

PRAKTIKUM DUA BELAS MATA KULIAH BAHASA BAHASA PEMROGRAMAN : PHP

Oleh: Fauzan Zacky Firmansyah (23106050047)

Visibility Publik

```
01Visibility-public.php > ...
1  <?php
   8 references | 1 implementation
2  class Produk {
   13 references
3     public $merek;
   4 references | 0 overrides
4     public function hello(): string {
5         return "ini adalah Produk";
6     }
7 }
8
9 $produk01 = new Produk(); Missing argument 1..2 for __construct(
10 $produk01->merek = "Asus";
11
12 echo $produk01->merek;
13 echo "<br>";
14 echo $produk01->hello();
```

Asus
ini adalah Produk

Visibility Private

```
02Visibility-privat.php > ...
1  <?php
   8 references | 1 implementation
2  class Produk {
   13 references
3     private $merek;
   4 references
4     private function hello(): string {
5         return "ini adalah Produk";
6     }
7 }
8
9 $produk01 = new Produk(); Missing argument 1..2 for __construct(
10
11 // fatal error: cannot access private property
12 $produk01->merek = "Asus";
13
14 // fatal error: cannot access private property
15 echo $produk01->merek;
16 echo "<br>";
17 // fatal error: cannot access private property
18 echo $produk01->hello();
```

Fatal error: Uncaught Error: Cannot access private property Produk::\$merek in D:\xampp\htdocs\BBP\prak12\02Visibility-privat.php:12 Stack trace: #0 {main} thrown in D:\xampp\htdocs\BBP\prak12\02Visibility-privat.php on line 12

Kode diatas error karena hal berikut:

Properti merek: Properti \$merek dideklarasikan sebagai private di dalam kelas Produk, yang artinya hanya bisa diakses dari dalam kelas itu sendiri. Pada kode ini, Anda mencoba mengakses \$produk01->merek di luar kelas (yaitu pada bagian \$produk01->merek = "Asus"; dan echo \$produk01->merek;), sehingga muncul error karena properti ini tidak dapat diakses dari luar.

Metode hello: Sama seperti \$merek, metode hello() juga dideklarasikan sebagai private. Jadi, metode ini hanya dapat dipanggil dari dalam kelas Produk. Karena pada kode di atas Anda mencoba memanggilnya di luar kelas (echo \$produk01->hello();), maka terjadi error.

```
03Visibility-privat2.php > ...
1 <?php
2 class Produk {
3     private $merek;
4
5     public function setMerek($merek): void {
6         $this->merek = $merek;
7     }
8
9     public function getMerek(): string{
10        return $this->merek;
11    }
12
13    $produk01 = new Produk(); Missing argument 1..2 for __construct(
14    $produk01->setMerek("Asus");
15
16    echo $produk01->getMerek();
```

Asus

Visibility Protected

```
04Visibility-protected.php > ...
1 <?php
2 class Produk {
3     protected $merek;
4     protected function hello(): string {
5         return "ini adalah Produk";
6     }
7 }
8
9 $produk01 = new Produk(); Missing argument 1..2 for __construct(
10 // Fatal error: Uncaught Error: Cannot access protected property
11 $produk01->merek = "Asus";
12
13 // Fatal error: Uncaught Error: Cannot access protected property
14 echo $produk01->merek;
15 echo "<br>";
16
17 // Fatal error: Uncaught Error: Cannot access protected property
18 echo $produk01->hello();
```

Fatal error: Uncaught Error: Cannot access protected property Produk::\$merek in D:\xampp\htdocs\BBP\prak12\04Visibility-protected.php:11 Stack trace: #0 {main} thrown in D:\xampp\htdocs\BBP\prak12\04Visibility-protected.php on line 11

Kode diatas error karena properti \$merek dan metode hello() dideklarasikan dengan aksesibilitas protected. Dalam PHP, aksesibilitas protected berarti bahwa properti dan metode hanya bisa diakses dari dalam kelas itu sendiri atau dari kelas turunan (subclass). Namun, pada kode di atas, Anda mencoba mengaksesnya dari luar kelas Produk, sehingga muncul error.

```
05Visibility-protected2.php > ...
1 <?php
2 class Produk {
3     protected $merek = "Asus";
4     protected function hello(): string {
5         return "ini adalah Produk";
6     }
7 }
8
9 class Laptop extends Produk {
10     public function helloLaptop(): string {
11         return $this->hello()." Laptop ".$this->merek;
12     }
13 }
14 $produk01 = new Laptop(); Missing argument 1..2 for __construct(
15 echo $produk01->helloLaptop();
```

ini adalah Produk Laptop Asus

Getter dan Setter

```
06Getter&Setter.php > ...
1 <?php
2 class Produk {
3     private $merek;
4     private $stok;
5     public function setMerek($merek): void {
6         $this->merek = $merek;
7     }
8     public function setStok($stok): void {
9         $this->stok = $stok;
10    }
11    public function getMerek(): mixed {
12        return $this->merek;
13    }
14    public function getStok(): mixed {
15        return $this->stok;
16    }
17 }
18 $produk01 = new Produk();
19 $produk01->setMerek("Acer");
20 $produk01->setStok(10);
21
22 echo $produk01->getMerek();
23 echo "<br>";
24 echo $produk01->getStok();
```

Acer
10

Contoh penerapan getter dan setter

```
07Getter&Setter-implementation.php > ...
1 <?php
2 class Produk {
3     private $stok = 0;
4     public function setStok($stok): void {
5         if(is_int(value: $stok)) {
6             $this->stok = $stok;
7         }
8         else {
9             echo "Error: stok harus angka bulat <br>";
10        }
11    }
12    public function getStok(): mixed {
13        return $this->stok;
14    }
15 }
16
17 $produk01 = new Produk();
18 echo $produk01->getStok();
19 echo "<br>";
20
21 $produk01->setStok(10.5);
22 echo $produk01->getStok();
23 echo "<br>";
24 $produk01->setStok("Satu");
25 echo $produk01->getStok();
26 echo "<br>";
27 $produk01->setStok(10);
28 echo $produk01->getStok();
```

0
Error: stok harus angka bulat
0
Error: stok harus angka bulat
0
10

Setter Getter dan constructor

```

1 <?php
2 class Produk {
3     private $merek;
4     private $stok;
5     private function setMerek($merek): void{
6         if(is_string(value: $merek)){
7             $this->merek = $merek;
8         }
9         else {
10             die("Error: merek harus berbentuk string <br>");
11         }
12     }
13     private function setStok($stok): void {
14         if(is_int(value: $stok)) {
15             $this->stok = $stok;
16         }
17         else {
18             die("Error: stok harus angka bulat <br>");
19         }
20     }
21     public function __construct($merek, $stok){
22         $this->setMerek($merek);
23         $this->setStok($stok);
24     }
25     public function getMerek(): string{
26         return strtoupper(string: $this->merek);
27     }
28     public function getStok(): mixed{
29         return $this->stok;
30     }
31 }
32
33 $produk01 = new Produk(merek: "Acer", stok: 10);
34 echo "Stok Produk ". $produk01->getMerek().": ". $produk01->getStok();

```

Stok Produk ACER: 10