

VERSION 1.3
20 OCTOBER 2024



[PEMROGRAMAN WEB]

Modul 3 - PHP STRUKTURAL DAN OOP (BACKEND)

DISUSUN OLEH:
OGYA ADYATMA PUTRA
GERALDI NATHAN TOMMY SAPUTRA

DIAUDIT OLEH:
AMINUDIN, S.KOM., M.CS.

LAB. INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

[PEMROGRAMAN WEBSITE]

PERSIAPAN MATERI

<https://www.w3schools.com/php/default.asp>

https://www.w3schools.com/php/php_oop_what_is.asp

TUJUAN

Mahasiswa mampu memahami konsep dasar bahasa pemrograman PHP secara Struktural, OOP dan Algoritma Pemrograman.

TARGET MODUL

Mahasiswa dapat memahami dasar bahasa pemrograman PHP secara struktural, OOp dan Algoritma Pemrograman.

PERSIAPAN SOFTWARE/APLIKASI

Hardware dan Infrastruktur

- Laptop/PC
- Koneksi Internet

Software

- Text Editor Visual Studio Code (Recommended)

Extension (VS Code)

- PHP Intelephense
- Prettier - Code Formatter | Prettier
- Chrome Extension: JSON Viewer

Programming Language

- PHP Versi 8.1 atau lebih

NOTES

Silahkan mencari tutorial penginstalan PHP pada device yang kalian miliki seperti Linux, Windows ataupun MacOS. Pada windows dan linux dapat menggunakan web server seperti **XAMPP** atau **LARAGON**, dan pada MacOS dapat menginstall melalui **Homebrew** atau **XAMPP**.

MATERI

PHP STRUKTURAL

PHP adalah bahasa scripting server-side yang digunakan untuk mengembangkan situs web statis, situs web dinamis atau aplikasi web. PHP singkatan dari Hypertext Preprocessor yang sebelumnya dikenal sebagai Personal Home Pages. Di era industri sekarang, PHP lebih banyak digunakan

untuk pembuatan layanan web backend, seperti pembuatan API dan berbagai aplikasi backend lainnya.

A. Fungsi PHP dalam Pemrograman Website

PHP adalah bahasa pemrograman sisi server yang paling populer dan banyak digunakan dalam pengembangan web. PHP memiliki berbagai fungsi dalam pembuatan web, diantaranya:

1. Menangani request pengguna: PHP digunakan untuk mengolah permintaan yang dikirim oleh pengguna dari browser ke server. Hal ini memungkinkan untuk membuat situs web interaktif dan responsif. Interaksi bisa melalui API atau melalui HTML jika monolith.
2. Interaksi dengan basis data: PHP memungkinkan pengembang web untuk berinteraksi dengan basis data, seperti MySQL, PostgreSQL dan Oracle untuk menyimpan dan mengambil informasi dari server.
3. Membuat dan mengelola sesi: PHP dapat digunakan untuk mengelola sesi pengguna, sehingga dapat melacak status login, pengaturan preferensi dan kegiatan pengguna lainnya.
4. PHP dapat digunakan untuk mengatur dan mengelola cookies, yang memungkinkan situs web untuk menyimpan informasi di sisi klien.
5. Integrasi dengan layanan pihak ketiga: PHP dapat digunakan untuk berinteraksi dengan API dan layanan pihak ketiga seperti layanan pembayaran, media sosial atau pengiriman pesan.

Masih banyak fungsi bahasa pemrograman PHP pada konteks pemrograman website. Dalam berbagai kasus dan beberapa praktek pemrograman website dengan arsitektur monolith, PHP bisa digabungkan dengan HTML untuk memberi logic agar website kita bisa berinteraksi ke server.

Bahasa pemrograman PHP banyak digunakan untuk membuat aplikasi web service backend yang berupa API (Application Programming Interface) yang merupakan aplikasi web service yang digunakan untuk berinteraksi ke server melalui API tersebut.

```
1 <?php
2 // Set header untuk memberitahu respons adalah JSON
3 header('Content-Type: application/json');
4
5 // Fungsi untuk membuat respons JSON dengan pesan "Hello World"
6 function helloWorld()
7 {
8     $response = array('message' => 'Hello World');
9     return json_encode($response);
10 }
11
12 // Memanggil fungsi helloWorld dan akan menampilkan respons message
13 echo helloWorld();
14
```

Pada gambar diatas terdapat sebuah source code yang ditulis pada sebuah file bernama "helloworld.php", dimana maksud dari code tersebut adalah untuk membuat sebuah API dengan format JSON yang akan mengembalikan nilai "Hello World"

Mari kita coba menjalankan file tersebut dengan php server melalui terminal kita dengan cara mengetikkan syntax "php -S <HOST>:<PORT> <DIRECTORY KE FILE PHP>" pada kasus kali ini seperti berikut "php -S localhost:8888 helloworld.php".

```
PS C:\Kuliah\Code Modul Pemrograman Website\Modul2\Website\Modul3> php -S localhost:8888 helloworld.php
[Fri Aug 2 05:24:31 2024] PHP 8.2.4 Development Server (http://localhost:8888) started
[Fri Aug 2 05:24:34 2024] [::1]:61271 Accepted
[Fri Aug 2 05:24:34 2024] [::1]:61272 Accepted
[Fri Aug 2 05:24:34 2024] [::1]:61271 Closing
[Fri Aug 2 05:24:34 2024] [::1]:61272 Closing
```

Kalian bisa membuka localhost tersebut untuk melihat hasilnya seperti berikut.

```
1 // 20240802052434
2 // http://localhost:8888/
3
4 {
5   "message": "Hello World"
6 }
```



B. SYNTAX PHP

Pengertian Syntax PHP adalah aturan penulisan pada bahasa pemrograman yang benar agar dapat dimengerti oleh compiler. Dalam penulisan PHP diawali dengan "<?php" dan diakhiri dengan "?>". Di dalam file PHP juga dapat berisi tag seperti HTML dan skrip sisi klien seperti halnya pada bahasa pemrograman javascript.

Untuk menjalankan file PHP, kalian bisa gunakan command line pada terminal text editor kalian dan masuk file directory php yang ingin dijalankan, contohnya seperti "phpPublic/index.php" dimana dalam contoh tersebut kalian akan menjalankan file index.php yang berada di dalam folder phpPublic.

1. Echo



2. Variable

Beberapa aturan variabel pada bahasa pemrograman PHP

- Penulisan variabel dimulai dengan simbol "\$".
- Nama variabel harus dimulai dengan huruf atau garis bawah.
- Nama variabel tidak bisa dimulai dengan angka.



```
1 <?php
2
3 $var = 150;
4
5 echo $var;
```

3. Tipe Data

a. String



```
1 <?php
2
3 $varString = "Pemrograman Website";
4
5 echo $var;
```

b. Integer

Beberapa aturan dalam penulisan tipe data integer:

- Integer harus mempunyai paling tidak 1 digit angka.
- Integer tidak bisa berisi koma atau kosong
- Integer memiliki 3 format yaitu decimal, heksadesimal dan oktal.

```
1 <?php
2
3 $i = 404;
4 echo "Menampilkan Angka Positif = $i";
5 echo "\n";
6
7 $i = -404;
8 echo "Menampilkan Angka Negatif = $i";
9 echo "\n";
10
11 $i = 0xD1;
12 echo "Menampilkan Angka Hexadecimal = $i";
13 echo "\n";
14
15 $i = 0324;
16 echo "Menampilkan Angka Octal dari 034 = $i";
17 echo "\n";
```

4. Operator

```
<?php
$angka1 = 10;
$angka2 = 5;

// Operator tambah (+)
$tambah = $angka1 + $angka2;
echo "Hasil penjumlahan $angka1 + $angka2 = $tambah \n";

// Operator kurang (-)
$kurang = $angka1 - $angka2;
echo "Hasil pengurangan $angka1 - $angka2 = $kurang \n";

// Operator kali (*)
$kali = $angka1 * $angka2;
echo "Hasil perkalian $angka1 * $angka2 = $kali \n";

// Operator bagi (/)
$bagi = $angka1 / $angka2;
echo "Hasil pembagian $angka1 / $angka2 = $bagi \n";

// Operator modulus (%)
$modulus = $angka1 % $angka2;
echo "Hasil modulus $angka1 % $angka2 = $modulus \n";
```

5. Conditional Statement (IF ELSE)

```
1  <?php
2
3  $i = 3;
4
5  if ($i == 0) {
6      echo "Angka 0";
7      echo "\n";
8  } else if ($i % 2 == 0) {
9      echo "Angka Genap";
10     echo "\n";
11 } else {
12     echo "Angka Ganjil";
13     echo "\n";
14 }
15
```

6. Perulangan

```
1  <?php
2
3  echo "Perulangan : \n";
4  for ($i = 5; $i >= 1; $i--) {
5      echo $i . "\n";
6  }
```

7. Fungsi Parameter


```
1  <?php
2
3  function kali($a, $b)
4  {
5      echo $a * $b;
6  }
7
8  kali(5, 5);
```

8. Array

```
1  <?php
2
3  $mobil = ["BMW", "VW", "HONDA"];
4  echo "Mobil Pertama: $mobil[0] \n";
5
6  echo "Daftar Mobil \n";
7  for ($i = 0; $i < count($mobil); $i++) {
8      echo "$mobil[$i] \n";
9  }
10
```

PHP OOP (Object Oriented Programming)

OOP (Object Oriented Programming) merupakan paradigma penulisan kode yang memecah alur program menjadi berbagai modul sederhana yang disebut objek. Setiap objek akan memiliki fungsi dan tugas khusus, sehingga memudahkan dalam pembuatan aplikasi.

A. CLASS

Dalam OOP, class digunakan untuk membuat kerangka kerja. Class berisi property dan method, yang berfungsi sebagai wadah yang menyimpan keduanya. Objek yang merupakan hasil instansiasi dari kelas akan

digunakan dalam aplikasi..Penulisan class diawali dengan keyword *class*, kemudian diikuti dengan nama dari *class* tersebut.

A screenshot of a code editor with a dark background and three colored window control buttons (red, yellow, green) at the top left. The code is written in PHP and defines a class named 'mobil'.

```
1  <?php
2
3  class mobil
4  {
5      // code
6  }
```

B. PROPERTY / ATTRIBUTE

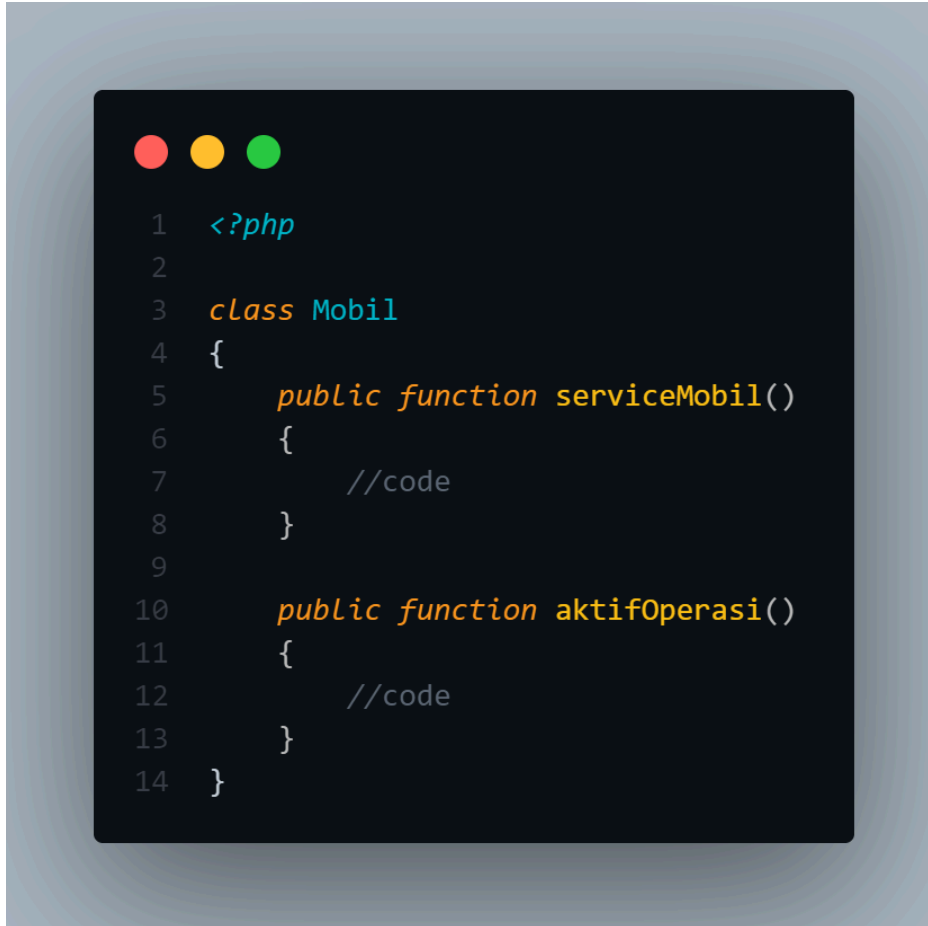
Property adalah variabel yang terdapat di dalam class. Property berfungsi untuk menyimpan data dan lain-lain. Aturan dan tipe data yang biasa diterapkan ke dalam variabel bisa juga diterapkan ke dalam property.

A screenshot of a code editor with a dark background and three colored window control buttons (red, yellow, green) at the top left. The code is written in PHP and defines a class named 'Mobil' with two properties: '\$tipe' and '\$harga'.

```
1  <?php
2
3  class Mobil
4  {
5      var $tipe;
6      var $harga;
7  }
8
```

C. METHOD

Method adalah sebuah aksi yang terdapat di dalam class. Method pada dasarnya adalah function yang berada di dalam class. Seluruh fungsi dan sifat function bisa diterapkan ke dalam method, seperti argumen atau parameter, mengembalikan nilai dan lain-lain.

A screenshot of a code editor with a dark background and light-colored text. The code is written in PHP and defines a class named 'Mobil'. The class has two public methods: 'serviceMobil()' and 'aktifOperasi()'. The code is numbered from 1 to 14. The first two lines are the PHP opening tag and a blank line. Line 3 starts the class definition. Line 4 opens the class curly brace. Line 5 defines the 'serviceMobil()' method. Line 6 opens the method curly brace. Line 7 contains a comment '//code'. Line 8 closes the method curly brace. Line 9 is a blank line. Line 10 defines the 'aktifOperasi()' method. Line 11 opens the method curly brace. Line 12 contains a comment '//code'. Line 13 closes the method curly brace. Line 14 closes the class curly brace. There are three colored circles (red, yellow, green) in the top left corner of the code editor window.

```
1  <?php
2
3  class Mobil
4  {
5      public function serviceMobil()
6      {
7          //code
8      }
9
10     public function aktifOperasi()
11     {
12         //code
13     }
14 }
```

D. OBJEK

Object adalah hasil cetak atau output dari class. Proses menampilkan hasil cetak atau output objek dari class ini disebut dengan instansiasi. Penulisan object diawali dengan syntax new dan kemudian diikuti dengan nama class yang ingin di instansiasi. Hasil cetakan akan disimpan kedalam variabel untuk melanjutkan proses program.



```
1  <?php
2
3  class RentalMobil
4  {
5      public function serviceMobil()
6      {
7          //code
8      }
9
10     public function aktifOperasi()
11     {
12         //code
13     }
14 }
15
16 $tipeMobil = new RentalMobil();
17
```

E. ABSTRACT CLASS DAN ABSTRACT METHOD

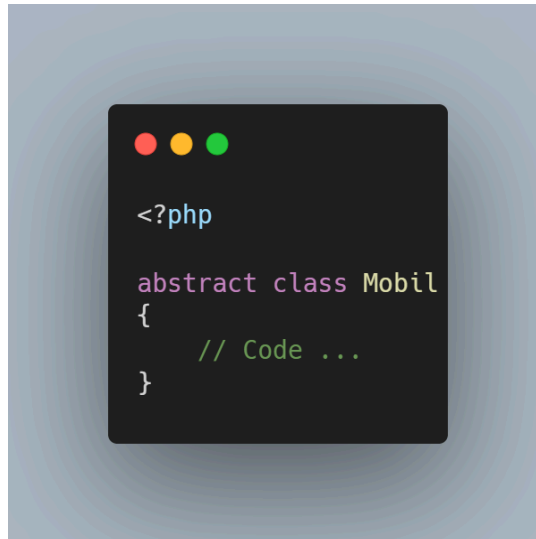
Abstract Class adalah sebuah class yang tidak dapat di instansikan. Abstract Class tidak dapat dilakukan pembuatan object pada umumnya untuk memanggil method maupun property. Abstract Class berperan menggunakan konsep pewarisan, sehingga Abstract Class dijadikan sebagai parent class dan akan diwarisi pada child class.

Abstract Method adalah sebuah method atau metode yang dideklarasikan berupa nama method dan parameter tetapi tidak diimplementasikan dalam kode tersebut. Abstract method harus diimplementasikan di dalam Child Class.

- **MEMBUAT ABSTRACT CLASS**

Pembuatan abstract class harus berlawanan dengan keyword abstract sebelum class. Sebuah abstract class bisa memiliki property dan method seperti biasa selayaknya class normal tetapi juga bisa memiliki abstract method.

Contoh abstract class:



- **MEMBUAT ABSTRACT METHOD**

Jika sebuah method dinyatakan sebagai abstract method, maka kita tidak perlu membuat isi methodnya, tetapi hanya signature dari method tersebut. Jika sebuah method dinyatakan sebagai abstract method, isi dari method tersebut akan dibuat dalam class turunan. Abstract method harus berada di dalam abstract class.

Contoh pembuatan dan pengimplementasian abstract method pada PHP:



```
<?php

abstract class Mobil
{
    abstract public function mobilBerkendara();
}
```

F. OBJEK INTERFACE

Object Interface adalah sebuah abstrak kelas yang sama sekali tidak memiliki implementasi dimana nantinya membuat perjanjian dengan class yang mengimplementasikan interface tersebut. Interface tidak seperti abstract dimana interface tidak boleh memiliki suatu properti dengan visibility apapun. Artinya interface hanya membuat method saja. Namun dapat memiliki konstanta. Bagi class yang menggunakan object interface, class tersebut harus mengimplementasikan seluruh method yang ada di dalam interface.

Untuk membuat interface di dalam PHP, penulisannya sama seperti membuat class, hanya saja dengan menggunakan keyword *"interface"*, seperti contoh berikut.

```
1  <?php
2
3  interface kuliner
4  {
5      //code
6  }
7
```

G. INHERITANCE

Inheritance atau pewarisan adalah konsep pemrograman dimana sebuah class dapat menurunkan atau mewarisi property dan method yang dimilikinya kepada class lain. Konsep ini digunakan untuk memanfaatkan fitur code reuse untuk menghindari duplikasi kode program.

Di dalam PHP, inheritance dari sebuah class pada class lain menggunakan penulisan "Extends", seperti contoh berikut

```
1  <?php
2
3  class Hewan
4  {
5      var $jenis;
6      var $gender;
7
8      function berjalan()
9      {
10         echo "Kucing itu ";
11     }
12 }
13
14 class Kucing extends Hewan
15 {
16     function tampilData()
17     {
18         echo "berjenis " . $this->jenis . " " . $this->gender . ", ";
19     }
20
21     function tidur()
22     {
23         echo "kucing itu juga sering tidur \n";
24     }
25 }
26
27 $kucing = new Kucing();
28
29 $kucing->jenis = "Anggora";
30 $kucing->gender = "Betina";
31 $kucing->berjalan();
32 $kucing->tampilData();
33 $kucing->tidur();
34
```

Kode PHP di atas menunjukkan konsep **inheritance (pewarisan)** dalam OOP, di mana kelas **Kucing** mewarisi properti dan method dari kelas **Hewan**. Kelas **Hewan** memiliki properti **\$jenis** dan **\$gender**, serta method **berjalan()**, sementara kelas **Kucing** menambahkan method **tampilData()** dan **tidur()**. Dengan menggunakan objek **\$kucing**, kita dapat menetapkan nilai jenis dan gender kucing, lalu memanggil method untuk menampilkan

informasi dan perilaku kucing, seperti berjalan dan tidur. Pewarisan ini memungkinkan penggunaan kembali kode dari kelas induk sambil menambahkan fungsionalitas khusus.

H. PHP TRAIT

Trait adalah suatu mekanisme dimana suatu class diizinkan untuk menggunakan kembali kode program yang berasal dari hirarki yang berbeda. Trait dapat memiliki method dan abstract method yang dapat digunakan di beberapa kelas, dan method tersebut dapat memiliki access modifier apapun.

Membuat Trait

A screenshot of a code editor with a dark background and light-colored text. The code is PHP and defines a trait named 'Mengendarai'. It includes a public function 'berkendara()' which returns the string 'Berkendara menggunakan mobil'. The code is numbered from 1 to 9 on the left side of the editor.

```
1  <?php
2
3  trait Mengendarai
4  {
5      public function berkendara()
6      {
7          return "Berkendara menggunakan mobil";
8      }
9  }
```

Pada kode di atas, terdapat trait **Mengendarai**. Trait **Mengendarai** mendefinisikan sebuah method bernama **berkendara()** yang mengembalikan string **"Berkendara menggunakan mobil"**. Pembuatan trait sama seperti pembuatan class, tetapi diganti dengan "trait".

Menggunakan Trait

```
1  <?php
2
3  trait Mengendarai
4  {
5      public function berkendara()
6      {
7          return "Berkendara menggunakan mobil";
8      }
9  }
10
11  trait Tujuan
12  {
13      public function destinasi()
14      {
15          return "untuk berlibur menuju bali";
16      }
17  }
```

Menggunakan Multiple Trait

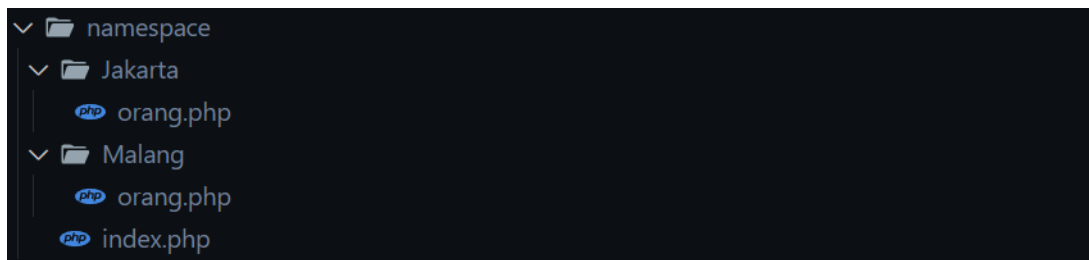
```
1  <?php
2
3  trait Mengendarai
4  {
5      public function berkendara()
6      {
7          return "Berkendara menggunakan mobil";
8      }
9  }
10
11 trait Tujuan
12 {
13     public function destinasi()
14     {
15         return "untuk berlibur menuju bali";
16     }
17 }
18
19 class Liburan
20 {
21     use Mengendarai, Tujuan;
22 }
23
24 $aktivitas = new Liburan();
25 echo $aktivitas->berkendara() . " ";
26 echo $aktivitas->destinasi();
27
```

Kode di atas menunjukkan penggunaan **multiple traits** dalam PHP, di mana dua trait, **Mengendarai** dan **Tujuan**, digunakan di dalam kelas **Liburan** dengan kata kunci **use**. Trait **Mengendarai** memiliki method **berkendara()** yang mengembalikan teks "Berkendara menggunakan mobil", sedangkan trait **Tujuan** memiliki method **destinasi()** yang mengembalikan teks "untuk berlibur menuju bali". Dengan menggabungkan kedua trait ini dalam kelas **Liburan**, objek **\$aktivitas** dari kelas tersebut dapat memanggil kedua method dari trait yang berbeda, sehingga memungkinkan **Liburan** memiliki perilaku dari kedua trait tanpa perlu mewarisi kelas lain. Ini mendemonstrasikan cara menggunakan multiple traits untuk

menggabungkan beberapa fungsionalitas ke dalam satu kelas, tanpa terikat oleh pewarisan tunggal.

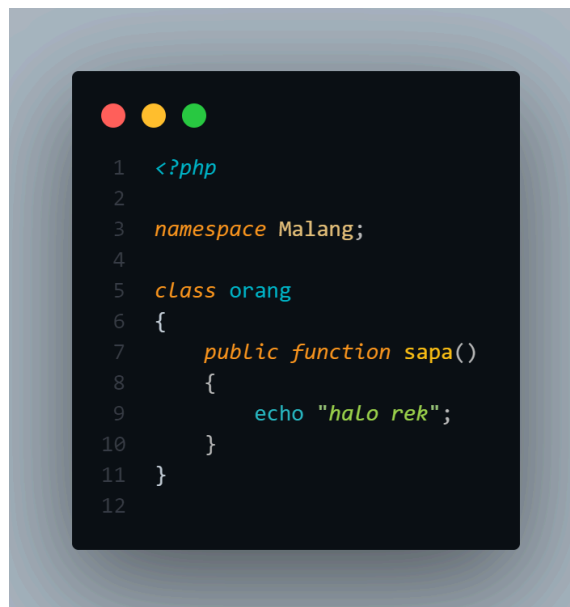
I. PHP NAMESPACE

Kegunaan namespace dapat mengorganisir kode dengan lebih rapi dan terstruktur, sehingga lebih mudah di maintenance. Cara kerja namespace mirip dengan cara kerja folder dan file. Dalam folder tidak bisa membuat 2 buah file dengan nama yang sama, tetapi hal tersebut bisa dilakukan jika foldernya berbeda.

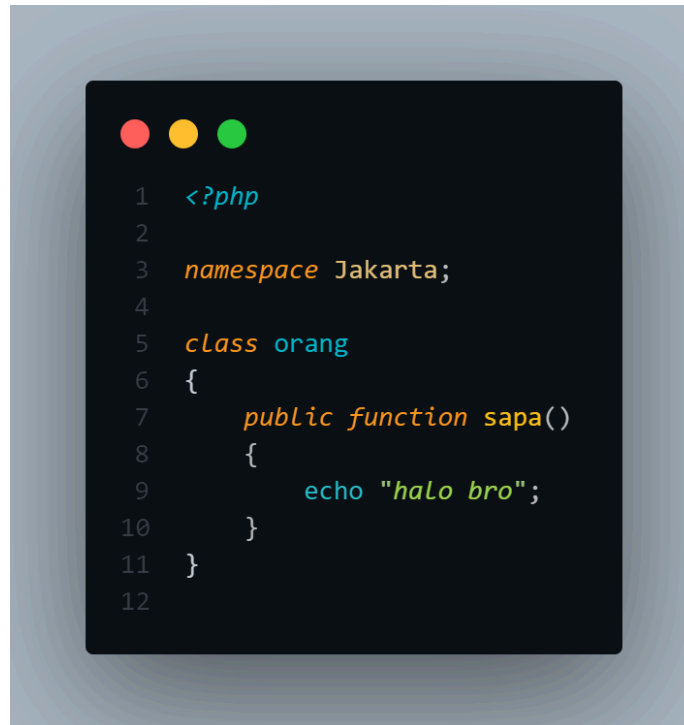


Dalam folder namespace terdapat 2 folder dan 1 file index.php. Detail code pada masing-masing file bisa kalian lihat dibawah ini.

File Malang/orang.php



File Jakarta/orang.php

A screenshot of a code editor with a dark background and light-colored text. The code is written in PHP and defines a namespace and a class. The code is as follows:

```
1  <?php
2
3  namespace Jakarta;
4
5  class orang
6  {
7      public function sapa()
8      {
9          echo "halo bro";
10     }
11 }
12
```

Pada kode di atas, **namespace** dideklarasikan menggunakan kata kunci **namespace**, diikuti oleh nama namespace, yaitu **Malang**. Namespace ini digunakan untuk mengelompokkan kelas, fungsi, atau konstanta agar memiliki ruang lingkup yang terpisah dari kode lainnya. Dalam hal ini, kelas **orang** berada di dalam namespace **Malang**, sehingga jika ada kelas dengan nama yang sama di namespace lain, tidak akan terjadi konflik. Namespace sangat penting dalam proyek besar karena membantu menghindari name collision).

File index.php



```

1  <?php
2
3  include "MaLang/orang.php";
4  include "JaKarta/orang.php";
5
6  $iqbal = new \MaLang\orang();
7  $iqbal->sapa();
8  echo "\n";
9
10 $yovi = new \JaKarta\orang();
11 $yovi->sapa();
12

```

Kode di atas menggunakan dua file eksternal yang masing-masing berisi kelas dengan nama yang sama, yaitu `orang`, tetapi berada di dalam namespace yang berbeda: `MaLang` dan `JaKarta`. Dengan menggunakan perintah `include`, file `MaLang/orang.php` dan `JaKarta/orang.php` dimasukkan ke dalam kode utama. Kelas `orang` dari namespace `MaLang` diinstansiasi dengan objek `$iqbal`, dan method `sapa()` dipanggil untuk menampilkan pesan dari kelas `orang` yang berada di namespace `MaLang`. Begitu juga, kelas `orang` dari namespace `JaKarta` diinstansiasi dengan objek `$yovi`, dan method `sapa()` dipanggil untuk menampilkan pesan dari kelas `orang` di namespace `JaKarta`. Penggunaan namespace dengan `\MaLang\orang` dan `\JaKarta\orang` memungkinkan kedua kelas `orang` dengan nama yang sama digunakan tanpa konflik, karena masing-masing berada dalam ruang lingkup yang berbeda.

J. MAGIC METHOD

Dalam pemrograman class dengan PHP kita akan menjumpai method yang diawali dengan double underscore. Method ini akan digunakan untuk memberikan respons pada suatu event atau kejadian tertentu.

Kalian bisa melihat dokumentasi lengkap pada <https://www.php.net/manual/en/language.oop5.magic.php> untuk mempelajari lebih lanjut mengenai magic method.

1. __construct()

Method ini akan diakses pada saat pembuatan project

2. __destruct()

Method ini akan diakses saat object dihapus

3. __call()

Method ini akan diakses saat object menjalankan method yang invisible atau tidak terdefiniskan

4. __callStatic()

Method ini akan diakses jika kita menjalankan suatu static method yang invisible atau tidak didefinisikan.

5. __get()

Method ini akan diakses jika program ingin mengambil nilai dari properti yang invisible.

6. __set()

Method ini akan diakses jika program ingin memberikan nilai dari suatu property.

7. __isset()

Method ini akan diakses jika program menjalankan isset() atau empty() pada property yang tidak bisa diakses.

8. __unset()

Method ini akan diakses jika program menjalankan unset() pada property yang tidak bisa diakses

9. __sleep()

Berhubungan dengan serialisasi object.

10. __wakeup

Berhubungan dengan unserialisasi object.

11. __toString()

Method ini akan diakses jika kita memperlakukan class sebagai string.

12. __invoke()

Method ini akan dijalankan ketika program memperlakukan object sebagai function.

13. __set_state()

Method yang dipakai saat mengeksport class dengan function `var_export()` yaitu function yang mengeksport properti dalam method ke dalam array.

14.__clone()

Method ini berhubungan dengan cloning object. Saat selesai proses cloning object, method ini akan dijalankan.

15.__debugInfo()

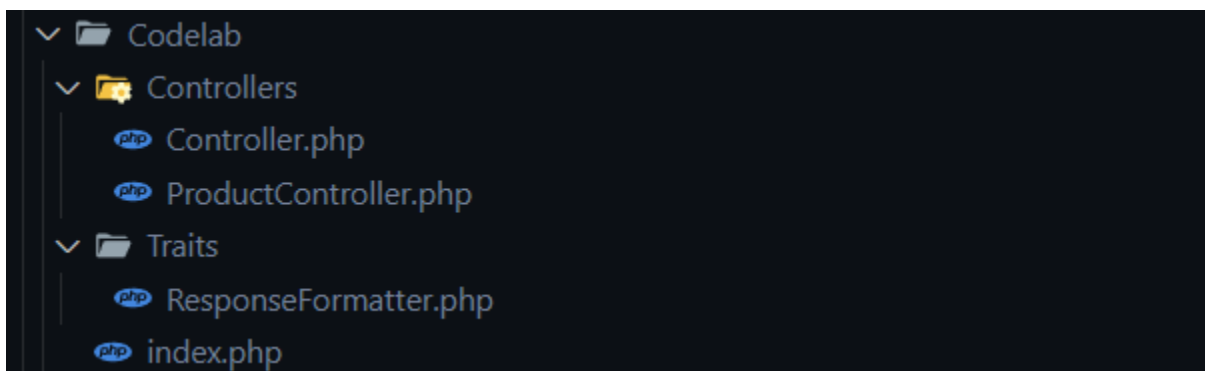
Method ini diakses ketika program menjalankan `var_dump()` pada suatu object

CODELAB

BUATLAH PERULANGAN MENGGUNAKAN PHP DENGAN OUTPUT SEPERTI DIBAWAH INI

1. SEGITIGA SAMA SISI**2. SEGITIGA SAMA SISI TERBALIK****PEMBUATAN REST API DENGAN FORMAT JSON**

Buatlah folder dan file seperti dibawah ini



File Controller/Controller.php



```
<?php

namespace Controller;

class Controller
{
    // Variable Attributes
    var $controllerName;
    var $controllerMethod;

    // Method untuk mengambil semua atribut
    public function getControllerAttribute()
    {
        return [
            "ControllerName" => $this->controllerName,
            "Method" => $this->controllerMethod,
        ];
    }
}
```

File Traits/ResponseFormatter.php

```
<?php

namespace Traits;

// Disimulasikan ini train untuk formatter json response
trait ResponseFormatter{
    public function responseFormatter($code, $message, $data = null){
        return json_encode([
            "code" => $code,
            "message" => $message,
            "data" => $data
        ]);
    }
}
```

File Controller/ProductController.php

```
<?php

namespace Controller;

include "Traits/ResponseFormatter.php";
include "Controllers/Controller.php";

use Traits\ResponseFormatter;

class ProductController extends Controller{
    use ResponseFormatter;

    public function __construct(){
        $this->controllerName = "Get All Product";
        $this->controllerMethod = "GET";
    }

    public function getAllProduct(){
        // Array Dummy Data
        $dummyData = [
            "Air Mineral",
            "Kebab",
            "Spaghetti",
            "Jus Jambu"
        ];

        $response = [
            "controller_attribute" => $this->getControllerAttribute(),
            "product" => $dummyData
        ];

        return $this->responseFormatter(200, "Success", $response);
    }
};
```

File Index.php



Setelah mengikuti langkah-langkah diatas, buka terminal pada directory project dan jalankan command berikut :

`"php -S localhost:8888 index.php"`

Setelah itu buka "localhost:8888" untuk melihat hasilnya.

```
1 // 20241009174780
2 // http://localhost:8888/
3
4 {
5   "code": 200,
6   "message": "Success",
7   "data": {
8     "controller_attribute": {
9       "controllerName": "Get All Product",
10      "Method": "GET"
11    },
12    "product": [
13      "Air Mineral",
14      "Kebab",
15      "Spaghettil",
16      "Jus Jambu"
17    ]
18  }
19 }
```



TUGAS

TUGAS 1

Buatlah sebuah program menggunakan bahasa PHP sesuai dengan kreativitas kalian dengan syarat seperti berikut.

- Menerapkan Paradigma OOP (Object Oriented Programming)
- Gunakan Traits, Method, Class, Inheritance, Namespace, Abstract Class dan Abstract Method.
- Gunakan Magic Method jika diperlukan (Optional)
- Class minimal mempunyai 2 property/attribute.

Dilarang menggunakan kode yang sama dengan **CODELAB**.

TUGAS 2

Buatlah sebuah program yang didalamnya memiliki fungsi yang menerapkan sebuah perulangan dan dan percabangan dengan ketentuan seperti berikut:

- Fungsi menerima sebuah bilangan bulat positif n sebagai input.
- Fungsi harus mencetak setiap bilangan bulat dari 1 hingga n secara berurutan, satu nilai per baris.
- Jika sebuah bilangan habis dibagi 4 dan 6, cetak "**Pemrograman Website 2024**".
- Jika sebuah bilangan habis dibagi 5, cetak "**2024**".
- Jika sebuah bilangan hanya habis dibagi 4 tetapi tidak habis dibagi 6, cetak "**Pemrograman**".
- Jika sebuah bilangan hanya habis dibagi 6 tetapi tidak habis dibagi 4, cetak "**Website**".
- Jika bilangan tidak memenuhi salah satu kondisi di atas, cetak bilangan itu sendiri.

```

1
2
3
Pemrograman
2024
Website
7
Pemrograman
9
2024
11
Pemrograman Website 2024
13
14
2024
Pemrograman
17
Website
19
2024
21
22
23
Pemrograman Website 2024
2024

```

KRITERIA & DETAIL PENILAIAN

Kriteria	Persentase Penilaian
Dapat menerapkan OOP dengan bahasa pemrograman	30%
Dapat mengimplementasikan algoritma pemrograman dengan PHP	30%
Pemahaman	40%