Mohammad Rizky Arfiyanto
105220033

1. Penggunaan PHP Object-Oriented Programming (OOP) menjadi sangat penting dalam pengembangan aplikasi web modern karena memberikan struktur kode yang lebih terorganisir dan mudah dipahami. Dengan memisahkan fungsionalitas ke dalam objek-objek, OOP meningkatkan keterbacaan dan pemeliharaan kode. Pengembangan aplikasi yang penuh dapat digunakan kembali OOP memungkinkan pengembangan aplikasi yang lebih dapat digunakan kembali dengan menggunakan polimorfisme dan pewarisan. Klasifikasi Kelas dan objek OOP menggunakan konsep kelas dan objek, yang memungkinkan mengelola data dan fungsi yang terkait dengannya.

   Sumber :
   - https://www.phptutorial.net/php-oop/

2. Konsep dasar Pemrograman Berorientasi Objek (OOP) merupakan pendekatan pemrograman yang menekankan pada objek dan kelas yang terdiri dari mereka. Berikut adalah beberapa konsep dasar OOP, yaitu ;

   Kelas dan objek, OOP menggunakan konsep kelas dan objek, di mana kelas adalah bentuk dasar atau cetak biru yang mendefinisikan variabel, metode, dan sifat yang akan dimiliki oleh objek yang dihasilkan dari kelas tersebut. Kelas dan objek memungkinkan mengelola data dan fungsi yang terkait dengannya. Pewarisan adalah konsep di mana kelas ikut (anak) mewarisi sifat dan metode dari kelas kakep (orang tua).

   Pewarisan memungkinkan untuk menciptakan aplikasi yang lebih dapat digunakan kembali dan dinamis.

   Polimorfisme adalah kemampuan objek untuk menangani objek yang berbeda dengan satu sama dalam OOP, objek dapat menangani objek lain melalui polimorfisme.

   Enkapsulasi adalah kemampuan untuk memasukkan data dan perilaku yang terkait dengannya dalam satu unit, seperti kelas. Enkapsulasi membantu mengelola kompleksitas dan membuat kode lebih mudah dikelola.

   Metode pembuatan Dalam OOP, metode adalah fungsi yang terkait dengan kelas dan objek. Metode mengatur bagaimana objek menangani operasi dan menyimpan data.

   Implementasi PHP dalam konsep dasaf OOP melalui fitur seperti pewarisan, polimorfisme, dan enkapsulasi. Dalam versi 5.0, PHP mendukung pemrograman berbasis objek, memungkinkan pengembang untuk menggunakan fitur OOP dalam pengembangan aplikasi web

   Sumber :
   - https://www.simplilearn.com/tutorials/php-tutorial/oops-in-php
   - https://www.phptutorial.net/php-oop/
   - https://www.w3schools.com/php/php_oop_what_is.asp

3. Menggunakan paradigma Object-Oriented Programming (OOP) dalam pengembangan aplikasi PHP menawarkan sejumlah keuntungan dibandingkan dengan pendekatan procedural. Pertama, OOP memungkinkan pengorganisasian kode yang lebih terstruktur dengan menggunakan konsep objek, kelas, dan pewarisan. Ini membantu dalam menciptakan hierarki yang jelas antara berbagai entitas dalam aplikasi, memudahkan pemeliharaan dan pengembangan lebih lanjut. Selain itu, konsep encapsulation dalam OOP memungkinkan penyembunyian detail implementasi dan pengelompokan fungsi-fungsi terkait bersama-sama, meningkatkan keamanan dan pemahaman kode. Polimorfisme dalam OOP juga mempermudah adaptasi dan perluasan kode tanpa merusak fungsionalitas yang sudah ada.

   Sumber :
   - Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Prentice Hall.

4. Konsep pewarisan (inheritance) dalam PHP OOP adalah ketika suatu kelas merwarisi sifat dan metode dari kelas lain. Hal ini penting dalam pengembangan berorientasi objek karena memungkinkan Anda untuk menciptakan kelas baru berdasarkan kelas yang ada, dengan menggunakan sifat dan metode yang ada pada kelas asli.

   Sumber :
   - https://www.w3schools.com/php/php_oop_inheritance.asp

5. Dalam paradigma Object-Oriented Programming (OOP) PHP, kelas dan objek memiliki peran kunci dalam membentuk struktur aplikasi. Kelas adalah blueprint atau cetak biru yang mendefinisikan atribut dan metode yang akan dimiliki oleh objek. Sebuah objek, di sisi lain, adalah instansi konkret dari sebuah kelas yang memiliki propertinya sendiri dan dapat melakukan operasi sesuai dengan metodenya. Penerapan kelas dan objek memungkinkan pengelompokan data dan perilaku terkait ke dalam satu entitas yang koheren, meningkatkan modularitas dan memudahkan pemeliharaan kode.

   Sumber :
   - Welling, L., & Thomson, L. (2016). PHP and MySQL Web Development (5th ed.). Addison-Wesley.

6. Perbedaan nya sebagai berikut ;
   - Kelas (class): Kelas adalah blueprint atau cetakan yang mendefinisikan variabel dan fungsi yang umum pada semua objek. Kelas digunakan untuk membuat kerangka dasar dan menggambarkan kumpulan objek yang sama.

   - Objek (object): Obyek adalah kumpulan variabel dan fungsi yang dihasilkan dari template khusus atau disebut Obyek adalah elemen pada saat run-time yang akan diciptakan, dimanipulasi, dan dibuang/di-destroy ketika eksekusi. Objek

merupakan instansi dari kelas yang dihasilkan dengan menjalankan konstruktor kelas.

Keduanya saling berinteraksi melalui konsep polymorphism, di mana objek dapat mengakses dan menggunakan metode yang ada di dalam kelasnya. Contoh penggunaan kelas dan objek dalam PHP OOP:

- Buat kelas Person yang memiliki property seperti nama, usia, dan alamat.
- Buat objek person1 dari kelas Person.
- Mengakses property dan metode dari objek person1 untuk mengambil atau mengubah data pengguna.

Sumber :

- https://www.duniailkom.com/tutorial-belajar-oop-php-pengertian-class-object-property-dan-method/

7. Enkapsulasi dalam PHP OOP dapat diimplementasikan menggunakan pengubah akses publik, terproteksi, dan privat. Pengubah ini mengontrol visibilitas properti dan metode dalam suatu kelas. Dengan tahapan sebagai berikut:
    - Public : Properti dan metode yang dideklarasikan sebagai public dapat diakses dari luar kelas.
    - Protected : Properti dan metode yang dinyatakan dilindungi hanya dapat diakses di dalam kelas itu sendiri dan oleh kelas mana pun yang mewarisinya.
    - Private: Properti dan metode yang dideklarasikan sebagai private hanya dapat diakses di dalam kelas itu sendiri. Berikut ini contoh bagaimana enkapsulasi dapat diimplementasikan di PHP OOP:

```
class User {
        private $username;
        protected $email;

        public function setUsername($username) {
                $this->username = $username;
}

public function getUsername() {
        return $this->username;
}

protected function setEmail($email) {
        $this->email = $email;
}

protected function getEmail() {
        return $this->email;
 }
}
```

Dari contoh di atas, properti dinyatakan sebagai pribadi, sehingga hanya dapat diakses di dalam kelas. Properti tersebut dinyatakan dilindungi, sehingga dapat diakses di dalam kelas dan kelas mana pun yang mewarisinya.

Sumber :

https://www.malasngoding.com/php-oop-part-4-pengertian-enkapsulasi-public-private-protected/

8. Polimorfisme dalam PHP OOP adalah sebuah konsep yang memungkinkan objek dari kelas yang berbeda merespons secara berbeda berdasarkan pesan yang sama. Hal ini memungkinkan kelas yang berbeda untuk mengimplementasikan nama metode yang sama dengan fungsionalitas berbeda, yang meningkatkan kegunaan kembali dan fleksibilitas kode. Polimorfisme dapat diimplementasikan dalam PHP menggunakan kelas atau antarmuka abstrak. Untuk mengimplementasikan polimorfisme di PHP, Anda dapat menggunakan kelas abstrak atau antarmuka. Kelas abstrak digunakan untuk mendefinisikan cetak biru metode yang harus diimplementasikan oleh subkelas, sedangkan antarmuka digunakan untuk mendefinisikan sekumpulan metode yang harus diimplementasikan oleh kelas.

Sumber :

- https://www.phptutorial.net/php-oop/php-polymorphism/

9. Abstraksi adalah konsep kunci dalam OOP karena memungkinkan pengembang untuk membuat representasi sederhana dari permasalahan dunia nyata yang kompleks. Abstraksi melibatkan identifikasi fitur-fitur penting dari suatu objek dan mengabaikan fitur-fitur yang tidak penting. Hal ini membantu mengurangi kompleksitas dan meningkatkan efisiensi kode. Abstraksi juga membantu meningkatkan keterbacaan kode dengan menyembunyikan detail implementasi suatu kelas dan hanya memperlihatkan informasi yang diperlukan kepada user.

Dengan menggunakan abstraksi, pengembang dapat membuat antarmuka yang disederhanakan dan terstandarisasi untuk sistem yang kompleks, sehingga membuat kode lebih mudah dipahami dan dipelihara. Abstraksi juga membantu mengurangi duplikasi kode dan meningkatkan penggunaan kembali kode, sehingga menghasilkan kode yang lebih efisien dan modular. Secara keseluruhan, abstraksi adalah konsep kunci dalam OOP yang membantu menyederhanakan sistem yang kompleks, meningkatkan efisiensi, dan meningkatkan keterbacaan kode.

Sumber :

- https://parsinta.com/articles/pemrograman-berorientasi-objek-di-php-hzcxd

10. Antarmuka dalam PHP OOP adalah mekanisme yang memungkinkan untuk menyebutkan setiap kelas yang menerima kontrak keseluruhan yang ditentukan sebelumnya. Antarmuka berkaitan dengan abstraksi, yang merupakan pengalaman utama dalam pemrograman berorientasi objek. Dengan menggunakan antarmuka, Anda dapat mendefinisikan kontrak keseluruhan yang harus dipatuhi oleh kelas yang menerima

antarmuka tersebut, sehingga Anda dapat mengelola ketergantungan dan polimorfisme dengan lebih efisien dan kontrol. Peran antarmuka dalam PHP OOP meliputi:

- Menyediakan mekanisme untuk mengelola kemandirian dan polimorfisme.
- mengisyaratkan bahwa semua kelas yang menerima antarmuka tersebut mendukung kontrak yang ditentukan.
- Meningkatkan efisiensi dan keterbacaan kode dengan mengeliminasi detail implementasi ke dalam kelas-kelas yang sesuai.

Sumber :

- Rais, Muh. 2019. Penerapan Konsep Object Oriented Programming Untuk Aplikasi Pembuat Surat. Jurnal PROTEK.