ketika sebuah class mempunyai 2 atau lebih method yang memiliki nama yang sama tetapi memiliki parameter yang berbeda.



Overriding:

Method yang terjadi ketika sebuah subclass memiliki sebuah implementasi spesifik sebuah metod yang telah didefinisikan oleh superclass. Harus memiliki nama metode yang sama, return type yang sama, dan parameter yang sama seperti superclass.

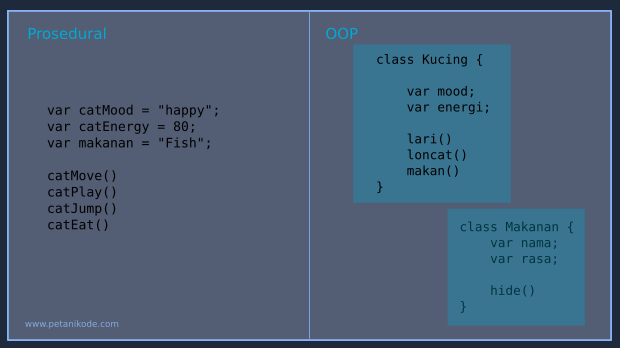
# VERSI PETANICODE

Mengapa harus belajar oop: biasanya pada pemrogramman prosedural, biasanya memecah program menjadi beberapa prosedur. Lalu variable global dan lokal digunakan untuk menyimpan data. Tapi ketika programmnya jadi semakin besar, variable dan prosedur yang dibuat juga semakin banyak. Ketika variable dan prosedur semakin banyak, maka akan sulit untuk di maintance. Jadi pake oop untuk memisahkannya.

OOP ?: teknik pemrograman yang berorientasi objek. Pada OOP, fungsi dan variable dibungkus ke dalam sebuah objek atau class, yang dapat saling berinteraksi. Jadi gak ada lagi kode yang berantakan, semua dibungkus di objek kayak kresek.

Class dan Objek:

Class bertugas untuk mengumpulkan prosedur/fungsi, dan variable dalam satu tempat. Sebagai contoh,



Kita bisa menentukan mana variable dan prosedur untuk si kucing dan makanan.

Class ini nanti dipake untuk membuat objek. Yang berarti class adalah blueprint dari sebuah objek.

Objek : adalah sebuah variable yang merupakan instance atau perwujudan dari class. Instance bisa diartikan sebagai wujud dari class.

# VERSI PZN