**Encapsulasi**

Encapsulasi adalah konsep dalam OOP yang di gunakan untuk menyembunyikan detail internal suatu class dan hanya memperlihatkan apa yang perlu di ketahui oleh pengguna eksternal. Berikut beberapa tujuan penggunaan encapsulasi :

1. Abstraksi
2. Keamanan dan integritas data’
3. Pemeliharaan dan perubah yang aman
4. Reusabilitas dan fleksibilitas
5. Meningkatkan keterbacaan dan kesadaran

Membuat semua variabel private dan untuk mengambil data yang kita butuhkan kita menggunakan getter dan setter. Getter ( mengambil variabel ) dan setter ( mensetting variabel ).

Sebagai contoh dalam sebuah game, kita membuat getter private adalah nama dari object. Nama dari object itu tidak boleh berganti atau berubah selama game itu berjalan. Misal namanya adalah sniper, maka akan tetap sniper selama game itu berjalan. Ada juga yang bisa berubah, misal kita mensetting darah hero itu boleh berkurang tetapi tidak boleh berkurang sampai minus ( - ), karna kalau sampai minus maka heronya akan mati.

Berikut cara menggunakan enkapsulasi :

class Hero:

    def \_\_init\_\_(self, name, health, attactPower):

*self*.\_\_name = name

*self*.\_\_health = health

*self*.\_\_attPower = attactPower

*# getter*

    def getName(self):

        return *self*.\_\_name

    def getHealth(self):

        return *self*.\_\_health

*# setter*

    def diserang(self, serangPower):

*self*.\_\_health -= serangPower

    def setAttPower(self, nilaiBaru):

*self*.\_\_attPower = nilaiBaru

*# awal game*

earhshaker = Hero("earhshaker", 50, 5)

*# game berjalan*

print(earhshaker.getName())

print(earhshaker.getHealth())

earhshaker.diserang(5)

print(earhshaker.getHealth())

kode di atas memperlihatkan kalau kode itu menjaga kode berikut agar tidak bisa di ubah semena-mena :

*self*.\_\_name = name

*self*.\_\_health = health

*self*.\_\_attPower = attactPower