

**LAPORAN
UTS PEMOGRAMAN WEB 2**



**Oleh:
Ade Hikmat Pauji Ridwan
22552011130**

**Dosen Pengampu:
RUDHI WAHYUDI FEBRIANTO**

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI TEKNOLOGI BANDUNG**

2024

Pendahuluan: Laporan ini memberikan gambaran umum tentang aplikasi web yang dikembangkan sebagai bagian dari kursus pemrograman. Aplikasi ini menunjukkan penggunaan prinsip Pemrograman Berorientasi Objek (OOP) dalam PHP, yang dikemas dalam antarmuka web yang ramah pengguna yang menyoroti kode sumber dan fungsionalitas berbagai kelas dan antarmuka PHP.

- [Kelas Abstrak Product](#)
- [Interface IElectronic](#)
- [PHP Class: Electronic](#)
- [PHP Class: Television](#)
- [Implementation](#)
- [Result](#)

Kelas Abstrak Product

Di sini kita mendefinisikan `Product` sebagai kelas abstrak. Kelas abstrak ini tidak dapat diinstansiasi secara langsung dan bertujuan untuk menyediakan kerangka dasar bagi kelas-kelas turunan. Kelas ini mendefinisikan properti dasar seperti nama, merek, dan stok, serta metode abstrak `getInfo()` yang harus diimplementasikan oleh setiap kelas turunan.

```
<?php
abstract class Product {
    // access modifier
    public $nama;
    public $merek;
    protected $stock;

    // encapsulation
    abstract public function getInfo();
}
```

Interface IElectronic

Interface `IElectronic` mendefinisikan kontrak yang harus diikuti oleh kelas-kelas yang mengimplementasikannya. Interface ini memastikan bahwa semua kelas elektronik memiliki metode untuk mengatur merek, nama, dan stok. Hal ini memudahkan pengelolaan atribut elektronik secara konsisten di berbagai jenis kelas elektronik.

```
<?php
// interfaces
interface IElectronic {
    public function setMerek($merek);
    public function setName($nama);
    public function setStock($stock);
}
```

PHP Class: Electronic

Kelas `Electronic` adalah implementasi konkret dari kelas abstrak `Product` dan interface `IElectronic`. Kelas ini menyediakan implementasi spesifik untuk metode yang didefinisikan dalam interface, mengatur bagaimana atribut seperti merek, nama, dan stok dihandle. Kelas ini juga memperkenalkan metodologi yang spesifik untuk produk elektronik.

```
<?php
require_once 'IElectronic.php';
require_once 'Product.php';
class Electronic extends Product implements IElectronic {
    public function __construct(string $nama, string $merek, int $stock) {
        $this->setName($nama);
        $this->setMerek($merek);
        $this->setStock($stock);
    }

    public function setMerek($merek){
        $this->merek = $merek;
    }
    public function setName($nama){
        $this->nama = "Product Electronic ". $nama;
    }
    public function setStock($stock){
        $this->stock = $stock;
    }

    // overriding func
    public function getInfo() {
        return "nama product ". $this->nama." merek product ".$this->merek;
    }
}
```

PHP Class: Television

Kelas `Television` memperluas kelas `Electronic` dengan menambahkan atribut dan fungsionalitas spesifik untuk televisi, seperti tipe televisi. Ini adalah contoh dari spesialisasi lebih lanjut dalam hierarki kelas, dimana kelas `Television` menyesuaikan dan memperluas fungsi dari kelas dasarnya untuk memenuhi kebutuhan yang lebih spesifik.

```
<?php
require_once 'Electronic.php';
class Television extends Electronic {
    private $type;
    public function __construct(string $nama, string $merek, int $stock, string $type) {
        parent::__construct($nama, $merek, $stock);
        $this->type = $type;
    }

    public function __destruct() {
        echo "Destructor Menghancurkan " . $this->nama . ". televisi tipe " . $this->type . " .";
    }

    //overriding
    public function getInfo() {
        return parent::getInfo()." Type ". $this->type;
    }
}
```

Implementasi

```
<?php
require_once 'Television.php';

function main() {
    // with inheritance
    $television = new Television("TV LG 123", "LG", 1000, "LCD");
    echo $television->Info();
}
```

Result

```
nama product Product Electronic TV LG 123 merek product LG Type LCD Destructor Menghancurkan Product Electronic TV LG 123, televisi tipe LCD.
```

Struktur Aplikasi Web: Aplikasi web terdiri dari beberapa komponen kunci yang menggambarkan konsep OOP seperti abstraksi, enkapsulasi, warisan, dan polimorfisme. Komponen-komponen ini terintegrasi ke dalam antarmuka web yang memungkinkan navigasi dan interaksi yang mudah.

1. Navigasi Sidebar:

- Menyediakan tautan ke berbagai bagian aplikasi, masing-masing didedikasikan untuk kelas atau antarmuka tertentu.
- Meningkatkan pengalaman pengguna dengan memungkinkan akses cepat ke informasi.

2. Penyorotan Kode:

- Menggunakan Highlight.js untuk meningkatkan keterbacaan kode PHP yang ditampilkan di halaman.
- Penyorotan sintaks membuatnya lebih mudah untuk membedakan antara elemen-elemen kode yang berbeda, seperti kata kunci, variabel, dan string.

3. Ikhtisar Bagian:

- **Kelas Abstrak Produk:**
 - Membahas kelas abstrak **Produk** yang berfungsi sebagai cetak biru untuk kelas turunan.
 - Mendefinisikan properti dasar seperti nama, merek, dan stok, serta metode abstrak **getInfo()**.
- **Antarmuka IElectronic:**
 - Mendefinisikan kontrak (**IElectronic**) yang harus diimplementasikan oleh semua kelas elektronik.
 - Memastikan penanganan atribut elektronik yang konsisten di berbagai kelas.
- **Kelas PHP: Electronic:**
 - Implementasi konkret dari kelas **Produk** dan antarmuka **IElectronic**.
 - Mengelola atribut seperti merek, nama, dan stok yang spesifik untuk produk elektronik.
- **Kelas PHP: Television:**
 - Memperluas kelas **Electronic**, menambahkan atribut dan fungsionalitas spesifik untuk produk televisi, seperti tipe.
 - Contoh spesialisasi lebih lanjut dalam hierarki kelas.
- **Implementasi & Hasil:**
 - Menampilkan implementasi file **main.php** dan hasilnya.
 - **main.php** mencakup fungsi **main()** yang dipanggil untuk menampilkan hasil.

Kesimpulan: Aplikasi ini secara efektif menunjukkan bagaimana Pemrograman Berorientasi Objek dapat diterapkan dalam PHP untuk mengelola kompleksitas dalam desain perangkat lunak. Ini memberikan contoh praktis tentang bagaimana kelas abstrak, antarmuka, dan warisan dapat digunakan untuk membuat basis kode yang terstruktur dan modular.

Rekomendasi untuk Masa Depan:

- Integrasi elemen interaktif lebih lanjut untuk memanipulasi properti objek langsung melalui antarmuka web.
- Perluasan hierarki kelas untuk mencakup lebih banyak perangkat elektronik dan fitur.