JOBSHEET 2

Implementasi Prinsip OOP dalam PHP

1. Membuat Class dan Object

* Membuat class Mahasiswa yang memiliki atribut nama, nim, dan jurusan.

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

Sintaks yang digunakan untuk membuat class Mahasiswa sebagai berikut:

Class Mahasiswa {}

Sintaks yang digunakan untuk membuat atribut nama, nim dan jurusan sebagai berikut:

public $nama; = menyimpan nama mahasiswa

public $nim; = menyimpan nim mahasiswa

public $jurusan; = menyimpan jurusan mahasiswa

Public atribut digunakan agar atribut tersebut dapat diakses dari kelas mana saja.

* Buat metode tampilknData() dalam class Mahasiswa

Metode ini digunakan untuk menampilkan data.

Sintaks yang digunakan dalam penulisan metode ini sebagai berikut:

    public function tampilkanData() {

        return "Nama = $this->nama NIM = $this->nim Jurusan = $this->jurusan.";

    }

$this-> nama digunakan untuk mengatur nilai dari properti objek.

* Instansiasi objek dari Class mahasiswa dan tampilkan data mahasiswa tersebut

1. Membuat objek baru

Membuat objek $nama1 dari kelas Mahasiswa dengan nama ”Hafidziyah Arrohmah”, NIM “230302082”, dan jurusan “Teknik Informatika”.

Sintaks yang digunakan dalam penulisan metode ini sebegai berikut:

$nama1 = new Mahasiswa();

$mahasiswa1->nama = "Hafidziyah Arrohmah";

$mahasiswa1->nim = "230302082";

$mahasiswa1->jurusan = "TI";

$mahasiswa1->nama digunakan untuk memberikan nilai dari properti objek.

1. Menampilkan data

Menampilkan data mahasiswa dengan menggunakan metode tampilkanData().

Sintaks yang digunakan dalam memanggil data ini sebegai berikut:

echo $mahasiswa1->tampilkanData();

* Output

Output yang dihasilkan sebagai berikut:

Nama = Hafidziyah Arrohmah NIM = 230302082 Jurusan = TI.

1. Encapsulation

* Mengubah atribut dalam class Mahasiswa menjadi private

Privat atribut artinya atribut hanya bisa diakses dari dalam kelas Mahasiswa.

Sintaks untuk membuat private atribut sebagai berikut:

private $nama;

private $nim;

private $jurusan;

* Membuat metode getter dan setter untuk atribut nama, nim, dan jurusan

1. Metode setNama() , setNim() , setJurusan() merupakan metode setter yang berfungsi untuk mengatur atau mengganti nilai dari atribut privat.

Sintaks yang digunakan adalah sebagai berikut:

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

public function setNim($nim) { $this->nim = $nim; }

public function setJurusan($jurusan) { $this->jurusan = $jurusan; }

$this->nama digunakan untuk mengatur nilai dari properti objek.

1. Metode getNama() , getNim() , getJurusan() merupakan metode getter yang berfungsi untuk mengembalikan nilai dari atribut privat.

Sintaks yang digunakan adalah sebagai berikut:

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

public function getNim() { return $this->nim; }

public function getJurusan() { return $this->jurusan; }

$this->nama digunakan untuk mengatur nilai yang akan dikembalikan dari properti objek.

* Demonstrasi akses ke atribut menggunakan metode getter dan setter

1. Menentukan nilai awal dari objek menggunakan bantuan konstruktor

Konstruktor akan dipanggil secara otomatis saat objek mahasiswa dibuat.

Konstruktor menerima 3 parameter yaitu $nama,$nim, dan $jurusan kemudian akan menginisialisasi atribut kelas dengan nilai-nilai tersebut.

Sintaks yang digunakan untuk membuat konstruktor sebagai berikut:

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

$this-> nama digunakan untuk mengatur nilai dari properti objek.

1. Membuat objek baru

Membuat objek $nama1 dari kelas Mahasiswa dengan nama ”Hafidziyah Arrohmah”, NIM “230302082”, dan jurusan “Teknik Informatika”.

Sintaks yang digunakan dalam penulisan metode ini sebegai berikut:

$mahasiswa1 = new Mahasiswa ("Hafidziyah Arrohmah", "230302082", "Teknik Informatika");

echo $mahasiswa1->tampilkanData();

1. Demonstrasi metode getter dan setter

Sintaks yang digunakan untuk demostrasi metode setter dan getter sebagai berikut:

$mahasiswa1->setNama("Arrohmah");

$mahasiswa1->setNim("230402082");

$mahasiswa1->setJurusan("Teknik Listrik");

echo $mahasiswa1->tampilkanData();

echo $mahasiswa1->getNama();

echo $mahasiswa1->getNim();

echo $mahasiswa1->getJurusan();

$mahasiswa1->setNama(“Arrohmah”) = untuk mengatur nilai objek

echo $mahasiswa1->tampilkanData() = untuk memanggil metode yang akan dilakukan

* Output

Output yang dihasilkan sebagai berikut:

Nama = Hafidziyah Arrohmah NIM = 230302082 Jurusan = Teknik Informatika.  
Nama = Arrohmah NIM = 230402082 Jurusan = Teknik Listrik.  
Arrohmah  
230402082  
Teknik Listrik

1. Inheritance

* Membuat class Pengguna dengan atribut nama dan metode getNama()

1. Membuat kelas Pengguna

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

Sintaks yang digunakan untuk membuat kelas sebagai berikut:

class Pengguna{}

1. Membuat protected atribut

Atribut yang dibuat bersifat protected yang artinya atribut ini hanya dapat diakses dalam kelas Pengguna dan kelas turunan dari pengguna.

Sintaks yang digunakan sebagai berikut:

protected $nama;

1. Membuat metode getNama()

Metode ini digunakan untuk mengembalikan nilai dari atribut $nama.

Sintaks yang digunakan sebagai berikut:

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

$this->nama berfungsi untuk mengatur nilai yang akan dikembalikan dari atribut $nama

* Membuat class Dosen yang mewarisi class Pengguna dan menambahkan atribut mataKuliah

1. Membuat kelas Dosen yang mewarisi kelas Pengguna

Sintaks yang digunakan sebagai berikut:

Class Dosen extends Pengguna {}

extends Pengguna menandakan bahwa kelas Dosen adalah cabang atau subclass dari kelas Pengguna, sehingga kelas Dosen mewarisi semua atribut dan metode dari kelas Pengguna.

1. Membuat atribut mataKuliah

Sintaks yang digunakan sebagai berikut:

private $mataKuliah;

atribut mataKuliah bersifat private yang artinya atribut tersebut hanya dapat diakses oleh kelas Dosen.

* Instansiasi objek dari class dosen dan tampilkan data dosen

1. Membuat konstruktor pada kelas Pengguna

Sintaks yang digunakan sebagai berikut:

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

Konstruktor akan dipanggil saat objek Pengguna dibuat.

Konstruktor akan menerima atribut $nama untuk menginisialisasi atribut tersebut.

1. Membuat konstruktor pada kelas Dosen

Sintaks yang digunakan sebagai berikut:

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

Konstruktor kelas Dosen menerima 2 parameter atribut yaitu $nama dan $mataKuliah.

Konstruktor ini memanggil konstruktor dari kelas induk menggunakan sintaks parent::\_\_construct dan menginisialisasi atribut mataKuliah

1. Membuat objek baru

Membuat objek $dosen1 dari kelas Dosen dengan nama ”Annisa”, dan mata kuliah “Bahasa Indonesia”.

Sintaks yang digunakan dalam penulisan metode ini sebegai berikut:

$dosen1 = new Dosen("Annisa", "Bahasa Indonesia");

1. Menampilkan data

Menampilkan data yang diperbarui dengan memanggil metode tampilkanDosen().

Sintaks yang digunakan dalam penulisan metode ini sebegai berikut:

echo $dosen1->tampilkanDosen();

1. Menggunakan metode getNama

Sintaks yang digunakan sebagai berikut:

echo $dosen1->getNama();

$dosen1->getNama() digunakan untuk memanggil metode

1. Output

Nama Dosen: Annisa Nama Mata Kuliah: Bahasa Indonesia  
Annisa

1. Polimorphism

* Membuat kelas Pengguna dengan metode aksesFitur()
* Menerapkan aksesFitur dengan cara berbeda di kelas Dosen dan Mahasiswa
* Melakukan instansiasi objek dari kelas Dosen dan Mahasiswa lalu panggil metode aksesFitur()