**POST TEST**

**PEMROGRAMAN WEB 1**

**MODUL 8**



NAMA : Aef Nur Hidayah

NIM : 20230910054

KELAS : SINFC – 2023 - 03

FAKULTAS ILMU KOMPUTER

UNIVERSITAS KUNINGAN

TAHUN 2024

# **D. PRE-TEST**

1. Apa itu PHP dan apa fungsinya dalam pengembangan web?

PHP (Hypertext Preprocessor) adalah bahasa pemrograman sisi server yang dirancang khusus untuk pengembangan web. PHP digunakan untuk membuat situs web dinamis dan interaktif dengan memungkinkan pengembang menjalankan logika di server sebelum mengirimkan hasilnya ke browser pengguna.

Fungsi Utama PHP dalam Pengembangan Web:

1. Pengolahan Data Formulir  
   PHP dapat memproses data yang dikirimkan melalui formulir (form), seperti login, pendaftaran, atau masukan pengguna lainnya. Contoh: memvalidasi data, menyimpan ke database, atau menampilkan kembali informasi.
2. Manajemen Basis Data  
   PHP sering digunakan untuk berinteraksi dengan database seperti MySQL, PostgreSQL, atau MongoDB. Hal ini memungkinkan pengembang untuk:
   1. Menyimpan dan mengambil data.
   2. Menampilkan data secara dinamis.
   3. Melakukan operasi seperti pembaruan, penghapusan, dan pencarian data.
3. Pembuatan Konten Dinamis  
   PHP memungkinkan situs web menampilkan konten yang berubah berdasarkan konteks atau masukan pengguna. Misalnya, menampilkan halaman pengguna yang berbeda tergantung pada apakah pengguna telah login.
4. Manajemen Session dan Cookies  
   PHP membantu melacak data pengguna antara berbagai halaman, seperti informasi login, keranjang belanja, atau preferensi pengguna.
5. Integrasi dengan Layanan Lain  
   PHP dapat mengakses API eksternal, seperti API pembayaran, API media sosial, atau API cuaca, untuk menambah fungsionalitas pada situs web.
6. Pembangunan CMS (Content Management System)  
   Banyak CMS populer seperti WordPress, Joomla, dan Drupal dibangun menggunakan PHP. Dengan ini, pengguna dapat mengelola konten situs tanpa harus menulis kode.
7. Membuat Aplikasi Web Full-Stack  
   PHP dapat digunakan untuk membuat aplikasi berbasis web yang memiliki frontend dan backend, sering kali digabungkan dengan teknologi HTML, CSS, dan JavaScript.
8. Bagaimana cara mendeklarasikan dan menginisialisasi variable dalam PHP? Berikan satu contoh.

Dalam PHP, variabel dideklarasikan dengan diawali tanda dolar ($), diikuti dengan nama variabel yang valid. Variabel dapat langsung diinisialisasi dengan nilai tertentu. PHP adalah bahasa yang **loosely typed**, sehingga Anda tidak perlu menentukan tipe data variabel saat mendeklarasikannya.

Aturan Penamaan Variabel:

1. Harus diawali dengan tanda $ diikuti huruf atau underscore (\_).
2. Tidak boleh diawali dengan angka.
3. Hanya boleh mengandung huruf, angka, dan underscore.
4. Bersifat case-sensitive (huruf besar dan kecil dibedakan).

|  |
| --- |
| $nama = "Aef Nur Hidayah"; // String  $usia = 19; // Integer  $tinggi = 1.69; // Float  $status = true; // Boolean |

1. Sebutkan dan jelaskan tiga tipe data yang umum digunakan dalam PHP.

Ada tiga tipe data yang paling umum digunakan yaitu :

* 1. Integer (Bilangan Bulat)

Tipe data ini digunakan untuk menyimpan angka bulat, baik positif maupun negative. Bilangan ini tidak mengandung angka decimal. Nilai integer bergantung pada platform, tetapi umumnya antara -2^31 dan 2^31-1 pada sistem 32-bit, dan antara -2^63 dan 2^63-1 pada sistem 64-bit.

* 1. String (Text)

Tipe data string digunakan untuk menyimpan teks, seperti kata, kalimat, atau kombinasi karakter. String dapat ditulis dalam tanda kutip tunggal (') atau ganda (").String bisa berisi huruf, angka, atau simbol lainnya. PHP mendukung operasi seperti concatenation (penggabungan string) menggunakan operator titik (.).

* 1. Boolean (Benar / Salah)

Tipe data boolean digunakan untuk menyimpan nilai logika yang hanya memiliki dua kemungkinan nilai: true (benar) atau false (salah). Tipe ini biasanya digunakan dalam kondisi percabangan atau untuk menentukan apakah suatu kondisi terpenuhi. Tipe data boolean sering digunakan dalam pernyataan kontrol, seperti if, untuk menentukan apakah suatu kondisi benar atau tidak.

1. Apa perbedaan antara echo dan print dalam PHP?

echo dan print adalah dua konstruksi dalam PHP yang digunakan untuk menampilkan output ke layar. Meskipun keduanya memiliki fungsi yang mirip, ada beberapa perbedaan antara keduanya.

* 1. Perbedaan sintaksis dan penggunaan

**echo**:

* echo adalah **konstruksi** (bukan fungsi) yang digunakan untuk mencetak satu atau lebih string.
* Dapat digunakan dengan atau tanpa tanda kurung (parentheses).
* Dapat mencetak beberapa argumen secara bersamaan dengan memisahkan mereka menggunakan koma.

**print**:

* print adalah **fungsi** yang digunakan untuk mencetak satu string.
* Hanya menerima satu argumen.
* Selalu mengembalikan nilai 1, yang berarti dapat digunakan dalam ekspresi
  1. Kinerja

Echo sedikit lebih cepat diibandingkan dengan print. Hal ini karena echo tidak mengembalikan nilai apapun, sedangkan print mengembalikan nilai 1. Namun, Perbedaan kinerja ini biasanya sangat kecil dan tidak terlalu signifikan untuk aplikasi web skala kecil hingga menengah.

1. Jelaskan fungsi dari variable super global $\_POST dan berikan contoh penggunaannya.

Variabel super global $\_POST dalam php digunakan untuk mengakses data yang dikirimkan melalui methode HTTP POST. Variable ini sangat berguna dalam menangani data yang dikirimkan dari formulir HTML, misalnya saat pengguna mengisi data dan mengirimkan formulir pendaftaran, login atau kontak.

* 1. Fungsi dari $\_POST:
* **Menerima data dari formulir**: Variabel $\_POST menyimpan data yang dikirimkan oleh pengguna melalui formulir HTML menggunakan metode POST. Data ini tidak terlihat di URL, yang menjadikannya lebih aman untuk mengirimkan informasi sensitif seperti kata sandi atau data pribadi.
* **Memungkinkan pengolahan data**: Dengan $\_POST, Anda dapat memproses data yang dikirimkan ke server, seperti memvalidasi, menyimpan ke database, atau menampilkan hasil kembali kepada pengguna.

**Contoh Penggunaanya** :

|  |
| --- |
| <form method="POST" action="proses.php">  <label for="nama">Nama:</label>  <input type="text" id="nama" name="nama" required><br>    <label for="email">Email:</label>  <input type="email" id="email" name="email" required><br>    <input type="submit" value="Kirim">  </form> |

Proses.php

|  |
| --- |
| <?php  // Mengecek apakah data dikirimkan melalui metode POST  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  // Mengambil data yang dikirimkan melalui $\_POST  $nama = $\_POST['nama'];  $email = $\_POST['email'];  // Menampilkan data yang diterima  echo "Nama: " . $nama . "<br>";  echo "Email: " . $email;  }  ?> |

# **F. POST-TEST**

1. Dalam konteks instruksi output di PHP, bandingkan penggunaan echo dan print dari segi performa dan fungsionalitas. Jelaskan situasi di mana Anda lebih memilih salah satu dari keduanya dan alasan di balik pilihan anda. Sertakan contoh kode untuk ilustrasi.

a. Performa

* **echo lebih cepat dibandingkan print**. Hal ini karena echo adalah bahasa konstruk (language construct), bukan fungsi, dan tidak mengembalikan nilai.
* **print lebih lambat dibandingkan echo** karena print mengembalikan nilai (integer 1), yang dapat digunakan dalam ekspresi.

Dalam aplikasi besar atau situasi di mana kecepatan sangat penting, menggunakan echo bisa sedikit lebih efisien, meskipun perbedaannya biasanya kecil dan tidak signifikan pada skala kecil.

* 1. Fungsionalitas

**echo**:

* Dapat menerima lebih dari satu argumen (meskipun jarang digunakan).

|  |
| --- |
| echo "Hello", " World"; // Output: Hello World |

**print**:

* Hanya dapat menerima satu argumen.
* Dapat digunakan sebagai bagian dari ekspresi karena mengembalikan nilai.

|  |
| --- |
| if (print("Hello")) {  print(" World"); // Output: Hello World  } |

* 1. Situasi Penggunaan

**Lebih memilih echo:**

1. **Saat performa penting**:
   * Karena lebih cepat, echo lebih cocok untuk output yang sering digunakan, seperti menghasilkan banyak elemen HTML.
2. **Saat menghasilkan beberapa argumen**:
   * echo mendukung beberapa argumen, yang tidak dimiliki print.

|  |
| --- |
| <?php  echo "Hello", " World", "!"; // Output: Hello World! |

2. Jelaskan bagaimana PHP menangani penggabungan string dan variable Ketika menggunakan instruksi echo. Apa perbedaaan antara menggunakan tanda kutip ganda (“ ”) dan tanda kutip Tunggal (` `) dalam penggabungan string dengan variable? Berikan contoh kode yang menunjukkan perbedaan ini.

**Perbedaan Tanda Kutip Ganda dan Tunggal**

**a. Tanda Kutip Ganda (")**

* PHP akan mem-parsing string untuk mencari variabel yang disisipkan.
* Variabel dalam tanda kutip ganda akan dievaluasi dan digantikan dengan nilainya.
* Mendukung karakter escape seperti \n, \t, dll.

|  |
| --- |
| $name = "Aef";  echo "Hello, $name!"; // Output: Hello, Aef! |

b. **Tanda Kutip Tunggal (')**

* PHP tidak mem-parsing isi string.
* Variabel dianggap sebagai teks literal, bukan nilainya.
* Tidak mendukung karakter escape kecuali untuk \' dan \\.

|  |
| --- |
| $name = "Aef";  echo 'Hello, $name!'; // Output: Hello, $name! |

**Penggabungan String dan Variabel**

1. **Dengan Operator . (Titik)**

* Baik untuk tanda kutip tunggal maupun ganda.
* Variabel tetap dievaluasi, dan string digabung secara eksplisit

|  |
| --- |
| $name = "Aef";  echo "Hello, " . $name . "!"; // Output: Hello, Aef!  echo 'Hello, ' . $name . '!'; // Output: Hello, Aef! |

1. **Dengan Penyisipan Variabel (Hanya dalam Tanda Kutip Ganda)**

* Lebih sederhana karena variabel otomatis dievaluasi.
* Tidak perlu operator penggabungan.

|  |
| --- |
| $name = "Aef";  echo "Hello, $name!"; // Output: Hello, Aef! |
| $name = "Aef";  echo 'Hello, $name!'; // Output: Hello, $name! |

1. **Menggunakan Kurung Kurawal {}**

Jika variabel berada di dalam string yang kompleks, gunakan kurung kurawal {} untuk memastikan PHP mengenali batas variabel.

|  |
| --- |
| $name = "Aef";  echo "Hello, {$name}Nur!"; // Output: Hello, AefNur! |

3. Anda diminta untuk menulis sebuah script PHP yang mencetak hasil dari operasi matematika yang melibatkan variable dan string. Misalnya, jika anda memiliki variable $a dengan nilai 5 dan $b dengan nilai 10, tuliskan kode yang menghasilkan output berupa string yang menyatakan hasil dari penjumlahan kedua variable tersebut, dan jelaskan cara PHP menangani

|  |
| --- |
| <?php  // Mendefinisikan variabel  $a = 5;  $b = 10;  // Melakukan operasi matematika dan mencetak hasilnya  echo "Hasil dari $a + $b adalah " . ($a + $b) . ".";  ?> |

* **Tanda Kutip Ganda**: PHP mem-parsing string untuk menemukan variabel seperti $a dan $b, lalu menggantinya dengan nilai aktual (5 dan 10).
* **Operator . (Titik)**: PHP menggabungkan string pertama ("Hasil dari $a + $b adalah ") dengan hasil operasi matematika ($a + $b) yang menghasilkan angka (15), lalu menambahkan string terakhir (".").

# **G. LATIHAN / TUGAS**

1.Buat Aplikasi Tracer Alumni sederhana menggunakan PHP dan jQuery.

|  |
| --- |
| Membuat Table |
| CREATE TABLE alumni (  id INT AUTO\_INCREMENT PRIMARY KEY,  name VARCHAR(100) NOT NULL,  graduation\_year INT NOT NULL,  email VARCHAR(100) NOT NULL,  job\_status VARCHAR(50) NOT NULL  ); |

|  |
| --- |
| db.php |
| <?php  $host = "localhost";  $user = "root";  $password = ""; // Password MySQL default Laragon adalah kosong  $database = "tracer\_alumni";  $conn = new mysqli($host, $user, $password, $database);  if ($conn->connect\_error) {  die("Connection failed: " . $conn->connect\_error);  }  ?> |

|  |
| --- |
| Index.php |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Tracer Alumni</title>  <link rel="stylesheet" href="css/styles.css">  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  <script src="js/scripts.js"></script>  </head>  <body>  <div class="container">  <h1>Tracer Alumni</h1>  <form id="alumniForm">  <input type="text" name="name" placeholder="Nama Lengkap" required>  <input type="number" name="graduation\_year" placeholder="Tahun Lulus" required>  <input type="email" name="email" placeholder="Email" required>  <select name="job\_status" required>  <option value="">Status Pekerjaan</option>  <option value="Bekerja">Bekerja</option>  <option value="Belum Bekerja">Belum Bekerja</option>  </select>  <button type="submit">Simpan</button>  </form>  <h2>Data Alumni</h2>  <table id="alumniTable">  <thead>  <tr>  <th>Nama</th>  <th>Tahun Lulus</th>  <th>Email</th>  <th>Status Pekerjaan</th>  </tr>  </thead>  <tbody>  <!-- Data alumni akan dimuat di sini -->  </tbody>  </table>  </div>  </body>  </html> |

|  |
| --- |
| Process.php |
| <?php  include 'db.php';  if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {  $name = $\_POST['name'];  $graduation\_year = $\_POST['graduation\_year'];  $email = $\_POST['email'];  $job\_status = $\_POST['job\_status'];  $stmt = $conn->prepare("INSERT INTO alumni (name, graduation\_year, email, job\_status) VALUES (?, ?, ?, ?)");  $stmt->bind\_param("siss", $name, $graduation\_year, $email, $job\_status);  if ($stmt->execute()) {  echo "Data berhasil disimpan!";  } else {  echo "Gagal menyimpan data.";  }  $stmt->close();  }  if ($\_SERVER['REQUEST\_METHOD'] === 'GET') {  $result = $conn->query("SELECT \* FROM alumni");  $data = [];  while ($row = $result->fetch\_assoc()) {  $data[] = $row;  }  echo json\_encode($data);  }  ?> |

|  |
| --- |
| Script.js |
| $(document).ready(function() {  // Muat data alumni  function loadAlumni() {  $.get("process.php", function(data) {  const alumni = JSON.parse(data);  const rows = alumni.map(alumnus => `  <tr>  <td>${alumnus.name}</td>  <td>${alumnus.graduation\_year}</td>  <td>${alumnus.email}</td>  <td>${alumnus.job\_status}</td>  </tr>  `);  $("#alumniTable tbody").html(rows.join(""));  });  }  loadAlumni();  // Tangani submit form  $("#alumniForm").submit(function(e) {  e.preventDefault();  const formData = $(this).serialize();  $.post("process.php", formData, function(response) {  alert(response);  loadAlumni();  });  $(this).trigger("reset");  });  }); |

|  |
| --- |
| Style.css |
| body {  font-family: Arial, sans-serif;  margin: 20px;  }  .container {  max-width: 800px;  margin: 0 auto;  }  form {  margin-bottom: 20px;  }  form input, form select, form button {  margin: 5px 0;  padding: 10px;  width: 100%;  }  table {  width: 100%;  border-collapse: collapse;  }  table, th, td {  border: 1px solid #ddd;  }  th, td {  padding: 10px;  text-align: left;  } |

Analisis :

1. db.php

Fungsi:  
File ini digunakan untuk menghubungkan aplikasi dengan database MySQL.

Penjelasan:

* $conn: Variabel ini menyimpan objek koneksi ke database menggunakan mysqli. Jika ada masalah saat koneksi (seperti salah nama pengguna atau kata sandi), maka aplikasi akan menghentikan eksekusi dan menampilkan pesan kesalahan.

### **2.** index.php

**Fungsi:**  
File utama yang digunakan untuk menampilkan form input data alumni serta menampilkan data alumni yang telah disimpan.

3. process.php

Fungsi:  
File ini menangani pengolahan data, baik untuk menyimpan data baru ke dalam database maupun mengambil data alumni yang sudah ada.

Penjelasan:

* Pengolahan Data POST:  
  Ketika form dikirimkan dengan metode POST, data nama, tahun lulus, email, dan status pekerjaan akan disimpan ke dalam database menggunakan query INSERT INTO.
* Pengolahan Data GET:  
  Ketika permintaan GET dilakukan (misalnya saat memuat data alumni), file ini akan mengambil semua data alumni dari database dan mengirimkannya dalam format JSON untuk ditampilkan di halaman.

### **4.** scripts.js

**Fungsi:**  
File ini mengatur interaksi pengguna di sisi klien, termasuk mengirimkan data form ke server dan memuat data alumni ke dalam tabel.

Penjelasan:

* Fungsi loadAlumni:  
  Fungsi ini akan memuat data alumni dari server dengan permintaan GET ke process.php. Data yang diterima dalam format JSON kemudian ditampilkan di tabel HTML.
* Pengiriman Form:  
  Ketika form disubmit, data akan dikirimkan ke server dengan metode POST. Setelah data berhasil disimpan, halaman tidak akan di-refresh, tetapi data alumni akan dimuat ulang di tabel.

### **5.** styles.css

**Fungsi:**  
File ini mengatur desain dan tampilan antarmuka pengguna aplikasi.

Penjelasan:

* Desain Form:  
  Menata form agar input dan tombol terlihat rapi dan mudah digunakan.
* Tabel Data:  
  Mengatur tabel untuk menampilkan data alumni dengan gaya yang bersih dan mudah dibaca.