

MODUL 5

PHP

5.1 Tujuan Praktikum

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Memahami apa itu PHP.
2. Memahami dasar-dasar PHP dan menggunakan sintaks-sintaks PHP.
3. Mengetahui berbagai macam fungsi pada PHP lanjutan.
4. Membuat halaman *login* sebuah situs web.

5.2 Alat & Bahan

Alat & Bahan yang digunakan adalah sebagai berikut:

1. Laptop (Windows, Linux, atau MacOS)
2. PHP 8.x
3. Perangkat lunak XAMPP
4. Editor teks (VS Code)

5.3 Dasar Teori

5.3.1 XAMPP

XAMPP adalah perangkat lunak *open-source* yang digunakan untuk membuat server lokal di komputer. Perangkat lunak ini membantu pengembang dan desainer web menguji proyek sebelum dipublikasikan ke internet. XAMPP terdiri dari beberapa komponen utama:

- **X (Cross Platform):** Dapat dijalankan di berbagai sistem operasi.
- **A (Apache):** Web server yang berfungsi untuk mengelola dan menyajikan halaman web.
- **M (MySQL/MariaDB):** Sistem manajemen basis data untuk menyimpan dan mengolah data.
- **P (PHP):** Bahasa pemrograman yang digunakan untuk membangun aplikasi web dinamis.
- **P (Perl):** Bahasa pemrograman tambahan yang digunakan untuk kebutuhan *scripting*.

5.3.2 PHP (Hypertext Preprocessor)

PHP adalah bahasa skrip *open source* yang dirancang khusus untuk pengembangan web dan dapat disematkan dalam HTML. Dikembangkan awalnya oleh Rasmus Lerdorf pada tahun 1994, PHP kini dikelola oleh The PHP Group dan menjadi salah satu bahasa pemrograman *server-side* paling populer di dunia. PHP memiliki beberapa fungsi utama, yaitu:

1. Memproses dan menampilkan konten dinamis dalam sebuah situs web.
2. Mengelola data dari formulir dan sesi pengguna.
3. Berinteraksi dengan *database* seperti MySQL dan PostgreSQL.
4. Menghasilkan dan mengedit berkas seperti gambar, PDF, dan XML.

5.3.3 Sejarah PHP

PHP pertama kali muncul sebagai sekumpulan skrip Perl yang dibuat oleh Rasmus Lerdorf untuk memantau kunjungan ke resume onlinenya, yang disebut "Personal Home Page Tools" (PHP Tools). Versi publik pertama dirilis pada 1995 (PHP/FI). Pada 1997, Zeev Suraski dan Andi Gutmans menulis ulang *parser* PHP, menciptakan PHP 3.0, yang menjadi fondasi bahasa modern. Versi utama terkini adalah PHP 8.x, dirilis dengan fitur seperti *JIT compiler* dan *union types*.

5.3.4 Syarat Menjalankan PHP

1. Web Server: Perangkat lunak yang menangani permintaan HTTP dan menjalankan kode PHP. Contoh: Apache (www.apache.org).
2. PHP Interpreter: Diperlukan untuk menjalankan skrip PHP, bisa dikonfigurasi sebagai modul Apache atau aplikasi CGI (www.php.net).
3. Database Server: Digunakan untuk menyimpan dan mengelola data. PHP sering digunakan dengan MySQL/MariaDB (www.mysql.com).

5.4 Sintaks dan Semantik

5.4.1 Delimiter

Kode PHP diapit oleh tag khusus:

```
1  <?php
2  |  // Kode PHP
3  ?>
4  |  _____
```

5.4.2 Komentar

PHP mendukung tiga jenis komentar:

```
1 // Komentar satu baris
2
3 # Alternatif komentar satu baris
4
5 /*
6 | Komentar
7 | multi-baris
8 */
```

5.5 Variabel dan Tipe Data

5.5.1 Variabel

Variabel dideklarasikan dengan prefix “\$” dan bersifat *case-sensitive*.



```
$auraPradnya = "Hello Friend"; // $auraPradnya adalah variabel
```

5.5.2 Tipe Data Saklar

1. **String:** Teks, diapit tanda kutip ("..." atau '...')
2. **Integer:** Bilangan bulat (-3, 42)
3. **Float:** Bilangan desimal (3.14, -0.001)
4. **Boolean:** *true* atau *false*

```
1 <?php
2 // String
3 $teks = "Halo, Dunia!";
4
5 // Integer
6 $angka = 42;
7
8 // Float
9 $desimal = 3.14;
10
11 // Boolean
12 $isActive = true;
13
14 echo $teks;
15 ?>
```

5.5.3 Tipe Data Komposit

1. Array Indeks: Menggunakan indeks numerik.

```

1 <?php
2 $array = [1, 2, "tiga"];
3 echo $array[2]; // output tiga
4 ?>

```

2. Array Asosiatif: Menggunakan kunci (*key*) yang dapat berupa string.

```

1 <?php
2 $user = ["nama" => "Fauzan", "usia" => 19];
3 echo $user["nama"]; // Output Fauzan
4 ?>
5

```

3. Object: Tipe data yang dibuat dari sebuah kelas.

```

1 <?php
   1 reference | 0 implementations
2 class Kucing {
   2 references
3     public $nama;
4 }
5 $kucingSaya = new Kucing();
6 $kucingSaya->nama = "Momo";
7 echo $kucingSaya->nama; // Output: Momo

```

5.5.4 Tipe Data Khusus

1. **NULL**: Dalam PHP, sebuah variabel yang diatur ke NULL berarti variabel tersebut tidak memiliki nilai yang terdefinisi. NULL sering digunakan untuk menandakan bahwa variabel belum diinisialisasi atau tidak memiliki nilai yang valid.
2. **Resource**: Tipe data ini merupakan referensi ke sumber eksternal, seperti koneksi ke *database*, file, atau layanan jaringan. *Resource* digunakan untuk mengelola dan berinteraksi dengan sumber daya di luar memori program, memungkinkan PHP untuk melakukan operasi yang memerlukan akses ke data atau layanan eksternal. *Resource* biasanya dihasilkan oleh fungsi tertentu dan tidak dapat diubah secara langsung.

5.6 Operator

5.6.1 Aritmatika

Operator	Deskripsi	Contoh	Output
+	Penjumlahan	5 + 3	8
-	Pengurangan	5 - 3	2
*	Perkalian	5 * 3	15

/	Pembagian	6 / 3	2
%	Modulus (Sisa Bagi)	7 % 3	1

5.6.2 Perbandingan

Operator	Deskripsi	Contoh	Output
==	Sama dengan	5 == "5"	True
===	Identik (sama nilai & tipe data)	5 === "5"	False
!=	Tidak sama	5 != 3	True
!==	Tidak identik	5 !== "5"	True
>	Lebih besar	10 > 5	True
<	Lebih kecil	3 < 5	True
>=	Lebih besar atau sama dengan	5 >= 5	True
<=	Lebih kecil atau sama dengan	4 <= 5	True

5.6.3 Logika

- && (AND): Mengembalikan nilai *true* jika kedua operandnya *true*.
- || (OR): Mengembalikan nilai *true* jika salah satu atau kedua operandnya *true*.
- ! (NOT): Membalikkan nilai boolean dari operandnya, mengembalikan *true* jika operandnya *false*, dan sebaliknya.

5.6.4 Penggabungan String

Menggunakan operator "." (titik):

```

1 <?php
2 $nama = "John" . " " . "Doe";
3 echo $nama; // Output: John Doe
4 ?>
```

5.7 Struktur Kontrol

5.7.1 if-elseif-else:

Struktur percabangan yang digunakan untuk menjalankan kode berdasarkan suatu kondisi. Jika kondisi dalam "if" terpenuhi (*true*), maka kode di dalamnya akan dijalankan. Jika tidak, maka kode dalam "else" akan dieksekusi sebagai alternatif. Contoh:

```

1  <?php
2  $x = -5;
3  if ($x > 0) {
4  |     echo "Positif";
5  } elseif ($x == 0) {
6  |     echo "Nol";
7  } else {
8  |     echo "Negatif"; // output Negatif
9  }
10 ?>

```

5.7.2 Switch

Struktur percabangan yang digunakan untuk memeriksa nilai sebuah variabel dan menjalankan perintah berdasarkan kondisi yang sesuai. *Switch* lebih efisien dibandingkan “if-else” ketika ada banyak kemungkinan nilai yang harus diperiksa. Contoh:

```

1  <?php
2  $hari = "Senin"; // output Hari libur
3  switch ($hari) {
4  |     case 'Senin':
5  |     case 'Selasa':
6  |     case 'Rabu':
7  |     case 'Kamis':
8  |     case 'Jumat':
9  |         echo "Hari kerja";
10 |         break;
11 |     default:
12 |         echo "Hari libur";
13 }
14 ?>

```

5.8 Perulangan

5.8.1 For

Digunakan ketika jumlah perulangan sudah diketahui. Contoh:

```

1  <?php
2  for ($i = 0; $i < 5; $i++) { //output 0 1 2 3 4
3  |     echo $i . " ";
4  }
5  ?>

```

5.8.2 While

Digunakan ketika jumlah perulangan tidak diketahui dan bergantung pada kondisi tertentu. Contoh:

```

1 <?php
2 $x = 0;
3 while ($x < 3) {
4     echo "Angka: $x "; // output Angka: 0 Angka: 1 Angka: 2
5     $x++;
6 }
7 ?>

```

5.8.3 Foreach

Digunakan khusus untuk mengulang setiap elemen dalam *array*. Contoh:

```

1 <?php
2 $buah = ["Apel", "Mangga", "Jeruk"];
3 foreach ($buah as $item) {
4     echo $item . " "; // output Apel Mangga Jeruk
5 }
6 ?>

```

5.9 Fungsi

5.9.1 Deklarasi Fungsi

Deklarasi fungsi dalam PHP adalah pernyataan yang digunakan untuk mendefinisikan sebuah *function*, yang merupakan blok kode yang dapat dipanggil untuk menjalankan serangkaian instruksi tertentu.

```

1 <?php
2 1 reference
3 function hitungLuas($panjang, $lebar): float|int {
4     return $panjang * $lebar;
5 }
6 $panjang = 5;
7 $lebar = 3;
8 echo "Panjang: $panjang <br>";
9 echo "Lebar: $lebar <br>";
10 echo "Luas: " . hitungLuas(panjang: $panjang, lebar: $lebar);
11 ?>

```

Parameter *default*:

```

1 <?php
2 2 references
3 function salam($nama = "Tam"): string {
4     return "Halo, $nama!";
5 }
6 echo salam();
7 echo "<br>";
8 echo salam(nama: "Fauzan");
9 ?>

```

5.9.2 Fungsi Variadik

Menerima jumlah argumen variabel menggunakan "...". Contoh:

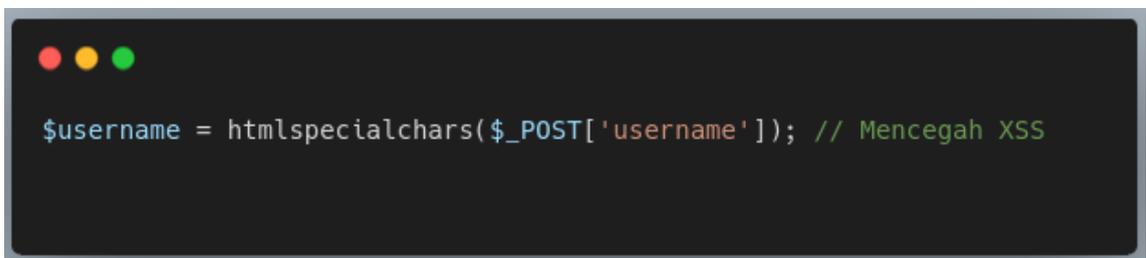
```
1  <?php
   2 references
2  function jumlahkan(...$angka): float|int {
3      |   return array_sum(array: $angka);
4  }
5  echo jumlahkan(angka: 1, 2, 3);
6  echo "<br>";
7  echo jumlahkan(angka: 10, 20, 30, 40);
8  ?>
```

5.10 Penanganan Formulir

PHP mengakses data formulir melalui *superglobals*:

- `$_GET`: Data dari metode GET digunakan untuk mengirimkan data melalui URL. Data yang dikirimkan akan terlihat di alamat URL setelah tanda "?". Metode ini cocok untuk pencarian atau navigasi, tetapi kurang aman untuk data sensitif karena bisa terlihat dan disimpan di riwayat *browser*.
- `$_POST`: Data dari metode POST digunakan untuk mengirimkan data yang diinputkan pengguna ke server tanpa menampilkannya di URL. Metode ini lebih aman dibandingkan GET karena data tidak terlihat oleh pengguna. POST biasanya digunakan untuk login, pendaftaran, atau pengiriman data yang bersifat rahasia.

Contoh:

A screenshot of a code editor with a dark background and light-colored text. The code is a single line of PHP: `$username = htmlspecialchars($_POST['username']); // Mencegah XSS`. The text is color-coded: `$username` is blue, `=` is black, `htmlspecialchars` is green, `(` is black, `$_POST` is red, `['username']` is black, `);` is black, `//` is green, and `Mencegah XSS` is green. There are three colored circles (red, yellow, green) in the top left corner of the editor window.

5.11 Interaksi Database (MySQLi)

5.11.1 Koneksi ke MySQL

Koneksi ke MySQL diperlukan agar PHP dapat berkomunikasi dengan *database*. Dalam PHP, koneksi bisa dibuat menggunakan MySQLi.

```
$koneksi = new mysqli("localhost", "user", "password", "database");
if ($koneksi->connect_error) {
    die("Koneksi gagal: " . $koneksi->connect_error);
}
```

5.11.2 Query Aman dengan Prepared Statements

Prepared Statements adalah cara aman untuk menjalankan *query* dengan cara memisahkan perintah SQL dari data yang dikirimkan pengguna.

```
$stmt = $koneksi->prepare("INSERT INTO users (email) VALUES (?)");
$stmt->bind_param("s", $email); // 's' = tipe string
$stmt->execute();
```

5.12 Penanganan Kesalahan

5.12.1 Exception Handling

Exception Handling adalah mekanisme dalam pemrograman yang memungkinkan pengembang untuk menangani kesalahan atau kondisi tidak terduga yang terjadi selama eksekusi program.

```
try {
    if (!file_exists("file.txt")) {
        throw new Exception("File tidak ditemukan");
    }
} catch (Exception $e) {
    echo "Error: " . $e->getMessage();
}
```

5.12.2 Error Reporting

Dalam PHP, fungsi `error_reporting(E_ALL);` digunakan untuk menampilkan semua jenis kesalahan, sementara `ini_set('display_error', 1);` mengatur agar kesalahan ditampilkan di layar, yang berguna untuk tujuan *debugging* selama pengembangan.

```
error_reporting(E_ALL); // Menampilkan semua error
ini_set('display_errors', 1);
```

5.13 Keamanan

1. Validasi Input:

```
$email = filter_var($_POST['email'],
FILTER_VALIDATE_EMAIL);
```

2. Gunakan *prepared statements* (MySQLi atau PDO) untuk mencegah *SQL Injection*.
3. Mencegah XSS:

```
echo htmlspecialchars($input_pengguna);
```

5.14 Manajemen Sesi

Manajemen sesi di PHP adalah mekanisme yang digunakan untuk menyimpan dan mengelola data pengguna selama interaksi mereka dengan aplikasi web. Sesi memungkinkan penyimpanan informasi yang dapat diakses di berbagai halaman tanpa perlu mengirimkan data tersebut melalui URL atau formulir.

```
session_start();
$_SESSION['pengguna'] = "rical"; // Menyimpan data sesi
session_unset(); // Menghapus semua variabel sesi
session_destroy(); // Menghancurkan sesi
```

Panduan Praktikum

Membuat Sistem Login Sederhana Menggunakan PHP

```
Struktur file:

/login-system/
├── config.php
├── register.php
├── login.php
├── dashboard.php
├── logout.php
└── style.css
```

1. Pastikan XAMPP sudah berjalan
2. Jalankan query berikut di phpMyAdmin atau CLI MySQL untuk membuat database:

```
CREATE DATABASE login_system;
USE login_system;

CREATE TABLE users (
  id INT PRIMARY KEY AUTO_INCREMENT,
  username VARCHAR(50) UNIQUE NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  password VARCHAR(255) NOT NULL
);
```

3. Simpan semua file di folder htdocs/login-system
4. Akses melalui browser:
 - Registrasi: <http://localhost/login-system/register.php>
 - Login: <http://localhost/login-system/login.php>

Catatan: Kode lengkap akan ditampilkan saat sesi praktikum.