**#10 OOP DASAR pada PHP Setter dan Getter**

Topik mengenai setter dan getter ini sangat erat kitan nya dengan Visibility :

* **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.

Saya sebut erat kaitan nya karena kita menggunakan setter dan getter untuk menghindari ketika kita membuat sebuah property dengan visibility public, karena sebaik nya kita tidak membiar kan bagian dari luar class kita bisa mengakses property secara langsung nah untuk itu kita harus merubah private atau protected ya.

Nah setelah kita mengubah protected atau private artinya kita tidak punya akses secara langsung jika kita mau mendapat kan isi dari sebuah property atau menset nilai nya. Untuk itulah kita butuh sebuah method untuk bisa mmebaca atau melihat isi dari property tersebut atau pun mengubah isi nya.

Nah method itu nanti nya kita akan sebut dengan setter dan getter atau method untuk ngeset dan method untuk get gampangnya. Kita balik ke studi kasus sebelum nya ya.

Nah sebetulnya pada saat kita belajar visibility kita sudah membuat metho yg tugas nya melihat nilai dan set nilai nah itu bisa kita bisa sebut dengan setter dan getter, nah tapi kalo kalian liat masih ada beberapa property visibility nya public.

Nah itu yg akan kita hindari jadi sebaik nya property itu protected atau ga private nanti tergantung dari dia mau dipake di class child nya atau tidak gitu.

**STUDI KASUS**

Nah misal sekarang saya ganti yg tadi nya property public sekarang jadi private gini :

class Produk{

  private $judul ,$penulis ,$penerbit, $harga;

  protected $diskon = 0;

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

    }

    public function getLabel() {

        return "$this->penulis, $this->penerbit";

    }

    public function getInfoLengkap() {

        $str = "{**$this**->judul} | {**$this**->getLabel()}, (Rp. {**$this**->harga})";

        return $str;

    }

}

Inget kan kalo udah private gitu artinya Cuma bisa di pake oleh class Produk gabisa dipake class turunan nya seperti Komik dan Game.

Nah ini codinga nya sama aja kaya sebelum nya Cuma diganti jadi private untuk property yg di atas itu, nah ternyata tidak ada error sama sekali karena di function turunan nya itu dia memanggil fungsi didalam fungsi yg constructor misal nya nah itu makannya ga error. Dan karena juga kita menggunakan property yg private itu digunakan nya hanya di class Produk itu sisa nya hanya memanggil function saja.

Nah sekarang kalo saya mau coba panggil produk1 yg komik ya panggil judul nya :

echo $produk1->judul;

harus nya itu error ya karena kan property judul nya di private class anak nya yaitu si komik gabisa ngeakses ya.

Sekarang kita instasiasi class produk() gini :

$produk4 = new Produk("BarangBaru");

echo $produk3->judul;

error juga tapi sekarang error nya cannot access property Produk, jadi bahkan kita gabisa ngeakses dari luar padahal instasiasi nya sudah class itu nya langsung si Produk().

Sekarang biar kita bisa mengakses kita harus sediakan celah nya, kita buat method baru yg public getJudul() jadi kalo method nya dipanggil ke ambil property judul nya sama kaya sebelum nya ini tuh.

class Produk{

  private $judul ,$penulis ,$penerbit, $harga;

  protected $diskon = 0;

*// 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 getJudul() {

        return **$this**->judul;

    }

}

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

echo $produk2->getJudul();

nah kalo dengan seperti ini kita panggil bisa karena yg kita panggil itu method nya ya, itu kita panggil class anak yaitu class Komik() sedangkan getJudul() ada di class ibu nya yaitu Class Produk() jadi semua anak dari class Produk() bisa akses getJudul() itu ya.

**NGAPAIN SIH CAPE-CAPE BUAT FUNCTION GITU /////**

**a**lesan nya kita menggunakan getter seperti itu supaya nanti kita bisa melakukan sesuatu didalam function nya misal nya kalian validasi yg mau di input hanya string saja misal, nanti akan lebih jelas ketika kita cobain yg setter.

**Sekarang kalo kita mau mengubah judul nya gimana //// setter**

Kita juga gabisa kalo mau mengubah judul nya kita harus buat juga didalam function nya ya, kita harus buat setter method gini :

class Produk{

  private $judul ,$penulis ,$penerbit, $harga;

  protected $diskon = 0;

*// 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 setJudul($judul) {

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

    }

    public function getJudul() {

        return **$this**->judul;

    }

}

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

$produk2->setJudul("pisang goreng");

echo $produk2->getJudul();

nah kalo dengan begitu maka bisa, simpel aja tapi itu ya. Jadi yg setJudul() adalah setter dan yg getJudul() adalah getter.

**KENAPA HARUS CAPE-CAPE BUAT SETTER DAN GETTER /// PROBLEM 2**

Dlihat function setJudul() codingan nya hanya buat ngeset judul baru aja emang kalo Cuma gitu membuat banyak codingan nya aja ya. Tapi gimana kalo kita mau lakukan validasi misalnya yg hanya boleh di tulis oleh judul itu hanya string .

Jadi kan kita bisa ngeset gini :

class Produk{

  private $judul ,$penulis ,$penerbit, $harga;

  protected $diskon = 0;

*// 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 setJudul($judul) {

         if(!is\_string($judul)) {

            throw new Exception("Judul harus string");

        }

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

    }

    public function getJudul() {

        return **$this**->judul;

    }

}

$produk2->setJudul("pisang goreng");

echo $produk2->getJudul();

kita ngeset judul nya masih berupa string tapi gimana kalo bukan string :

$produk2->setJudul(2521);

echo $produk2->getJudul();

Nah error katanya Judul harus string, jadi menggunakan setter dan Getter memungkin kan kita melakukan validasi.

Nah itu baru property judul belum property penulis, penerbit, harga dan diskon kita buat :

class Produk{

  private $judul ,$penulis ,$penerbit, $harga;

  protected $diskon = 0;

*// 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;

    }

*// judul*

    public function setJudul($judul) {

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

    }

    public function getJudul() {

        return **$this**->judul;

    }

*//penulis*

    public function setPenulis($penulis) {

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

    }

    public function getPenulis() {

        return **$this**->penulis;

    }

*//penerbit*

    public function setPenerbit($penerbit) {

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

    }

    public function getPenerbit() {

    return **$this**->penerbit;

    }

}

Nah untuk yg dikon itu kita kasih visibikity nya Protected karena diskon itu biar bisa di akses buat class komik dan class game, karena diskon nya itu hanya bisa digunakan di class game di class komik ngga kita kasih.

Biar diskon nya fleksibil setDiskon() nya di taro di class Produk biar kalo class komik mau diskon gampang tinggal panggil aja gitu. Jadi kita taro aja setDiskon() di class Produk() ya jadi property $diskon itu kita bisa ganti jadi private ya.

class Produk{

  private $judul ,$penulis ,$penerbit, $harga,  $diskon = 0;

*// 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;

    }

*// judul*

    public function setJudul($judul) {

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

    }

    public function getJudul() {

        return **$this**->judul;

    }

*//penulis*

    public function setPenulis($penulis) {

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

    }

    public function getPenulis() {

        return **$this**->penulis;

    }

*//penerbit*

    public function setPenerbit($penerbit) {

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

    }

    public function getPenerbit() {

    return **$this**->penerbit;

    }

*// harga*

    public function setHarga() {

        return **$this**->harga;

    }

    public function getHarga() {

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

    }

*// diskon*

    public function setDiskon() {

        return **$this**->diskon;

    }

    public function setDiskon($diskon) {

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

    }

}

echo $produk2->getPenerbit();

nah enak nya kalo mau panggil penerbit atau penulis tinggal get sesuaikan yg kita mau.

Sebenar nya setter dan getter didalam php itu bisa kita lakukan menggunakan \_\_set() & \_\_get() fungsi nya langsung ya dinamakan magic method ya. Di php itu banyak ya magic method kita baru menggunakan constructor aja.

Set dan get nya akan lebih jelas pada saat kita membahas OVERLOADING ya dimana kita membuat function yg sama tapi kita dengan parameter yg berbeda.

Nah untuk video selanjut nya kita akan membahas Static Keyword, yg bisa kita pake untuk property dan method kita.