JOBSHEET 3

Menerapkan Konsep Inheritance, Polymorphism, Encapsulation, dan Abstraction dalam PHP

1. Latihan
2. Inheritance

* Membuat kelas person dengan atribut nama dan metode getName()

1. Membuat kelas person

Kelas merupakan struktur atau blueprint yang mendefinisikan atribut dan metode yang dimiliki oleh suatu objek.

Untuk membuat kelas baru, digunakanlah sintaks sebagai berikut:

class Person{}

1. Membuat protected properties

Atribut atau properties adalah variabel yang menyimpan data untuk untuk objek.

Protected properties merupakan properties yang hanya dapat di akses di kelas itu sendiri dan kelas turunannya.

Untuk membuat protected properties, digunakanlah sintaks sebagai berikut:

protected $nama;

1. Membuat konstruktor

Konstruktor digunakan untuk mengatur nilai awal dari objek saat pertama kali dibuat.

Konstruktor juga bisa digunakan untuk menginisialisasi properti dengan nilai yang diberikan saat instansiasi.

Sintaks yang digunakan dalam penulisan metode ini sebegai berikut:

public function \_\_construct($nama) { $this->nama = $nama; }

1. Membuat metod getName()

Metod ini mengembalikan nilai dari atribut nama.

Sintaks yang digunakan sebagai berikut:

public function getName() { return $this->nama; }

* Membuat kelas Student yang mewarisi dari Person dan menambahkan atribut studentID serta metode getStudentID()

1. Membuat kelas Student yang merupakan kelas turunan dari kelas Person

Kelas Student mewarisi kelas person sehinggan akan memiliki semua atribut dan metod yang ada di kelas Person.

Sintaks yang digunakan sebagai berikut:

class Student extends Person {}

1. Membuat konstruktor

Konstruktor di kelas Student memanggil konstruktor di kelas person untuk menginisialisasi atribut/properties nama, kemudian menginisialisasi studentID.

Sintaks yang diguanakan sebagai berikut:

public function \_\_construct($nama, $studentID) { parent::\_\_construct($nama); $this->studentID = $studentID; }

1. Membuat metod getStudentID()

Metod getStudentID() berfungsi untuk mengembalikan nilai dari atribut studentID

Sintaks yang digunakan sebagai berikut:

public function getStudentID() { return $this->studentID; }

* Instansiasi dan penggunaan

1. Membuat objek student

Objek student dibuat dengan nama (“Alifia”) dan ID Mahasiswa “2309”

Sintaks yang digunakan sebagai berikut:

$student1 = new Student("Alifia", "2309");

1. Memanggil metode

Memanggil metod getName() untuk menampilkan nama mahasiswa

Sintaks yang digunakan sebagai berikut:

echo $student1->getName();

Memanggil metod getStudentID() untuk menampilkan ID mahasiswa

Sintaks yang digunakan sebagai berikut:

echo $student1->getStudentID();

1. Output

Output yang dihasilkan sebagai berikut:

Alifia  
2309

1. Polymorphism

* Membuat kelas Teacher yang juga mewarisi dari Person dan menambahkan atribut teacherID

1. Membuat kelas Person

Kelas merupakan struktur atau blueprint yang mendefinisikan atribut dan metode yang dimiliki oleh suatu objek.

Untuk membuat kelas baru, digunakanlah sintaks sebagai berikut:

class Person {}

1. Membuat protected properties

Atribut atau properties adalah variabel yang menyimpan data untuk untuk objek.

Protected properties merupakan properties yang hanya dapat di akses di kelas itu sendiri dan kelas turunannya.

Untuk membuat protected properties, digunakanlah sintaks sebagai berikut:

protected $nama;

1. Membuat konstruktor

Konstruktor digunakan untuk mengatur nilai awal dari objek saat pertama kali dibuat.

Konstruktor juga bisa digunakan untuk menginisialisasi properti dengan nilai yang diberikan saat instansiasi.

Sintaks yang digunakan dalam penulisan metode ini sebegai berikut:

public function \_\_construct($nama) { $this->nama = $nama; }

1. Membuat metod getName()

Metod ini mengembalikan nilai dari atribut nama.

Sintaks yang digunakan sebagai berikut:

public function getName() { return $this->nama; }

1. Membuat kelas Student yang merupakan turunan dari kelas Person

Kelas Students mewarisi kelas Person sehingga dapat menggunakan atribut dan metode dari kelas Person.

Sintaks yang digunakan sebagai berikut:

class Student extends Person {}

1. Membuat public properties

Atribut atau properties adalah variabel yang menyimpan data untuk untuk objek.

Public properties merupakan properties yang dapat di akses oleh semua kelas.

Untuk membuat protected properties, digunakanlah sintaks sebagai berikut:

public $studentID;

1. Membuat konstruktor

Konstruktor digunakan untuk mengatur nilai awal dari objek saat pertama kali dibuat.

Konstruktor juga bisa digunakan untuk menginisialisasi properti dengan nilai yang diberikan saat instansiasi.

Sintaks yang digunakan dalam penulisan metode ini sebagai berikut:

public function \_\_construct($nama, $studentID) { parent::\_\_construct($nama); $this->studentID = $studentID; }

1. Membuat metod getStudentID()

Metod ini digunakan untuk mengembalikan nilai studentID

Sintaks yang digunakan sebagai berikut:

public function getStudentID() { return "Student ID: $this->studentID"; }

1. Membuat kelas Teacher yang merupakan turunan dari kelas Person

Kelas Teacher mewarisi kelas Person sehingga dapat menggunakan atribut dan metod dari kelas Person.

Sintaks yang digunakan sebagai berikut:

class Teacher extends Person {}

1. Membuat private properties

Atribut atau properties adalah variabel yang menyimpan data untuk untuk objek.

Private properties merupakan properties yang hanya dapat digunakan oleh kelas itu sendiri

Untuk membuat protected properties, digunakanlah sintaks sebagai berikut:

private $teacherID;

1. Membuat konstruktor

Konstruktor digunakan untuk mengatur nilai awal dari objek saat pertama kali dibuat.

Konstruktor juga bisa digunakan untuk menginisialisasi properti dengan nilai yang diberikan saat instansiasi.

Sintaks yang digunakan dalam penulisan metode ini sebagai berikut:

public function \_\_construct($nama, $teacherID) { parent::\_\_construct($nama); $this->teacherID = $teacherID; }

1. Membuat metod getTeacherID()

Metod ini digunakan untuk mengembalikan nilai teacherID

Sintaks yang digunakan sebagai berikut:

public function getTeacherID() { return "Teacher ID is: $this->teacherID"; }

* Override metode getName() di kelas Student dan Teacher untuk menampilkan format berbeda

1. Membuat metod getName() pada kelas Teacher

Metod ini digunakan untuk mengembalikan nilai nama di kelas Teacher.

Sintaks yang digunakan sebagai berikut:

public function getName() { return "Teacher name is: $this->nama"; }

1. Membuat metod getName() pada kelas Student

Metod ini digunakan untuk mengembalikan nilai nama di kelas Student.

Sintaks yang digunakan sebagai berikut:

public function getName() { return "Student: $this->nama"; }

* Instansiasi objek dan penggunaan metod

Sintaks yang digunakan sebagai berikut:

$student1 = new Student("Alifia", "2309"); echo $student1->getName(); echo "<br>"; echo $student1->getStudentID(); echo "<br>"; $teacher1 = new Teacher("Manda", "2890"); echo $teacher1->getName(); echo "<br>"; echo $teacher1->getTeacherID(); echo "<br>";

* Output

Output yang dihasilkan sebagai berikut

Student: Alifia  
Student ID: 2309  
Teacher name is: Manda  
Teacher ID is: 2890

1. Encapsulation

* Mengubah atribut nama dan studentID dalam kelas Student menjadi privat

Atribut atau properties adalah variabel yang menyimpan data untuk untuk objek.

Private properties merupakan properties yang hanya dapat digunakan oleh kelas itu sendiri

Untuk membuat protected properties, digunakanlah sintaks sebagai berikut:

private $studentID;

* Menambahkan metode setter dan getter untuk mengakses dan mengubah nilai atribut nama dan studentID

1. Metod setter

Metod ini digunakan untuk mengubah nilai.

Sintaks yang digunakan sebagai berikut:

public function setStudentID($studentID) { $this->studentID = $studentID; }

public function setName($nama) { $this->nama = $nama; }

1. Metod getter

Metod ini digunakan untuk mengembalikan nilai.

Sintaks yang digunakan sebagai berikut:

public function getStudentID() { return "Student ID: $this->studentID"; }

public function getName() { return "Student: $this->nama"; }

* Instansiasi dan pemanggilan metod

Sintaks yang digunakan sebagai berikut:

$student1 = new Student("Alifia", "2309");

echo $student1->getName(); echo "<br>";

echo $student1->getStudentID();

echo "<br>";

echo $student1->setName("Ayu");

echo $student1->getName();

echo "<br>";

echo $student1->setStudentID("1309");

echo $student1->getStudentID();

echo "<br>";

* Output

Output yang dihasilkan sebagai berikut:

Student: Alifia  
Student ID: 2309  
Student: Ayu  
Student ID: 1309

1. Abstraction

* Membuat kelas abstrak Course dengan metode abstrak getCourseDetails()

1. Membuat kelas abstrak Course

Kelas abstrak Course adalah kelas yang tidak dapat diinstansiasi secara langsung.

Kelas ini berfungsi sebagai template untuk kelas-kelas turunannya

1. Membuat metode abstrak getCourseDetails

* Membuat kelas OnlineCourse dan OfflineCourse yang mengimplementasikan getCourseDetails() untuk memberikan detail yang berbeda

1. Membuat kelas OnlineCourse
2. Implementasi metod getCourseDetails
3. Membuat kelas OfflineCourse
4. Implementasi metod getCourseDetails

* Instansiasi dan penggunaan metod
* Output

Output yang dihasilkan sebagai berikut:

Kursus Online: Privat Al-Faiz dilakukan pada hari Minggu  
Kursus Offline: Brain Academy dilakukan di tempat Cilacap