

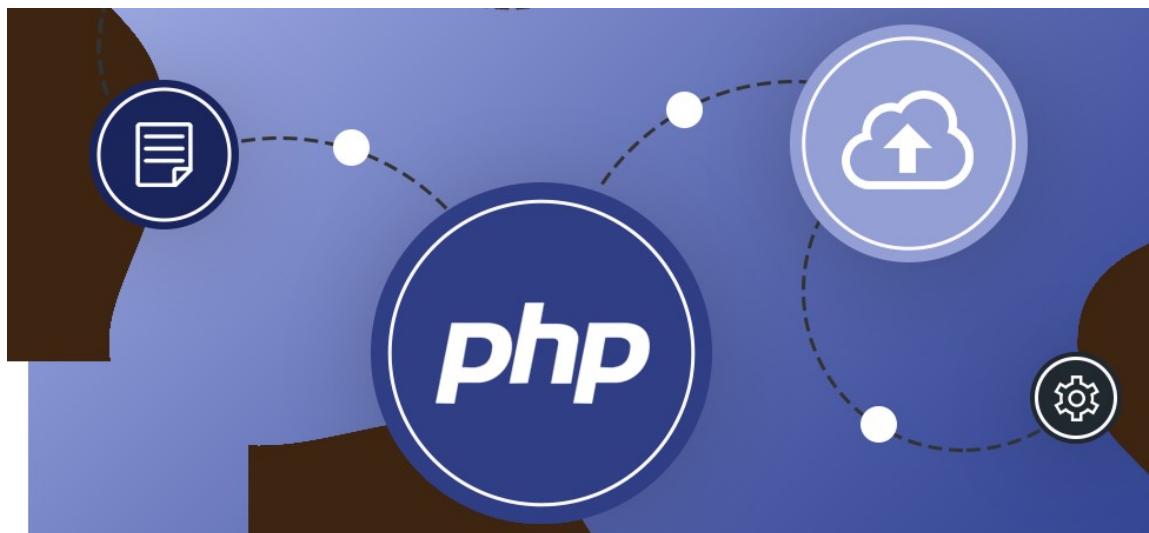
MODUL PERKULIAHAN

INSTITUT WIDYA PRATAMA



PEMROGRAMAN BERBASIS WEB 2

MENGENAL PHP



DESKRIPSI MATAKULIAH

Mahasiswa mampu mengembangkan aplikasi web dinamis menggunakan PHP dan framework CodeIgniter dengan konsep MVC. Mahasiswa mempelajari implementasi CRUD, autentikasi, pengelolaan database, serta integrasi Bootstrap untuk tampilan responsif. Melalui pendekatan berbasis proyek, mahasiswa diharapkan mampu merancang dan membangun aplikasi web secara utuh dan profesional.

CAPAIAN PEMBELAJARAN

DOSEN PENGAMPU

Taufikqurrohman, M.Kom

INSTITUT WIDYA PRATAMA

MENGENAL PHP

CAPAIAN PEMBELAJARAN

Mahasiswa memahami dasar sintaks PHP dan dapat menuliskan program sederhana.

MATERI PEMBELAJARAN

- Mengenal PHP
- Apa itu XAMPP
- Mengenal Struktur Dasar PHP
- Aturan Menulis PHP
- Mengenal Variabel
- Tipe Data PHP
- Konversi Tipe Data
- Menghapus Variabel dari Memori
- Praktikum

MATERI

Setelah kita belajar tentang HTML, CSS dan Bootstrap (selama sebelum UTS) selanjutnya kita belajar Javascript dan sekarang membahas PHP. PHP adalah bahasa pemrograman yang berjalan di server, karena itu kita membutuhkan sebuah web server. PHP juga merupakan bahasa pemrograman yang bersifat *open source*. Artinya pengguna PHP bebas memodifikasi dan mengembangkan sesuai dengan kebutuhan.

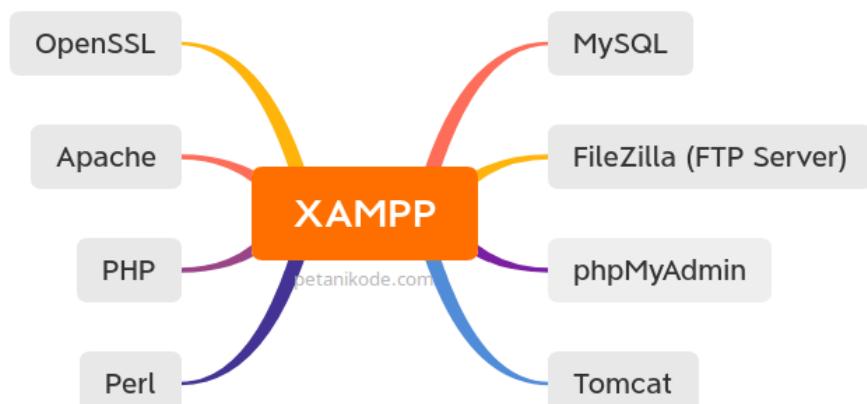
1. Apa itu PHP

PHP (PHP: *Hypertext Preprocessor*) adalah sebuah bahasa pemrograman server *side scripting* yang bersifat *open source*. PHP Pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1994. Setiap satu *statement* (perintah) pada PHP biasanya diakhiri dengan titik-koma (;). PHP merupakan bahasa pemrograman yang *case sensitive*, untuk nama *identifier* yang dibuat oleh user (seperti: variable, konstanta, fungsi dll), namun tidak *case sensitive*, untuk identifier *built-in* dari PHP. Sebagai sebuah *scripting language*, PHP menjalankan instruksi pemrograman saat proses *runtime*.

PHP merupakan bahasa pemrograman *server-side*, maka *script* dari PHP nantinya akan diproses di server. Cara menjalankan file PHP tidak seperti menjalankan file HTML, kita diharuskan untuk mengetikkan pada bagian address bar dari browser yang digunakan, seperti: <http://namaserver/folder/namfile.php>. Untuk **namaserver**: kita bisa menggantinya dengan IP Address atau *localhost*, untuk nama **folder**: menyesuaikan dengan folder yang kita buat di web server, sedangkan untuk **namafile.php**: merupakan nama file yang akan dipanggil, defaultnya adalah **index.php**, sama seperti penamaan di html. Web server di windows bisa dibuat dengan berbagai macam cara, ada yang menggunakan PHP secara manual dengan web server bawaan PHP, ada yang menggunakan WSL lalu menginstal LAMPP Stack, ada yang menggunakan XAMPP, ada juga yang menggunakan Docker, dan lain sebagainya. Untuk pembelajaran ini kita menggunakan XAMPP.

2. Apa itu XAMPP

XAMPP adalah program untuk membuat server. XAMPP menyediakan paket server yang kita butuhkan untuk pengembangan web dengan PHP. Mulai dari PHP, MySQL, PHPmyadmin, dll.



XAMPP merupakan singkatan dari **X** (*cross-platform*), Apache, MySQL atau MariaDB, **PHP**, **Perl**. Huruf **X** di XAMPP memiliki makna *cross-platform* yang artinya XAMPP bisa digunakan di berbagai sistem operasi seperti Linux, Windows, dan MacOS.

Yang belum punya XAMPP silahkan download di: <https://www.apachefriends.org/download.html>. Sedangkan untuk cara install XAMPP di Windows 10 bisa mengikuti tutorial yang ada di website berikut: <https://pureinfotech.com/install-xampp-windows-10/>

3. Stuktur Program PHP yang Paling Dasar

Ini adalah bentuk paling dasar program PHP:

```
<?php
    echo "Hello World!";
?>
```

Keterangan:

- `<?php;` ini adalah pembuka program PHP. Pembuka ini wajib ada di setiap program PHP.
- `echo "Hello World!";` adalah sebuah statement atau perintah untuk menampilkan teks.
- `?>;` ini adalah penutup program PHP. Tutup sebenarnya bersifat *opsional*. Tutup program dibutuhkan saat kita menggabungkan kode PHP dengan HTML.

4. Menulis Kode HTML dan PHP

Saat kita menulis kode PHP di dalam HTML, maka wajib hukumnya membuat tutup program. Contoh:

```
<!DOCTYPE html>
<html>
    <head>
        <title><?php echo "Belajar PHP" ?></title>
    </head>
    <body>
        <?php
            echo "saya sedang belajar PHP<br>";
            echo "<p>Belajar PHP hingga jadi master</p>";
        ?>
    </body>
</html>
```

Perhatikan contoh di atas! Apa yang akan terjadi bila kita menghapus tutup PHP (`?>`)? Tentunya program akan error. Jika sintak PHP ditulis di dalam HTML, maka file tersebut harus disimpan dengan ekstensi `.php` bukan `.html` meskipun isinya HTML dan PHP.

Jika ada yang menulis sintak PHP seperti ini:

```
<?php
    echo "<html>";
    echo "<head>";
    echo "<title>Judul Web</title>";
    echo "</head>";
    echo "<body>";
    echo "<h1>Selamat datang</h1>";
    echo "</body>";
    echo "</html>";
```

maka sintak PHP yang seperti itu bisa tidak ditutup, karena kode HTML-nya ditulis di dalam sintak PHP.

a. Penulisan Statement dan Ekspresi

Statement dan ekspresi adalah intruksi yang akan diberikan ke komputer. Setiap statement dan ekspresi di PHP harus diakhiri dengan titik koma (;). Contoh:

```
<?php
    echo "ini statement 1";
    echo "ini statement 2";

    $a = $b + $c;
```

Jika kita lupa menuliskan titik koma? Maka programnya akan error.

b. Aturan Penulisan Case PHP

PHP adalah bahasa pemrograman yang bersifat *case sensitive*. Artinya, huruf besar (kapital) dan huruf kecil akan dibedakan. Contoh:

```
<?php
$makul = "Pemrograman";
$MAKUL = "Berbasis Web 2";
?>
```

Variabel \$nama dan \$NAMA adalah dua variabel yang berbeda. Mereka tidak sama. Penulisan huruf besar dan kecil dalam program harus diperhatikan, karena bisa menyebabkan error bila salah. Contoh: sintak program yang salah tulis.

```
$sumur = 19;
echo "Umur saya adalah $Umur";
```

Pada kode diatas, kita membuat variabel \$umur dengan huruf kecil. Lalu saat menggunakan variabel kita menggunakan \$Umur. Jelas sekali ini akan menyebabkan error. Untuk menghindari ini, kita harus konsisten dalam penamaan variabel dan konstanta. Gunakan nama variabel dengan huruf kecil saja dan konstanta dengan huruf kapital. Contoh:

```
constINI_KONSTANTA = 123;
$ini_variabel = 23;
$iniJugaVariabel = 49;
```

Kenapa ini ada huruf kapital di `$iniJugaVariabel`? Karena, ini yang namanya *camelCase*. Jika variabel terdiri dari dua atau lebih suku kata, maka kita bisa memisahnya dengan huruf kapital atau bisa juga dengan garis bawah (*underscore*).

c. Penulisan Komentar di PHP

Komentar adalah bagian yang tidak akan dieksekusi oleh komputer. Biasanya digunakan untuk keterangan, penjelasan, dan dokumentasi kode program. Komentar di PHP dapat ditulis dengan dua cara:

- Menggunakan tanda `//` untuk komentar satu baris;
- Menggunakan tanda `/*` untuk komentar lebih dari satu baris.

Contoh:

```
<?php

    // ini adalah komentar
    echo "Hello world";

    /*
        ini adalah komentar
        yang lebih dari satu
        baris
    */

?>
```

d. Penulisan Blok Program

Blok program merupakan kumpulan dari statement dan ekspresi. Blok program di PHP dibungkus dengan kurung kurawal `{ ... }`. Contoh:

```
if ($umur > 18){
    echo "Kamu sudah dewasa";
    echo "Selamat datang";
    echo "Kamu boleh minum kopi";
}
```

Program di atas adalah contoh blok kode `if` yang berisi tiga statement.

e. Fungsi Mencetak Teks

PHP memiliki beberapa fungsi untuk mencetak teks ke layar, fungsi-fungsi ini akan sering kita gunakan dalam pemrograman PHP, diantaranya:

- fungsi `echo()`;
- fungsi `print()`;
- fungsi `printf()`;

1) Fungsi echo()

Fungi `echo()` adalah fungsi untuk menampilkan teks ke layar. Fungsi ini dapat digunakan dengan tanda kurung maupun tanpa tanda kurung. Contoh:

```
<?php
    echo "<h2>Belajar PHP itu mudah!</h2>";
    echo ("Hello world!<br>");
    echo "Aku sedang belajar PHP!<br>";
    echo "Ini ", "teks ", "yang ", "dibuat ",
"terpisah.";
?>
```

Hasilnya:



Fungsi `echo()` tidak akan mengembalikan apa-apa setelah dieksekusi. Dia hanya bertugas menampilkan teks saja.

2) Fungsi print()

Fungsi `print()` sama seperti fungsi `echo()`. Dia juga digunakan untuk menampilkan teks ke layar. Fungsi `print()` juga bisa digunakan tanpa tanda kurung. Perbedaannya dengan `echo()`:

- Fungsi `print()` akan selalu mengembalikan nilai `1` saat dieksekusi, sedangkan `echo()` tidak mengembalikan apa-apa.
- Fungsi `print()` hanya boleh diberikan satu parameter saja, sedangkan `echo()` boleh lebih dari satu.

Contoh:

```
<?php
    print "<h2>Belajar PHP dari Nol!</h2>";
    print "Hello world!<br>";
    print "Belajar mencetak teks di PHP!";
?>
```

Hasilnya:



Belajar PHP dari Nol!

Hello world!
Belajar mencetak teks di PHP!

Apabila kita membuat variabel lalu mengisinya dengan fungsi `print()` seperti ini:

```
$cetak = print("Hello World!");
```

Maka variabel `$cetak` akan bernilai 1. Apabila kita memberikan dua parameter ke dalam fungsi `print()`, Maka akan terjadi error. Contoh:

```
$cetak = print("Hello World!");
print("Hello", "World"); // <-- ini akan error
```

Hasilnya:



Parse error: syntax error, unexpected '' in C:\xampp\htdocs\php\index.php
on line 4

3) Fungsi printf()

Fungsi `printf()` adalah fungsi untuk memformat teks atau string. Fungsi ini akan mengembalikan panjang dari teks saat dieksekusi. Biasanya saat kita menggunakan fungsi `echo`, kita akan menulis seperti ini:

```
$txt = "STMIK Widya Pratama";
echo "Aku kuliah di " . $txt . "<br>";
```

Apabila kita ingin menggunakan fungsi `printf()`, maka kita bisa lakukan seperti ini:

```
$txt = "STMIK Widya Pratama";
printf("Aku kuliah di %s<br>", $txt);
```

Simbol `%s` adalah sebuah *placeholder* untuk teks (string). Selain simbol `%s` ada juga simbol:

- `%d` untuk bilangan desimal (integer);
- `%f` untuk pecahan (float);
- `%b` untuk boolean;

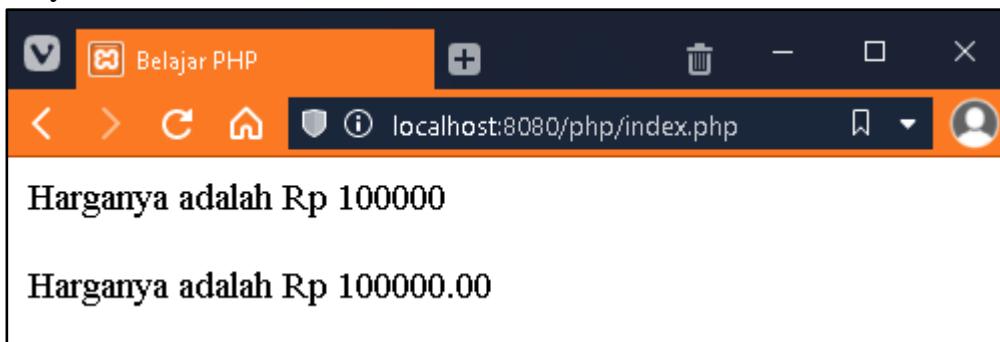
Salah satu yang patut kita coba adalah `%f`, karena dengan ini kita bisa mengatur bagaimana bilangan pecahan ditampilkan. Contoh:

```
// misalkan kita punya bilangan dengan notasi E seperti
ini
$harga = 100000;

// jika kita cetak dengan echo:
echo "Harganya adalah Rp $harga";

// jika kita cetak dengan printf
printf("Harganya adalah Rp %.2f", $harga);
```

Hasilnya:



Simbol `.2f` artinya kita akan menetak bilangan dengan dua angka di belakang koma.

f. Menggabungkan String atau Teks di PHP

Pada fungsi `echo()`, kita menggabungkan teks atau string dengan memberikan sebagai argumen (dipisah dengan tanda koma).

```
echo "Ini ", "teks ", "yang ", "dibuat ", "terpisah.;"
```

Selain cara ini, kita juga bisa melakukannya dengan tanda titik (.). Titik adalah operator untuk menggabungkan dua teks di PHP. Kenapa tidak menggunakan simbol plus (+)? Pada PHP, simbol plus digunakan untuk penjumlahan, bukan untuk menggabungkan string. Jadi:

```
echo "1" + "1"; // akan menghasilkan 2
echo "1" . "1"; // akan menghasilkan 11
```

Contoh:

```
<?php
$txt1 = "Belajar PHP";
$txt2 = "STMIK Widya Pratama";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Aku Kuliah di " . $txt2 . "<br>," . "<br>";
echo $x + $y;
?>
```

Hasilnya:



g. Mencetak Tanda Kutip

Kadang kita ingin menetak tanda kutip seperti ini:

```
echo 'hari ini hari jum'at';
```

Maka akan terjadi error! Karena di sana dianggap tutup teksnya berakhir pada teks `jum`. Lalu bagaimana caranya kita mencetak tanda kutip? Kita bisa menggunakan tanda *back slash* (\). Contoh:

```
echo 'hari ini hari jum\'at';
```

Maka akan menghasilkan:

```
hari ini hari jum'at
```

5. Mengenal Variabel dan Tipe Data

Apa kamu pernah menemukan `x` dan `y` dalam perlajaran matematika? Mereka berdua sebenarnya adalah variabel yang menyimpan sesuatu. Kadang, kita sering diberi tugas untuk mencari tahu isi dari `x` dan `y`. Contohnya:

```
jika x + 3 = 5, Berapakah x?
```

Variabe dalam pemrograman juga memiliki arti yang sama seperti dalam matematika. Variabel adalah tempat kita menyimpan nilai sementara. Variabel akan ada selama program dijalankan. Namun kita juga bisa menghapusnya dari memori.

a. Membuat Variabel di PHP

Pada PHP, kita membuat variabel dengan tanda dolar (\$), lalu diikuti dengan nama variabelnya serta nilai yang ingin kita simpan. Contoh:

```
<?php  
    $harga = 1000;
```

Kita baru saja membuat variabel bernama `$harga` dengan isi `1000`. Tanda sama dengan (`=`) adalah simbol atau operator yang digunakan untuk mengisi nilai ke variabel. Contoh lainnya:

```
<?php  
    $nama_barang = "Kopi C++";  
    $harga = 4000;  
    $stok = 40;
```

Variabel juga dapat diisi ulang dengan nilai yang baru. Contoh:

```
<?php  
    // membuat variabel baru  
    $stok = 40;  
  
    // mengisi ulang variabel dengan nilai baru  
    $stok = 34;
```

Dalam membuat nama variabel ada beberapa hal yang harus diperhatikan:

- Awal dari nama variabel tidak boleh menggunakan angka dan simbol, kecuali *underscore* (_).

Contoh: ✗ Salah

```
$!nama = "";  
$46rosi = "";
```

Contoh: ✓ Benar

```
$nama = "";  
$rosi46 = "";
```

- Nama variabel yang terdiri dari dua suku kata, bisa dipisah dengan *underscore* () atau menggunakan *camelCase*. Contoh:

```
$nama_barang = "Buku PHP";  
$namaPembeli = "Wachid"; // <-- menggunakan camelCase
```

- Variabel harus diisi saat pembuatannya. Bila kita tidak ingin mengisi, cukup isi dengan nilai kosong. Contoh:

```
$nama_barang = "";  
$namaPembeli = "";  
$harga = 0;
```

- Nama variabel bersifat *Case Sensitive*, artinya huruf besar dan huruf kecil dibedakan. Contoh: Tiga variabel ini akan dianggap berbeda.

```
$Belajar = "";  
$BELAJAR = "";  
$belajar = "";
```

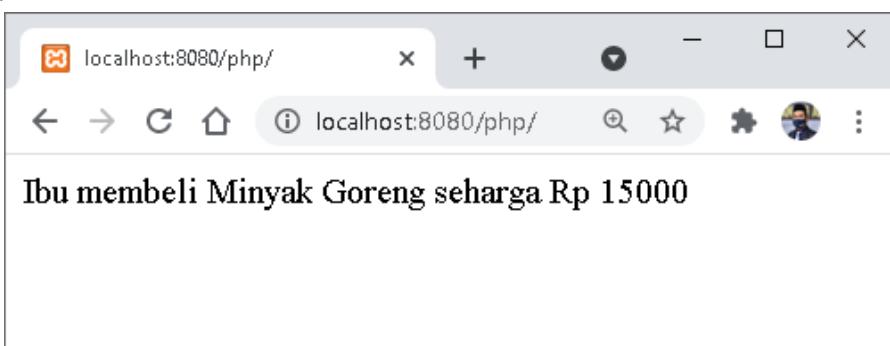
b. Mengambil Nilai dari Variabel

Setelah kita membuat variabel, biasanya akan kita gunakan pada proses berikutnya dengan mengambil nilainya. Mengambil nilai dari variabel bisa kita lakukan dengan menuliskan namanya dalam perintah `echo` maupun ekspresi yang lain. Contoh:

```
<?php
    // membuat variabel baru
    $nama_barang = "Minyak Goreng";
    $harga = 15000;

    // menampilkan isi variabel
    echo "Ibu membeli $nama_barang seharga Rp $harga";
```

Hasilnya:



Jika kita menggunakan tanda petik ganda ("") dalam `echo`, maka kita bisa menuliskan langsung nama variabelnya seperti ini:

```
$judul = "Belajar PHP dari nol sampai expert";
echo "Judul artikel: $judul";
```

Tetapi, Apabila kita menggunakan tanda petik tunggal ('), maka kita harus menggunakan titik untuk menggabungkan teks dengan variabelnya. Contoh:

```
$judul = "Tutorial PHP untuk Pemula";
echo 'Judul artikel: '.$judul;
```

Tanda titik pada perintah `echo` berfungsi untuk menggabungkan teks yang ada di dalam variabel dengan teks yang akan kita cetak.

6. Mengenal Tipe Data di PHP

Variabel yang sudah kita buat bisa kita simpan dengan berbagai jenis data. Jenis-jenis data ini disebut tipe data. Ada beberapa macam tipe data yang dapat disimpan dalam variabel:

1. Tipe data char (karakter)
2. Tipe data string (teks)
3. Tipe data integer (angka)
4. Tipe data float (pecahan)
5. Tipe data boolean
6. Tipe data objek
7. Tipe data Array
8. NULL
9. dll.

Kita tidak harus mendeklarasikan tipe datanya secara eksplisit. Karena PHP sudah mampu mengenali tipe data dari nilai yang kita berikan. Contoh:

```
<?php

// tipe data char (karakter)
$jenis_kelamin = 'L';

// tipe data string (teks)
$nama_lengkap = "Taufikqurrohman";

// tipe data integer
$umur = 33;

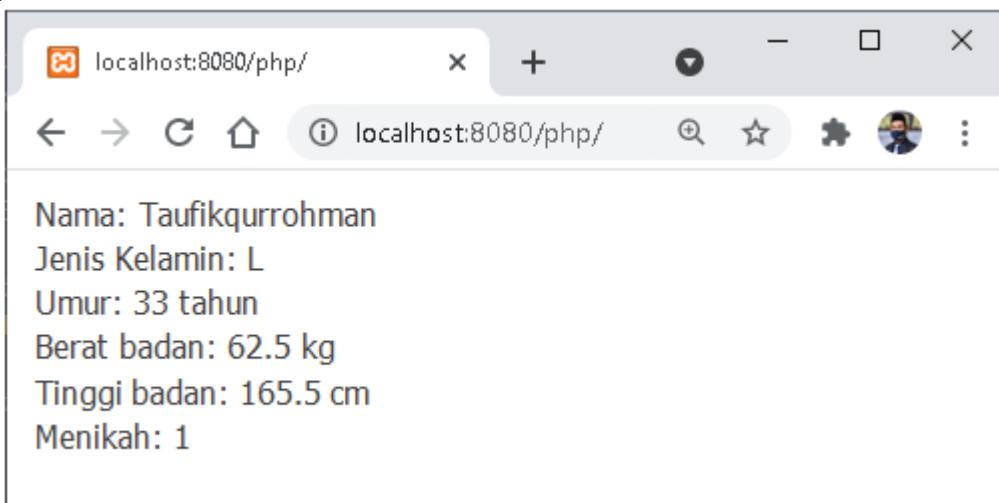
// tipe data float
$berat = 62.5;

// tipe data float
$tinggi = 165.5;

// tipe data boolean
$menikah = true;

echo "Nama: $nama_lengkap<br>";
echo "Jenis Kelamin: $jenis_kelamin<br>";
echo "Umur: $umur tahun<br>";
echo "berat badan: $berat kg<br>";
echo "tinggi badan: $tinggi cm<br>";
echo "menikah: $menikah";
```

Hasilnya:



Nama: Taufikqurrohman
Jenis Kelamin: L
Umur: 33 tahun
Berat badan: 62.5 kg
Tinggi badan: 165.5 cm
Menikah: 1

Variabel `$menikah` akan ditampilkan kosong, karena nilai `false` akan dikonversi menjadi kosong dalam string.

a. Tipe Data Char dan String di PHP

Char adalah tipe data yang terdiri dari karakter. Penulisannya diapit dengan tanda petik satu. Contoh:

```
$huruf = 'E';
```

Lalu, *String* adalah tipe data yang terdiri dari kumpulan karakter. Penulisannya diapit dengan tanda petik ganda. Contoh:

```
$alamat = "Jl. Patriot 25, Pekalongan";
```

b. Tipe Data Integer di PHP

Integer adalah tipe data angka. Penulisanya tidak menggunakan tanda petik.

Contoh:

```
$nilai = 98; // angka positif  
$poin = -31; // angka negatif
```

c. Tipe Data Float di PHP

Float adalah tipe data bilangan pecahan. Sama seperti integer, tipe data ini ditulis tanpa tanda petik. Contoh:

```
$panjang = 233.12;  
$kapasitas = 13232.12;
```

Ada juga tipe data *float* ditulis dalam notasi seperti ini: (_{E-5} artinya eksponen dari ₁₀)

```
$jarak = 1.2E-5;
```

Contoh di atas akan sama dengan 1.2×10^{-5} . Kalau kita jabarkan akan menjadi 0.000012 . Agar format *float* tidak tercetak dalam notasi _E, kita bisa gunakan fungsi *sprintf()*. Contoh:

```
echo sprintf('%f', $a);  
// batasi angka di belakang koma  
echo sprintf('.3f', $a);
```

d. Tipe data Boolean di PHP

Tipe data *boolean* adalah tipe data yang hanya bernilai *true* dan *false*. Penulisan *true* dan *false* tidak diapit dengan tanda petik. Contoh:

```
$isActive = false;  
$menikah = true;
```

e. Tipe Data Array dalam PHP

Array adalah tipe data yang berisi sekumpulan data. Contoh:

```
$minuman = array("Kopi", "Teh", "Jus Jeruk");  
$makanan = ["Nasi Goreng", "Soto", "Bubur"];
```

f. Tipe Data Objek di PHP

Tipe data objek adalah tipe data abstrak yang berisi data dan *method*. Contoh:

```
$user = new User();
```

Tipe data objek lebih sering disebut *instance* dari sebuah *class*. Pada contoh di atas `User()` adalah *class* yang di-*instance* di variabel `$user`. Setiap pembuatan *instance* harus dihului dengan kata kunci `new`.

g. Tipe Data NULL di PHP

Tipe data `NULL` adalah tipe data yang menyatakan kosong. Artinya: Jika kita mengisi variabel dengan nilai `NULL`, maka variabel tersebut akan dianggap kosong atau tidak punya nilai. Contoh:

```
$nama = NULL;
```

7. Konversi Tipe data di PHP

Apa yang akan terjadi bila kita melakukan pembagian dengan tipe data teks dengan angka seperti ini:

```
$a = 3;  
$b = "angka 10";  
  
$c = $b / $a;
```

Tentunya akan terjadi error saat kita ingin mencetak isi variabel `$c`. Karena itu, kita harus konversi dulu tipe datanya agar bisa dilakukan operasi yang lain. Pada PHP konversi tipe data bisa dilakukan dengan operator kali (*). Contoh:

```
<?php  
$foo = "1"; // mula-mula $foo dalam bentuk string (ASCII 49)  
$foo *= 2; // $foo sekarang adalah integer (2)  
$foo = $foo * 1.3; // $foo sekarang adalah float (2.6)  
$foo = 5 * "10 Little Piggies"; // $foo sekarang adalah integer (50)  
$foo = 5 * "10 Small Pigs"; // $foo sekarang adalah integer (50)  
?>
```

Selain cara ini, kita juga bisa melakukannya seperti di bahasa C:

```
<?php  
  
$a = "32";  
$a = (int) $a; // ubah nilai a menjadi integer  
$a = (float) $a; // ubah nilai a menjadi float  
$a = (string) $a; // ubah nilai a menjadi string  
  
?>
```

8. Menghapus Variabel dari Memori

Apabila ada variabel yang sudah tidak dibutuhkan lagi, maka kita bisa menghapusnya untuk meningkatkan performa program. Cara menghapus variabel di PHP dapat menggunakan fungsi `unset()`. Contoh:

```
// membuat variabel $tmp
$tmp = 2901;

// menghapus variabel $tmp
unset($tmp);

// mencoba mengakses variabel $tmp
echo $tmp;
```

Jika kita eksekusi kode di atas, maka akan terjadi error:

```
PHP Notice: Undefined variable: tmp
```

Karena variabel `$tmp` sudah kita hapus. Kenapa harus menghapus variabel? Untuk mengoptimasi program, variabel-variabel yang tidak terpakai harus dihapus agar tidak membebani memori pada server.

9. Praktikum

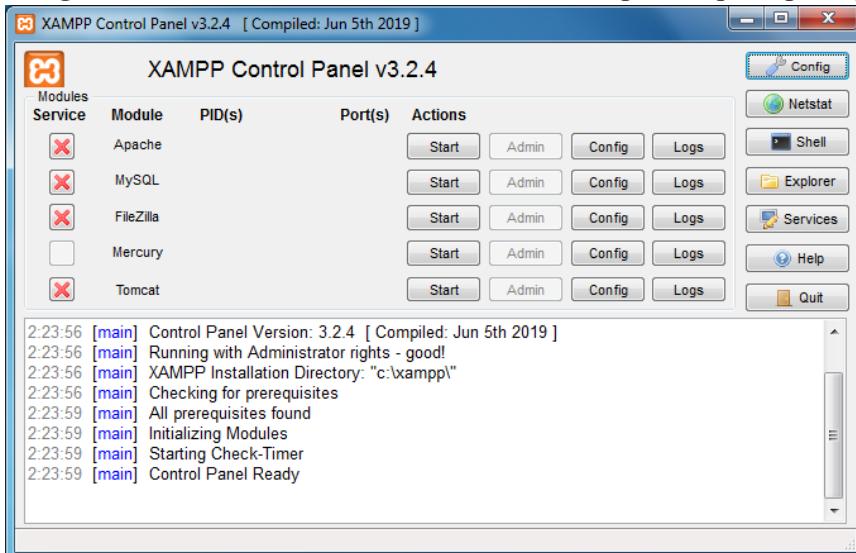
A. Alat dan Bahan

Praktikum 1 berisi materi tentang pengenalan instalasi perangkat lunak yang akan digunakan selama kuliah/praktikum, mencakup:

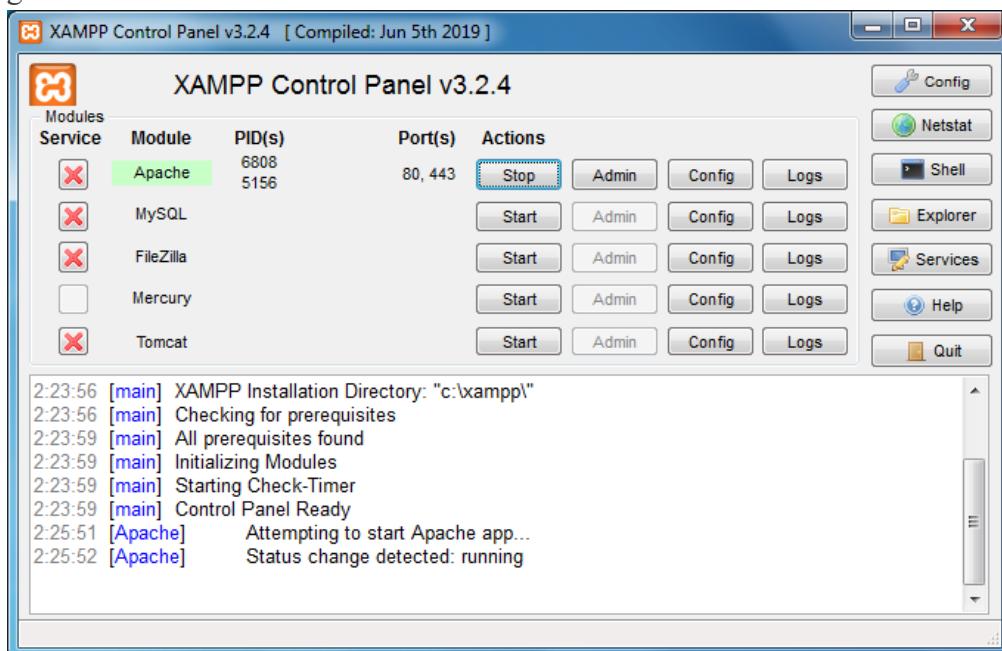
- 1) Server Lokal (XAMPP)
- 2) Teks Editor (VS Code)
- 3) Web Browser

B. Praktikum

- 1) Instal dan nyalakan XAMPP
 - a) Yang belum instal silahkan diinstal dulu
 - b) Yang sudah ada silahkan buka file XAMPP sampai tampil seperti gambar dibawah



- c) Jalankan Apache dengan cara tekan tombol start sehingga tampil seperti pada gambar berikut



- 2) Buat folder **Pertemuan 1** pada folder XAMPP/htdocs/PBW2
 3) Ketikan kode berikut dan simpan pada folder XAMPP/htdocs/PBW2/Pertemuan 1 dengan nama file index.html

```

<!DOCTYPE html>
<html>
<head>
    <title>PBW2 - Pertemuan 1</title>
</head>
<body>
    <h3>PHP</h3>
    <ol>
        <li>Variabel dan Tipe Data</li>
        <li>Operator</li>
    </ol>
</body>
</html>

```

- 4) Tambahkan kode berikut pada file index.html yang ada di folder XAMPP/htdocs/PBW2

```

<!DOCTYPE html>
<html>
<head>
    <title>PBW2</title>
</head>
<body>
    <h2>Selamat Datang di Pemrograman Berbasis Web 2</h2>
    <ol>
        <li><a href="Pertemuan 1">Pertemuan 1 - Mengenal PHP</a></li>
    </ol>
</body>
</html>

```

- 5) Ketikan kode berikut dan simpan pada folder XAMPP/htdocs/PBW2/Pertemuan 1 dengan nama file **phpvtd.php**

```
<!DOCTYPE html>
<html>
<head>
    <title>PBW2 - Pertemuan 1</title>
</head>
<body>
    <h3>PHP</h3>
    <ol>
        <li>Variabel dan Tipe Data</li>
        <li><a href="oprtr.php">Operator</a></li>
    </ol>
    <?php
        echo "Selamat datang di pemrograman berbasis web 2 <br>";
        // variabel dg Tipe data integer
        $pertemuan = 1;
        // variabel dg Tipe data String
        $materi = "Mengenal PHP";
        // variabel dg Tipe data boolean atau bool
        $proses = true;
        $belajar = $proses ? "Hadir" : "Tidak Hadir";

        echo "Pertemuan $pertemuan <br>";
        echo "Materi $materi <br>";
        echo "Kehadiran $belajar <br>";
    ?>
</body>
</html>
```

- 6) Tambahkan kode berikut pada file **index.html** yang ada di folder XAMPP/htdocs/PBW2/Pertemuan 1

```
<!DOCTYPE html>
<html>
<head>
    <title>PBW2 - Pertemuan 1</title>
</head>
<body>
    <h3>PHP</h3>
    <ol>
        <li><a href="vtd.php">Variabel dan Tipe Data</a></li>
        <li>Operator</li>
    </ol>
</body>
</html>
```

- 7) Ketikan kode berikut dan simpan pada folder XAMPP/htdocs/PBW2/Pertemuan 1 dengan nama file **operator.php**

```
<!DOCTYPE html>
<html>
<head>
    <title>PBW2 - Pertemuan 1</title>
</head>
<body>
    <h3>PHP Operator</h3>
    <ol>
        <li><a href="vtd.php">Variabel dan Tipe Data</a></li>
        <li>Operator</li>
    </ol>
    <?php
        echo    "<h4>Operator Aritmatika</h4>";
        $a = 12;
        $b = 5;
        //penjumlahan
        $c = $a + $b;
        echo "Nilai $a + $b = $c <br>";
        //pengurangan
        $c = $a - $b;
        echo "Nilai $a - $b = $c <br>";
        //perkalian
        $c = $a * $b;
        echo "Nilai $a * $b = $c <br>";
        //pembagian
        $c = $a / $b;
        echo "Nilai $a / $b = $c <br>";
        //pangkat
        $c = $a ** $b;
        echo "Nilai $a$b = $c <br> ";
        //modulus atau sisa hasil bagi
        $c = $a % $b;
        echo "Sisa hasil bagi dari nilai $a / $b = $c";
```

```
echo      "<h4>Operator Assignment</h4>";
// Operator Assignment / penugasan
// atau menetapkan nilai ke suatu variabel
$c = 5;
echo "Nilai c = $c <br>";
// kode diatas menetapkan nilai 5 ke variabel c
// melakukan operasi dan menetapkan nilai ke variabel
// secara bersamaan contoh
$c += 5;
echo "Nilai c = $c <br>";
echo      "<h4>Operator Perbandingan</h4>";
$a = 5;
$b = 3;
//Operator lebih besar
$c = $a > $b;
echo $c."<br>";
//Operator lebih kecil
$c = $a < $b;
echo "Nilai $a < $b adalah $c <br>";
//Operator lebih besar sama dengan
$c = $a >= $b;
echo "Nilai $a >= $b adalah $c <br>";
//Operator lebih kecil sama dengan
$c = $a <= $b;
echo "Nilai $a <= $b adalah $c <br>";
//Operator sama dengan
$c = $a == $b;
echo "Nilai $a = $b adalah $c <br>";
//Operator tidak sama dengan
$c = $a != $b;
echo "Nilai $a != $b adalah $c";
// nilai yang ditampilkan adalah
// nilai 1 untuk true sedangkan
// false tidak ditampilkan atau 0
// Nilai dengan tipe data boolean
// biasanya tidak untuk ditampilkan.
// Biasanya digunakan untuk pembuatan
// kondisi pada percabangan.
echo      "<h4>Operator Logika</h4>";
// membandingkan nilai boolean
$a = true;
$b = false;
$c = $a && $b;
echo "Nilai c adalah $c <br>";
$c = $a || $b;
echo "Nilai c adalah $c <br>";
$c = !$b;
echo "Nilai c adalah $c <br>";
echo      "<h4>Operator Increment dan Decrement</h4>";
$a = 0;
// Increment
$a++;
$a++;
$c++;
echo "Nilai a = $a <br>";
// Decrement
$c--;
$c--;
echo "Nilai a = $a <br>";
?>
</body>
</html>
```

- 8) Tambahkan kode berikut pada file **index.html** yang ada di folder XAMPP/htdocs/PBW2/Pertemuan 1

```
<!DOCTYPE html>
<html>
<head>
    <title>PBW2 - Pertemuan 1</title>
</head>
<body>
    <h3>PHP</h3>
    <ol>
        <li><a href="vtd.php">Variabel dan Tipe Data</a></li>
        <li><a href="operator.php">Operator</a></li>
    </ol>
</body>
</html>
```

- 9) Tampilan folder PBW2

Komputer > Local Disk (C:) > xampp > htdocs > PBW2			
Name	Date modified	Type	Size
📁 Pertemuan 1	9/28/2025 12:07 PM	File folder	
📄 index	9/28/2025 8:25 AM	Chrome HTML Do...	1 KB

- 10) Tampilan folder pertemuan 1

Komputer > Local Disk (C:) > xampp > htdocs > PBW2 > Pertemuan 1			
Name	Date modified	Type	Size
📄 index	9/28/2025 12:05 PM	Chrome HTML Do...	1 KB
📄 operator	9/29/2025 7:08 AM	PHP Source File	3 KB
📄 vtd	9/28/2025 12:03 PM	PHP Source File	1 KB

RANGKUMAN

- **Kunci sukses** belajar pemrograman adalah **sering praktik**, jangan hanya membaca saja, mulailah tulis kodemu sendiri.
- Tiga fungsi `echo()`, `print()`, `printf()` memiliki kegunaan yang sama, yaitu mencetak teks ke layar, silahkan sesuaikan dengan kebiasaan anda sendiri-sendiri yang mana yang akan digunakan.
- Kalau kita ringkas, 5 hal dasar yang harus dipahami tentang variabel dalam PHP adalah:
 - Membuat variabel
 - Mengambil nilai dari variabel
 - Mengetahui tipe data apa saja yang dapat disimpan di variabel
 - Mengkonversi tipe data
 - Menghapus variabel dari memori

TES FORMATIF

1. Silahkan coba semua source code yang ada di materi yang ada, di usahakan untuk mengetik ulang. Jika ingin copy paste silahkan pahami maksud dan tujuan dari source code yang ada.

DAFTAR PUSTAKA

Awwabiin, Salmaa. (2021). *Mengenal PHP, Fungsi dan Sintaks Dasarnya untuk Pemula*. Jakarta: NiagaHoster

Muhardian, Ahmad. (2021). *Mengenal PHP dan Memahami Struktur Dasar Sintaks Program PHP untuk Pemula*. Lombok: PetaniKode

Muhardian, Ahmad. (2021). *Mengenal Fungsi di PHP dan Memahami Variabel dan Tipe Data di PHP untuk Pemula*. Lombok: PetaniKode