

MODUL 2

PEMROGRAMAN WEB

SYNTAX PHP

A. TUJUAN PERKULIAHAN

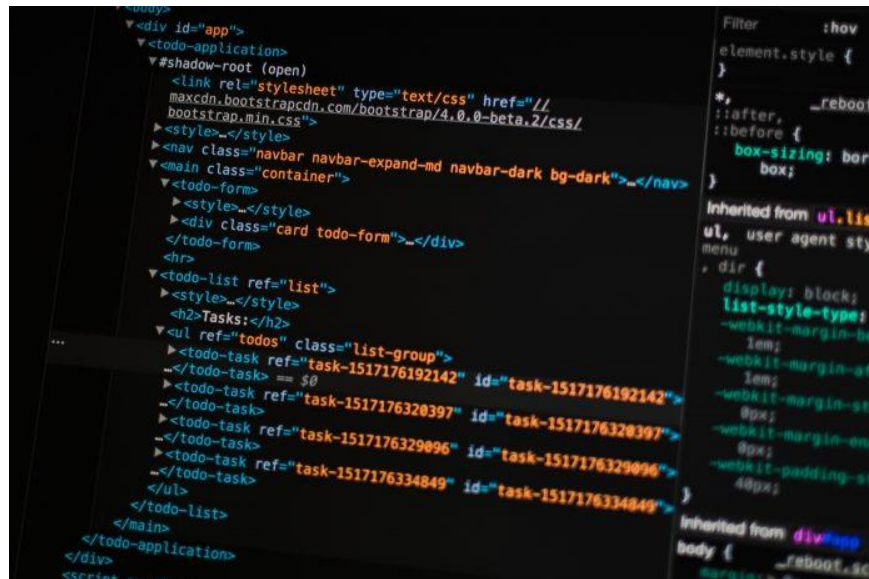
- Mahasiswa mengenali sintaks bahasa PHP
- Mahasiswa mampu mengimplementasikan sintaks bahasa PHP
- Mahasiswa mampu memahami perbedaan antar versi PHP

B. ALOKASI WAKTU 3 x 50 menit

C. DASAR TEORI

1. SYNTAX

Dalam dunia pemrograman, sintaks adalah rangkaian aturan atas susunan kode atau ekspresi secara terstruktur. Aturan tersebut mengacu pada bahasa pemrograman yang dipilih. Jika memenuhi aturan, maka sistem komputer akan mampu mengenali maksud kode program yang ditulis oleh programmer.



Saat kompilasi kode program, komputer akan membaca script sesuai dengan sintaks yang berlaku. Jika tidak memenuhi sintaks, tools yang digunakan biasanya memunculkan pesan jenis kesalahan. Contohnya adalah eror pada sintaks yang harus dibenahi atau peringatan

(warning) yang bersifat opsional untuk diperbaiki. PHP sebagai sebuah bahasa pemrograman juga memiliki sintaks yang harus dipenuhi agar rancangan algoritma dapat berjalan sesuai dengan tujuan pembuatan kode yang diharapkan.

a. Tag

PHP ditulis di dalam sebuah file dengan ekstensi *.php. Penulisan sintaks PHP harus berada di dalam tag / delimiter berikut .

```
<?php  
.....  
?>
```

b. Comment

Comment merupakan baris kode yang tidak dieksekusi sebagai sebuah perintah dan tidak memiliki output dalam bentuk apapun. Comment hanya bersifat sebagai label untuk memperjelas baris kode. Berikut penulisan comment pada satu baris.

```
<?php  
// .....  
?>
```

Atau jika dalam multiple line.

```
<?php  
/*  
.....  
*/  
?>
```

c. Variabel dan tipe data

Variabel adalah lokasi penyimpanan dan terkait nama simbolis yang berisi beberapa kuantitas yang diketahui atau tidak diketahui atau informasi, nilai. Tipe data yang disupport oleh PHP antara lain:

1. String
2. Integer
3. Float (floating point numbers / double)
4. Boolean

5. Array
6. Object
7. NULL
8. Resource

Penulisan variabel dapat dilakukan seperti berikut:

```
<?php
$a = "Hello world!";

$b = 123;

$c = ['saya', 'kamu', 'kita'];

?>
```

d. Standard Output

Standard output merupakan kumpulan perintah yang berfungsi untuk menampilkan tulisan maupun isi variabel. Berikut ini adalah beberapa perintah standard output.

- **Echo dan Print**

Digunakan untuk penulisan tulisan dan isi variabel yang inline. Penulisan tulisan dan isi variabel menggunakan petik satu ('...') atau petik dua("...").

```
<?php
echo "Halo halo";
print "Hai hai";

?>
```

Perbedaannya kecil: echo tidak memiliki nilai kembalian sedangkan print memiliki nilai kembalian 1 sehingga dapat digunakan dalam ekspresi. echo dapat mengambil beberapa parameter (walaupun penggunaan seperti itu jarang terjadi) sementara print dapat mengambil satu argumen. echo sedikit lebih cepat daripada print

- **Print_r**

Digunakan untuk menampilkan isi dari sebuah array

```
<?php
$a=[1,2,3,4,5,6,7];

print_r($a);

?>
```

- **Var_dump**

Digunakan untuk menampilkan isi dari sebuah variabel beserta tipe data yang dimilikinya

```
<?php
$a=[1,2,3,4,5,6,7];
var_dump($a);
?>
```

e. HTML dan PHP

HTML dan PHP dapat dituliskan dalam sebuah file secara bersamaan. Tag PHP dapat disisipkan di antara kode HTML, baik di awal tengah maupun akhir tag HTML.

```
<body>
<ul>
  <li>Nama : <?php echo $nama; ?></li>
  <li>NIM : <?php echo $nim; ?></li>
  <li>Alamat : <?php echo $alamat; ?></li>
</ul>
</body>
```

f. Operator Aritmatik (+, -, *, /, %)

PHP memiliki operator aritmatik untuk melakukan perhitungan matematis, seperti penjumlahan, pengurangan, perkalian, pembagian dan modulo.

```
<?php
echo 12 + 3;

echo "<br>";

echo 15 - 1;

?>
```

g. Operator assignment (=, +=, -=, *=, /=, %=)

Operator assignment digunakan untuk menetapkan nilai dari suatu variabel. Biasanya operator assignment digunakan setelah perhitungan aritmatik.

```
<?php

$a = 12;

$b = 15;

$c = $a + $b;

echo $c;

?>
```

h. Operator perbandingan (==, !=, >, <, >=, <=)

Operator perbandingan biasa digunakan pada control statements atau conditional. Tipe data hasil operator perbandingan ini adalah Boolean, yakni true atau false.

```
<?php

$a = 12;

$b = 15;

var_dump ($a > $b);

?>
```

i. Operator identitas (===, !==)

Operator identitas mirip dengan operator perbandingan dengan tipe data hasil Boolean. Bedanya adalah yang dibandingkan antara variabel tidak hanya nilainya, tetapi juga tipe datanya

```
<?php

$a = 12;

$b = 'dua belas';

var_dump ($a === $b);

?>
```

j. Operator logika (&&, ||, !)

Operator logika membandingkan dua atau lebih statement dengan tipe data Boolean

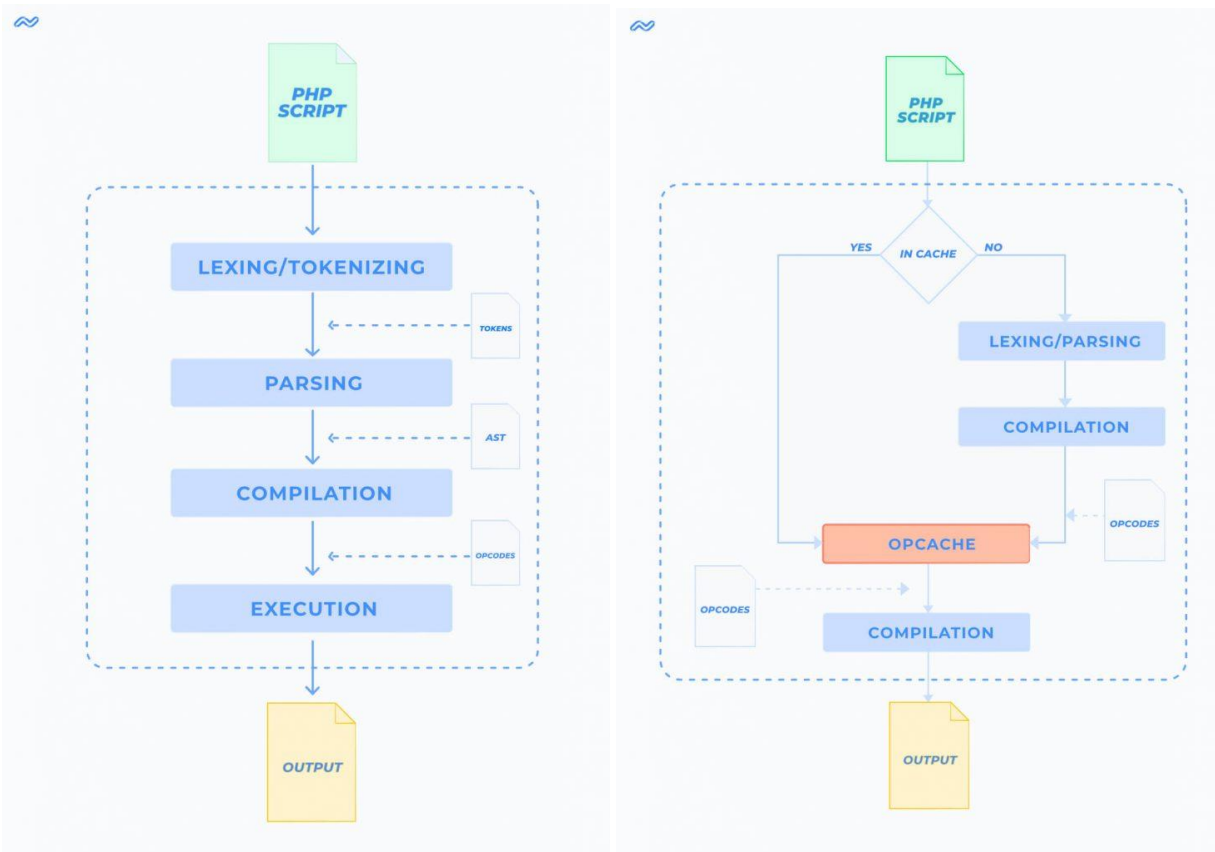
```
<?php  
  
$x = 100;  
  
$y = 50;  
  
if ($x == 100 && $y == 50) {  
    echo "Hello world!";  
}  
  
?>
```

2. PHP 8

PHP 8 merupakan versi PHP yang terbaru pada saat modul ini ditulis. PHP 8 merupakan pengembangan dari versi PHP sebelumnya, yakni PHP 7.4 yang memiliki penambahan fitur dan efisiensi dalam pemrosesan data. Lebih lengkap ada di <https://www.php.net/releases/8.0/en.php>. PHP 8 memiliki beberapa poin perubahan yang cukup signifikan, antara lain:

a. JIT (Just In Time) Compiler

JIT Compiler merupakan teknik untuk meningkatkan kinerja program yang diinterpretasikan. Sebab sebelum program dijalankan, JIT Compiler sudah mengkompilasi program menjadi kode bahasa mesin. Dengan teknik JIT Compiler, eksekusi kode pun menjadi jauh lebih cepat dibanding interpreter PHP versi sebelumnya.



Pada compiler PHP sebelumnya, interpreter akan melakukan proses mengurai, mengkompilasi dan eksekusi kode. Proses itu dilakukan berulang kali setiap ada request skrip. Hal itu menyebabkan pemborosan ruang CPU.

Namun JIT Compiler PHP 8 memiliki kode mesin aktual yang dapat bekerjasama dengan OPcache. Jadi, saat ada request terhadap skrip dan cache sudah tersedia di OPcache, skrip itu akan langsung diteruskan ke CPU untuk dikompilasi.

Selain itu, JIT Compiler PHP 8 memiliki dua mesin kompilasi JIT, yaitu **function** dan **tracing**. JIT berbasis function akan mengoptimalkan kode hanya dalam satu fungsi lingkup saja. Sedangkan, JIT berbasis tracing mengoptimalkan seluruh pelacakan dan pencarian kode.

Menurut ibexa.co, ketika fitur JIT Compiler tersebut digunakan, fitur itu dapat meningkatkan kinerja aplikasi berbasis PHP sebesar **9%**. Silakan buka link berikut untuk informasi lebih lanjut <https://developers.ibexa.co/blog/benchmarks-php-7.4-8.0-jit-opcache-preloading-symfony>

b. Weak Maps

Weak Maps adalah struktur data yang dapat menampung objek. Berbeda dengan Weak Reference pada PHP 7.4, Weak Maps juga dapat membersihkan cache dari objek yang tak direferensikan lagi.

Dalam PHP, ketika ada variabel dari objek dihapus, maka akan ada pengecekan referensi ke objek tersebut. Kalo tidak ada variabel lain, itu artinya aman untuk menghapus objeknya. Melalui fitur Weak Maps pada PHP 8, proses penghapusan itu jadi lebih otomatis. Berikut ini contoh penggunaanya:

```
class EmailSender {  
    private WeakMap $counter;  
  
    public function __construct() {  
        $this->counter = new WeakMap();  
    }  
  
    public function send(Account $account, EmailMessage  
$message) {  
        $this->counter[$account] ??= 0;  
        $this->counter[$account]++;  
        // Do the rest.  
    }  
  
    public function getStats(): iterable {  
        return $this->counter->getIterator();  
    }  
}
```

Pada contoh di atas, setiap method **EmailSender::send()** dipanggil, maka dia akan menghitung jumlah email yang dikirim oleh objek **Account**. Jumlah email tidak menjadi property dari objek Account. Method **EmailSender::getStats** dapat digunakan dengan

foreach untuk menampilkan daftar angka tiap akun yang di email karena menggunakan iterable. Jika Account di **unset** pada kode, maka **counter juga akan otomatis dihapus**. Dengan cara ini, penggunaan memori jadi lebih efisien karena proses caching dilakukan tanpa tambahan logic lain seperti **destructor** dan lainnya.

Buka link berikut untuk mengenal lebih banyak tentang praktek penggunaan weakmap.
<https://php.watch/articles/practical-weakmap>

c. Union Type

Union type adalah suatu nilai yang memiliki representasi sama di sebuah posisi dan dapat menerima beberapa jenis variabel. Karena PHP mendukung Union Type, Anda bisa mendeklarasikan variabel dalam beberapa tipe.

- **Type** atau **null**, menggunakan sintaks khusus ?Type
- **array** atau **Traversable**, menggunakan tipe khusus iterable

Pada PHP versi sebelumnya, jenis penyatuan arbitrer tidak didukung oleh bahasa itu. Sehingga programmer cenderung menggunakan anotasi PHPDoc sebagai solusi.

Untungnya saat ini union type telah hadir secara native di PHP 8. Jadi, Anda tak perlu lagi menggunakan anotasi **PHPDoc**. Anda bisa menetapkan beberapa tipe parameter dengan sintaks T1|T2|... di semua posisi.

Ada beberapa keuntungan dari union type. **Pertama**, karena tipe nilai parameter yang bisa ditetapkan sendiri, Anda dapat menghindari kemungkinan terjadinya error dengan lebih baik. **Kedua**, informasi tipe yang kurang update pun bisa dihindari.

Berikut contoh perbandingan penggunaan union type di PHP:

PHP 7

```
class Number {  
    /** @var int|float */  
    private $number;  
  
    /**  
     * @param float|int $number
```

```

*/

public function __construct($number) {
    $this->number = $number;
}
}

new Number('NaN'); // Ok

```

PHP 8

```

class Number {
    public function __construct(
        private int|float $number
    ) {}
}

new Number('NaN'); // TypeError

```

Dengan adanya fitur Union Type ini, penulisan sintaks menjadi lebih sederhana dan membuat deklarasi tipe lebih mudah dibaca.

d. Penggunaan ::Class For Objects

Pada versi sebelumnya, Anda harus menggunakan **get_class()** untuk menetapkan kelas ke objek. Namun pada PHP 8, Anda dapat menggunakan **::class to objects** untuk memastikan kode sumber jadi lebih ramping. Agar lebih jelas, perhatikan contoh di bawah ini:

```

$object = new stdClass;
var_dump($object::class); // "stdClass"
$object = null;
var_dump($object::class); // TypeError

```

e. NullSafe Operator

Sebelumnya, untuk pengecekan nilai null di PHP 7 Anda perlu memasukkan kode **is_null** pada beberapa baris kode berulang kali. Di bawah ini adalah contohnya:

PHP 7

```
<?php
$user = get_user($id);
if (!is_null($user)) {
    $address = $user->getAddress();
    if (!is_null($address)) {
        $state = $address->state;
        If (!is_null($state)) {
            // And so on.
        }
    }
} }
```

Melalui PHP 8 Anda bisa menggunakan fitur NullSafe Operator. Fitur ini memungkinkan Anda melakukan metode pemanggilan operator **?->** untuk mengecek nilai null.

```
$country = $session?->user?->getAddress()?->country;
```

Artinya, dengan menggunakan fitur ini, Anda bisa mendeteksi null dalam suatu elemen rantai. Jika ditemukan null dalam rantai itu, maka proses eksekusi akan langsung dibatalkan

f. Constructor Property Promotion

Fitur baru PHP 8 berikutnya adalah Constructor Property Promotion. Fitur ini berfungsi mengurangi penulisan atribut berulang pada class constructor. Di bawah ini adalah contohnya:

```
class Point {  
    public function __construct(  
        public float $x = 0.0,  
        public float $y = 0.0,  
        public float $z = 0.0,  
    ) {}  
}
```

Hal berbeda terjadi pada PHP 7 dimana semua properti harus diulang beberapa kali sebelum Anda dapat menggunakannya dengan objek. Contohnya seperti ini:

```
class Point {  
    public float $x;  
    public float $y;  
    public float $z;  
  
    public function __construct(  
        float $x = 0.0,  
        float $y = 0.0,  
        float $z = 0.0  
    ) {  
        $this->x = $x;  
        $this->y = $y;  
        $this->z = $z;  
    }  
}
```

Melalui fitur Constructor Property Promotion di PHP 8, penulisan atribut menjadi lebih pendek, lebih mudah dibaca dan tak terlalu rentan terhadap error. Namun fungsi ini hanya bisa digunakan pada promoted parameter seperti parameter yang diawali dengan public, protected, dan private.

g. Named Arguments

Pada versi PHP 7.4, Anda perlu memperhatikan urutan pemberian argumen untuk mengirimkan argumen ke sebuah fungsi. Namun pada PHP 8, Anda bisa mengirimkan argumen ke sebuah fungsi menggunakan nama pengenalan.

Misalnya saja jika ingin melakukan set nilai argumen ke-7 (`http_only`) pada PHP 7. Anda perlu mengirim nilai argumen sebelumnya. Hal ini karena argumen dibaca berdasarkan urutan. Sehingga Anda harus mengirim argumen ke-1 sampai ke-6.

```
setcookie(  
    "Selamat datang",    // ---> 1  
    "abc123",            // ---> 2  
    time() + 3600,       // ---> 3  
    "",                  // ---> 4  
    "",                  // ---> 5  
    false,               // ---> 6  
    true                 // ---> 7  
);
```

Namun hal ini tidak berlaku untuk named arguments PHP 8. Argumen yang dikirimkan memiliki nama sehingga urutan argumen jadi tidak berpengaruh. Artinya, Anda hanya perlu menulis nilai yang ingin Anda ubah.

```
setcookie(  
    expires: time() + 3600,  
    name: "selamat datang",  
    value: "abc123",  
    http_only: true  
);
```

Dengan adanya Name Arguments pada PHP 8 ini, proses coding menjadi lebih mudah. Sebab identifier ketika argumen dikirim ke fungsi akan memperjelas pembacaan kode. Selain itu, kode bisa terdokumentasi secara mandiri.

h. Match Expression

PHP 8 memperkenalkan fitur Match Expression. Fitur ini mirip dengan Switch pada PHP 7. Bedanya, Match Expression memiliki sintaks yang lebih ringkas dan aman.

Selain perbedaan utama tersebut, perbedaan lain antara Match dan Switch adalah sebagai berikut:

- Match expression dapat mengembalikan value
- Match expression hanya mendukung ekspresi tunggal dan tidak membutuhkan pernyataan break;
- Match expression melakukan perbandingan yang sangat ketat

Contoh perbedaan Switch dan Match pada penggunaannya adalah sebagai berikut:

PHP 7

```
switch (8.0) {  
    case '8.0':  
        $result = "Oh no!";  
        break;  
    case 8.0:  
        $result = "This is what I expected";  
        break;  
}  
echo $result;  
//> Oh no!
```

PHP 8

```
echo match (8.0) {  
    '8.0' => "Oh no!",  
    8.0  => "This is what I expected",  
};  
//> This is what I expected
```

Contoh penggunaan Switch di PHP 7 di atas menunjukkan hasil output yang mengacu pada valuenya, meskipun bukan tipe datanya, yaitu “Oh no!”.

Sedangkan pada Match PHP 8, ketika Anda memasukkan type data yang sama maka akan memunculkan outputnya, yaitu “This is what I expected”. Ketika Anda memasukan type data berbeda, maka akan memunculkan pesan error.

i. Attributes

Attributes adalah cara untuk memberi keterangan informasi tambahan dan konfigurasi, sehingga dapat dibaca oleh mesin pada deklarasi dalam kode.

Pada versi sebelumnya, Anda melakukan dokumentasi parameter dan nilai menggunakan PHPDoc. Fitur Attributes di PHP 8 memudahkan Anda untuk menggunakan metadata terstruktur dengan sintaks asli PHP. Berikut ini adalah perbandingannya:

PHP 7

```
class PostsController
{
    /**
     * @Route("/api/posts/{id}", methods={"GET"})
     */
    public function get($id) { /* ... */ }
}
```

PHP 8

```
class PostsController
{
    #[Route("/api/posts/{id}", methods: ["GET"])]
    public function get($id) { /* ... */ }
```

j. Numeric Strings

Pada versi sebelumnya, komparasi string dengan angka dilakukan dengan cara mengubah string ke angka. Kemudian baru melakukan komparasi antara integer atau float. Cara ini rentan bug atau masalah keamanan.

Numeric Strings menjadikan PHP 8 lebih baik daripada versi sebelumnya. Sebab numeric strings mengubah string menjadi angka sebelum melakukan komparasi. Tujuannya untuk menghindari kesalahan komparasi. Berikut contoh perbandingan antara PHP 7 dan PHP 8 yang menunjukkan hasil perbandingan sederhana:

PHP 7

```
<?php
0 == '0' // true
?>
```

PHP 8

```
<?php
0 == '0' // false
?>
```

Contoh PHP 7 menunjukkan bahwa integer 0 (nol) sama dengan '0'. Artinya type itu adalah type string dengan output true. Sedangkan pada PHP 8, kondisi yang sama akan menghasilkan output false karena adanya type data yang berbeda.

k. Consistent type errors for internal functions

Saat Anda melakukan kesalahan pengiriman tipe, fungsi user defined dan internal akan mengirimkan pesan yang berbeda. Fungsi user defined mengirimkan pesan TypeError. Sedangkan fungsi internal mengirimkan beragam pesan berbeda.

Hal ini mungkin membuat Anda kesulitan mengidentifikasi letak dan penyebab error atau kesalahan. Nah, PHP 8 menghadirkan solusinya dengan fitur consistent type errors for internal functions.

Fitur ini akan memunculkan pesan error dan letak error ketika ada kesalahan koding. Kesalahan ditunjukkan dengan **TypeError : ...** atau **ValueError : ...** .

Dengan adanya fitur ini, Anda dapat mengetahui letak error. Selain itu, Anda juga akan mengetahui cara memperbaiki error statement. Berikut ini contoh penggunaan fitur ini:

PHP 7

```
strlen([]); // Warning: strlen() expects parameter 1 to be string,  
array given  
  
array_chunk([], -1); // Warning: array_chunk(): Size parameter expected  
to be greater than 0
```

PHP 8

```
strlen([]); // TypeError: strlen(): Argument #1 ($str) must be of type  
string, array given  
  
array_chunk([], -1); // ValueError: array_chunk(): Argument #2  
($length) must be greater than 0
```

Seperti yang terlihat pada contoh di atas, pesan error yang muncul pada PHP 8 diperbaiki menjadi lebih konsisten untuk semua jenis parameter yang tidak valid.

D. PRAKTIKUM : Pengenalan Syntax PHP

1. Salin script berikut ini dan beri nama p0201.php:

```
<!DOCTYPE html>
<html>
    <head>Contoh</head>
    <body>
        <h1>Latihan 1</h1>
        <?php
            echo "<p>Hello world</p>";
        ?>
        <h1>Latihan 2</h1>
        <?php
            $a = '14';
            $b = '11';
            $c = $a + $b;
            echo (Hasil perhitungan adalah $c);
        ?>
        <h1>Latihan 3</h1>
        <?php
            $z = [nama->'agus',nim->'k1424567',alamat->'Solo',bidang->'RPL']
            echo $z[nama];
        ?>
        <h1>Latihan 4</h1>
        <?php
            $objek = new stdClass();
            $objek =>name = "Steven William";
            $objek =>age = 30;
            $objek =>city = "Kartosuro";

            $myJSON = json ($objek);

            echo $myJSON;
        ?>
    </body>
</html>
```

Jalankan script di atas di browser. Amati!

Berdasarkan hasil, apakah kode dapat dijalankan? Jika tidak bagaimana perbaikannya?

Tulis analisa hasil perbaikan per tiap Heading latihan.

Jawab:

.....

-
-
2. Buat script yang dapat menyimpan data berikut dalam bentuk variabel array dan menampilkannya isi dari array tersebut dan beri nama p0202.php.
 - Sulis | K3510001 | Solo | RPL
 - Rendi | K3510002 | Klaten | TKJ
 - Jatmiko | K3510003 | Boyolali | Multimedia
 - Feri | K3510004 | Wonogiri | RPL
 3. Berdasarkan hasil analisa Anda, bagaimana cara menentukan penggunaan versi PHP yang tepat pada suatu aplikasi berbasis web ?

E. REFERENSI

<https://adammuiz.com/syntax/>

<https://www.php.net/releases/8.0/en.php>

<https://www.niagahoster.co.id/blog/php-8-adalah/>

<https://php.watch/articles/practical-weakmap>