

TEORI PEMROGRAMAN BERORIENTASI OBJEK

23 April 2024

VISIBILITY

1. DASAR TEORI

a. Visibility

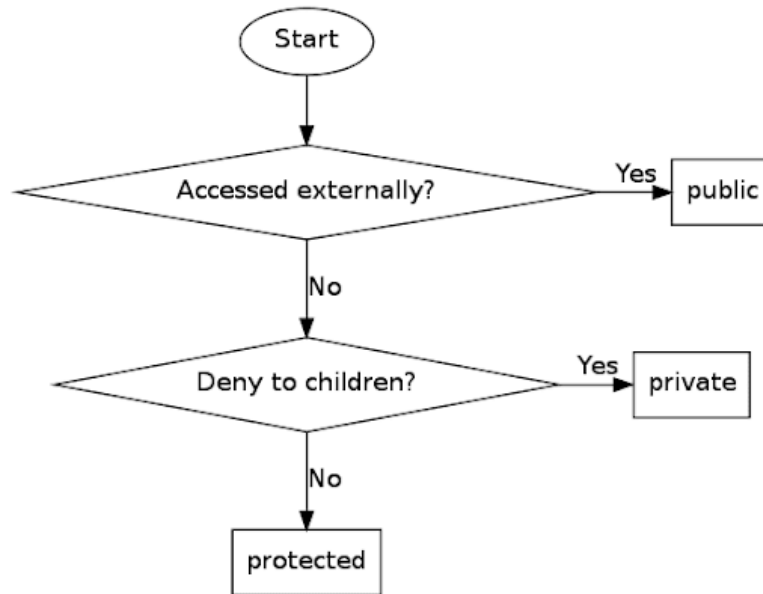
Visibility (atau kadang disebut juga access modifiers) adalah konsep dalam pemrograman berorientasi objek yang mengontrol akses ke properti dan metode dalam sebuah kelas.

b. Jenis-Jenis Visibility

Di Visibility sendiri di dalam pemrograman merujuk pada tingkat aksesibilitas dari suatu variabel, method, atau class dalam sebuah program. Terdapat beberapa tingkat visibility dalam pemrograman, yaitu:

- **Public:** Variabel, method, atau class dengan visibility public dapat diakses oleh semua kelas dan paket dalam program.
- **Private:** Variabel, method, atau class dengan visibility private hanya dapat diakses oleh kelas yang mendeklarasikan variabel, method, atau class tersebut.
- **Protected:** Variabel, method, atau class dengan visibility protected dapat diakses oleh kelas yang mendeklarasikan variabel, method, atau class tersebut, serta kelas turunan (subclass) dari kelas tersebut.

Contoh :



2. Contoh Penggunaan :

a. Public | Membuat class yang Propertynya adalah Public

```
class Parent {  
    public $property1; // properti public  
  
    public function getProperty() {  
        return $this->property;  
    }  
}
```

```
// Membuat objek dari kelas Parent  
$objek1 = new Parent("isian property");
```

```
// Mengakses properti public dari luar kelas  
echo "Hasil Objek: " . $Parent->property . "<br>";
```

```
// Memanggil metode public untuk mendapatkan merek mobil  
echo "Hasil Objek (dari property): " . $Parent->getProperty() . "<br>";  
?>
```

Untuk Public sendiri isi dari classnya bisa diambil langsung dari luar kelas.

b. Protected | Membuat class yang Propertynya adalah Protected

```
class ParentClass {  
    protected $property1; // properti protected  
  
    public function getProperty() {  
        return $this->property1;  
    }  
}  
  
// Subclass yang meng-extend kelas ParentClass  
class ChildClass extends ParentClass {  
    public function setProperty($value) {  
        // Mengakses properti protected dari dalam subkelas  
        $this->property1 = $value;  
    }  
}  
  
// Membuat objek dari kelas ChildClass  
$childObj = new ChildClass();  
  
// Mengatur nilai properti protected dari dalam subkelas  
$childObj->setProperty("Isian property");  
  
// Mengakses properti protected melalui metode public dari subkelas  
echo "Hasil Objek (dari metode): " . $childObj->getProperty() . "<br>";
```

Untuk Protected sendiri kita tidak bisa memanggil property secara langsung dari objek, tetapi masih bisa dipanggil melalui child class.

c. Private | Membuat class yang Propertynya Private

```
class ParentClass {
    private $property1; // properti private

    public function getProperty() {
        return $this->property1;
    }
}

// Subclass yang meng-extend kelas ParentClass
class ChildClass extends ParentClass {
    public function setProperty($value) {
        // Mengakses properti private dari dalam subkelas tidak diperbolehkan
        // $this->property1 = $value;
    }
}

// Membuat objek dari kelas ChildClass
$childObj = new ChildClass();

// Tidak dapat mengatur nilai properti private dari luar kelas
// $childObj->setProperty("Isian property");

// Tidak dapat mengakses properti private melalui metode public dari subkelas
// echo "Hasil Objek (dari metode): " . $childObj->getProperty() . "<br>";

// Memanggil metode public untuk mendapatkan properti private
echo "Hasil Objek (dari metode): " . $objek1->getProperty() . "<br>";
```

Untuk Private sendiri kita tidak bisa memanggil property secara langsung dari objek dan dari child class sendiri juga kita juga tidak bisa memanggilnya, jadi untuk memanggilnya kita harus membuat method sendiri di parent classnya agar bisa dipanggil propertynya.

3. PRAKTIK

Membuat class yang berisikan property public

1. Setelah file .php terbuat, buatlah Parent class dan buatlah property public dan methodnya.

```
3 references | 1 implementation
class Car {
    7 references
    public $brand; // properti public

    2 references | 0 overrides
    public function __construct($brand) {
        $this->brand = $brand;
    }

    1 reference | 0 overrides
    public function getBrand() {
        return $this->brand;
    }
} <- #2-12 class Car
```

2. Kemudian buatlah agar property public dari parent class bisa dipanggil.

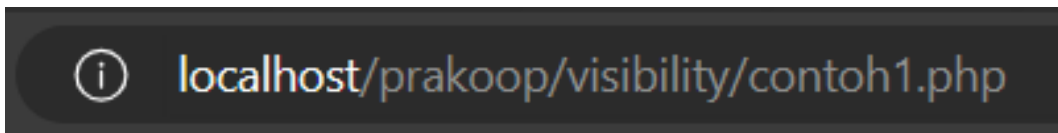
```
// Membuat objek dari kelas Car
$car = new Car("Toyota");

// Mengakses properti public dari luar kelas
echo "Merek mobil: " . $car->brand . "<br>";

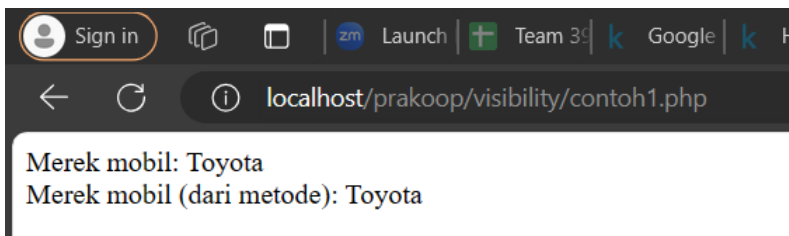
// Memanggil metode public untuk mendapatkan merek mobil
echo "Merek mobil (dari metode): " . $car->getBrand() . "<br>";
?> The closing ?> tag should be omitted from files containing only PHP.
```

3. Kemudian buka browser dan ketiklah url:

`http://localhost/namafolder(jika langsung htdocs bisa skip)/namafile.php`



4. Ketika url tersebut dibuka maka Hasil Akhir seperti ini :



Membuat class yang berisikan property protected

1. Setelah file .php terbuat, buatlah Parent class dan buatlah property public dan methodnya.

```
1 reference | 1 implementation
class Menu {
  2 references
  protected $name; // properti protected

  0 references | 0 overrides
  public function __construct($name) {
    $this->name = $name;
  }

  1 reference | 0 overrides
  protected function getName() {
    return $this->name;
  }
} <- #2-12 class Menu
```

2. Setelah itu buatlah Child Classnya beserta methodnya.

```
1 reference | 0 implementations
class NasiPadang extends Menu {
  1 reference | 0 overrides
  public function displayMenu() {
    // Memanggil metode protected dari kelas induk
    return "Menu Nasi Padang: " . $this->getName();
  }
} <- #14-19 class NasiPadang extends Menu
```

3. Kemudian buatlah agar property protected dari parent class bisa dipanggil.

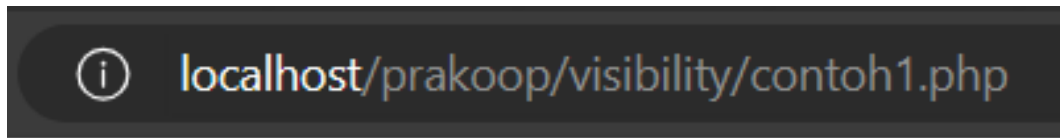
```
// Membuat objek dari kelas NasiPadang
$nasiPadang = new NasiPadang("Rendang");

// Tidak dapat mengakses properti protected dari luar kelas
// echo "Nama menu: " . $nasiPadang->name . "<br>";

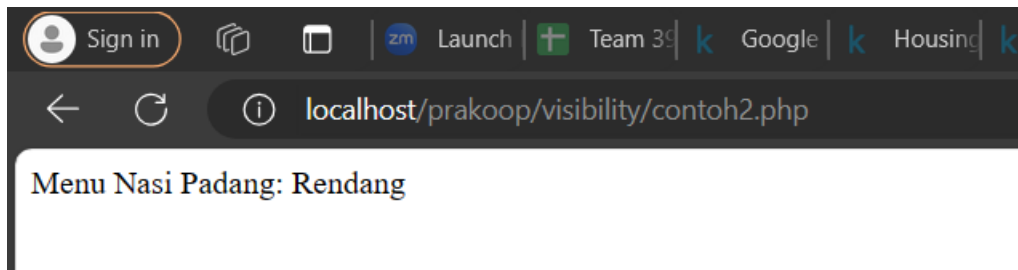
// Tidak dapat memanggil metode protected dari luar kelas
// echo "Nama menu (dari metode): " . $nasiPadang->getName() . "<br>";

// Memanggil metode public yang mengakses properti protected
echo $nasiPadang->displayMenu() . "<br>";
?> The closing ?> tag should be omitted from files containing only PHP.
```

4. Kemudian buka browser dan ketiklah url:
[http://localhost/namafolder\(jika langsung htdocs bisa skip\)/namafile.php](http://localhost/namafolder(jika langsung htdocs bisa skip)/namafile.php)



5. Ketika url tersebut dibuka maka Hasil Akhir seperti ini :



Membuat class yang berisikan property private

1. Setelah file .php terbuat, buatlah Parent class dan buatlah property private dan methodnya.

```
class Pakaian {  
    2 references  
    private $jenis; // properti private  
  
    0 references | 0 overrides  
    public function __construct($jenis) {  
        $this->jenis = $jenis;  
    }  
  
    1 reference | 0 overrides  
    public function getJenis() {  
        return $this->jenis;  
    }  
}
```

2. Setelah itu buatlah Child Classnya beserta methodnya.

```
// Subclass yang meng-extend kelas Pakaian  
1 reference | 0 implementations  
class PakaianAnak extends Pakaian {  
    0 references | 0 overrides  
    public function setJenis($value) {  
        // Mengakses properti private dari dalam subkelas tidak diperbolehkan  
        // $this->jenis = $value;  
    }  
}  
} <- #15-20 class PakaianAnak extends Pakaian
```

3. Kemudian buatlah agar property private dari parent class bisa dipanggil.

```
// Membuat objek dari kelas PakaianAnak
$pakaianAnak = new PakaianAnak("Baju Anak");

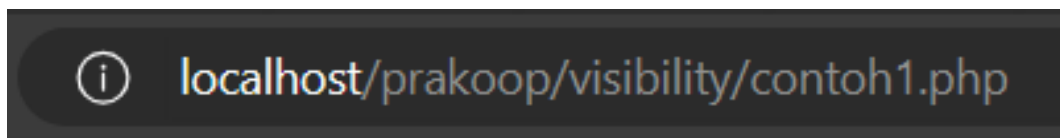
// Tidak dapat mengatur nilai properti private dari luar kelas
// $pakaianAnak->setJenis("Baju Anak Laki-laki");

// Tidak dapat mengakses properti private melalui metode public dari subkelas
// echo "Jenis Pakaian: " . $pakaianAnak->jenis . "<br>";

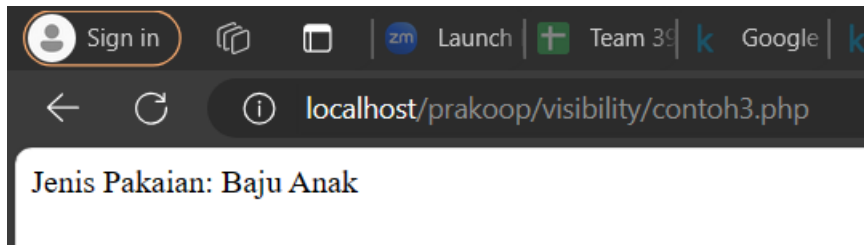
// Memanggil metode public untuk mendapatkan properti private
echo "Jenis Pakaian: " . $pakaianAnak->getJenis() . "<br>";
The closing ?> tag should be omitted from files containing only PHP.
```

4. Kemudian buka browser dan ketiklah url:

`http:localhost/namafolder(jika langsung htdocs bisa skip)/namafile.php`



5. Ketika url tersebut dibuka maka Hasil Akhir seperti ini :



Membuat class yang berisikan Kombinasi Property Visibility Dan Method Visibility [1]

1. Setelah file .php terbuat, buatlah Parent class dan buatlah propertynya.

```
Click here to ask Blackbox to help you code faster
1  <?php
   2 references | 1 implementation
2  class Person {
3
4  4 references
   public $name; // public property
5
6  2 references
   private $age; // private property
7
8  2 references
   protected $address; // protected property
9
```

2. Setelah itu buatlah __construct.

```
1 reference | 0 overrides
public function __construct($name, $age, $address) {
    $this->name = $name;
    $this->age = $age;
    $this->address = $address;
} <- #10-18 public function __construct($name, $age, $address)
```

3. Setelah itu buatlah Methodnya.

```
0 references | 0 overrides
public function greet() {
    echo "Hello, my name is " . $this->name;
} <- #20-24 public function greet()

0 references
private function getAge() {
    return $this->age;
} <- #26-30 private function getAge()

1 reference | 0 overrides
protected function getAddress() {
    return $this->address;
} <- #32-36 protected function getAddress()

} <- #2-38 class Person
```

4. Setelah itu buatlah Child Classnya.

```
class Student extends Person
{
    1 reference | 0 overrides
    public function showInfo()
    {
        echo "Name: " . $this->name; // bisa diakses karena property name public
        echo "<br>";

        // echo "Age: " . $this->getAge(); // tidak bisa diakses karena method getAge() private
        echo "Address: " . $this->getAddress(); // bisa diakses karena method getAddress() protected
    } <- #44-54 public function showInfo()
} <- #41-55 class Student extends Person
```

5. Kemudian buatlah agar property dari parent class bisa dipanggil.

```
$person = new Person("John", 25, "Jakarta");

$person->name = "Doe"; // bisa diakses karena property name public
// $person->age = 30; // tidak bisa diakses karena property age private
// $person->address = "Bandung"; // tidak bisa diakses karena property address protected

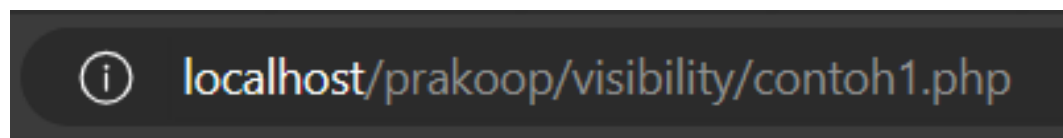
$student = new Student("Jane", 20, "Surabaya");

$student->showInfo(); // bisa mengakses method getAddress() karena inheritance dan protected
```

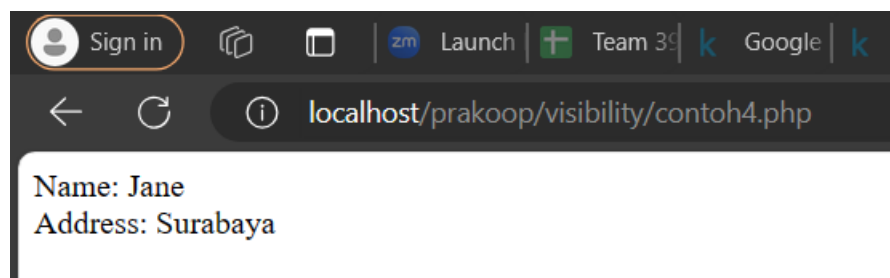
6. Kemudian buka browser dan ketiklah

url:

[http://localhost/namafolder\(jika langsung htdocs bisa skip\)/namafile.php](http://localhost/namafolder(jika langsung htdocs bisa skip)/namafile.php)



7. Ketika url tersebut dibuka maka Hasil Akhir seperti ini :



Membuat class yang berisikan Kombinasi Property Visibility Dan Method Visibility [2]

1. Setelah file .php terbuat, buatlah Parent class dan buatlah propertynya.

```
3 references | 1 implementation
class Car
{
    7 references
    public $brand;

    3 references
    protected $model;

    2 references
    private $year;
}
```

2. Setelah itu buatlah __construct.

```
2 references | 0 overrides
public function __construct($brand, $model, $year)
{
    $this->brand = $brand;

    $this->model = $model;

    $this->year = $year;
} <- #13-20 public function __construct($brand, $model, $year)
```

3. Setelah itu buatlah Methodnya.

```
0 references | 0 overrides
public function getDetails()
{
    echo "Brand: " . $this->brand . "<br>";

    echo "Model: " . $this->model . "<br>";

    echo "Year: " . $this->year . "<br>";
} <- #23-30 public function getDetails()
} <- #4-31 class Car
```

4. Setelah itu buatlah Child Classnya.

```
1 reference | 0 implementations
class SportsCar extends Car
{
    1 reference | 0 overrides
    public function showDetails()
    {
        echo "Brand: " . $this->brand . "<br>"; // bisa diakses karena public
        echo "Model: " . $this->model . "<br>"; // bisa diakses karena protected
        // echo "Year: " . $this->year . "<br>"; // tidak bisa diakses karena private
    } <- #37-45 public function showDetails()
} <- #34-46 class SportsCar extends Car
```

5. Kemudian buatlah agar property dari parent class bisa dipanggil.

```
$car = new Car("Toyota", "Camry", "2021");

$car->brand = "Honda"; // bisa diakses karena public
// $car->model = "Accord"; // tidak bisa diakses karena protected
// $car->year = "2022"; // tidak bisa diakses karena private

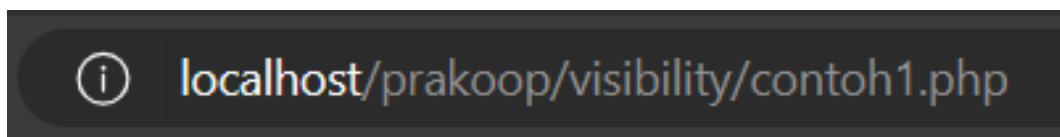
$sportsCar = new SportsCar("Ferrari", "488 GTB", "2020");

$sportsCar->showDetails(); // bisa mengakses property brand dan model karena public dan protected
```

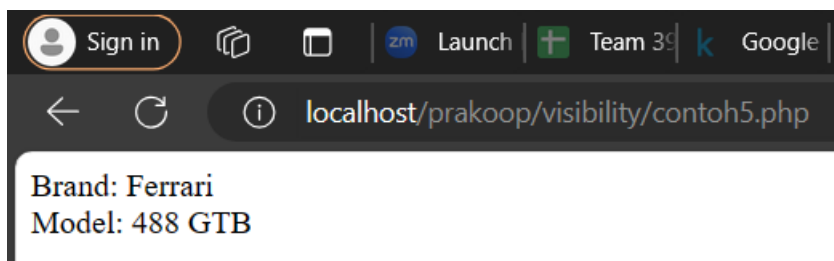
6. Kemudian buka browser dan ketiklah

url:

[http://localhost/namafolder\(jika langsung htdocs bisa skip\)/namafolder.php](http://localhost/namafolder(jika langsung htdocs bisa skip)/namafolder.php)



7. Ketika url tersebut dibuka maka Hasil Akhir seperti ini :



3. PERCOBAAN MANDIRI

Lakukanlah Percobaan Praktik Dimulai dari atas, setelah dilakukan buatlah variasi dengan tema yang berbeda dari contoh diatas.

4. TUGAS TAKE HOME

PENGUMPULAN TUGAS DAN DEADLINE

- a. DI Google Drive
- b. Deadline = 30 April 2024 | 23.59 WIB

Soal |

1. Buatlah code untuk melakukan sesuatu dengan menggunakan metode dan property yang sudah ada visibilitynya, setiap metode dan property setidaknya memiliki semua jenis property minimal 1 per jenis dan tampilkanlah hasilnya.
2. Buatlah Penjelasan dari masing masing Outputnya
3. Dikumpulkan file .php dan laporannya di google drive

DAFTAR PUSTAKA

- <https://amuofficial.net/web-programming/visibility-atau-access-modifier-dalam-php/>
- <https://torquemag.io/2016/05/understanding-concept-visibility-object-oriented-php/>
- https://www.youtube.com/watch?v=fMjvxnbmROk&list=PLFIM0718LjIWvxxll-6wLXrC_16h_Bl_p&index=9&ab_channel=WebProgrammingUNPAS