**#3 OOP DASAR pada PHP Property & Method**

Property Adalah….

* Sesuatu yang Meprepsentasikan data / keadaan dari sebuah object
* Atau dia bisa di sebut Variable yang ada di dalam object (atau disebut juga member variable)
* Nah nanti cara bikin nya sama seperti di dalam PHP, ditambah dengan visibility di depan nya (misal public, private, protectic).

Method Adalah…..

* Merepresentasikan perilaku dari sebuah object
* Function yang ada di dalam object (jadi kalo kalian bikin function didalam object itu disebut method)
* Sama seperti function di dalam PHP, ditambah dengan visibility di depannya (walaupun untuk method ini bisa tidak pake visibility nanti akan default nya public)

Nah sekarang ilustrasi nya gini sekarang kita punya contoh kasus ada sebuah mobil misal ada daftar property dan daftar method yg bisa kita terapkan di mobil nya :

Property :

* Nama
* Merk’
* Warna
* kecepatanMaksimal
* jumlahPenumpang

Method :

* tambahKecepatan
* kurangiKecepatan
* gantiTransmisi
* belokKiri
* belokKanan

jadi kalo untuk property itu apa yg menempel pada mobil itu nah kalo method itu apasi yg bisa dilakukan oleh mobil itu. Dan hal itu kalo kita representasikan ke coding PHP bisa seperti ini :

class Mobil {

    public $nama;

    public $merk;

    public $warna;

    public $kecepatanMaksimal;

    public $jumlahPenumpang;

    public function tambahKecepatan() {

    }

    public function kurangiKecepatan() {

    }

    public function gantiTransisi() {

    }

}

Jadi kita buat class Mobil, nah sama kan kaya buat variable Cuma beda nya kita tambahkan visibility atau akses modifier public itu artinya kita bisa gunakan variable nya dibagian manapun di program kita.

Itu contoh kasus buat class Mobil ya, nah sekarang kita balik ke kasus kita yg membuat class Produk nanti class ini akan kita gunakan untuk produk-produk yg akan kita jual. Sekarang kita akan membuat proerty dan method yg sesuai untuk class Produk.

Pertama apa kira-kira antara game dan komik hal yg sama, umum nya pasti mereka punya judul ya dan lain hal yg sama.

class Produk{

  public $judul, $penulis, $penerbit, $harga;

}

$produk1 = new Produk();

Sekarang saya punya object nama nya produk1 yg merupakan instance dari class Produk kalo kita var\_dump($produk1) maka yg tampil null karena belum kita definisikan atau belum kita isi property nya secara default.

class Produk{

  public $judul = "judul"

         , $penulis = "penulis"

         , $penerbit = "penerbit"

         , $harga = 0;

}

$produk1 = new Produk();

var\_dump($produk1);

jadi otomatis nilai nya ada. Nah cara ke-2 semua itu bisa kita timpa dengan menuliskan setelah kita bikin object nya. Misal saya ingin buat judul yg baru kalian hanya perlu panggil object nya lalu nama property nya :

class Produk{

  public $judul = "judul"

         , $penulis = "penulis"

         , $penerbit = "penerbit"

         , $harga = 0;

}

$produk1 = new Produk();

$produk1->judul = "Twenty Five Twenty One";

var\_dump($produk1);

nah jadi sekarang property judul nya di timpa sama judul baru, misal sekarang kita buat object baru nama nya $produk2 saya bikin dari instance class Produk lagi dan kalo saya var\_dump($produk2) hasil nya itu isi default dari class Produk nya :

class Produk{

  public $judul = "judul"

         , $penulis = "penulis"

         , $penerbit = "penerbit"

         , $harga = 0;

}

$produk1 = new Produk();

$produk1->judul = "Twenty Five Twenty One";

var\_dump($produk1);

$produk2 = new Produk();

var\_dump($produk2->judul);

saya hanya mencetak judul nya aja nah benar kan isi nya default dari class produk begitu cara maen nya. Kalo $produk2 judul nya mau beda tinggal timpa aja :

$produk2 = new Produk();

$produk2->judul = "Hometown cha cha";

var\_dump($produk2->judul);

jadi kalian bisa bikin property di dalam class nya lalu di isi dengan nilai default atau kalian bisa timpa saat buat object nya. Nah bahkan kalian bisa menambahkan property baru setelah object nya di instansiasi kalo sebelum nya kita timpa property judul yg sudah ada dari class nya sendiri, tapi gimana kalo kita panggil object nya dengan property yg belum ada di class produk itu :

class Produk{

  public $judul = "judul"

         , $penulis = "penulis"

         , $penerbit = "penerbit"

         , $harga = 0;

}

$produk1 = new Produk();

$produk1->judul = "Twenty Five Twenty One";

var\_dump($produk1);

$produk2 = new Produk();

$produk2->judul = "Hometown cha cha";

$produk2->tanggalTerbit = "2022-02-19";

var\_dump($produk2->tanggalTerbit);

Nah property tanggalTerbit itu tidak ada di class Produk, nah maka akan otomatis di tambahkan property baru. Jadi hati-hati kalo tujuan kalian menimpa property nama nya harus sama gaboleh sampe salah, karena typo sedikit itu ngaruh yg ada malah ke buat property baru. Nah makannya nanti kedepan nya visibility itu penting supaya kalian gabisa sembarangan nambah dan mengubah property yg ada.

class Produk{

  public $judul = "judul"

         , $penulis = "penulis"

         , $penerbit = "penerbit"

         , $harga = 0;

}

$produk3 = new Produk();

$produk3->judul = "Naruto";

$produk3->penulis = "Masashi Kishimoto";

$produk3->penerbit = "Shonen Jump";

$produk3->harga = 30000;

var\_dump($produk3);

nah itu kita buat yg lengkap nya, lalu kalo kita mau buat label misal nya :

echo "<br><br>Komik : $produk3->penulis, $produk3->penerbit";

kita buat function atau method, kalo function sebaik nya dia mengembalikan nilai yah.

class Produk{

  public $judul = "judul"

         , $penulis = "penulis"

         , $penerbit = "penerbit"

         , $harga = 0;

    public function sayHello() {

        return "Hello World";

    }

}

Nah sekarang kita mau panggil method nya :

echo $produk3->sayHello();

gitu simpel kan, nah sekarang saya mau penulis dan penerbit dipanggil dengan method, misal saya punya method getLabel() gini :

class Produk{

  public $judul = "judul"

         , $penulis = "penulis"

         , $penerbit = "penerbit"

         , $harga = 0;

    public function getLabel() {

        return "$penulis, $penerbit";

    }

}

Nah tapi kalo kalian jalankan getLabel() nya itu akan error karena mengingat variable scope yg ada didalam function kalo kita nulis variable yg didalam function itu artinya variable hanya untuk function itu aja bukan ngambil dari variable diluar function itu.

Nah untuk mengakses property yg ada di class Produk kalian bisa tambahkan ( $this ) gini :

class Produk{

  public $judul = "judul"

         , $penulis = "penulis"

         , $penerbit = "penerbit"

         , $harga = 0;

    public function getLabel() {

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

    }

}

Nah itu artinya variable yg ada di function getLabel() akan mengacu kepada variable di class Produk itu. Mari kita panggil function nya :

class Produk{

  public $judul = "judul"

         , $penulis = "penulis"

         , $penerbit = "penerbit"

         , $harga = 0;

    public function getLabel() {

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

    }

}

$produk3 = new Produk();

$produk3->judul = "Naruto";

$produk3->penulis = "Masashi Kishimoto";

$produk3->penerbit = "Shonen Jump";

$produk3->harga = 30000;

var\_dump($produk3);

echo "<br><br>Komik : " . $produk3->getLabel();

terakhir kita bikin produk berikut nya :

class Produk{

  public $judul = "judul"

         , $penulis = "penulis"

         , $penerbit = "penerbit"

         , $harga = 0;

    public function getLabel() {

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

    }

}

$produk3 = new Produk();

$produk3->judul = "Naruto";

$produk3->penulis = "Masashi Kishimoto";

$produk3->penerbit = "Shonen Jump";

$produk3->harga = 30000;

$produk4 = new Produk();

$produk4->judul = "Uncharted";

$produk4->penulis = "Neil Druckmann";

$produk4->penerbit = "Sony Computer";

$produk4->harga = 250000;

Jadi sekarang kita punya produk game dan komik dengan nama object yg berbeda untuk komik nama object nya $Produk3 dan untuk game nama object nya $produk4. Nah untuk nampilin penerbit dan penulis dari ke-2 object tersebut gini aja :

echo "<br><br>Komik : " . $produk3->getLabel();

echo "<br><br>Game : " . $produk4->getLabel();

nah kalo yg kalian liat ribet juga cara kita nimpa sebuah isi nya untuk property yg ada di class Produk, karena nanti jika produk nya bertambah lagi kita harus buat manual repot ya nanti nya. Nah nanti ada cara nya supaya kalian hanya buat satu instance nya saja.

Nah nanti itu dinamakan Constructor Method, ini adalah method yg ketika otomatis dipanggil ketika sebuah class itu dibikin object nya atau instance nya.