**#6 OOP DASAR pada PHP Inheritance (Bagian 1)**

Inheritance atau pewarisan :

* Menciptakan hierarki antar kelas ***(Parent & Child)*** nanti kedepannya kalo kalian ngoding pake oop itu pasti punya banyak class nah Inheritance memungkin kan kita menghubungkan class” itu dengan hierarki yg nanti nya kalo lebih dari 2 class pasti ada yg nama nya konsep parent dan child.
* ***Child Class***, nanti nya mewarissemua properti dan method dari ***parent***-nya (yang visible), mewarisi itu artinya nanti child class itu bisa menggunakan semua properti dan method yg ada di ***parent***-nya.
* ***Child Class,*** kita buat untuk memperluas atau mengembangkan (extends) fungsionalitas dari Class ***parent***-nya***.***

**WHY inheritance ??**

Nah nanti dari studi kasus sebelum nya kita kembangkan fungsionalitas baru supaya ada problem nya dan nanti kita selesaikan problem nya dengan inheritance.

Jadi sebelum nya kita sudah buat class baru yaitu CetakInfoProduk yg ada mempunyai function cetak bisa menerima parameter dari instance Class lain yaitu si class Produk.

Sekarang kita akan menambahkan fungsionalitas untuk menampilkan info produk secara lebih detail seperti ini :

*// Problem Inharitance*

*// Komik : Twenty Five Twenty One | Jeong Jiman, TVN, (Rp. 1500000) - 100 Halaman.*

*// Game : Uncharted | Neil Druckmann, Sony Computer (Rp. 250000) - 50 jam.*

Beda nya kalo komik nampilin jumlah halaman dan kalo game waktu untuk tamat nya. Kita butuh menambahkan property baru yaitu halaman dan waktu main ya.

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

Nah itu salah satu cara menyelesaikan proble nya, tapi semua property harus di isi saat instancie atau pas buat object nya kan, tapi saat menampilkan detail produk komik misal nya yg terisi hanya proprty $jmlHalaman saja dan sebalik nya kalo yg game juga hanya $waktuMain saja yg ke isi.

Nah mau gamau kita harus mengirimkan 2 parameter gini contoh nya :

class Produk{

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

*// Constructor Method*

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

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

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

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

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

**$this**->jmlHalaman = $jmlHalaman;

**$this**->waktuMain = $waktuMain;

    }

    public function getLabel() {

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

    }

}

class CetakInfoProduk{

    public function cetak(Produk $produk) {

        $str = "{$produk->judul} | {$produk->getLabel()}, (Rp. {$produk->harga}) - {$produk->jmlHalaman} Halaman";

        return $str;

    }

}

$produk2 = new Produk("Twenty Five Twenty One", "Jeong Jiman", "TVN", 1500000, 100, 0);

$produk3 = new Produk("Naruto", "Masashi Kishimoto", "Shonen Jump", 30000, 0, 50);

$newProduk1 = new CetakInfoProduk();

echo $newProduk1->cetak($produk2);

Nah problem nya itu kita gatau tipe nya apa Komik atau Game. Tapi ada solusi nya inget dalam programan solusi nya banyak sekali, gimana kalo kita akalin tambahin tipe saja jadi kita buat method baru di class produk getInfoLengkap() kek gini :

    public function getInfoLengkap() {

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

        if(**$this**->tipe == "komik") {

            $str .= "- {**$this**->jmlHalaman} Halaman";

        } else if(**$this**->tipe == "game") {

            $str .= "- {**$this**->waktuMain} Halaman";

        }

        return $str;

    }

Nah ini kalo kalian mau ngatasin nya bisa seperti ini, tapi ini bener-bener kurang efektif karena nanti jika produk nya sudah banyak misal ada ratusan kondisi nya makin banyak, nanti misal ada hal lain tipe nya nambah bukan halaman sama waktuMain saja ada berbagai macam itu jadi merepotkan bukan.

Nah jadi solusi dari masalah ini kita bisa gunakan **inheritance** untuk implementasi nya ada di video berikut nya.