

VERSION 4.1

AGUSTUS, 4, 2023



PEMROGRAMAN WEB

PHP STRUKTURAL DAN OOP (BACKEND) – MODUL 4

DISUSUN OLEH:

M. SYAUQI AMIQ AMRULLAH
ALIF FATWA RAMADHANI

DIAUDIT OLEH:

AMINUDIN, S.KOM., M.CS.

LAB. INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

PEMROGRAMAN WEB

PERSIAPAN MATERI

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

TUJUAN

Mahasiswa mampu memahami konsep dasar bahasa pemrograman PHP secara Struktural, OOP dan Algoritma Pemrograman, Modul ini telah disesuaikan dengan beberapa poin penilaian Materi Uji Kompetensi.

PERSIAPAN SOFTWARE/APLIKASI

Hardware dan Infrastruktur

- Laptop/PC
- Koneksi Internet

Software

- Text Editor: Visual Studio Code (Recomended)
- Extension VSCode: PHP Intelephense (Code Formatter agar Rapih)
- PHP Versi 8.1 atau lebih
- Chrome Extension: JSON Viewer

Settings:

- Default Formatter pada VSCode di setting dengan extension PHP Intelephense

NOTES

Silahkan mencari tutorial mengenai cara penginstalan PHP pada device yang kalian miliki seperti Linux, Windows ataupun MacOS. Pada windows dan linux dapat menggunakan web server seperti XAMPP yang sudah terdapat PHP didalamnya, dan pada MacOS dapat menginstall melalui Homebrew ataupun XAMPP.

Cara diatas adalah salah satu cara penginstalan dari banyak cara menginstall PHP, silahkan di eksplorasi secara mandiri untuk penginstalan PHP pada device kalian.

XAMPP:

<https://www.apachefriends.org/download.html>

Homebrew (MacOS Recommended):

<https://formulae.brew.sh/formula/php>

PASTIKAN PILIH VERSI PHP 8.1 ATAU LEBIH

MATERI

PHP STRUKTURAL

PHP adalah bahasa scripting server-side, Bahasa pemrograman yang digunakan untuk mengembangkan situs web statis atau situs web dinamis atau aplikasi Web. PHP singkatan dari Hypertext Pre-processor, yang sebelumnya disebut Personal Home Pages. Namun di era industri saat ini bahasa pemrograman PHP lebih banyak digunakan untuk pembuatan aplikasi web service backend seperti pembuatan API dan lain – lainnya.

A. FUNGSI PHP DALAM PEMROGRAMAN WEB

PHP (Hypertext Preprocessor) adalah bahasa pemrograman sisi server yang paling populer dan banyak digunakan dalam pengembangan web. PHP memiliki berbagai fungsi dalam pembuatan web, di antaranya:

1. Menangani *request* pengguna: PHP digunakan untuk mengolah permintaan yang dikirim oleh pengguna dari browser ke server. Hal ini memungkinkan untuk membuat situs web yang 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, atau 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

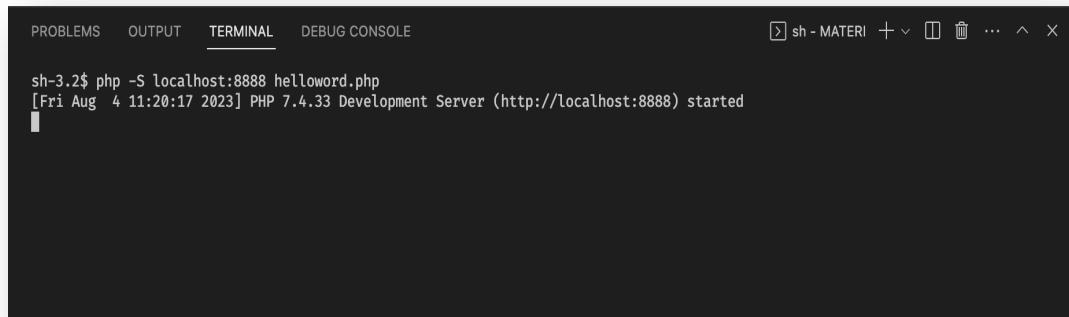
Dan masih banyak fungsi bahasa pemrograman PHP pada konteks pemrograman website. Dalam berbagai kasus dan beberapa praktik pemrograman website dengan arsitektur monolith, PHP bisa di gabungkan dengan HTML untuk memberi logic agar website kita bisa berinteraksi ke server.

Pada era industri saat ini PHP banyak digunakan untuk membuat aplikasi webservice backend yang berupa API (Application Programming Interface), singkatnya aplikasi webservice ini digunakan untuk berinteraksi dari Frontend ke Server melalui API tersebut. Mari kita coba untuk membuat API secara simple yang akan mengembalikan nilai "Hello World" dan berformat JSON.

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

Pada gambar, terdapat sebuah *source code* yang di tulis 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”.

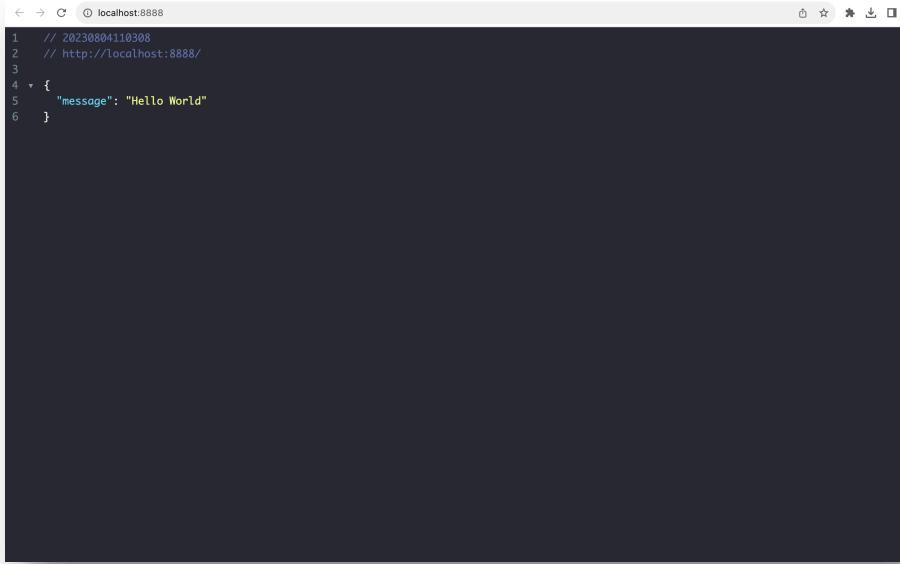


The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
sh-3.2$ php -S localhost:8888 helloworld.php
[Fri Aug 4 11:20:17 2023] PHP 7.4.33 Development Server (http://localhost:8888) started
```

The terminal tabs at the top are PROBLEMS, OUTPUT, TERMINAL (which is selected), and DEBUG CONSOLE. The status bar at the bottom right shows the path sh - MATERI and various icons.

Setelah php server berjalan mari kita coba buka URL Development Server tersebut pada browser kita. Maka hasilnya seperti berikut:



Pada gambar diatas, adalah contoh melalui aplikasi browser Chrome dengan Extension “JSON Viewer”. Pada gambar tersebut bisa lihat bahwa aplikasi kita berjalan dengan format JSON yang mempunyai key “message” dengan value “Hello World”.

B. SYNTAX PHP

Pengertian Syntax PHP adalah aturan penulisan agar mampu dimengerti dengan benar oleh compiler saat membaca bahasa pemrograman. Dalam penulisan PHP yang benar diawali dengan “<?php” dan diakhiri dengan “?>”. Dan di dalam File PHP juga dapat berisi tag seperti HTML dan skrip sisi klien seperti JavaScript.

Untuk menjalankan file php gunakan command line pada terminal berikut “php <Directory ke file php>” contohnya “php public/index.php”, maka akan menjalankan file php pada folder public/index.php.

1. Echo



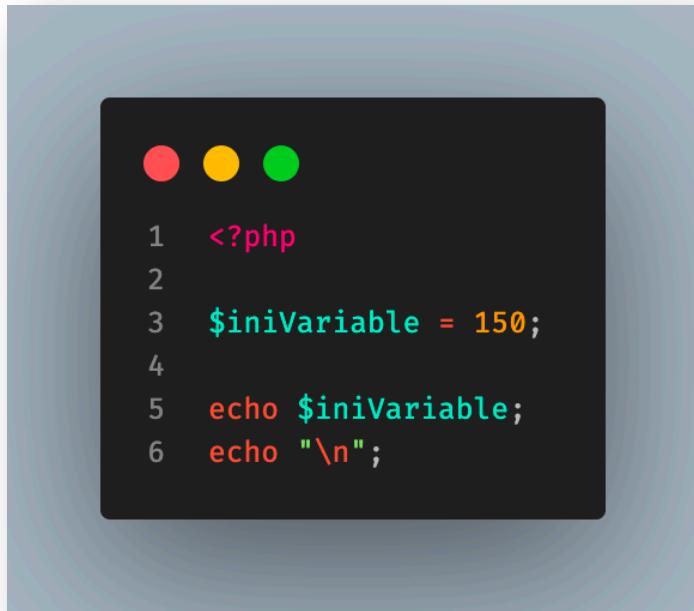
Setelah dijalankan:

```
sh-3.2$ php echo.php
Hellow World
sh-3.2$ []
```

2. Variable

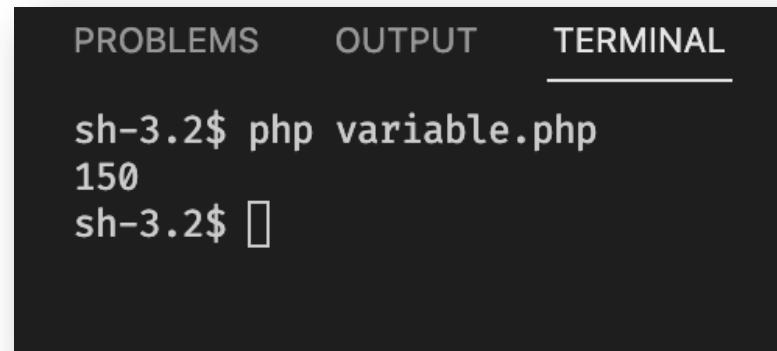
Terdapat aturan variable pada bahasa pemrograman PHP sebagai berikut:

- Variable di mulai dengan symbol “\$” kemudian nama variabelnya.
- Nama variable harus dimulai dengan huruf atau garis bawah.
- Nama variable tidak bisa dimulai dengan angka.
- Nama variable tidak bisa dimulai dengan angka.
- Nama variable tidak bisa dimulai dengan angka.



```
1 <?php
2
3 $iniVariable = 150;
4
5 echo $iniVariable;
6 echo "\n";
```

Setelah dijalankan:



PROBLEMS	OUTPUT	<u>TERMINAL</u>
sh-3.2\$ php variable.php 150 sh-3.2\$		

3. Tipe Data

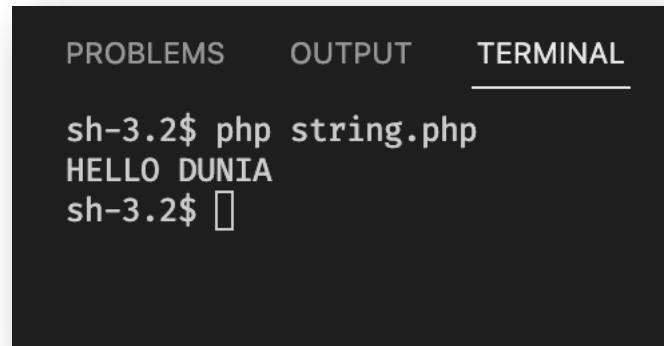
- **String**



A screenshot of a code editor window. At the top left are three colored circular icons: red, yellow, and green. Below them is a dark gray code block containing the following PHP code:

```
1 <?php
2
3 $iniVariable = "HELLO DUNIA";
4
5 echo $iniVariable;
6 echo "\n";
```

Setelah dijalankan:



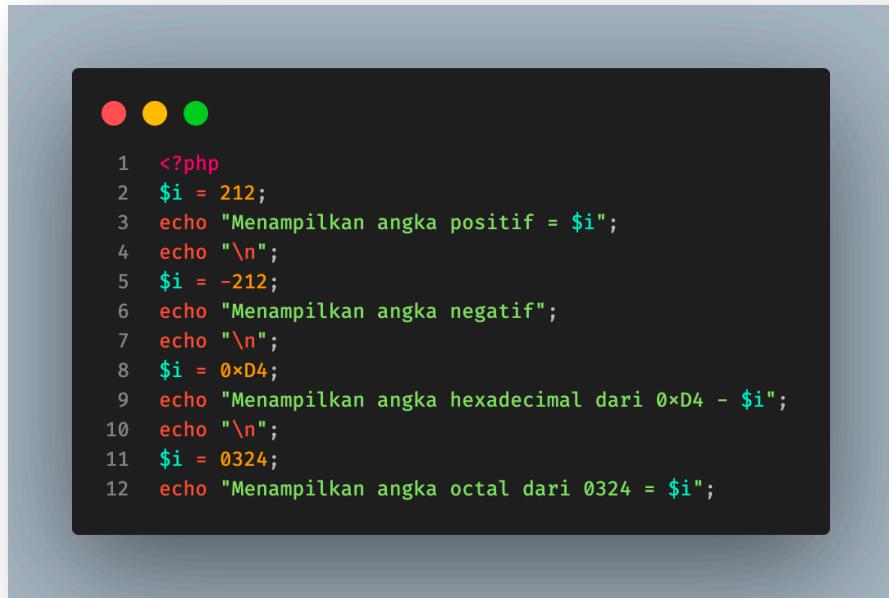
A screenshot of a terminal window. The tabs at the top are labeled PROBLEMS, OUTPUT, and TERMINAL. The TERMINAL tab is active, showing the following output:

```
sh-3.2$ php string.php
HELLO DUNIA
sh-3.2$
```

▪ **Integer**

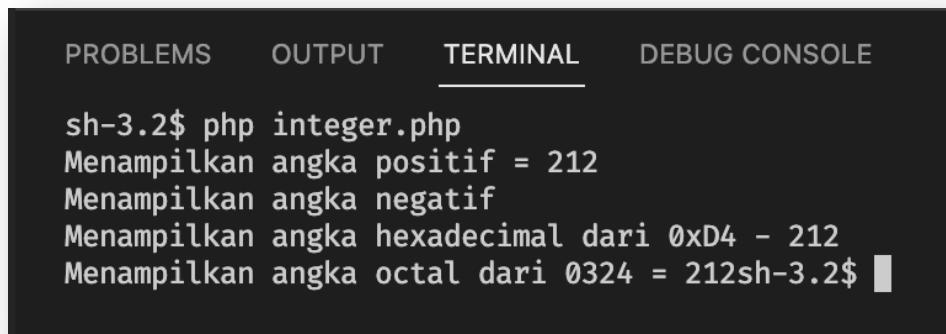
Terdapat Aturan Integer sebagai berikut:

- a. Integer harus mempunyai paling tidak 1 digit (0-9).
- b. Integer tidak bisa berisi koma atau kosong.
- c. Integer tidak berupa angka decimal.
- d. Integer bisa berupa bilangan positif atau negative.
- e. Integer memiliki 3 format yaitu decimal, heksadesimal, dan octal.



```
1 <?php
2 $i = 212;
3 echo "Menampilkan angka positif = $i";
4 echo "\n";
5 $i = -212;
6 echo "Menampilkan angka negatif";
7 echo "\n";
8 $i = 0xD4;
9 echo "Menampilkan angka hexadecimal dari 0xD4 - $i";
10 echo "\n";
11 $i = 0324;
12 echo "Menampilkan angka octal dari 0324 = $i";
```

Setelah dijalankan:



PROBLEMS	OUTPUT	TERMINAL	DEBUG CONSOLE
	sh-3.2\$ php integer.php Menampilkan angka positif = 212 Menampilkan angka negatif Menampilkan angka hexadecimal dari 0xD4 - 212 Menampilkan angka octal dari 0324 = 212sh-3.2\$ █		

4. Operator

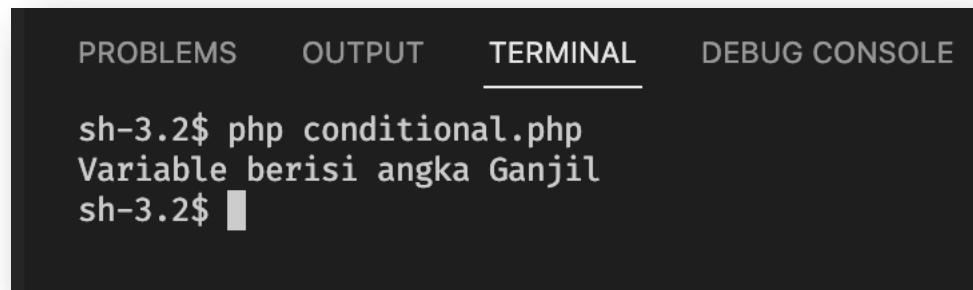
Operator	Contoh
+	\$a + \$b
-	\$a - \$b
*	\$a * \$b
/	\$a / \$b
%	\$a % \$b

5. Conditional Statement (IF ELSE)

```
● ● ●

1 <?php
2 $i = 3;
3 if ($i == 0) {
4     echo "Variable berisi angka 0";
5     echo "\n";
6 } else if ($i % 2 == 0) {
7     echo "Variable berisi angka genap";
8     echo "\n";
9 } else {
10    echo "Variable berisi angka Ganjil";
11    echo "\n";
12 }
```

Setelah dijalankan:



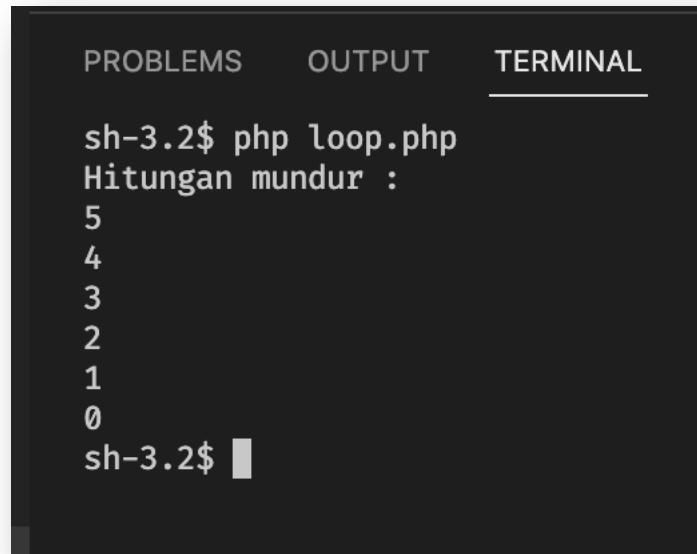
The screenshot shows a terminal window with tabs: PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. The TERMINAL tab is selected, displaying the following text:
sh-3.2\$ php conditional.php
Variable berisi angka Ganjil
sh-3.2\$ █

6. Perulangan



The screenshot shows a code editor with three colored circular icons at the top: red, yellow, and green. Below them is a PHP script:
1 <?php
2 echo "Hitungan mundur : \n";
3 for (\$i = 5; \$i ≥ 0; \$i--) {
4 echo \$i . "\n";
5 }
6

Setelah dijalankan:



PROBLEMS OUTPUT TERMINAL

```
sh-3.2$ php loop.php
Hitungan mundur :
5
4
3
2
1
0
sh-3.2$
```

7. Fungsi dengan parameter



```
1 <?php
2 function penjumlahan($a, $b)
3 {
4     echo $a + $b;
5     echo "\n";
6 }
7
8 function pangkatDua($a)
9 {
10    echo $a * $a;
11    echo "\n";
12 }
13
14 penjumlahan(5, 2);
15 pangkatDua(5);
```

Setelah dijalankan:



PROBLEMS OUTPUT TERMINAL

```
sh-3.2$ php function.php
7
25
sh-3.2$ []
```

8. Array



```
1 <?php
2
3 $buah = array("Mangga", "Apel", "Jeruk");
4 echo "Daftar buah: $buah[0], $buah[1], $buah[2] \n";
5
6 $mobil = ["BMW", "TOYOTA", "HONDA"];
7 echo "Daftar Mobil: \n";
8 for ($i = 0; $i < count($mobil); $i++) {
9     echo "$mobil[$i] \n";
10 }
11
```

Setelah dijalankan:

```
PROBLEMS      OUTPUT      TERMINAL      DE  
sh-3.2$ php array.php  
Daftar buah: Mangga, Apel, Jeruk  
Daftar Mobil:  
BMW  
TOYOTA  
HONDA  
sh-3.2$ █
```

PHP OOP (OBJECT ORIENTED PROGRAMMING)

OOP (Object Oriented Programming) merupakan suatu paradigma penulisan kode dengan memecah alur program menjadi modul-modul sederhana yang disebut dengan objek. Setiap objek akan memiliki fungsi dan tugas tersendiri. Sehingga akan lebih sangat memudahkan kita di dalam membuat aplikasi.

A. CLASS

Class di dalam OOP di gunakan untuk membuat sebuah kerangka kerja. Class berisi property dan method. Dapat diibaratkan bahwa class adalah sebuah wadah yang menyimpan property dan method. Hasil cetakan dari class, yakni object yang akan kita gunakan nantinya. Penulisan class diawali dengan keyword class, kemudian diikuti dengan nama dari class.

Contoh pembuatan Class pada PHP:

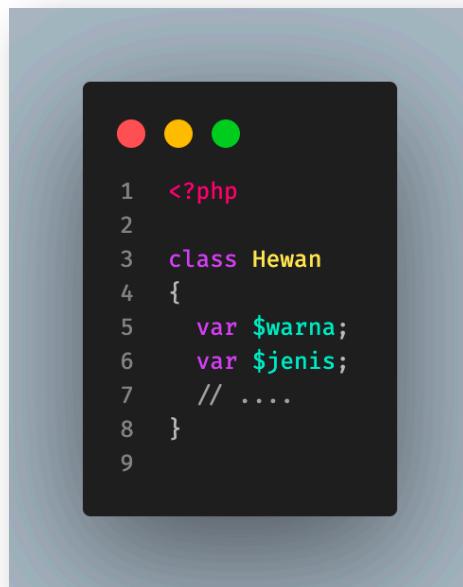


```
1 <?php
2
3 class Hewan
4 {
5     // your code ...
6 }
7
```

B. PROPERTY / ATTRIBUTE

Property merupakan variabel yang terletak di dalam class. Property berfungsi untuk meyimpan data dan lain-lain. Seluruh aturan dan tipe data yang biasa diinput ke dalam variabel, bisa juga diinput kedalam property.

Contoh pembuatan Property / Attribute:



```
1 <?php
2
3 class Hewan
4 {
5     var $warna;
6     var $jenis;
7     // ....
8 }
9
```

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/parameter, mengembalikan nilai, dan lain-lain. Penulisan method dimulai dengan keyword function lalu di ikuti oleh nama methodnya.

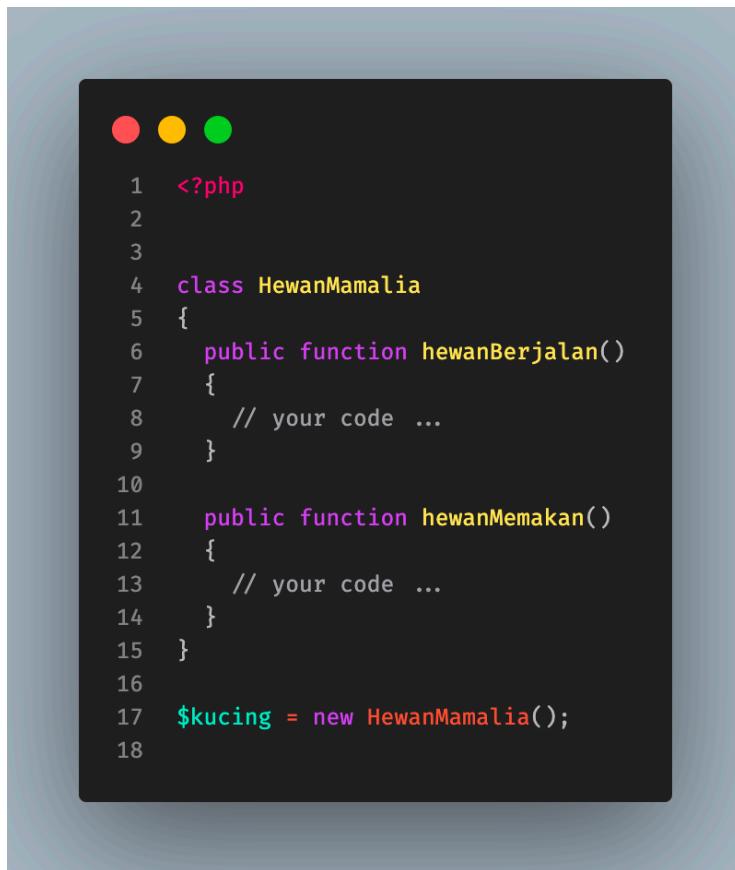
Contoh pembuatan method:

```
1 <?php
2
3 class Hewan
4 {
5     public function hewanBerjalan()
6     {
7         // your code ...
8     }
9
10    public function hewanMemakan()
11    {
12        // your code ...
13    }
14 }
15
```

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 di ikuti dengan nama class yang ingin di instansiasi. Hasil cetakan class akan disimpan dalam variabel untuk selanjutnya digunakan dalam proses program.

Contoh pembuatan objek dari sebuah Class:



```
1 <?php
2
3
4 class HewanMamalia
5 {
6     public function hewanBerjalan()
7     {
8         // your code ...
9     }
10
11    public function hewanMemakan()
12    {
13        // your code ...
14    }
15 }
16
17 $kucing = new HewanMamalia();
18
```

Dari contoh dapat dilihat bahwa variable “kucing” adalah objek dari class HewanMamalia. Objek tersebut memiliki seluruh property dan method yang telah dirancang dari class HewanMamalia.

E. ABSTRACT CLASS DAN ABSTRACT METHOD

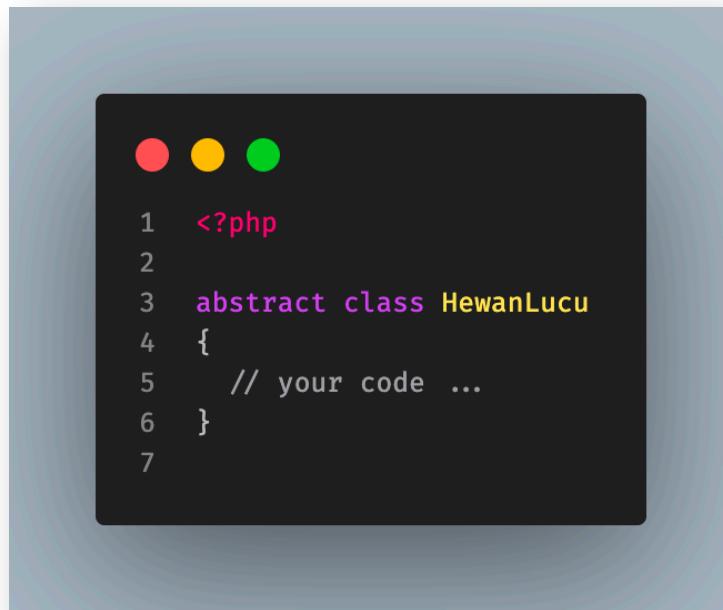
Abstract Class adalah sebuah class yang tidak dapat di instansiasikan. Abstract class tidak dapat dilakukan pembuatan object pada umumnya untuk memanggil method maupun property. Abstract class berperan sebagai kerangka dasarnya saja. Untuk implementasi penggunaan abstract class menggunakan konsep pewarisan, 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 paramaternya tetapi tidak diimplementasikan dalam kode tersebut. Abstract method harus diimplementasikan di dalam class anak (child class).

- **MEMBUAT ABSTRACT CLASS**

Untuk membuat abstract class Untuk membuat abstract class harus berawalan dengan keyword abstract sebelum class. Sebuah abstract class bisa memiliki property dan method biasa layaknya sebuah class ‘normal’, namun juga bisa memiliki abstract method.

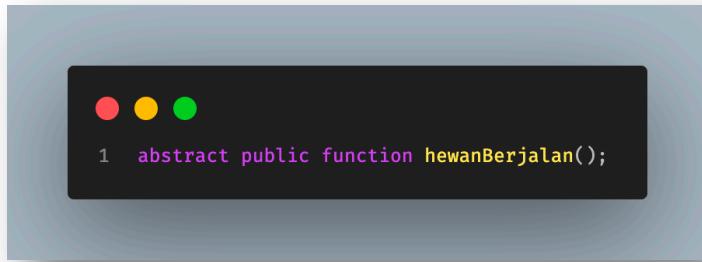
Contoh pembuatan abstract class:



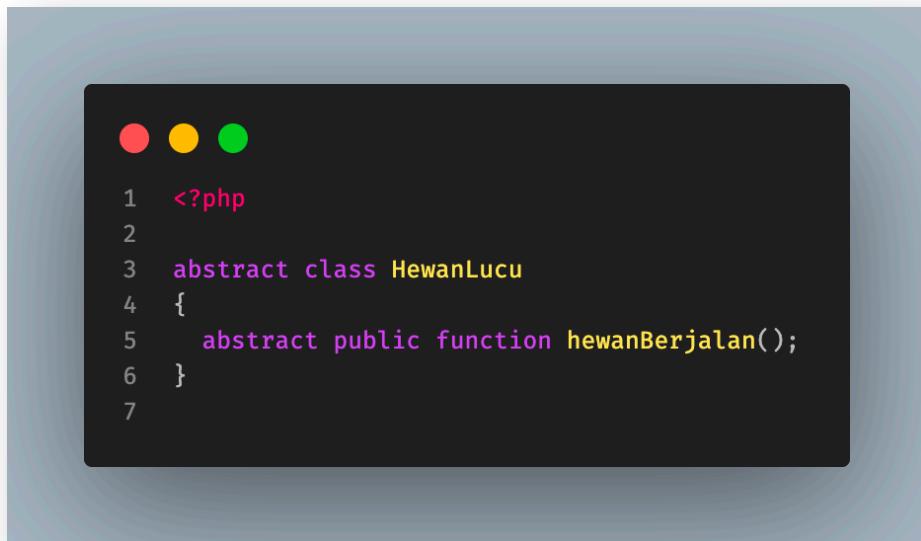
- **MEMBUAT ABSTRACT METHOD**

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. Signature terdiri dari nama method dan parameternya (jika ada).

Contoh pembuatan abstract method pada PHP:



Contoh penerapan abstract method didalam abstract class:



- **ABSTRACT CLASS BISA MEMILIKI METHOD DAN PROPERTY BIASA**

Jika sebuah class dinyatakan sebagai abstract class, class tersebut juga bisa memiliki property dan method ‘normal’. Namun kita hanya bisa mengakses property dan method ini dari class turunan, karena abstract class tidak bisa diinstansiasi.

- **CLASS TURUNAN HARUS MENGIMPLEMENTASIKAN ABSTRACT METHOD**

Jika sebuah class diturunkan dari abstract class, maka class tersebut harus membuat ulang seluruh abstract method yang terdapat dalam abstract class, dan juga harus sesuai dengan signature-nya.

Contoh class turunan yang harus mengimplementasi abstract method:

```
1  <?php
2
3  abstract class HewanLucu
4  {
5      abstract public function hewanBerjalan();
6  }
7
8  class Kucing extends HewanLucu
9  {
10     // Ini method yang harus di implementasi dari abstract class HewanLucu
11     public function hewanBerjalan()
12     {
13         // your code...
14     }
15 }
16
17 $kucingAnggora = new Kucing();
18
```

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 property dengan visibility apapun. Artinya interface hanya mem membuat method saja. Namun dapat memiliki konstanta. Bagi class yang menggunakan object interface, class tersebut harus mengimplementasikan ulang seluruh method yang ada di dalam interface.

Untuk membuat Interface di dalam PHP, penulisnya mirip seperti membuat class, tetapi menggunakan keyword interface, seperti contoh berikut:



Sesuai dengan tujuannya untuk membuat interface/antar muka bagi class, method di dalam perancangan interface harus memiliki hak akses public.

G. INHERITANCE

Inheritance atau Pewarisan/Penurunan adalah konsep pemrograman dimana sebuah class dapat ‘menurunkan’ property dan method yang dimilikinya kepada class lain. Konsep inheritance digunakan untuk memanfaatkan fitur ‘code reuse’ untuk menghindari duplikasi kode program.

Di dalam PHP, inheritance / penurunan dari sebuah class kepada class lain menggunakan kata kunci: ‘extends’, dengan penulisan dasar sebagai berikut:



```
1 <?php
2 class Manusia
3 {
4     var $nama;
5     var $umur;
6     function berjalan()
7     {
8         echo "Manusia Berjalan \n";
9     }
10 }
11
12 class Mahasiswa extends Manusia
13 {
14     function showData()
15     {
16         echo $this->nama . " berumur " . $this->umur . " \n";
17     }
18
19     function berbicara()
20     {
21         echo "Mahasiswa berbicara \n";
22     }
23 }
24 $mahasiswa = new Mahasiswa();
25
26 $mahasiswa->nama = "Jojon";
27 $mahasiswa->umur = 28;
28 $mahasiswa->berjalan();
29 $mahasiswa->berbicara();
30 $mahasiswa->showData();
31
```

Dalam contoh kode diatas, dibuat class Manusia dengan beberapa property dan sebuah method. Property class Manusia belum berisi nilai apa-apa. Dibawah class Manusia, dibuat class Mahasiswa extends class Manusia. Disini class Mahasiswa menurunkan class Manusia. Di dalam class Mahasiswa, kita bisa mengakses seluruh property dan method apapun dari class manusia selama memiliki hak akses public atau protected. Inilah yang dimaksud dengan inheritance/penurunan class dalam OOP.

H. PHP TRAIT

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

Perhatikan contoh di bawah ini, buat tiga buah class :



```
1 <?php
2
3 class Melihat
4 {
5     public function melihat()
6     {
7         return "Saya Melihat Benda Itu";
8     }
9 }
10
11 class Membaca
12 {
13     public function Membaca()
14     {
15         return "Saya Membaca Buku Itu";
16     }
17 }
18
19 class Mendengar
20 {
21     //
22 }
23
```

Jika kita menghendaki class Mendengar bisa menggunakan method membacaBuku() maka kita tinggal melakukan extends ke class Membaca. Bagaimana jika kita ingin class Mendengar dapat menggunakan method melihatBenda() dan membacaBuku()? Apakah class Mendengar meng-extends class Melihat dan class

Membaca tentu ini tidak diperbolehkan oleh PHP, PHP hanya memperbolehkan satu parent class . Untuk itulah adanya Trait.

- **MEBUAT TRAIT**

Sama seperti pembuatan class, hanya saja class diganti dengan trait.



The screenshot shows a code editor window with a dark theme. At the top left, there are three colored circular icons: red, yellow, and green. Below them is a line of code starting with the PHP opening tag. The main content of the code editor displays a trait named 'Melihat' with a single method 'melihat()' that returns a string. The code is numbered from 1 to 9 on the left side.

```
1 <?php
2
3 trait Melihat
4 {
5     public function melihat()
6     {
7         return "Saya Melihat Benda Itu";
8     }
9 }
```

- **MENGGUNAKAN TRAIT**

Cara memakai trait yaitu dengan menggunakan keyword use dalam sebuah class dan diiringi nama trait.



```
1 <?php
2
3 trait Melihat
4 {
5     public function melihat()
6     {
7         return "Saya Melihat Benda Itu";
8     }
9 }
10
11 class Mendengar
12 {
13     use Melihat;
14 }
15
```

- **MULTIPLE TRAIT**

Satu class boleh memakai lebih dari satu trait, cara memanggilnya dipisahkan oleh koma.

```
1 <?php
2
3 trait Melihat
4 {
5     public function melihat()
6     {
7         return "Saya Melihat Benda Itu";
8     }
9 }
10
11
12 trait Membaca
13 {
14     public function Membaca()
15     {
16         return "Saya Membaca Buku Itu";
17     }
18 }
19
20 class Mendengar
21 {
22     use Melihat, Membaca;
23 }
24
25 $aktivitas = new Mendengar();
26 echo $aktivitas→Membaca();
27 echo $aktivitas→melihat();
28
```

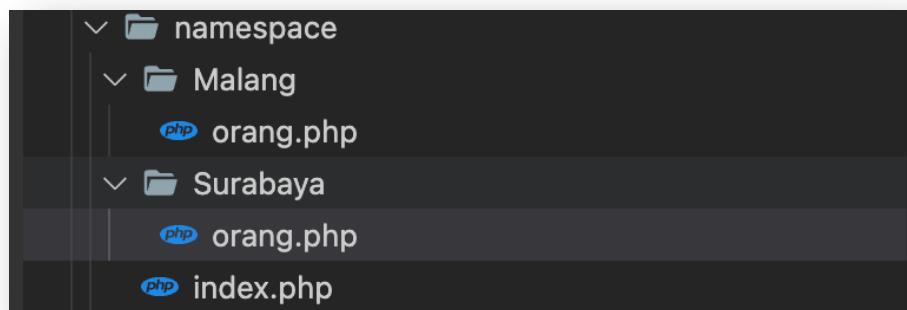
Setelah dijalankan:

```
PROBLEMS      OUTPUT      TERMINAL      DEBUG CONSOLE
sh-3.2$ php trait.php
sh-3.2$ php trait.php
Saya Membaca Buku ItuSaya Melihat Benda Itu
sh-3.2$ 
```

I. PHP NAMESPACE

Dengan namespace, bisa mengorganisasi kode dengan lebih rapi dan terstruktur, sehingga lebih mudah di-maintenance. Cara kerja namespace mirip dengan cara kerja folder dan file. Dalam satu folder tidak bisa membuat 2 buah file dengan nama yang sama. Tetapi hal tersebut bisa dilakukan jika foldernya berbeda.

Untuk mempelajari bagaimana membuat namespace, mari kita praktikkan dengan contoh sederhana. Pada folder htdocs buat folder baru yang berisi 2 folder .Buat beberapa file php dengan struktur seperti berikut:



Dalam folder namespace berisi folder malang, Surabaya dan index.php. Pada folder malang dan Surabaya berisi orang.php.

File Malang/orang.php :



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

File Surabaya/orang.php :



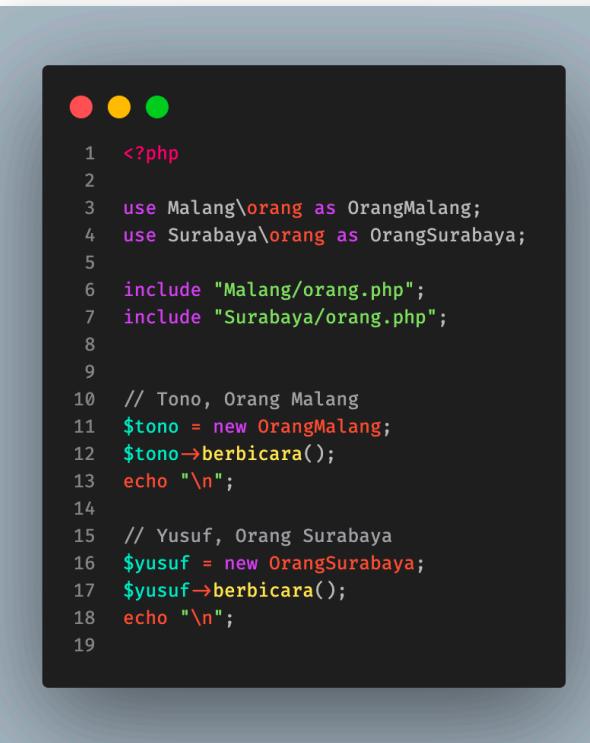
```
1 <?php
2
3 namespace Surabaya;
4
5 class orang
6 {
7     public function berbicara()
8     {
9         echo "HAI";
10    }
11 }
12
```

File index.php :



```
1 <?php
2 include "Malang/orang.php";
3 include "Surabaya/orang.php";
4
5
6 // Tono, Orang Malang
7 $tono = new \Malang\orang();
8 $tono->berbicara();
9 echo "\n";
10
11 // Yusuf, Orang Surabaya
12 $yusuf = new \Surabaya\orang();
13 $yusuf->berbicara();
14 echo "\n";
15
```

Atau bisa juga seperti berikut:



```
1 <?php
2
3 use Malang\orang as OrangMalang;
4 use Surabaya\orang as OrangSurabaya;
5
6 include "Malang/orang.php";
7 include "Surabaya/orang.php";
8
9
10 // Tono, Orang Malang
11 $tono = new OrangMalang();
12 $tono->berbicara();
13 echo "\n";
14
15 // Yusuf, Orang Surabaya
16 $yusuf = new OrangSurabaya();
17 $yusuf->berbicara();
18 echo "\n";
19
```

Hasilnya adalah sebagai berikut:

```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE
sh-3.2$ php index.php
HALO
HAI
sh-3.2$
```

J. MAGIC METHOD

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

Sesuai dengan dokumentasi PHP pada <http://php.net/manual/en/language.oop5.magic.php> kita mengenal method-method magic seperti berikut ini:

1. `__construct()`

Method ini akan diakses pada saat pembuatan object.

2. `__destruct()`

Method ini akan diakses saat object dihapus.

3. `__call()`

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

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 property yang invisible.

6. `__set()`

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

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 unserialize object.

11. `__toString()`

Method ini akan diakses jika kita memperlakukan class sebagai string.

12. `__invoke()`

Method ini akan diakses jika 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 kedalam array ('property' => value, ...).

14. `__clone()`

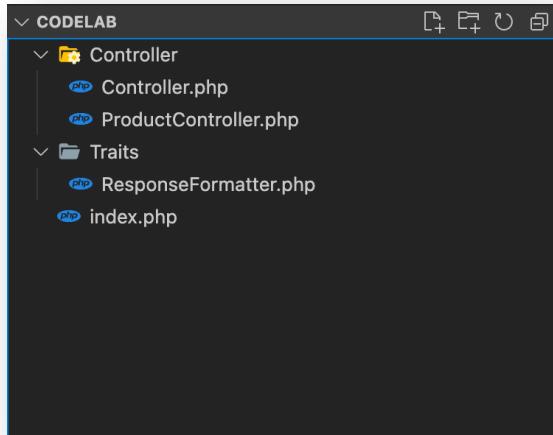
Method ini berhubungan dengan cloning object. Saat selesai proses cloning object, method ini akan dijalankan.

15. `__debugInfo()`

Method ini akan diakses jika program menjalankan `var_dump()` pada suatu object.

CODELAB**LATIHAN**

Buatlah file dan folder seperti berikut:



Terdapat Folder Controller, Traits, dan index.php, didalam folder Controller terdapat file Controller.php dan ProductController.php, dan di dalam folder Traits terdapat file bernama ResponseFormatter.php

File Controller/Controller.php:



```
1 <?php
2
3 namespace Controller;
4
5 class Controller
6 {
7     // VARIABLE ATRIBUT
8     var $controllerName;
9     var $controllerMethod;
10
11    // METHOD UNTUK MENGAMBIL SEMUA ATRIBUT
12    public function getControllerAttribute()
13    {
14        return [
15            "ControllerName" => $this->controllerName,
16            "Method" => $this->controllerMethod
17        ];
18    }
19 }
20
```

File Traits/ResponseFormatter.php:



```
1 <?php
2
3 namespace Traits;
4
5 // DISIMULASIKANINI TRAIT UNTUK FORMATTER JSON RESPONSE
6 trait ResponseFormatter
7 {
8     public function responseFormatter($code, $message, $data = null)
9     {
10         return json_encode([
11             "code" => $code,
12             "message" => $message,
13             "data" => $data
14         ]);
15     }
16 }
17
```

File Controller/ProductController.php:

```
1 <?php
2
3 namespace Controller;
4
5 include "Traits/ResponseFormatter.php";
6 include "Controller/Controller.php";
7
8 use Traits\ResponseFormatter;
9
10 // DISIMULASIKAN INI ADALAH CLASS CONTROLLER PRODUCT
11 class ProductController extends Controller
12 {
13     // gunakan trait yang sudah dibuat
14     use ResponseFormatter;
15
16     // MAGIC METHOD, Method ini akan diakses pada saat pembuatan object.
17     public function __construct()
18     {
19         // SET ATRIBUT controllerName PADA PARENT CLASS CONTROLLER
20         $this->controllerName = "Get All Product";
21         // SET ATRIBUT controllerMethod PADA PARENT CLASS CONTROLLER
22         $this->controllerMethod = "GET";
23     }
24
25     // DISIMULASIKAN INI METHOD UNTUK GET DATA PRODUCT
26     public function getAllProduct()
27     {
28         // ARRAY DUMMY DATA
29         $dummyData = [
30             "Air Mineral",
31             "Kebab",
32             "Spaghetti",
33             "Jus Jambu"
34         ];
35
36         $response = [
37             "controller_attribute" => $this->getControllerAttribute(), // ambil method getControllerAttribute dari parent class
38             "product" => $dummyData
39         ];
40
41         // gunakan method dari trait yang sudah dibuat
42         return $this->responseFormatter(200, "Success", $response);
43     }
44 }
```

File index.php:



```

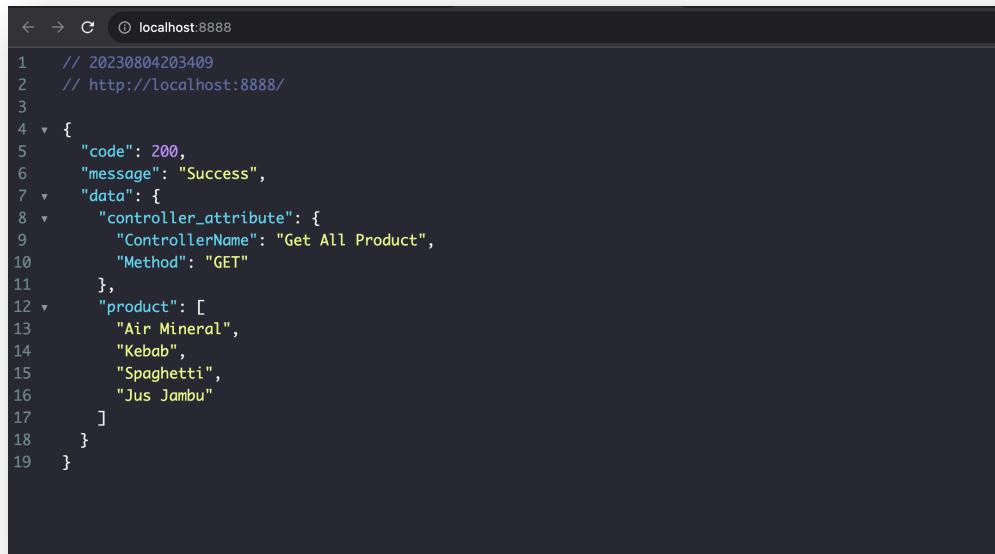
1 <?php
2 include "Controller/ProductController.php";
3
4 use Controller\ProductController;
5
6 // Deklarasikan objek class
7 $productController = new ProductController();
8
9 // Tampilkan hasil kembalian dari method getAllProduct menggunakan echo
10 echo $productController->getAllProduct();
11

```

Setelah semua sudah siap, buka terminal pada directory project dan jalankan php server kalian dengan command berikut:

“php -S localhost:8888 index.php”

Setelah itu buka “localhost:8888” pada browser kalian. Dan hasilnya akan seperti berikut:



```

1 // 20230804203409
2 // http://localhost:8888/
3
4 {
5   "code": 200,
6   "message": "Success",
7   "data": [
8     {
9       "controller_attribute": {
10         "ControllerName": "Get All Product",
11         "Method": "GET"
12       },
13       "product": [
14         "Air Mineral",
15         "Kebab",
16         "Spaghetti",
17         "Jus Jambu"
18       ]
19     }

```

Selamat kalian berhasil mensimulasikan pembuatan REST API dengan format JSON secara sederhana dengan menerapkan paradigma PHP OOP (Object Oriented Programming). Komunikasi antara Frontend Website dan Backend Server umumnya terjadi seperti ini dengan format JSON salah satunya. Masih banyak format lainnya seperti XML, RPC, dll.

TUGAS

TUGAS 1

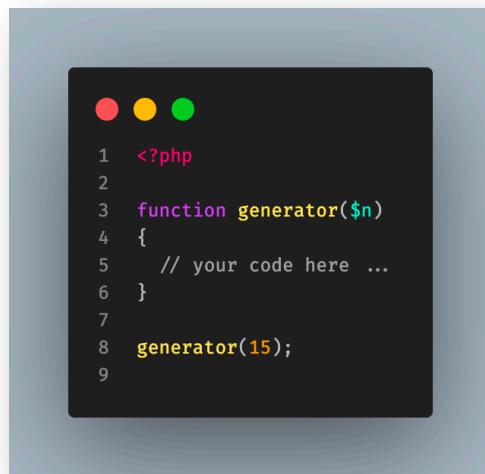
Diberi angka n , untuk setiap bilangan bulat dalam rentang dari 1 hingga n inklusif, cetak satu nilai per baris dengan ketentuan sebagai berikut:

- Jika i habis dibagi dengan 3 dan 5, maka cetak “HelloWorld”
- Jika i habis dibagi 3 (tidak habis dibagi 5), maka cetak “Hello”
- Jika i habis dibagi 5 (tidak habis dibagi 3), maka cetak “World”

Nilai i adalah variable contoh, tidak harus menggunakan huruf i

Deskripsi Fungsi:

Lengkapi fungsi dibawah ini:



PERHATIKAN PENJELASAN BERIKUT:

Parameter $\$n$ adalah batas nilai akhir untuk diuji (inklusif)

Returns: NONE (fungsi tidak mengembalikan apapun)

Print:

Contoh jika $n = 15$, maka nilai yang di uji $[1, 2, 3, 4, 5, 6, \dots n]$ secara ascending (dari kecil ke besar), masing-masing di print pada baris yang berbeda.

Sample Output dengan n = 15:

```
PROBLEMS      OUTPUT      TERMINAL
sh-3.2$ php generator.php
1
2
Hello
4
World
Hello
7
8
Hello
World
11
Hello
13
14
HelloWorld
sh-3.2$ █
```

TUGAS 2

Buatlah sebuah code dengan bahasa PHP dan kreasikan 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.

NOTES:

Asisten akan mengecek pemahaman anda mengenai konsep OOP dan pemahaman bahasa PHP anda.

KRITERIA & DETAIL PENILAIAN

Kriteria	Presentase Penilaian
Dapat menerapkan paradigma OOP dengan bahasa PHP	50%
Dapat mengimplementasikan algoritma pemrograman dengan PHP	40%
Dapat menjawab pertanyaan dari asisten laboratorium	10%