

**Modul Ajar**  
**Pemrograman Web**  
**Pertemuan V Pengenalan PHP**

---

---

## 5.1 Dasar PHP

PHP (Hypertext Preprocessor) pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994 sebagai sekumpulan skrip CGI (Common Gateway Interface) sederhana yang ditulis dalam bahasa C. Awalnya, PHP digunakan untuk memantau pengunjung pada halaman web pribadinya dan dinamai "Personal Home Page Tools". Seiring dengan meningkatnya kebutuhan dan potensi dari skrip tersebut, Lerdorf mengembangkan PHP menjadi bahasa pemrograman yang lebih lengkap dan dirilis secara publik sebagai PHP/FI (Personal Home Page / Forms Interpreter) pada tahun 1995. Versi awal ini memungkinkan integrasi dengan HTML dan akses ke database sederhana, yang menjadikannya populer di kalangan pengembang web.

Seiring waktu, PHP berkembang menjadi bahasa scripting yang lebih kompleks dan kuat, dengan dukungan komunitas open-source yang besar. Pada tahun 1997, perusahaan Zend Technologies mengambil alih pengembangan inti dan merilis PHP 3, kemudian PHP 4 dan PHP 5 yang memperkenalkan fitur seperti pemrograman berorientasi objek, ekstensi yang lebih banyak, serta peningkatan performa. PHP 7 membawa peningkatan besar dalam kecepatan eksekusi dan efisiensi memori, sedangkan PHP 8 yang dirilis tahun 2020 memperkenalkan fitur modern seperti *just-in-time compilation* dan *named arguments*. Saat ini, PHP tetap menjadi salah satu bahasa pemrograman web paling populer di dunia, digunakan oleh platform besar seperti WordPress, Facebook (awal mula), dan Wikipedia.

## 5.2 Sintaks Dasar PHP

```
<?php
// Ini komentar satu baris
echo "Halo dunia!";
?>
```

Gambar 5.1 Sintaks Dasar PHP

- Semua kode PHP ditulis di dalam tag `<?php ... ?>`
- `echo` digunakan untuk mencetak output ke browser

- Komentar:
  - `//` untuk satu baris
  - `/* ... */` untuk komentar multi-baris

### 5.3 Variable dan Tipe Data

Variabel dalam PHP adalah wadah untuk menyimpan data yang dapat berubah-ubah selama program berjalan. Setiap variabel diawali dengan tanda dolar (\$) diikuti dengan nama variabel, misalnya \$nama atau \$usia. Variabel dapat menyimpan berbagai jenis data, yang disebut **tipe data**, seperti **string** (teks), **integer** (bilangan bulat), **float** (bilangan desimal), **boolean** (nilai logika true/false), **array** (kumpulan data), dan **null** (tidak bernilai). PHP secara otomatis menentukan tipe data berdasarkan nilai yang diberikan, yang disebut dengan **dynamic typing**. Memahami variabel dan tipe data sangat penting karena ini menjadi dasar dari logika dan manipulasi data dalam pemrograman PHP.

#### 5.3.1 Aturan Penamaan Variable

- Harus diawali dengan tanda \$
- Nama hanya boleh terdiri dari huruf, angka, dan underscore
- Tidak boleh diawali dengan angka
- Bersifat **case-sensitive** (\$Nama berbeda dengan \$nama)

#### 5.3.2 Tipe Data Umum pada PHP

Tipe Data	Contoh	Keterangan
String	"Halo"	Teks
Integer	10	Bilangan Bulat
Float	10.5	Bilangan Pecahan
Boolean	True, false	Logika
Array	[1,2,3]	Kumpulan Data

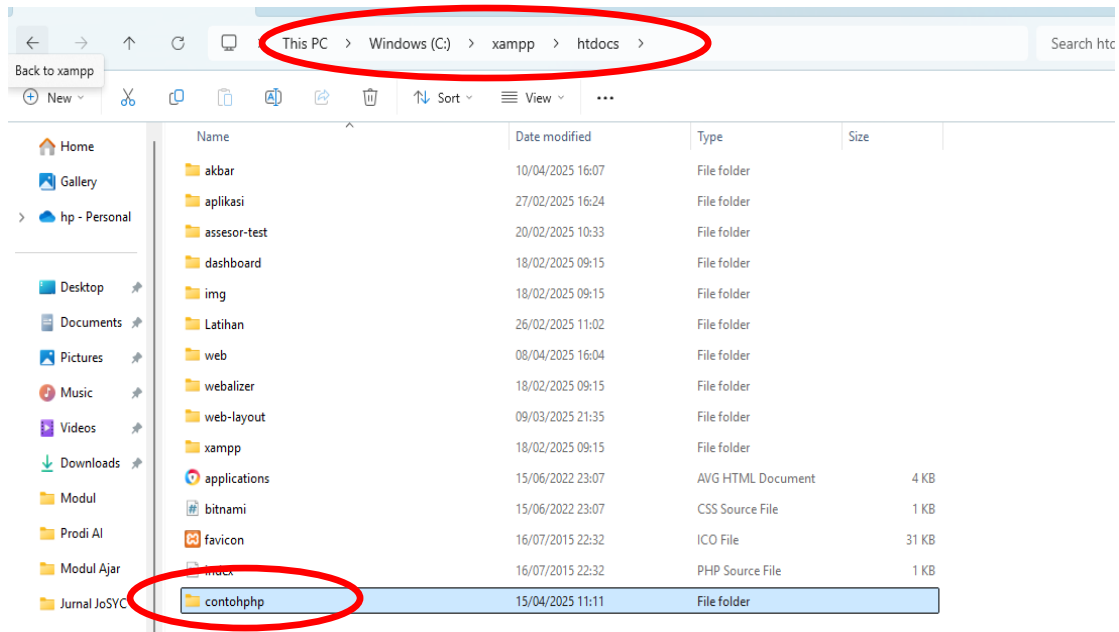
Table 1. Tipe Data Umum pada PHP

PHP memiliki beberapa tipe data dasar yang digunakan untuk menyimpan dan mengelola informasi. **String** adalah tipe data yang merepresentasikan teks, misalnya "Halo", dan biasanya digunakan untuk menyimpan nama, alamat, atau pesan. **Integer** menyimpan bilangan bulat seperti 10, tanpa desimal, cocok untuk data seperti umur atau jumlah barang. **Float** (atau double) menyimpan bilangan pecahan seperti 10.5, digunakan untuk nilai dengan

koma seperti harga atau berat. **Boolean** memiliki dua nilai: true atau false, dan digunakan untuk logika, seperti menentukan apakah pengguna sudah login atau belum. Terakhir, **Array** adalah tipe data yang menyimpan kumpulan nilai, seperti [1, 2, 3], dan sangat berguna untuk mengelola data dalam jumlah banyak seperti daftar produk atau nilai-nilai ujian.

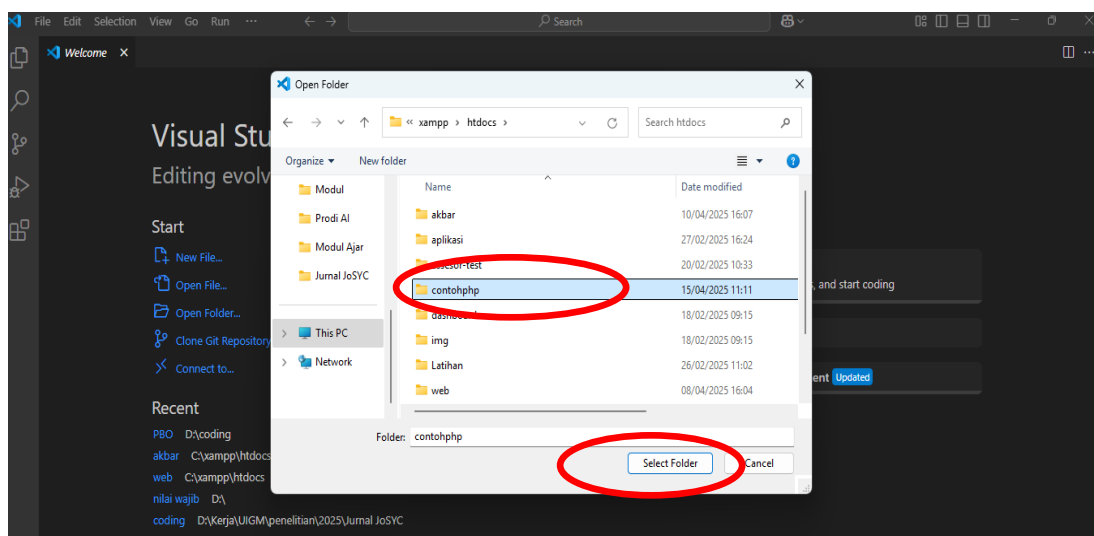
### Contoh Penggunaan

1. Buat folder baru pada folder htdocs, beri nama **contohphp**



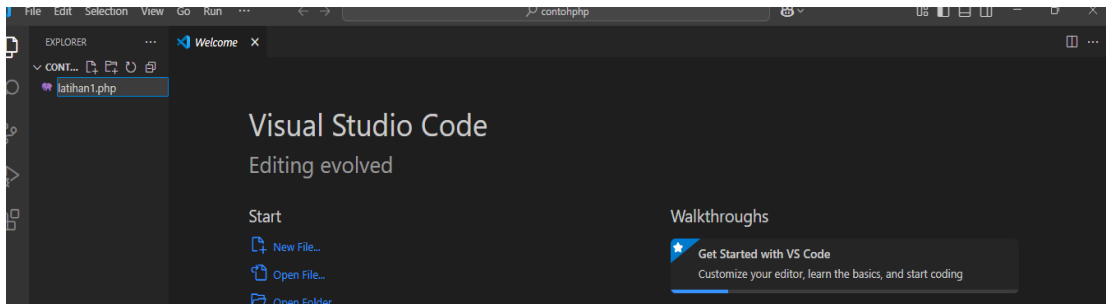
Gambar 5.2 Folder contohphp

2. Buka Visual Studio code dan buka folder, contohphp yang barusan dibuat.



Gambar 5.3 Select Folder

3. Buat file baru, beri nama latihan1.php



Gambar 5.4 File latihan1.php

4. Masukkan kode berikut

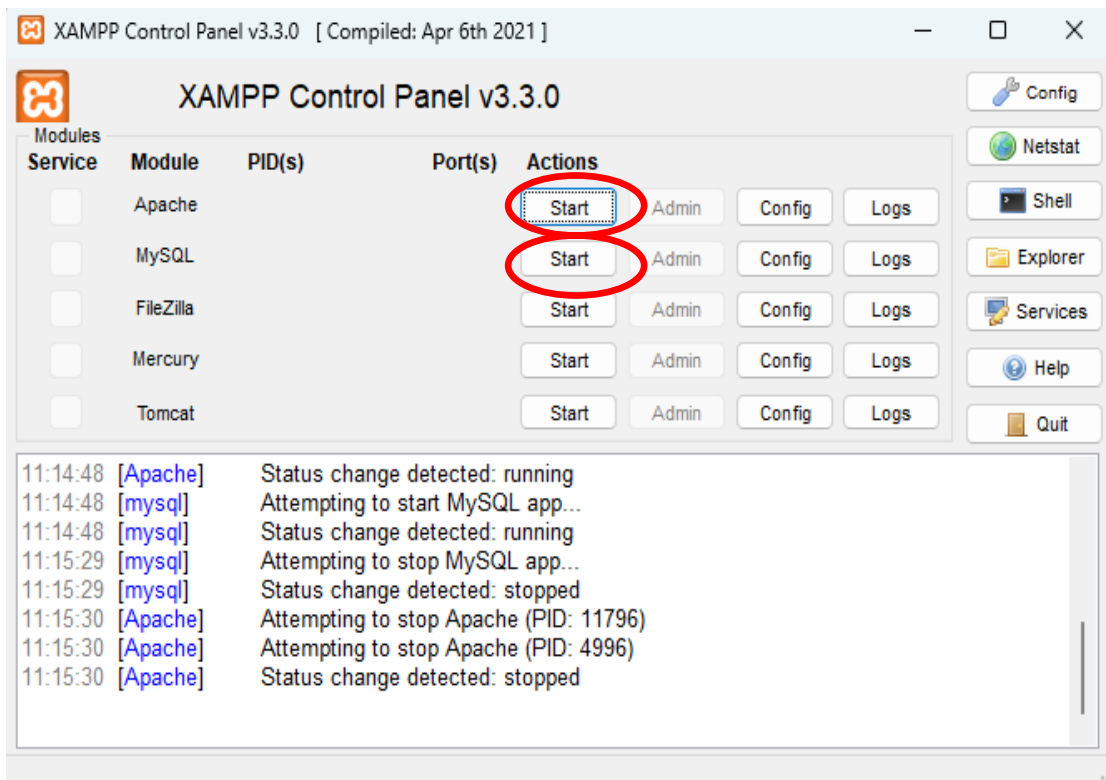
```

latihan1.php
1  <?php
2  $nama = "Dani";
3  $umur = 25;
4  $isLulus = false;
5
6  echo "Nama: $nama<br>";
7  echo "Umur: $umur<br>";
8  echo "Status Lulus: " . ($isLulus ? "Ya" : "Tidak");
9  ?>
10

```

Gambar 5.5 Kode latihan1.php

5. Lalu jalankan xampp, dan start tombol apache dan mysql



Gambar 5.6 Jalankan xampp

6. Lalu arahkan browser pada Alamat localhost/contohphp/latihan1.php



Gambar 5.7 hasil localhost

#### 5.4 Kontrol Program IF

Struktur kontrol **if** dalam PHP digunakan untuk membuat keputusan berdasarkan kondisi tertentu. Dengan menggunakan pernyataan if, program dapat menjalankan blok kode tertentu hanya jika kondisi yang diberikan bernilai **true**. Struktur dasar if dapat diperluas dengan else untuk menangani kondisi alternatif, dan elseif untuk mengevaluasi beberapa kondisi secara berurutan. Misalnya, `if ($nilai >= 80)` akan mengeksekusi blok kode jika nilai lebih besar atau sama dengan 80. Jika tidak, program bisa berlanjut ke blok elseif atau else. Struktur ini sangat penting untuk mengatur alur logika program agar dapat merespons input atau situasi yang berbeda secara dinamis. Di dalam pemrograman PHP terdapat 4 jenis percabangan If diantaranya adalah 1. If sederhana, 2. If, ... Else, 3. If, ...else if, else..., 4. If dengan operator logika (AND dan OR).

##### 5.4.1 If Sederhana

Php akan menjalankan perintah hanya jika kondisi bernilai true. Seperti, jika hari hujan maka berteduh. Kondisi berteduh akan dilakukan ketika hari hujan. Contoh

```

latihanif.php
1  <?php
2  $umur=20;
3
4  if ($umur>=18)
5  {
6      echo "Anda sudah cukup umur untuk membuat KTP";
7  }
8  ?>

```

Gambar 5.8 If sederhana

Tulisan “Anda sudah cukup umur untuk membuat KTP” akan dieksekusi Ketika \$umur lebih besar atau sama dengan 18. Pada kode diatas, tulisan “Anda sudah cukup umur untuk membuat KTP” karena \$umur adalah 20.

#### 5.4.2 If... Else

PHP akan menjalankan suatu blok kode jika kondisi benar dan blok lain jika kondisi salah

```

latihanif.php
1  <?php
2  $nilai = 65;
3
4  if ($nilai >= 75) {
5      echo "Selamat, Anda lulus!";
6  } else {
7      echo "Maaf, Anda belum lulus.";
8  }
9  ?>

```

Tidak Dieksekusi

Dieksekusi

Gambar 5.9 If Else

Tulisan “Maaf, Anda belum lulus” akan dieksekusi karena \$nilai adalah 65, untuk lulus \$nilai harus lebih besar atau sama dengan 75.

#### 5.4.3 If...ElseIF...Else

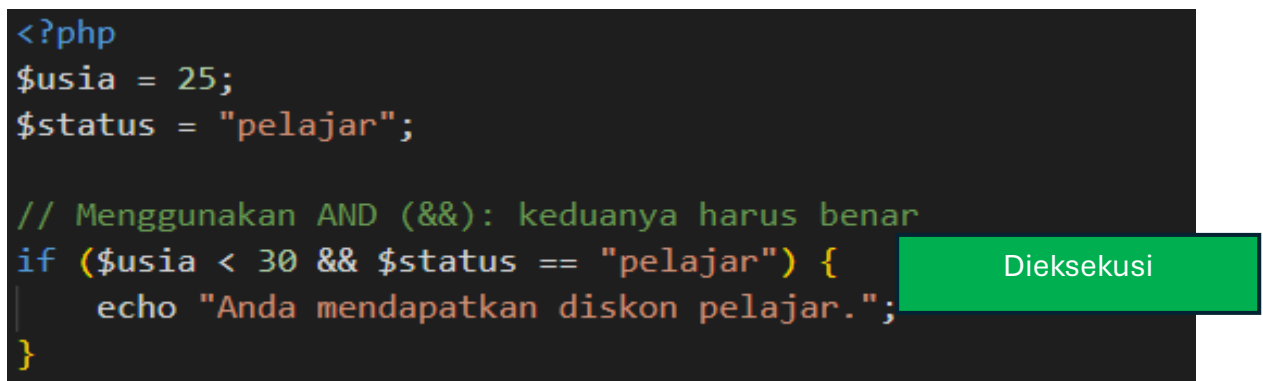
Menangani beberapa kondisi berbeda secara berurutan



Gambar 5.10 contoh If elseif, else

#### 5.4.4 If dengan operator Logika (AND dan OR)

Menggabungkan beberapa kondisi dalam satu logika




Gambar 5.11 Logika IF and Or

Kode diatas akan dieksekusi karena usia adalah 25(dibawah 30) dan status pelajar sehingga tulisan “Anda mendapatkan diskon pelajar”.

```
<?php
$usia = 25;
$status = "pelajar";

// Menggunakan OR (||): salah satu kondisi cukup benar
if ($usia < 18 || $status == "pelajar") {
    echo "<br>Anda berhak mengikuti program khusus.";
}
?>
```



Gambar 5.12 Kode If dengan Or

Kode diatas akan dieksekusi karena walaupun usia diatas 18 tahun, tapi status adalah pelajar. Sehingga tulisan “Anda berhak mengikuti program khusus”.

### 5.5 Switch Case

Struktur **switch...case** dalam PHP adalah salah satu bentuk kontrol alur program yang digunakan untuk menggantikan banyak pernyataan if...elseif yang memeriksa nilai dari satu variabel yang sama. Switch bekerja dengan membandingkan nilai sebuah variabel terhadap daftar kemungkinan nilai yang ditentukan oleh case. Jika ditemukan case yang cocok, maka blok kode di dalamnya akan dieksekusi hingga perintah break ditemukan. Jika tidak ada case yang cocok, maka blok default akan dijalankan sebagai alternatif.

Keunggulan utama dari switch adalah membuat kode menjadi lebih bersih, mudah dibaca, dan lebih terstruktur, terutama saat harus mengevaluasi banyak kemungkinan nilai dari satu variabel. Misalnya, jika kita ingin menampilkan pesan berbeda berdasarkan hari, menggunakan switch jauh lebih efisien dan rapi dibanding menggunakan banyak if...elseif. Dengan begitu, switch sangat cocok digunakan dalam situasi di mana nilai yang diperiksa terbatas dan eksplisit, seperti angka menu, pilihan user, atau nama hari.

Namun, perlu diperhatikan bahwa switch dalam PHP hanya cocok digunakan saat membandingkan nilai **secara tepat (==)**, bukan untuk kondisi logika kompleks seperti rentang nilai atau ekspresi logika. Jika logika yang dibutuhkan melibatkan operator perbandingan seperti >, <, atau kondisi AND/OR, maka lebih baik tetap menggunakan struktur if...elseif...else. Oleh karena itu, pemilihan antara switch atau if tergantung pada kompleksitas logika yang ingin diterapkan.



```

<?php
$hari = "Senin";

switch ($hari) {
    case "Senin":
        echo "Hari ini adalah awal pekan.";
        break;
    case "Selasa":
        echo "Hari ini adalah hari kedua kerja.";
        break;
    case "Rabu":
        echo "Hari ini adalah pertengahan minggu.";
        break;
    case "Kamis":
        echo "Hari ini semakin dekat ke akhir pekan.";
        break;
    case "Jumat":
        echo "Hari ini adalah hari terakhir kerja.";
        break;
    case "Sabtu":
    case "Minggu":
        echo "Selamat akhir pekan!";
        break;
    default:
        echo "Hari tidak dikenali.";
}
?>

```

Gambar 5.13 Kode Switch

- switch (\$hari) memeriksa nilai variabel \$hari.
- case mencocokkan nilai tertentu. Jika cocok, perintah di dalamnya akan dijalankan.
- break digunakan untuk menghentikan eksekusi setelah satu case cocok.
- default akan dijalankan jika tidak ada nilai yang cocok sama sekali.

## 5.5 Perulangan

Perulangan dalam PHP adalah mekanisme yang memungkinkan suatu blok kode dijalankan berulang kali selama kondisi tertentu masih terpenuhi. Struktur perulangan sangat berguna saat kita ingin melakukan proses yang sama secara berulang, seperti menampilkan data dalam daftar, menghitung jumlah, atau memproses input pengguna dalam jumlah banyak. PHP menyediakan beberapa jenis perulangan, seperti for, while, dan do...while. Perulangan for biasanya digunakan saat jumlah pengulangan sudah diketahui, sedangkan while dan do...while cocok digunakan ketika pengulangan tergantung pada kondisi yang baru diketahui selama

program berjalan. Dengan memahami perulangan, kita dapat membuat program menjadi lebih efisien dan dinamis.

#### 5.5.1 Perulangan For

Digunakan saat jumlah pengulangan sudah diketahui.

```
<?php
for ($i = 1; $i <= 5; $i++) {
    echo "Perulangan ke-$i<br>";
}
?>
```

Gambar 5.14 Kode perulangan for

Penjelasan : Mulai dari \$i=1, terus berulang selama \$i<=5, dan setiap kali \$i akan ditambah 1 sampai dengan \$i lebih dari 5.

#### 5.5.2 Perulangan While

Digunakan saat pengulangan bergantung pada kondisi

```
<?php
$i = 1;
while ($i <= 5) {
    echo "Perulangan ke-$i<br>";
    $i++;
}
?>
```

Penjelasan : Selama kondisi \$i<=5 bernilai true, perulangan akan terus berjalan.

#### 5.5.3 Perulangan do .. while

Menjalankan blok kode **setidaknya sekali**, lalu terus berulang selama kondisi terpenuhi.

```
<?php
$i = 1;
do {
    echo "Perulangan ke-$i<br>";
    $i++;
} while ($i <= 5);
?>
```

Penjelasan kode : kode dalam block **do** akan dijalankan terlebih dahulu, baru kemudian kondisi dicek.

#### Tugas Mandiri

1. Buatlah program PHP menghitung nilai mahasiswa, lalu tampilkan keterangan:
  - A jika nilai  $\geq 85$
  - B jika nilai  $\geq 75$
  - C jika nilai  $\geq 65$
  - D jika nilai  $< 65$
2. Buat program PHP menggunakan Switch-Case yang menampilkan menu makanan.
  - 1. Nasi Goreng
  - 2. Mie Ayam
  - 3. Soto Ayam
  - 4. Sate Ayam
3. Buat program PHP yang mencetak angka genap dari 2 hingga 20 menggunakan perulangan for