



Modul Lanjutan Basic Codeigniter

Jefri Maruli H

Pendahuluan

Saya akan membahas lebih detail tentang dasar Pemrograman PHP berbasis OOP (Object Oriented Programming) , mulai dari Class ,Object, Methods,Construct, Inheritance dan lain-lain.

Karena Codeigniter sendiri menggunakan metode berbasis Objek (OOP)



CLASS DAN OBJECT

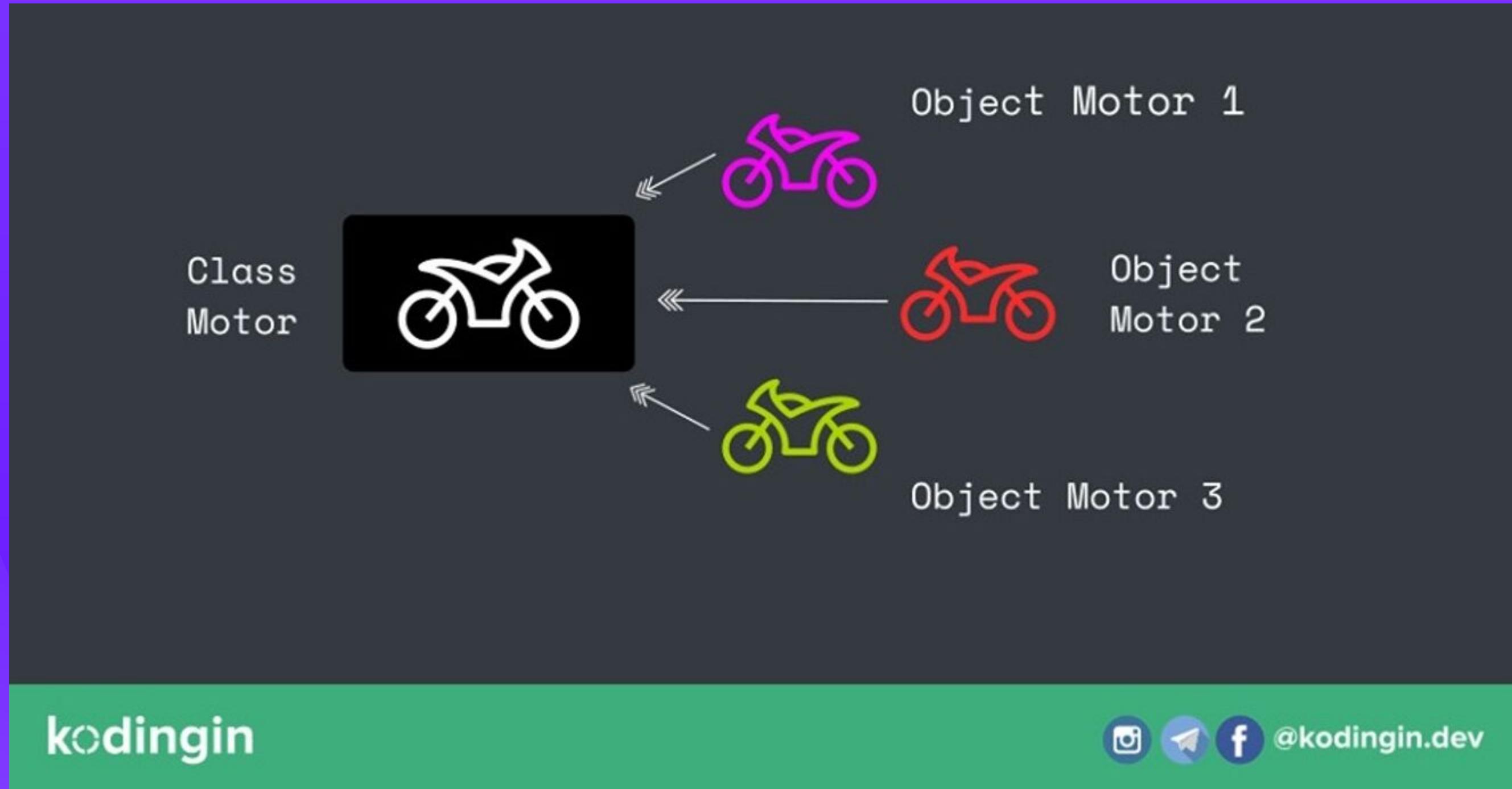
Class adalah Blueprint / kerangka kerja untuk membuat instance dari object.

Object adalah instance dari sebuah Class.
object

dan pengertian instance sendiri adalah salinan yang berbeda dari class, artinya instance menggambarkan sebuah objek dari class itu sendiri.



PENGGAMBARAN CLASS DAN OBJECT ITU SEPERTI APA?



kodingin



SUMBER : KODINGIN.COM



FIELD/ATTRIBUTE/PROPERTY

Field / Attribute adalah data dari Class dan object itu sendiri.



Warna : Merah

Tipe : Sport

Roda : Dua

PHP Motor.php ×

application > controllers > PHP Motor.php > PHP Intelephense > Motor

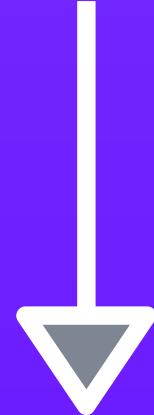
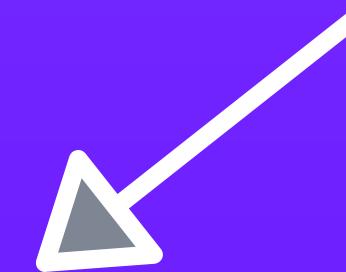
	Attribute	Value/Nilai
1	<?php	
2		
3	class Motor {	
4	public \$warna = "merah";	
5	public \$jumlah_roda = "dua";	
6	public \$tipe = "sport";	
7	}	
8		

METHOD/FUNCTION

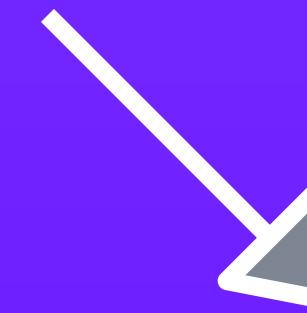
Method / Function adalah Perilaku dari sebuah Class dan Object



Berjalan



Berhenti



Belok Kiri/Kanan

```
<?php

class Motor {

    public $warna = "merah";
    public $jumlah_roda = "dua";
    public $tipe = "sport";

    public function index() {
    }
    public function berjalan() {
        return "berjalan";
    }
    public function berhenti() {
        return "berhenti";
    }
    public function belok_kiri() {
        return "belok kiri";
    }
    public function belok_kanan() {
        return "belok kanan";
    }
}
```



PEMBUATAN OBJECT

Motor1



Objek

Class

```
$motor1 = new Motor;
```

Motor2



```
$motor2 = new Motor;
```

Motor3

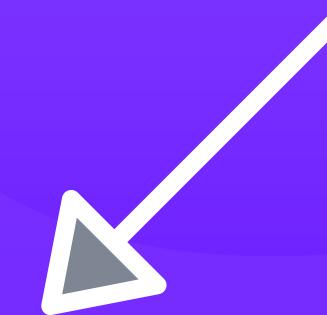


```
$motor3 = new Motor;
```

Object \$motor2 mempunyai semua attribute dan Method dari Class Motor

```
class Motor {
```

```
$motor2 = new Motor;
```



Motor2



Memanggil Atribut
warna merah pada
object \$motor2

```
echo $motor2->warna;
```



```
echo $motor2->berjalan();
```



Memanggil Method
Berjalan

IMPLEMENTASI KE PENULISAN KODE

```
1 <?php
2 class Motor {
3     public $warna = "merah";
4     public $jumlah_roda = "dua";
5     public $tipe = "sport";
6
7     public function index() {
8     }
9     public function berjalan() {
10        return "berjalan";
11    }
12    public function berhenti() {
13        return "berhenti";
14    }
15    public function belok_kiri() {
16        return "belok kiri";
17    }
18    public function belok_kanan() {
19        return "belok kanan";
20    }
21 }
22
23 $motor2 = new Motor;
24 echo 'Motor2 ini berwarna ' . $motor2->warna . '<br> jumlah rodanya ada <br>' . $motor2->jumlah_roda .
25 ' dan tipenya adalah motor ' . $motor2->tipe;
26
27 echo '<br>Motor2 ini <br> dapat ' . $motor2->berjalan() . '<br> dapat ' . $motor2->berhenti() . '<br> dapat ' .
28 ' $motor2->belok_kiri() . '<br> dapat ' . $motor2->belok_kanan();
```

← → ⌂ ⓘ localhost/mardira/index.php/motor
Motor2 ini berwarna merah
jumlah rodanya ada
dua dan tipenya adalah motor sport
Motor2 ini
dapat berjalan
dapat berhenti
dapat belok kiri
dapat belok kanan



LATIHAN

Buatlah 2 Objek yaitu \$motor1 dan \$motor 3 sama seperti membuat object \$motor2 dan menghasilkan output

Motor1 ini berwarna ungu
jumlah rodanya ada dua
dan tipenya adalah motor matic
Motor1 ini
dapat berjalan
dapat berhenti
dapat belok kiri
dapat belok kanan



Motor1

Motor2 ini berwarna merah
jumlah rodanya ada dua
dan tipenya adalah motor sport
Motor2 ini
dapat berjalan
dapat berhenti
dapat belok kiri
dapat belok kanan



Motor2

motor3 ini berwarna hijau
jumlah rodanya ada dua
dan tipenya adalah motor bebek
motor3 ini
dapat berjalan
dapat berhenti
dapat belok kiri
dapat belok kanan



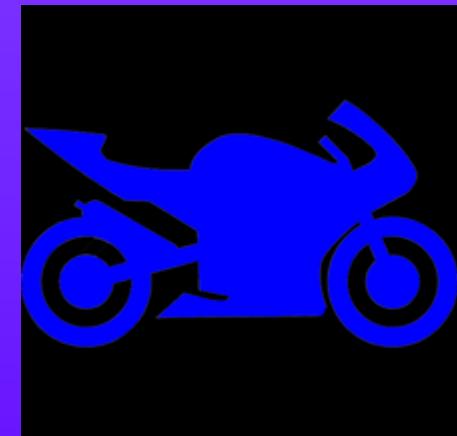
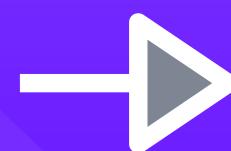
Motor3

MERUBAH NILAI ATTRIBUTE PADA OBJECT

Perubahan nilai/value attribute berlaku ketika object tersebut telah dibuat

Motor1

```
$motor1->warna1 = "biru";
```



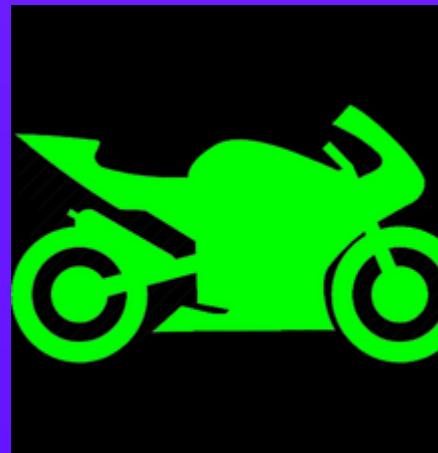
Motor2

```
$motor2->warna2 = "kuning";
```



Motor3

```
$motor3->warna3 = "coklat";
```



CONSTRUCTOR

Constructor adalah method khusus yang otomatis dijalankan pertama kali ketika pembuatan object dan pemanggilan method/function



The screenshot shows a code editor window titled "Mobil.php" with the file path "application > controllers > Mobil.php". The code is as follows:

```
<?php  
class Mobil {  
    public function __construct()  
    {  
        echo "jalan";  
    }  
}  
$mobil = new Mobil;
```

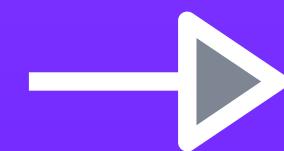
The code defines a class "Mobil" with a constructor. The constructor contains a single line of code "echo "jalan"" which outputs the string "jalan" when the object is created. The code editor uses syntax highlighting to distinguish between different parts of the code.

MEMBERIKAN NILAI AWAL PROPERTY TERHADAP VALUE DARI SUATU FIELD

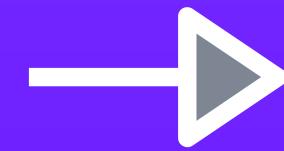
```
1 <?php
2
3 class Mobil
4 {
5
6     public $warna;
7     public function __construct($warna)
8     {
9         $this->warna = $warna;
10    }
11 }
12 $mobil1 = new Mobil("merah");
```

PENGGUNAAN CONSTRUCTOR

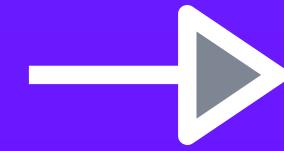
```
$mobil1 = new Mobil("merah");
```



```
$mobil2 = new Mobil("hijau");  
|
```



```
$mobil3 = new Mobil("biru");
```



IMPLEMENTASI CONSTRUCT

```
❶ Mobil.php ×  
application > controllers > Mobil.php > ...  
1  <?php  
2  
3  class Mobil  
4  {  
5  
6      public $warna;  
7      public function __construct($warna)  
8      {  
9          $this->warna = $warna;  
10     }  
11 }  
12  
13 $mobil1 = new Mobil("merah");  
14 $mobil2 = new Mobil("hijau");  
15 $mobil3 = new Mobil("biru");  
16  
17 echo 'mobil ini warnanya '. $mobil1->warna . '<br>';  
18 echo 'mobil ini warnanya '. $mobil2->warna . '<br>';  
19 echo 'mobil ini warnanya '. $mobil3->warna . '<br>';
```

Pengertian Inheritance (Pewarisan) dalam Pemrograman OOP

Inheritance atau Pewarisan atau penurunan adalah konsep pemrograman dimana sebuah Class dapat "menurunkan" **Property/Attribute** dan **Method/Function** kepada Class lainnya.

Konsep Inheritance Class dalam penulisan program, Class yang akan "menurunkan" pewarisan bisa disebut sebagai "Parent Class / Induk Class", dan sedangkan yang menerima "penurunan" bisa disebut sebagai "Child Class/Anakan Class".

```
class Bapak  
{  
}  
● ● ○
```

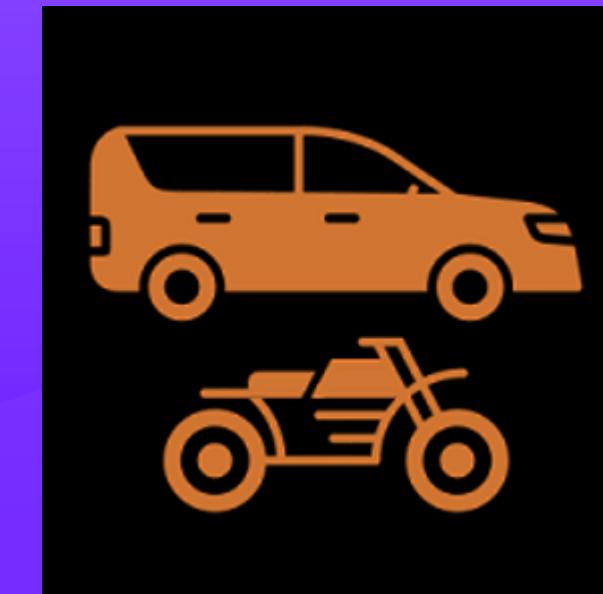


GAMBARAN PENGGUNAAN INHERITANCE

penggunaan Inheritance / penurunan dari sebuah class kepada class lain menggunakan kata 'extends' setelah pendeklarasian sebuah class.

Jika class menggunakan kata 'extends' ,Class tersebut akan menjadi Child/Anakan dari Induk Class , sebaliknya jika class tanpa menggunakan 'extends' artinya class tersebut adalah sebuah parent class

Konsep Inheritance



Class Kendaraan (Parent Class)



Class Mobil (Child)



Class Motor (Child)



Class Bus(Child)



Class Kendaraan
Attribute :

1. Roda

2. Warna

Method:

Berjalan()

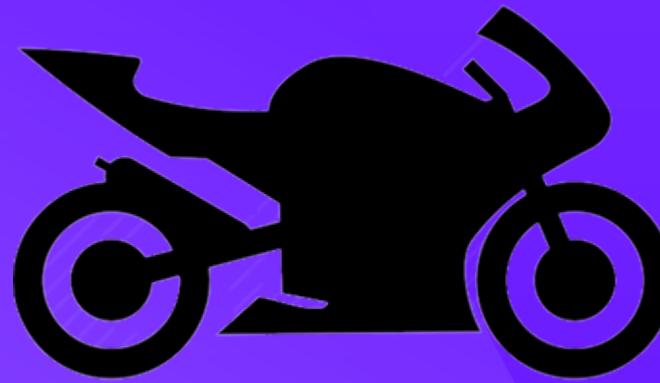
Berhenti()

```
<?php  
  
class Kendaaran  
{  
    public $rodamotor = "dua";  
    public $warna = "merah";  
  
    public function berjalan()  
    {  
        return "berjalan";  
    }  
  
    public function berhenti()  
    {  
        return "berhenti";  
    }  
}
```



Class Motor mewarisi Class kendaraan, tidak perlu mendefinisikan atribut atau method yang sudah di definisikan oleh Class Kendaraan

Class Kendaraan



Class Motor

Extends

Class Kendaraan

```
class Motor extends Kendaraan{  
}  
$motor = new Motor;  
echo 'motor ini mempunyai roda '.$motor->rodamotor;  
echo '<br> dan motor ini '.$motor->berjalan();|
```



Child class juga bisa mendefinisikan
attribute & methods yang tidak ada di parent
class

Attribute & Method class Motor:
warisan dari Class

Atribut

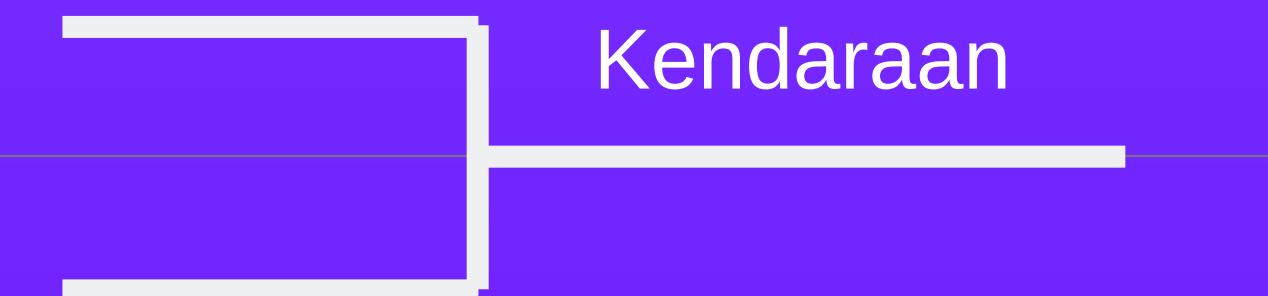
1. Roda

2. Warna

Method:

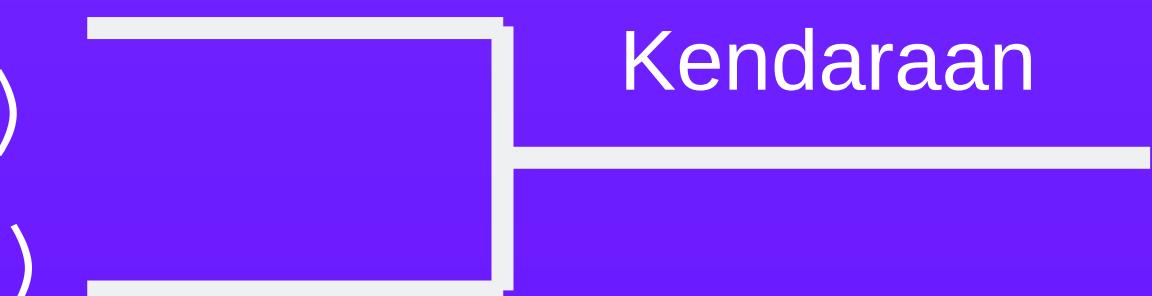
1. berjalan()

2. berhenti()



warisan dari Class

Kendaraan



Method asli di Class Motor

belok_kiri()

```
class Motor extends Kendaraan
{
    public function belok_kiri()
    {
        return "belok kiri";
    }
}
```

IMPLEMENTASI INHERITANCE



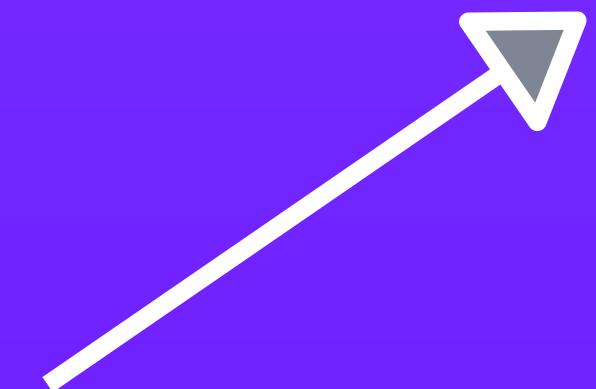
```
1 <?php
2 class Kendaraan {
3     public $rodamotor = "dua";
4     public $warna = "merah";
5     public function index()
6     { }
7
8     public function berjalan() {
9         return "berjalan";
10    }
11
12    public function berhenti() {
13        return "berhenti";
14    }
15 }
16 class Motor extends Kendaraan {
17     public function belok_kiri() {
18         return "belok kiri";
19     }
20 }
21
22 $motor = new Motor;
23 echo 'motor ini mempunyai roda ' . $motor->rodamotor;
24 echo '<br> dan motor ini ' . $motor->berjalan();
25 echo '<br> motor ini bisa ' . $motor->belok_kiri();
```

The screenshot shows a browser window with the URL `localhost/mardira/index.php/kendaraan`. The page content displays the following text:
motor ini mempunyai roda dua
dan motor ini berjalan
motor ini bisa belok kiri

LATIHAN



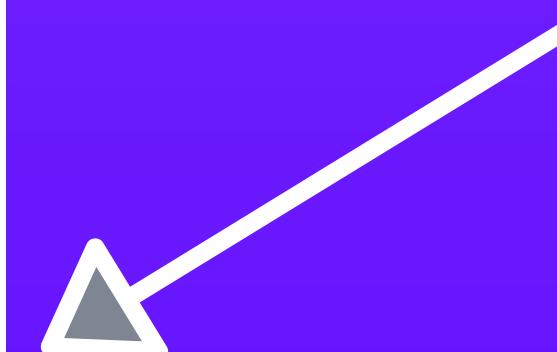
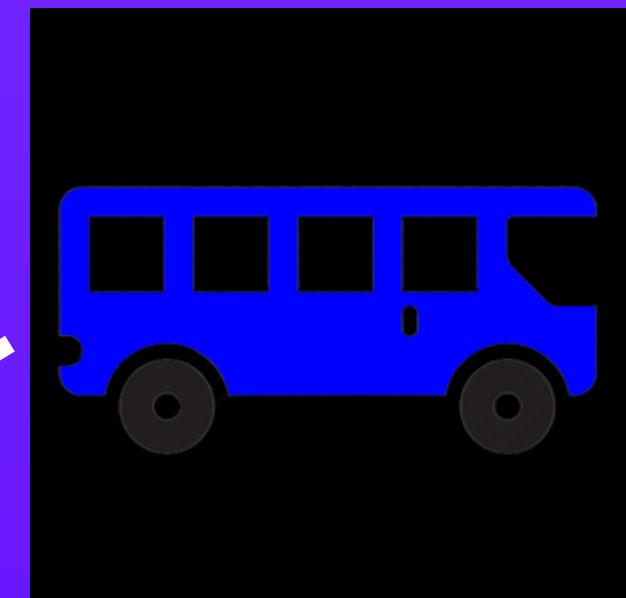
Buatlah 2 Class Turunan(Child Class)
yaitu Class Mobil dan Class Bus dari parent
Class Kendaraan dan menghasilkan output



mobil ini mempunyai roda empat
motor ini berwarna hijau
dan mobil ini berjalan
mobil ini bisa belok kiri
mobil ini bisa belok kanan

motor ini mempunyai roda dua
motor ini berwarna merah
dan motor ini berjalan
motor ini bisa belok kiri
motor ini bisa belok kanan

bus ini mempunyai roda enam
motor ini berwarna biru
dan bus ini berjalan
bus ini bisa belok kiri
bus ini bisa belok kanan





TERIMAKASIH