**#9 OOP DASAR pada PHP Visibility**

Tema nya Visibility atau Access Modifier, definisi singkat nya :

* Konsep yang digunakan untuk mengatur akses dari **property** dan **method** pada sebuah objek atau **Class**
* Ada 3 keyword visibility : **public**, **protected**, dan **private**
* **Public** : dapat digunakan di mana saja bahkan di luar kelass nya.
* **Protected** : Property dan Method nya hanya bisa digunakan di dalam sebuah kelas beserta turunan nya. Misalnya kalian punya class inheritence maka class tersebut bisa menggunakan property atau method nya
* **Private** : hanya dapat digunakan di dalam sebuah kelas tertentu saja.

Kita perlu menerapkan Visibility ini karena

* Supaya kita hanya memperlihatkan aspek dari class yg dibutuhkan oleh *“client”* misal nanti kalo kalian bekerja berkalborasi lalu kalian tidak mengizinkan bagian lain dari program itu mengakses class kalian itu kalian butuh visibility, ketika propety dan method nya tidak digunakan dimana” kita pake public karena dengan melakukan itu method dan property nya nanti bisa di akses bahkan di modifikasi dari luar.
* Menetukan kebutuhan yang jelas untuk object. (jadi nanti object mana yg boleh menggunakan property dan method yg mana saja).
* Memberikan kendali pada kode untuk menghindari *‘bug’*

**Studi kasus**

Rasanya di kasus sebelum nya seperti baik-baik saja, tapi kita liat ini semua yg kita buat sampai saat ini visibility nya masih menggunakan public artinya kita bisa mengakses dimna pun, contoh nya kita sudah meninstaciasi 2 produk itu.

Misal saya mau mencetak harga dari game ( echo $produk2->harga ) gaada masalah ya kan, artinya kita bisa mengakses dari class produk dengan visibility nya pubic, kita bisa akses dari luar class nya gitu.

Perhatikan dengan melakukan itu kita jadi punya problem, kita bsia ubah harga seenak kita gini :

$produk3->harga = 4000;

echo $produk3->harga;

jadi yg harusnya harga udah di set dari awal bisa berantakan jika di ubah gitu kan. Untuk lebih baik kita mau sesuatu di luar class tidak bisa mengubah yg ada didalam class, cara nya :

  public $judul ,$penulis ,$penerbit;

  protected $harga;

kita ubah nya di class produk() ya, jadi dengan begitu kita tidak bisa lagi mengubah harga nya seenak kita. Nah tapi masalahnya kita juga gabisa nampilin harga nya karena harga nya sekarang dilindungi, karena sekarang yg punya akses hanya class nya saja kalo mau langsung manggil di dalam class nya.

Nah tapi harga itu bisa di panggil untuk anak nya turunan nya misal class Komik dan class Game bisa langsung manggil variable nya tanpa error. Nah supaya kita bisa pake variable harga itu kita coba buat public function getHarga() di class Game misal nya :

    public function getHarga() {

        return **$this**->harga;

    }

Jadi getHarga hanya bisa digunakan oleh class produk beserta class turunan nya cara nampilin nya gini :

echo $produk3->getHarga();

aman ya harga nya sudah tampil. Sekarang kita ubah jadi private :

  public $judul ,$penulis ,$penerbit;

  private $harga;

inget private itu visibility yg memungkin kan sebuah property dan method hanya bisa di akses oleh class tertentu aja jadi kalo saya tulis private nya di harga berarti yg bisa di pake hanya class itu aja, class turunan nya atau anak nya gabisa pake.

Maka kalo kita liat hasil nya error gayss, karena harga hanya bisa dilihat di class product saja., jadi kalo masih mau tetep jalan pindahin aja getHarga itu di class produk gini :

class Produk{

  public $judul ,$penulis ,$penerbit;

  private $harga;

*// Constructor Method*

    public function \_\_construct($judul = "judul", $penulis = "penulis", $penerbit = "penerbit", $harga = 0) {

**$this**->judul = $judul;

**$this**->penulis = $penulis;

**$this**->penerbit = $penerbit;

**$this**->harga = $harga;

    }

    public function getHarga() {

        return **$this**->harga;

    }

}

$produk2 = new Komik("Twenty Five Twenty One", "Jeong Jiman", "TVN", 1500000, 100);

$produk3 = new Game("Naruto", "Masashi Kishimoto", "Shonen Jump", 30000, 50);

echo $produk3->getHarga();

jadi kita bisa set private property nya itu agar kita bisa pastikan property itu hanya bisa di pake di class itu saja.

Misal nya saya mau menambahkan vasilitas diskon, jadi saya punya sebuah property baru nama nya diskon, kita set public awal nya dan nilai defalut nya 0. Nah diskon ini akan berisi persentase diskon dari sebuah produk.

Nah misalnya untuk memberikan diskon itu kita punya sebuah method namanya setDiskon() ada parameter nya $diskon, nah method ini akan mengisi property diskon dari class produk ini

Nah terus untuk mendapatkan harga nya saya akan mengubah method getHarga jadi nanti harga setelah diskon. Jadi harga akhirnya harga di kurangi diskon nya.

class Produk{

  public $judul ,$penulis ,$penerbit, $diskon;

  private $harga;

*// Constructor Method*

    public function \_\_construct($judul = "judul", $penulis = "penulis", $penerbit = "penerbit", $harga = 0) {

**$this**->judul = $judul;

**$this**->penulis = $penulis;

**$this**->penerbit = $penerbit;

**$this**->harga = $harga;

    }

    public function setDiskon($diskon) {

**$this**->diskon = $diskon;

    }

    public function getHarga() {

        return **$this**->harga - (**$this**->harga \* **$this**->diskon / 100);

    }

}

$produk3->setDiskon(50);

echo $produk3->getHarga();

Sekarang $diskon nya itu masih public ya nah itu kita setDiskon nya 50% ya berarti nanti harga nya jadi 15ribu, nah tapi karena property method itu public kita bisa memodifikasi variable $diskon nya seterah kita misal 90% jadi kita paksa yg seharusnya ga segitu jadi salah juga.

Jadi jangan biarkan ada yg bisa langsung mengakses property nya kita kasih aja private ya gini :

  public $judul ,$penulis ,$penerbit;

  private $harga, $diskon = 0;

:dengan mengubah itu lalu kita panggil setDiskon tidak ada error masih sama saja kecuali kalian mau akses variable diskon baru gabisa, misal function setDiskon() nya mau kalian taro di class anak nya maka disitu kasih aja visibility protected kalo di class produk kasih aja private.

Misal nya hanya class game yg punya diskon ganti aja ya taro function nya di class game gitu.

Dan nanti di video selanjut nya kita akan perbaikin lagi supaya property nya tidak bisa di akses dari luar, kita akan masuk ke topik Accessor Methods (setters & Getters).