



REIKA AMALIA SYAHPUTRI

2341720173

TI 2D

Topik

- Konsep Form Proccessing dengan PHP dan JQuery

Tujuan

Mahasiswa diharapkan dapat:

1. Mahasiswa mampu membuat form menggunakan PHP
2. Mahasiswa mampu membuat form menggunakan jQuery

Perhatian

Jobsheet ini harus dikerjakan step-by-step sesuai langkah-langkah praktikum yang sudah diberikan. Soal dapat dijawab langsung di dalam kolom yang disediakan dengan menggunakan PDF Editor.

Pendahuluan

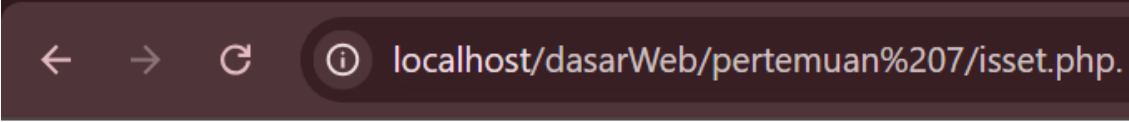
Function isset

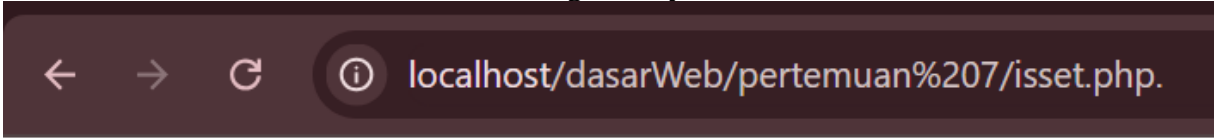
isset() adalah sebuah fungsi dalam PHP yang digunakan untuk memeriksa apakah suatu variabel telah didefinisikan (ada) atau tidak. Fungsi ini mengembalikan **true** jika variabel telah didefinisikan dan memiliki nilai, dan **false** jika variabel tidak ada atau bernilai **null**. Berikut adalah contoh script dan penjelasan minimal mengenai **isset()**:

Kode di atas memeriksa apakah variabel **\$nama** telah didefinisikan. Jika ya, maka pesan "Variabel 'nama' telah didefinisikan." akan ditampilkan, jika tidak, maka pesan "Variabel 'nama' tidak ditemukan atau bernilai null." akan ditampilkan.

Praktikum 1. Function Isset

Langkah	Keterangan
1	Buatlah satu file baru di dalam direktori dasarWeb, beri nama <code>isset.php</code> .
2	Ketikkan ke dalam file <code>isset.php</code> tersebut kode di bawah ini.

3	<pre>\$umur; if (isset(\$umur) && \$umur >= 18) { echo "Anda sudah dewasa."; } else { echo "Anda belum dewasa atau variabel 'umur' tidak ditemukan."; }</pre>
4	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/isset.php. Apa yang anda pahami dari penggunaan isset pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 1.1)</p> <ul style="list-style-type: none"> - Fungsi isset Digunakan untuk memeriksa apakah suatu variabel telah didefinisikan dan nilainya bukan null. Jika variabel ada dan bukan null, maka isset() akan mengembalikan nilai true. - Pada kode ini, isset(\$umur) memastikan bahwa variabel \$umur ada sebelum dilakukan pengecekan lebih lanjut apakah nilai \$umur >= 18. - Jika variabel \$umur ada dan bernilai 18 atau lebih, akan dicetak pesan "Anda sudah dewasa." - Jika variabel \$umur tidak ada atau nilainya kurang dari 18, akan muncul pesan "Anda belum dewasa atau variabel 'umur' tidak ditemukan."  <p>Anda belum dewasa atau variabel 'umur' tidak ditemukan.</p>
5	<p>Tambahkan isi dari file isset.php tersebut dengan kode di bawah ini.</p> <pre>\$data = array("nama" => "Jane", "usia" => 25); if (isset(\$data["nama"])) { echo "Nama: " . \$data["nama"]; } else { echo "Variabel 'nama' tidak ditemukan dalam array."; }</pre>
6	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/isset.php buat hasil tidak menjadi 1 baris, jadi tampilan dari echo harus terpisah. Apa yang anda pahami dari penggunaan isset pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 1.2)</p> <ul style="list-style-type: none"> - isset(\$data["nama"]) digunakan untuk memeriksa apakah kunci "nama" ada dalam array \$data dan nilainya bukan null. - Jika kunci "nama" ada, akan dicetak nilai yang terkait dengan kunci tersebut, yaitu "Jane". - Jika kunci "nama" tidak ditemukan dalam array, akan dicetak pesan "Variabel 'nama' tidak ditemukan dalam array."

	<ul style="list-style-type: none"> - Pada kedua kasus, fungsi <code>isset()</code> memastikan bahwa variabel atau kunci array telah didefinisikan sebelum mencoba mengaksesnya.  <p>Anda belum dewasa atau variabel 'umur' tidak ditemukan. Nama: Jane</p>
--	---

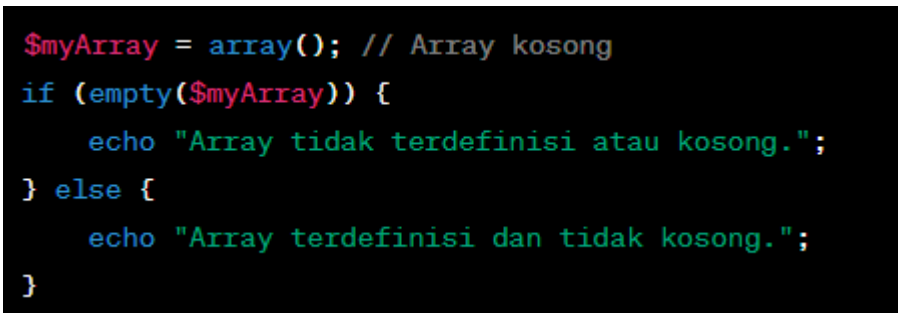
Function empty

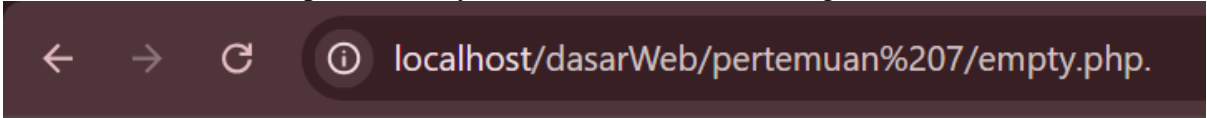
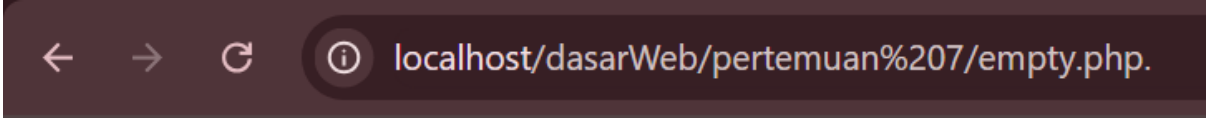
Fungsi `empty()` pada PHP digunakan untuk memeriksa apakah suatu variabel kosong atau tidak terdefinisi. Fungsi ini mengembalikan **true** jika variabel kosong atau tidak terdefinisi, dan **false** jika variabel memiliki nilai atau telah didefinisikan. Berikut adalah penjelasan dan contoh penggunaan `empty()`:

```
$nama = ""; // Variabel string kosong
if (empty($nama)) {
    echo "Nama tidak terdefinisi atau kosong.";
} else {
    echo "Nama terdefinisi dan tidak kosong.";
}
```

Fungsi `empty()` dapat digunakan untuk memeriksa apakah suatu string kosong atau tidak.

Praktikum 2. Function empty

Langkah	Keterangan
1	Buat satu file baru bernama <code>empty.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>empty.php</code>
2	
3	<p>Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/dasarWeb/empty.php</code>. Apa yang anda pahami dari penggunaan <code>empty</code> pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.1)</p> <ul style="list-style-type: none"> - Fungsi <code>empty()</code> di sini digunakan untuk mengecek apakah array <code>\$myArray</code> berisi elemen atau tidak. Jika kosong, pesan yang relevan akan ditampilkan.

	<ul style="list-style-type: none"> - Sebuah array dianggap kosong jika tidak memiliki elemen sama sekali (array kosong) atau jika variabelnya tidak didefinisikan atau bernilai null. - Variabel \$myArray dideklarasikan sebagai array kosong (array()), yang berarti tidak mengandung elemen apapun. - Fungsi empty(\$myArray) akan mengembalikan nilai true, karena array tersebut kosong. - Karena kondisi ini, pesan "Array tidak terdefinisi atau kosong." akan dicetak. - Jika array tersebut mengandung elemen (tidak kosong), maka bagian else akan dieksekusi, dan pesan "Array terdefinisi dan tidak kosong." akan dicetak.  <p>Array tidak terdefinisi atau kosong.</p>
4	<p>Tambahkan isi dari file empty.php tersebut dengan kode di bawah ini.</p> <pre>if (empty(\$nonExistentVar)) { echo "Variabel tidak terdefinisi atau kosong."; } else { echo "Variabel terdefinisi dan tidak kosong."; }</pre>
5	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/empty.php buat hasil tidak menjadi 1 baris, jadi tampilan dari echo harus terpisah. Apa yang anda pahami dari penggunaan empty pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.2)</p> <ul style="list-style-type: none"> - Pada kode ini, variabel \$nonExistentVar belum didefinisikan (belum ada nilai yang diberikan). - Fungsi empty(\$nonExistentVar) akan memeriksa apakah variabel tersebut tidak ada atau nilainya kosong. - Karena variabel tersebut tidak ada, empty() akan mengembalikan nilai true, dan pesan "Array tidak terdefinisi atau kosong." akan dicetak. - Jika variabel tersebut ada dan memiliki nilai yang tidak kosong, maka bagian else akan dieksekusi, dan pesan "Array terdefinisi dan tidak kosong." akan dicetak.  <p>Array tidak terdefinisi atau kosong.</p>

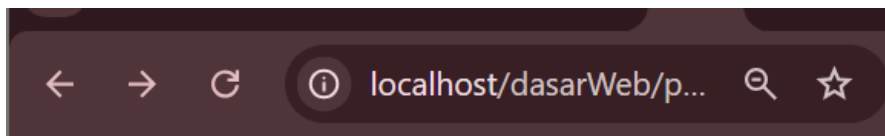
Praktikum Bagian 3 : Form Input PHP

Langkah	Keterangan
1	Buat satu file baru bernama <code>form.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>form.php</code>
2	<pre data-bbox="263 280 1228 974"> <!DOCTYPE html> <html> <head> <title>Form Input PHP</title> </head> <body> <h2>Form Input PHP</h2> <form method="post" action="proses_form.php"> <label for="nama">Nama:</label> <input type="text" name="nama" id="nama" required>

 <label for="email">Email:</label> <input type="email" name="email" id="email" required>

 <input type="submit" name="submit" value="Submit"> </form> </body> </html> </pre>
3	<p>Buat satu file baru bernama <code>proses_form.php</code> di dalam folder <code>dasarWeb</code>. Ketikkan kode pada langkah 2 di dalam <code>proses_form.php</code></p> <pre data-bbox="263 1086 1228 1556"> <?php if (\$_SERVER["REQUEST_METHOD"] == "POST") { \$nama = \$_POST["nama"]; \$email = \$_POST["email"]; echo "Nama: " . \$nama . "
"; echo "Email: " . \$email; } ?> </pre>
4	<p>Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/dasarWeb/proses_form.php</code>. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 3.1)</p> <ul style="list-style-type: none"> - Input teks untuk mengisi nama, dengan atribut <code>name="nama"</code> dan <code>id="nama"</code>, serta atribut <code>required</code> yang memastikan pengguna tidak bisa mengirimkan form tanpa mengisi nama. - Input email untuk mengisi alamat email, dengan atribut <code>name="email"</code>, <code>id="email"</code>, dan juga menggunakan atribut <code>required</code>. - Form menggunakan metode POST dengan atribut <code>method="post"</code>. Ini berarti data yang dimasukkan pengguna akan dikirim melalui metode POST (bukan GET) ketika tombol submit diklik.

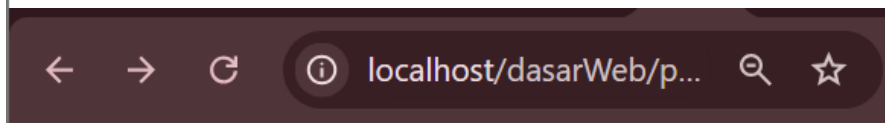
- Atribut `action="proses_form.php"` menentukan bahwa data form akan dikirim dan diproses oleh file PHP bernama `proses_form.php`.
- Di bagian PHP, kode mengecek apakah permintaan yang diterima adalah metode POST dengan menggunakan `$_SERVER["REQUEST_METHOD"] == "POST"`.
- `$nama = $_POST["nama"]` mengambil data dari input nama.
- `$email = $_POST["email"]` mengambil data dari input email.
- Setelah itu, data tersebut ditampilkan menggunakan `echo`, di mana nama dan email pengguna dicetak pada halaman dengan format:
`"Nama: [nama pengguna]"`
`"Email: [email pengguna]"`
- Form digunakan untuk mengumpulkan input dari pengguna (nama dan email), yang dikirim melalui metode POST ke file PHP untuk diproses.
- PHP memproses data tersebut dan menampilkannya kembali ke layar.



Form Input PHP

Nama:

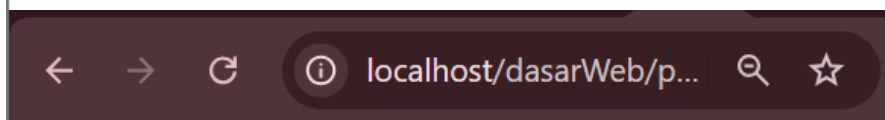
Email:



Form Input PHP

Nama:

Email:



Nama: rere
Email: reikaamalia17@gmail.com

5

Buat satu file baru bernama `form_self.php` di dalam folder `dasarWeb`. Ketikkan kode pada langkah 2 di dalam `form_self.php`

```

<!DOCTYPE html>
<html>
<head>
    <title>Form Input PHP</title>
</head>
<body>
    <h2>Form Input PHP</h2>
    <?php
    // Inisialisasi variabel
    $namaErr = "";
    $nama = "";

    // Cek apakah form sudah disubmit
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        // Validasi nama (contoh: pastikan nama tidak kosong)
        if (empty($_POST["nama"])) {
            $namaErr = "Nama harus diisi!";
        } else {
            $nama = $_POST["nama"];
            echo "Data berhasil disimpan!";
        }
    }
    ?>

    <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
        <label for="nama">Nama:</label>
        <input type="text" name="nama" id="nama" value="<?php echo $nama; ?>">
        <span class="error"><?php echo $namaErr; ?></span><br><br>

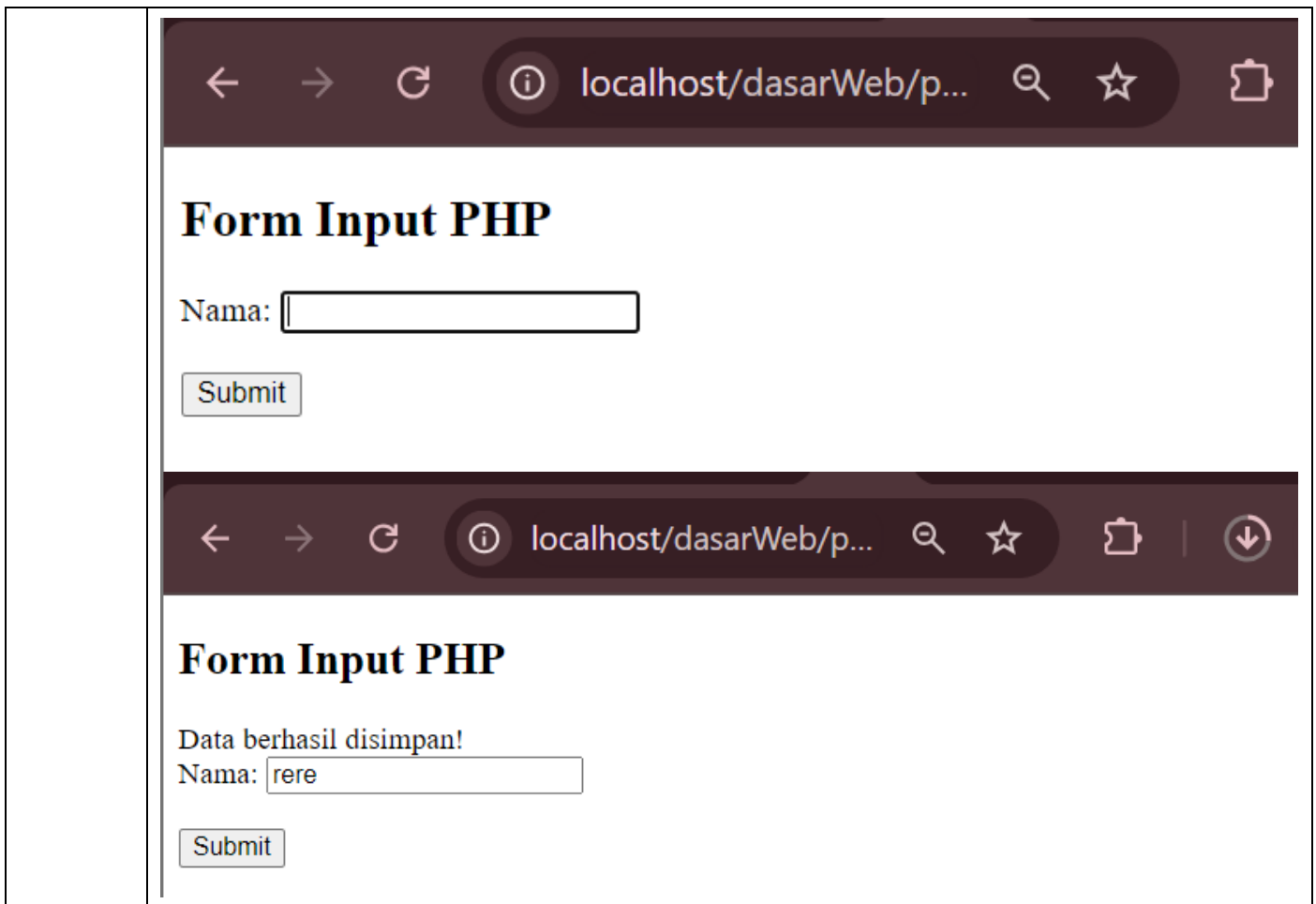
        <input type="submit" name="submit" value="Submit">
    </form>
</body>
</html>

```

Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/form_self.php. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 3.2)

6

- Form terdiri dari satu input teks untuk nama, dengan atribut name="nama" dan id="nama".
- Form menggunakan metode POST, ditentukan oleh atribut method="post", yang mengirimkan data ke file PHP yang sama (action="<?php echo htmlspecialchars(\$_SERVER["PHP_SELF"]); ?>"). Fungsi htmlspecialchars() digunakan untuk menghindari serangan XSS (Cross-site Scripting) dengan mengubah karakter khusus menjadi entitas HTML.
- Pada bagian PHP, dua variabel diinisialisasi: \$namaErr untuk menyimpan pesan kesalahan dan \$nama untuk menyimpan input yang dimasukkan pengguna.
- Blok if (\$_SERVER["REQUEST_METHOD"] == "POST") digunakan untuk memeriksa apakah form telah dikirimkan menggunakan metode POST.
- Jika input nama kosong, PHP akan memberikan pesan kesalahan "Nama harus diisi", yang disimpan di variabel \$namaErr dan ditampilkan di samping input form.
- Jika input diisi dengan benar, nilai input tersebut disimpan dalam variabel \$nama, dan pesan "Data berhasil disimpan!" ditampilkan sebagai konfirmasi.



HTML Injection

HTML injection (atau disebut juga "HTML injection attack" atau "client-side injection") adalah serangan keamanan yang terjadi ketika seorang penyerang memasukkan kode HTML atau JavaScript yang berbahaya ke dalam input yang diterima oleh sebuah aplikasi web. Kode berbahaya tersebut akan dieksekusi oleh browser pengguna yang melihat halaman web yang terpengaruh, yang dapat mengakibatkan akses ilegal ke data, perubahan tampilan halaman, atau serangan lainnya.

Seorang penyerang dapat mencoba menyisipkan kode HTML atau JavaScript yang berbahaya ke dalam input yang diterima oleh aplikasi web. Jika aplikasi web tersebut tidak melakukan sanitasi atau escape input dengan benar, maka kode berbahaya tersebut akan dieksekusi oleh browser pengguna, yang dapat menyebabkan masalah keamanan.

Contoh HTML Injection:

Misalnya, kita memiliki formulir pencarian sederhana di situs web yang mencari kata kunci di antara ulasan pengguna:

```
<form action="search.php" method="post">
  <input type="text" name="keyword">
  <input type="submit" value="Search">
</form>
```

Sekarang, jika aplikasi web tidak memvalidasi input dengan benar, seorang penyerang dapat memasukkan input berbahaya seperti ini:


```
<script>alert('You have been hacked!');</script>
```

Jika aplikasi web tidak menghindari atau membersihkan input ini sebelum menampilkannya di halaman hasil pencarian, maka hasilnya akan menjadi seperti ini:

```
<div>Hasil pencarian untuk "<script>alert('You have been hacked!');</script>":</div>
```

Kode JavaScript berbahaya tersebut akan dieksekusi oleh browser pengguna dan akan memunculkan kotak peringatan "You have been hacked!". Ini adalah contoh HTML injection yang sederhana. Sebagai pengembang web, penting untuk selalu melakukan sanitasi dan validasi input dari pengguna, serta menghindari menampilkan input pengguna langsung di halaman web tanpa escaping atau sanitasi yang tepat untuk menghindari serangan semacam ini.

Cara Mengangani HTML Injection

Cara menangani HTML injection adalah dengan memvalidasi, menyaring, dan menghindari menampilkan input pengguna tanpa escaping yang tepat. Berikut adalah beberapa langkah untuk melindungi aplikasi Anda dari HTML injection:

1. **Validasi Input:** Validasi input dari pengguna untuk memastikan bahwa hanya data yang valid yang diterima. Anda bisa menggunakan fungsi seperti `filter_input()` atau `filter_var()` untuk melakukan validasi.
2. **Saring Input:** Saring input pengguna untuk menghapus atau menggantikan karakter yang berpotensi berbahaya, seperti `<`, `>`, `&`, dan lainnya. Anda bisa menggunakan fungsi seperti `htmlspecialchars()` atau `strip_tags()` untuk ini.
3. **Parameterized Statements (Query):** Jika Anda menghasilkan SQL query dengan input pengguna, gunakan parameterized statements atau prepared statements untuk mencegah SQL injection, yang bisa menjadi bentuk serangan yang mirip dengan HTML injection.
4. **Content Security Policy (CSP):** Terapkan Content Security Policy (CSP) pada header HTTP Anda untuk mengontrol sumber daya yang dapat digunakan dalam halaman web Anda.
5. **Escape Output:** Ketika Anda menampilkan data di halaman web, pastikan Anda menghindari injeksi JavaScript dengan menggunakan `htmlspecialchars()` atau metode serupa.

Berikut contoh sederhana dalam PHP untuk mengatasi HTML injection:

```
// Menggunakan filter_input untuk validasi input
$user_input = filter_input(INPUT_POST, 'user_input', FILTER_SANITIZE_STRING);

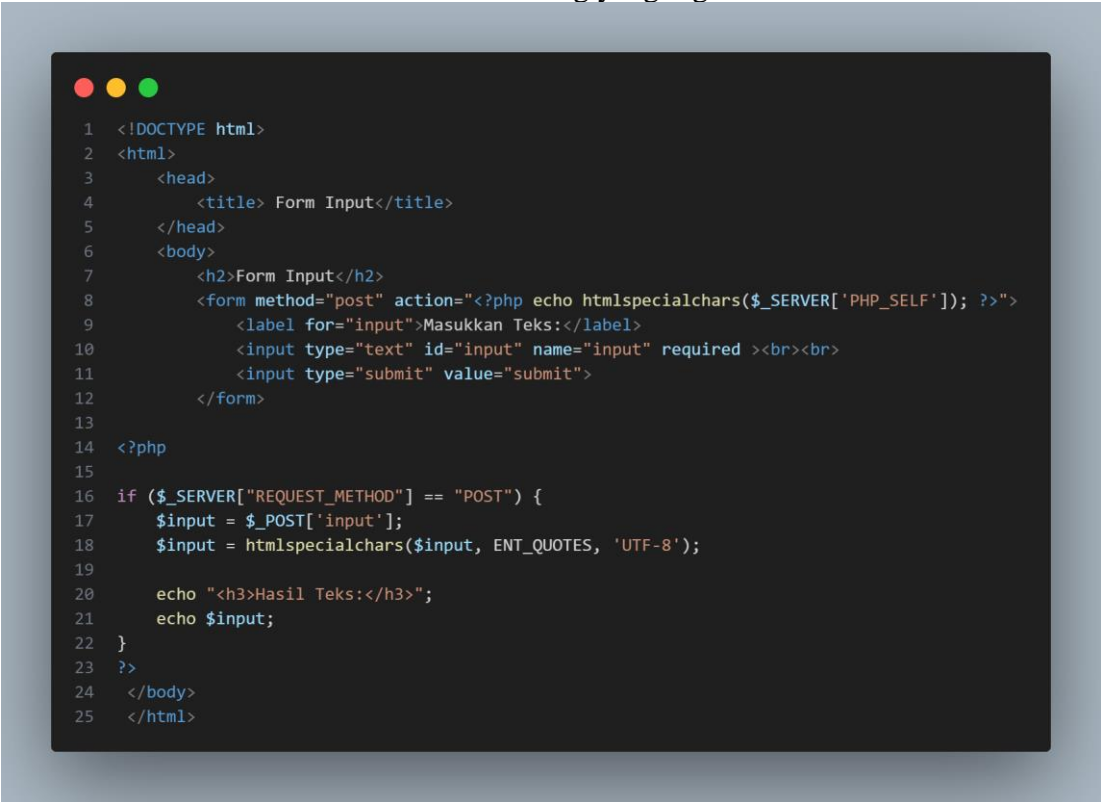
// Menggunakan htmlspecialchars untuk menghindari HTML injection saat menampilkan data
echo '<div>' . htmlspecialchars($user_input, ENT_QUOTES, 'UTF-8') . '</div>';
```

Dengan menggunakan langkah-langkah ini, Anda dapat memitigasi risiko HTML injection dalam aplikasi web Anda. Selalu penting untuk memvalidasi input, membersihkan data yang masuk, dan menghindari menampilkan data pengguna tanpa escaping yang benar.

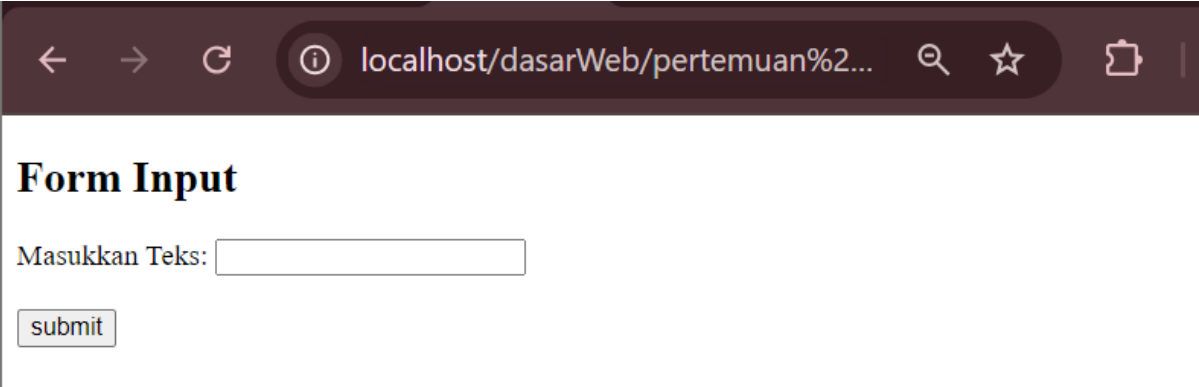
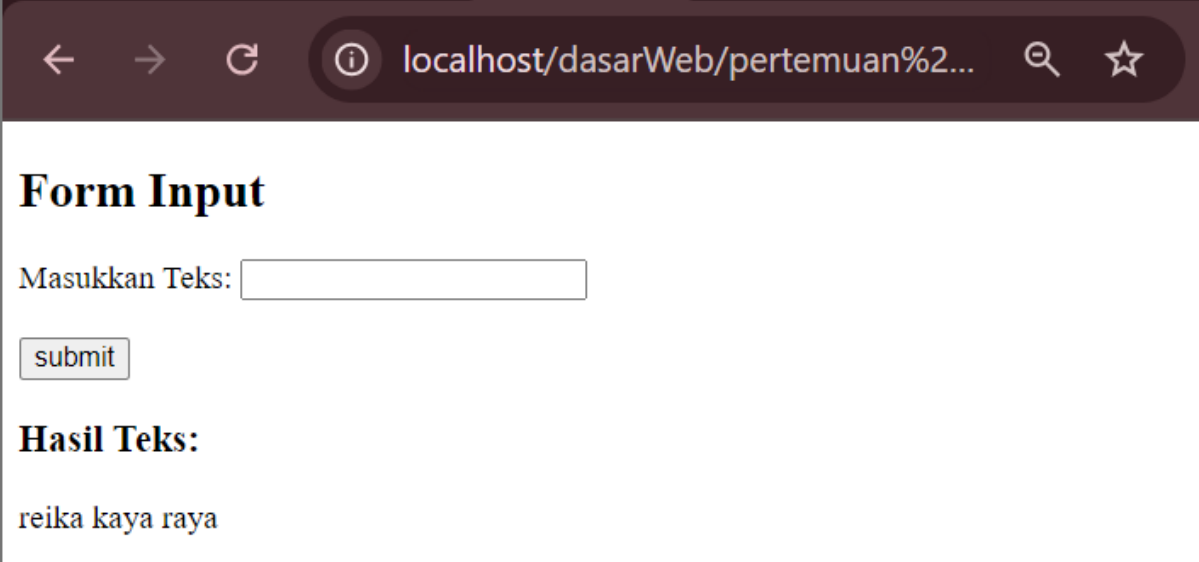
Praktikum 4 : HTML Injection

Langkah-langkah Praktikum :

Langkah	Keterangan
1	Buat satu file baru bernama <code>html_aman.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>html_aman.php</code>

2	<pre><code>\$input = \$_POST['input']; \$input = htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8');</code></pre>
3	<p>Tambahkan script lainnya supaya langkah 2 bisa berjalan normal. Lalu simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/html_aman.php</p>
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1)</p> <ul style="list-style-type: none"> - <form>: - method="post": Form ini akan mengirimkan data ke server menggunakan metode POST - action="<?php echo htmlspecialchars(\$_SERVER['PHP_SELF']); ?>": Bagian action menentukan ke mana data form akan dikirim. Dalam hal ini, data dikirim ke file PHP yang sama. \$_SERVER['PHP_SELF'] mereferensikan file saat ini. - \$input = \$_POST['input']; Mengambil nilai yang dikirim dari form melalui metode POST. \$_POST['input'] merujuk ke nilai dari kolom input yang memiliki name="input". - \$input = htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8'); - htmlspecialchars(): Fungsi ini digunakan untuk mencegah serangan XSS dengan mengonversi karakter spesial (seperti &, <, >, ', ") menjadi entitas HTML yang aman. - ENT_QUOTES: Opsi ini memastikan bahwa tanda kutip tunggal dan ganda juga diubah menjadi entitas HTML (&#039; untuk tanda kutip tunggal dan &quot; untuk tanda kutip ganda). - 'UTF-8': Menentukan bahwa encoding yang digunakan adalah UTF-8.  <pre><code>1 <!DOCTYPE html> 2 <html> 3 <head> 4 <title> Form Input</title> 5 </head> 6 <body> 7 <h2>Form Input</h2> 8 <form method="post" action="<?php echo htmlspecialchars(\$_SERVER['PHP_SELF']); ?>"> 9 <label for="input">Masukkan Teks:</label> 10 <input type="text" id="input" name="input" required >

 11 <input type="submit" value="submit"> 12 </form> 13 14 <?php 15 16 if (\$_SERVER["REQUEST_METHOD"] == "POST") { 17 \$input = \$_POST['input']; 18 \$input = htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8'); 19 20 echo "<h3>Hasil Teks:</h3>"; 21 echo \$input; 22 } 23 ?> 24 </body> 25 </html></code></pre>

	 
5	Ketikkan kode tambahan pada langkah 6 di dalam <code>html_aman.php</code>
6	<pre>// Memeriksa apakah input adalah email yang valid \$email = \$_POST['email']; if (filter_var(\$email, FILTER_VALIDATE_EMAIL)) { // Lanjutkan dengan pengolahan email yang aman } else { // Tangani input yang tidak valid }</pre>
7	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/html_aman.php</code>
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.2)</p> <ul style="list-style-type: none"> - <code><label></code> dan <code><input></code> untuk teks: - <code><label for="input">Masukkan Teks:</label></code>: Label ini memberikan deskripsi untuk input teks di bawahnya. Atribut <code>for</code> mengaitkan label dengan elemen input yang memiliki <code>id="input"</code>. - <code><input type="text" id="input" name="input" required></code>: Ini adalah input teks sederhana. Atribut <code>required</code> berarti pengguna harus mengisi kolom ini sebelum mengirim form.

	<ul style="list-style-type: none"> - <code><label></code> dan <code><input></code> untuk email: - <code><label for="email">Masukkan Email:</label></code>: Label ini memberikan deskripsi untuk input email. - <code><input type="email" id="email" name="email" required></code>: Input ini menerima alamat email. Atribut <code>type="email"</code> memastikan input ini hanya menerima email yang valid dari sisi browser. <code>required</code> juga memastikan kolom email harus diisi. - <code>if (\$_SERVER["REQUEST_METHOD"] == "POST") { ... }</code>: Blok if ini memeriksa apakah form dikirim dengan metode POST. Ini memastikan kode hanya dijalankan ketika form di-submit. - <code>\$input = \$_POST['input'];</code>: Mengambil data teks yang diisi oleh pengguna dari form. <code>\$_POST['input']</code> mereferensikan nilai dari input teks yang memiliki atribut <code>name="input"</code> di form. - <code>\$email = \$_POST['email'];</code>: Mengambil input email dari form melalui variabel superglobal <code>\$_POST</code>. Input ini berasal dari elemen form yang memiliki <code>name="email"</code>. - <code>filter_var(\$email, FILTER_VALIDATE_EMAIL)</code>: Fungsi ini memvalidasi apakah input <code>\$email</code> adalah format email yang valid. - Jika email valid, maka blok ini akan menampilkan input teks dan email yang valid. Jika tidak valid, akan memicu blok else yang menampilkan pesan error.
--	--

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Form Input</title>
5      </head>
6      <body>
7          <h2>Form Input</h2>
8          <form method="post" action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>">
9              <label for="input">Masukkan Teks:</label>
10             <input type="text" id="input" name="input" required><br><br>
11
12             <label for="email">Masukkan Email:</label>
13             <input type="email" id="email" name="email" required><br><br>
14
15             <input type="submit" value="submit">
16         </form>
17
18         <?php
19         if ($_SERVER["REQUEST_METHOD"] == "POST") {
20
21             $input = $_POST['input'];
22             $email = $_POST['email'];
23
24
25             $input = htmlspecialchars($input, ENT_QUOTES, 'UTF-8');
26
27             // Memeriksa apakah input adalah email yang valid
28             if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
29
30                 echo "<h3>Hasil Teks:</h3>";
31                 echo $input;
32
33                 echo "<h3>Hasil Email yang Valid:</h3>";
34                 echo $email;
35             } else {
36
37                 echo "<h3>Email tidak valid. Silakan masukkan email yang benar.</h3>";
38             }
39         }
40         ?>
41     </body>
42 </html>
43

```

<
>
↻
 ⓘ localhost/dasarWeb/pertemuan%207/ht... 🔍 ☆

Form Input

Masukkan Teks:

Masukkan Email:

	<div style="background-color: #333; color: white; padding: 5px;"> ← → ↻ localhost/dasarWeb/pertemuan%207/ht... 🔍 ☆ </div> <h2 style="margin-top: 10px;">Form Input</h2> <p>Masukkan Teks: <input type="text"/></p> <p>Masukkan Email: <input type="text"/></p> <p><input type="button" value="submit"/></p> <p>Hasil Teks:</p> <p>reika amalia</p> <p>Hasil Email yang Valid:</p> <p>reikaamalia17@gmail.com</p>
--	--

Regular Expression (Regex)

Regex adalah alat yang kuat untuk mencari, mencocokkan, atau memanipulasi teks berdasarkan pola tertentu. Anda dapat menggunakannya untuk berbagai tujuan seperti validasi input, pencarian teks, penggantian teks, dan banyak lagi.

Basic Regex Patterns:

- **Karakter Tunggal:** Karakter apa pun akan cocok dengan dirinya sendiri dalam Regex, kecuali karakter khusus yang harus dihindari dengan karakter pelarian \.
 - Contoh: Pola `/a/` akan cocok dengan huruf "a" dalam teks.
- **Karakter Spesifik:** Anda dapat mencocokkan karakter tertentu dengan menyebutkannya dalam pola.
 - Contoh: Pola `/hello/` akan cocok dengan teks "hello" dalam teks.
- **Karakter Khusus:** Beberapa karakter khusus dalam Regex harus dihindari dengan karakter pelarian (\) jika Anda ingin mencocokkannya secara harfiah. Contoh karakter khusus: `.`, `*`, `+`, `?`, `|`, `[`, `]`, `(`, `)`, `{`, `}`, `^`, `$`, `\`.
- **Karakter Set:** Anda dapat mencocokkan karakter dari satu set karakter dengan menggunakan `[]`. Misalnya, `/[aeiou]/` akan mencocokkan salah satu huruf vokal.
- **Rentang Karakter:** Anda dapat menggunakan `-` dalam karakter set untuk menentukan rentang karakter. Misalnya, `/[a-z]/` akan mencocokkan huruf kecil apa pun.

- **Kuantifier:** Anda dapat mengatur berapa kali karakter atau kelompok karakter sebelumnya harus muncul. Contoh kuantifier:
 - o *: 0 atau lebih kali
 - o +: 1 atau lebih kali
 - o ?: 0 atau 1 kali
 - o {n}: Persis n kali
 - o {n,}: Setidaknya n kali
 - o {n,m}: Setidaknya n kali, maksimal m kali

Contoh Penggunaan Regex:

1. Mencocokkan pola

```
$pattern = '/hello/';
$text = 'Hello, world!';
if (preg_match($pattern, $text)) {
    echo "Cocokkan ditemukan!";
} else {
    echo "Tidak ada cocokan!";
}
```

2. Mencocokkan karakter set

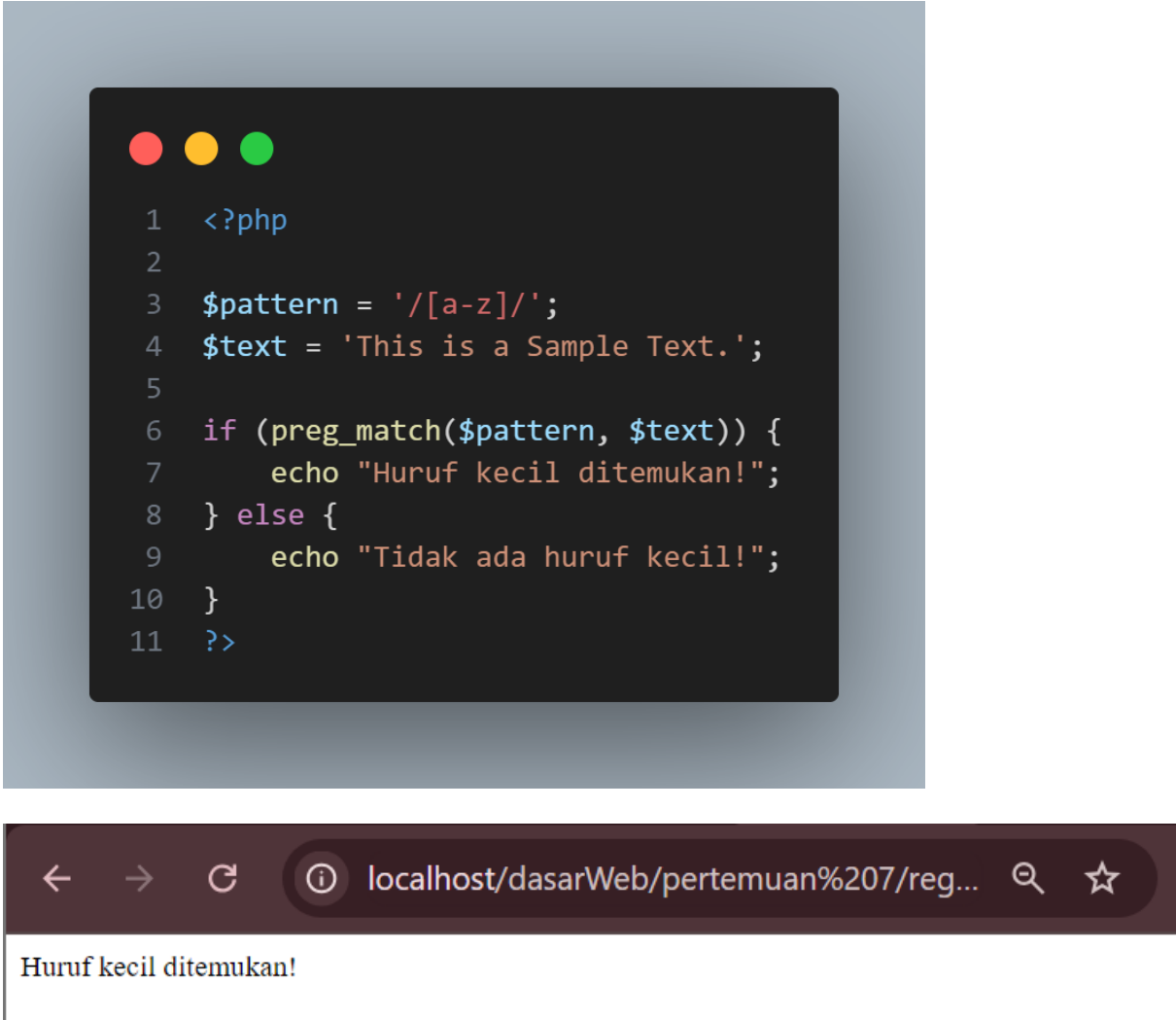
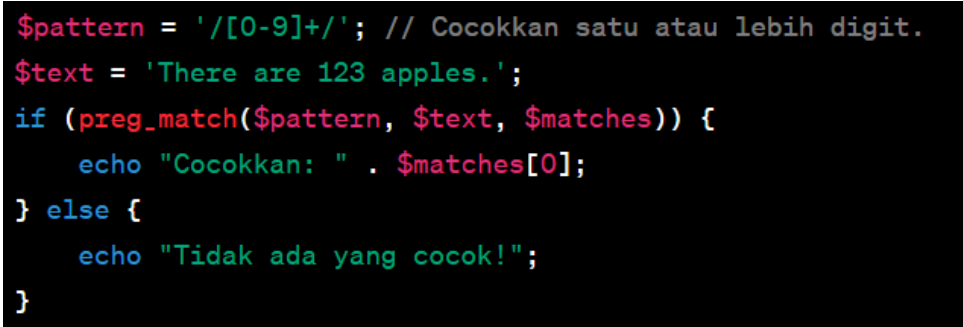
```
$pattern = '/[0-9]/'; // Cocokkan satu digit.
$text = 'There are 3 apples.';
if (preg_match($pattern, $text)) {
    echo "Digit ditemukan!";
} else {
    echo "Tidak ada digit!";
}
```

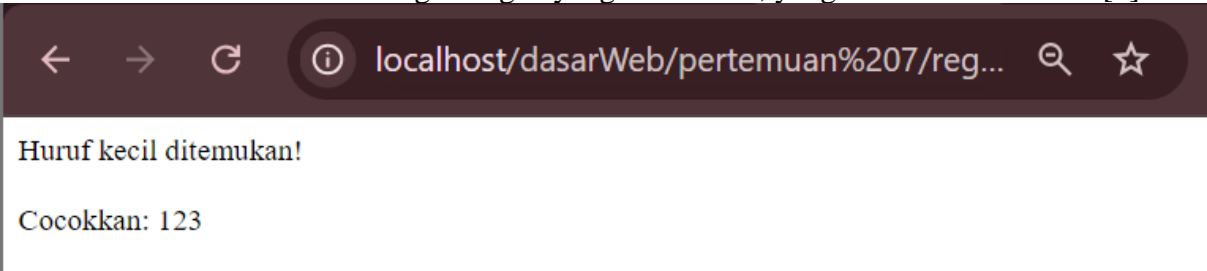
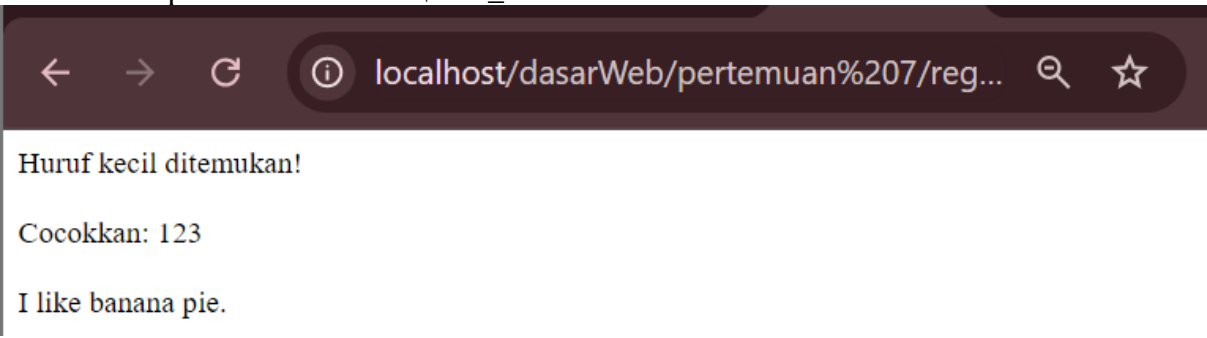
Praktikum 5 : Penggunaan Regex Pada PHP

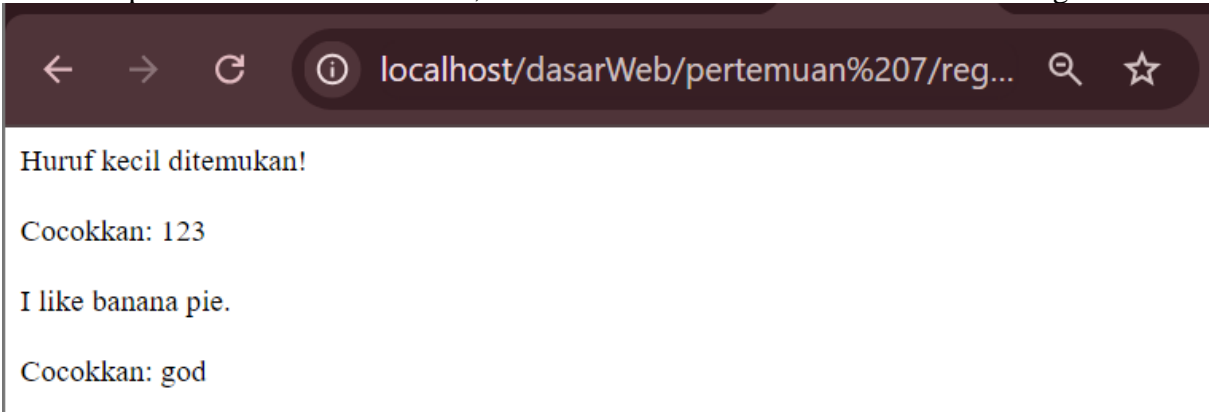
Langkah-langkah Praktikum :

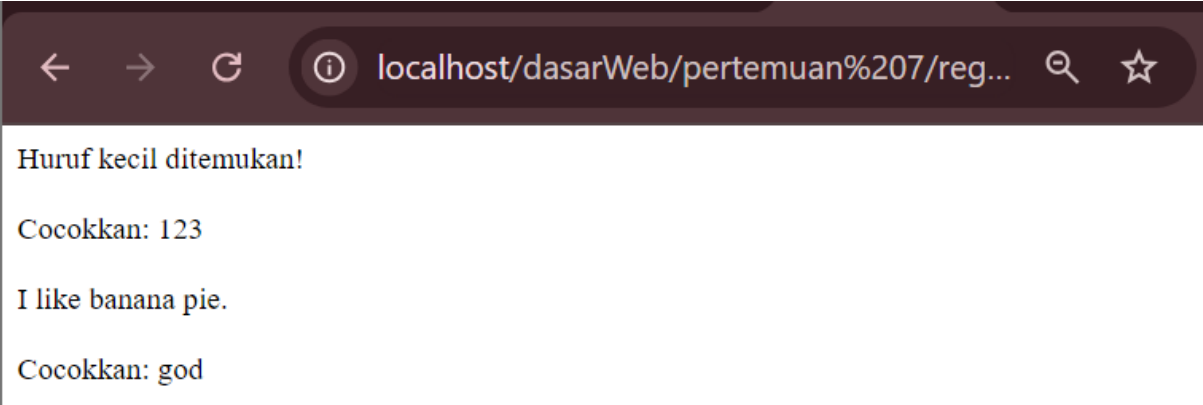
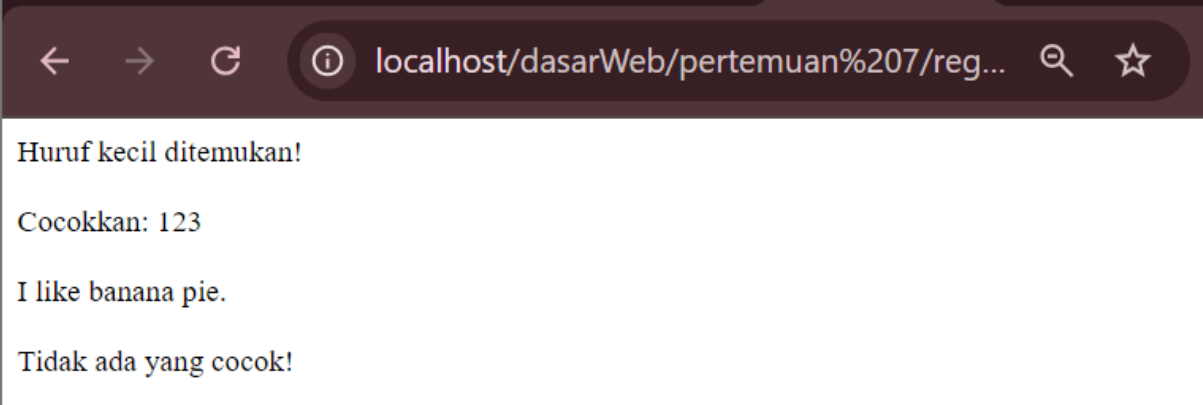
Langkah	Keterangan
1	Buat satu file baru bernama <code>regex.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>regex.php</code>

2	<pre> \$pattern = '/[a-z]/'; // Cocokkan huruf kecil. \$text = 'This is a Sample Text.'; if (preg_match(\$pattern, \$text)) { echo "Huruf kecil ditemukan!"; } else { echo "Tidak ada huruf kecil!"; } </pre>
3	<p>Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php</p>
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.1)</p> <ul style="list-style-type: none"> - \$pattern = '/[a-z]/';: - Variabel \$pattern menyimpan regex yang akan digunakan untuk mencari huruf kecil dalam string. - [a-z] adalah pattern dalam regex yang berarti "huruf kecil dari a sampai z". - Tanda / di awal dan akhir adalah pembatas untuk regular expression dalam PHP. - \$text = 'This is a Sample Text.';: - Variabel \$text menyimpan string yang akan diperiksa apakah mengandung huruf kecil atau tidak. Dalam hal ini, string-nya adalah 'This is a Sample Text.'. - if (preg_match(\$pattern, \$text)): - Jika fungsi preg_match() menemukan setidaknya satu huruf kecil dalam string, ia akan mengembalikan nilai true (atau 1) dan pernyataan di dalam blok if akan dijalankan. - Jika tidak ada huruf kecil yang ditemukan, ia akan mengembalikan false (atau 0) dan blok else akan dijalankan. - preg_match(\$pattern, \$text): - preg_match() adalah fungsi PHP yang digunakan untuk mencari pola (pattern) dalam sebuah string. - Argumen pertama, \$pattern, adalah pola yang dicari (regex [a-z]), yang berarti kita mencari setidaknya satu huruf kecil dalam string. - Argumen kedua, \$text, adalah string yang ingin kita periksa. - Return value: Fungsi preg_match() akan mengembalikan 1 jika pola ditemukan dalam string, dan 0 jika tidak ditemukan.

	 <pre> 1 <?php 2 3 \$pattern = '/[a-z]/'; 4 \$text = 'This is a Sample Text.'; 5 6 if (preg_match(\$pattern, \$text)) { 7 echo "Huruf kecil ditemukan!"; 8 } else { 9 echo "Tidak ada huruf kecil!"; 10 } 11 ?> </pre> <p>← → ↻ ⓘ localhost/dasarWeb/pertemuan%207/reg... 🔍 ☆</p> <p>Huruf kecil ditemukan!</p>
5	Ketikkan kode tambahan pada langkah 6 di dalam <code>regex.php</code>
6	 <pre> \$pattern = '/[0-9]+/'; // Cocokkan satu atau lebih digit. \$text = 'There are 123 apples.'; if (preg_match(\$pattern, \$text, \$matches)) { echo "Cocokkan: " . \$matches[0]; } else { echo "Tidak ada yang cocok!"; } </pre>
7	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/regex.php</code>
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.2)</p> <ul style="list-style-type: none"> - \$pattern adalah sebuah regular expression (regex) yang mencari satu atau lebih digit (angka) dalam teks. [] digunakan untuk mendefinisikan karakter yang ingin dicocokkan, dan + menunjukkan bahwa kita mencari satu atau lebih dari karakter yang ada dalam bracket tersebut. - \$text adalah string yang akan diperiksa untuk mencari angka. - Fungsi <code>preg_match</code> digunakan untuk mencari apakah ada kecocokan pola dalam \$text. Jika ada, hasilnya disimpan dalam array \$matches.

	<ul style="list-style-type: none"> - Jika pencocokan berhasil (artinya ada angka dalam teks), program akan mencetak "Cocokkan: " diikuti dengan angka yang ditemukan, yang terletak di <code>\$matches[0]</code>. 
9	Ketikkan kode tambahan pada langkah 10 di dalam <code>regex.php</code>
10	<pre><code>\$pattern = '/apple/'; \$replacement = 'banana'; \$text = 'I like apple pie.'; \$new_text = preg_replace(\$pattern, \$replacement, \$text); echo \$new_text; // Output: "I like banana pie."</code></pre>
11	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/regex.php</code>
12	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.3)</p> <ul style="list-style-type: none"> - <code>\$replacement</code> adalah string yang akan digunakan untuk menggantikan bagian dari teks yang cocok dengan pola. Dalam contoh ini, "apple" akan diganti dengan "banana". - <code>\$text</code> adalah string yang akan diperiksa dan dimodifikasi. Teks ini mengandung kata "apple". - Fungsi <code>preg_replace</code> digunakan untuk mencari pola yang ditentukan dalam <code>\$text</code> dan menggantinya dengan string yang ada di <code>\$replacement</code>, hasil dari penggantian ini disimpan dalam variabel <code>\$new_text</code>. 
13	Ketikkan kode tambahan pada langkah 14 di dalam <code>regex.php</code>

14	<pre> \$pattern = '/go*d/'; // Cocokkan "god", "good", "goood", dll. \$text = 'god is good.'; if (preg_match(\$pattern, \$text, \$matches)) { echo "Cocokkan: " . \$matches[0]; } else { echo "Tidak ada yang cocok!"; } </pre>
15	<p>Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php</p>
16	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.4)</p> <ul style="list-style-type: none"> - Secara keseluruhan, kode ini digunakan untuk mendeteksi dan mengekstrak kata-kata yang sesuai dengan pola tertentu dari string dalam PHP menggunakan regular expression. Dalam contoh ini, ia berhasil menemukan dan mencetak kata "god". 
17	<p>Soal 5.5: Buatlah script langkah 14 dengan variable pattern yang di ubah menggunakan `?` (0 atau 1 kali). Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.5)</p> <ul style="list-style-type: none"> - Pola Regex (/go?d/): - g - Harus ada karakter "g" di awal. - o? - Karakter "o" bisa ada 0 atau 1 kali. Ini berarti "g" diikuti oleh tidak ada "o" (seperti "gd") atau satu "o" (seperti "god"). - d - Harus ada karakter "d" di akhir. - Hasil Pencocokan - Untuk teks 'gd is god, good, and goood.', hasilnya akan: - "gd" cocok dengan pola (0 kali "o"). - "god" cocok dengan pola (1 kali "o"). - "good" tidak cocok karena memiliki dua "o". - "goood" tidak cocok karena memiliki lebih dari satu "o".

	
18	<p>Soal 5.6: Buatlah script langkah 14 dengan variable pattern yang di ubah menggunakan `{n,m}`. Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.6)</p> <ul style="list-style-type: none"> - Tidak ada yang cocok dikarenakan n,m seharusnya diisi dengan angka, berapa kali akan dikeluarkan, sehingga jika ditampilkan n,m akan membuat hasil tidak ada yang cocok. <pre data-bbox="272 763 1137 1301"> \$pattern = '/go{n,m}d/'; \$text = 'god is good.'; if (preg_match(\$pattern, \$text, \$matches)) { echo "Cocokkan: " . \$matches[0]; } else { echo "Tidak ada yang cocok!"; } ?> </pre> 

Praktikum 6 : Form Lanjut

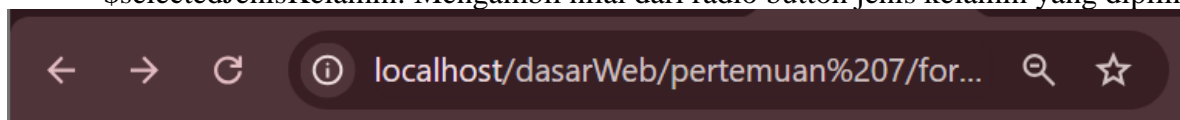
Langkah-langkah Praktikum :

Langka h	Keterangan
1	Buat satu file baru bernama <code>form_lanjut.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>form_lanjut.php</code>
2	<pre> <!DOCTYPE html> <html> <head> <title>Contoh Form dengan PHP</title> </head> <body> <h2>Form Contoh</h2> <form method="POST" action="proses_lanjut.php"> <label for="buah">Pilih Buah:</label> <select name="buah" id="buah"> <option value="apel">Apel</option> <option value="pisang">Pisang</option> <option value="mangga">Mangga</option> <option value="jeruk">Jeruk</option> </select>
 <label>Pilih Warna Favorit:</label>
 <input type="checkbox" name="warna[]" value="merah"> Merah
 <input type="checkbox" name="warna[]" value="biru"> Biru
 <input type="checkbox" name="warna[]" value="hijau"> Hijau

 <label>Pilih Jenis Kelamin:</label>
 <input type="radio" name="jenis_kelamin" value="laki-laki"> Laki-laki
 <input type="radio" name="jenis_kelamin" value="perempuan"> Perempuan

 <input type="submit" value="Submit"> </form> </body> </html> if (\$_SERVER["REQUEST_METHOD"] == "POST") { \$selectedBuah = \$_POST['buah']; if (isset(\$_POST['warna'])) { \$selectedWarna = \$_POST['warna']; } else { \$selectedWarna = []; } \$selectedJenisKelamin = \$_POST['jenis_kelamin']; echo "Anda memilih buah: " . \$selectedBuah . "
"; if (!empty(\$selectedWarna)) { echo "Warna favorit Anda: " . implode(", ", \$selectedWarna) . "
"; } else { echo "Anda tidak memilih warna favorit.
"; } echo "Jenis kelamin Anda: " . \$selectedJenisKelamin; } </pre>
3	Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/form_lanjut.php</code>
4	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 6.1)

- Form: Form ini menggunakan metode POST untuk mengirim data ke file proses_lanjut.php saat tombol "Submit" ditekan.
- Dropdown untuk Memilih Buah: Terdapat dropdown (<select>) untuk memilih salah satu dari empat jenis buah: apel, pisang, mangga, dan jeruk. Nilai yang dipilih akan dikirim dengan nama buah.
- Checkbox untuk Warna Favorit: Terdapat beberapa checkbox yang memungkinkan pengguna memilih lebih dari satu warna favorit. Nama warna[] menunjukkan bahwa ini adalah array, sehingga beberapa nilai dapat dikirim.
- Radio Button untuk Jenis Kelamin: Pengguna dapat memilih satu dari dua opsi jenis kelamin (laki-laki atau perempuan) menggunakan radio button.
- Kode ini pertama-tama memeriksa apakah permintaan yang diterima adalah metode POST (`$_SERVER["REQUEST_METHOD"] == "POST"`). Ini memastikan bahwa data hanya diproses jika form telah disubmit.
- `$selectedBuah`: Mengambil nilai dari dropdown buah yang dipilih.
- `$selectedWarna`: Memeriksa apakah warna favorit telah dipilih. Jika ya, ambil array warna dari form; jika tidak, buat array kosong.
- `$selectedJenisKelamin`: Mengambil nilai dari radio button jenis kelamin yang dipilih.



Form Contoh

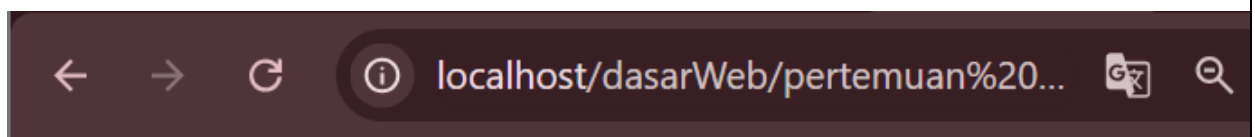
Pilih Buah:

Pilih Warna Favorit:

- ☒ Merah
☐ Biru
☐ Hijau

Pilih Jenis Kelamin:

- ☐ Laki-laki
☒ Perempuan



Anda memilih buah: apel
 Warna favorit Anda: merah
 Jenis kelamin Anda: perempuan

5

Buat satu file baru bernama `form_ajax.php` di dalam folder `dasarWeb`. Ketikkan kode pada langkah 2 di dalam `form_ajax.php`

```

<!DOCTYPE html>
<html>
<head>
    <title>Contoh Form dengan PHP dan jQuery</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
    <h2>Form Contoh</h2>
    <form id="myForm">
        <label for="buah">Pilih Buah:</label>
        <select name="buah" id="buah">
            <option value="apel">Apel</option>
            <option value="pisang">Pisang</option>
            <option value="mangga">Mangga</option>
            <option value="jeruk">Jeruk</option>
        </select>

        <br>

        <label>Pilih Warna Favorit:</label><br>
        <input type="checkbox" name="warna[]" value="merah"> Merah<br>
        <input type="checkbox" name="warna[]" value="biru"> Biru<br>
        <input type="checkbox" name="warna[]" value="hijau"> Hijau<br>

        <br>

        <label>Pilih Jenis Kelamin:</label><br>
        <input type="radio" name="jenis_kelamin" value="laki-laki"> Laki-laki<br>
        <input type="radio" name="jenis_kelamin" value="perempuan"> Perempuan<br>

        <br>

        <input type="submit" value="Submit">
    </form>

    <div id="hasil">
        <!-- Hasil akan ditampilkan di sini -->
    </div>

    <script>
        $(document).ready(function () {
            $("#myForm").submit(function (e) {
                e.preventDefault(); // Mencegah pengiriman form secara default

                // Mengumpulkan data form
                var formData = $("#myForm").serialize();

                // Kirim data ke server PHP
                $.ajax({
                    url: "proses_lanjut.php", // Ganti dengan nama file PHP yang sesuai
                    type: "POST",
                    data: formData,
                    success: function (response) {
                        // Tampilkan hasil dari server di div "hasil"
                        $("#hasil").html(response);
                    }
                });
            });
        });
    </script>
</body>
</html>

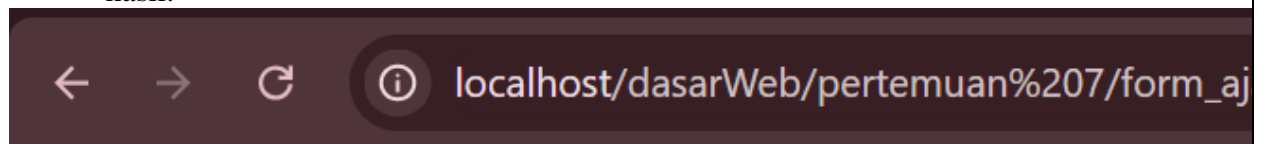
```

6

Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_ajax.php . Catat di sini apa yang anda amati dari penambahan kode program di atas.
(soal no 6.2)

- Dropdown untuk Memilih Buah: Memungkinkan pengguna memilih dari beberapa buah (apel, pisang, mangga, jeruk).

- Checkbox untuk Warna Favorit: Pengguna dapat memilih lebih dari satu warna favorit.
- Radio Button untuk Jenis Kelamin: Pengguna dapat memilih satu dari dua jenis kelamin (laki-laki atau perempuan).
- Div dengan ID hasil digunakan untuk menampilkan respon yang diterima dari server setelah form disubmit.
- `$(document).ready(...)`: Fungsi ini memastikan bahwa jQuery hanya menjalankan kode di dalamnya setelah halaman sepenuhnya dimuat.
- `$("#myForm").submit(function (e) {...})`: Menangkap event submit pada form dengan ID myForm. Fungsi ini akan dijalankan saat form disubmit.
- `e.preventDefault()`: Mencegah perilaku default form, yaitu pengiriman data dan reload halaman.
- `var formData = $("#myForm").serialize();`: Mengambil semua data dari form dan mengubahnya menjadi format URL-encoded yang siap untuk dikirim melalui AJAX.
- AJAX Request:
 - `url: "proses_lanjut.php"`: Menentukan URL ke mana data akan dikirim untuk diproses.
 - `type: "POST"`: Menentukan metode HTTP yang digunakan untuk mengirim data.
 - `data: formData`: Mengirim data yang telah diserialisasi.
 - `success: function (response) {...}`: Fungsi callback yang dijalankan jika permintaan AJAX berhasil. Respon dari server (dari proses_lanjut.php) akan ditampilkan di div dengan ID hasil.



Form Contoh

Pilih Buah:

Pilih Warna Favorit:

- ☒ Merah
☒ Biru
☐ Hijau

Pilih Jenis Kelamin

- ☐ Laki-laki
☒ Perempuan

Anda memilih buah: apel


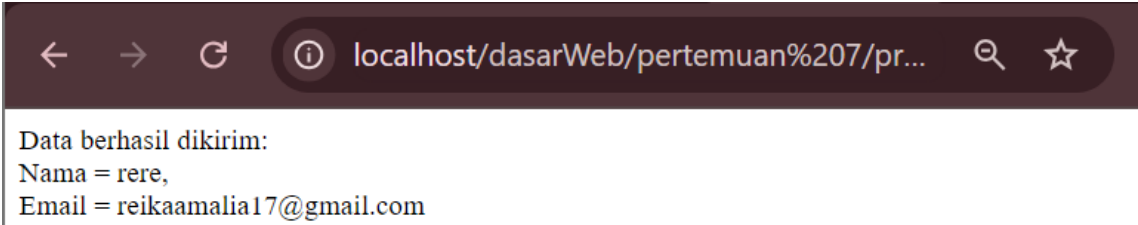
Warna favorit Anda: merah, biru

Jenis kelamin Anda: perempuan

Praktikum 7 : Validasi Form

Langkah-langkah Praktikum :

Langkah	Keterangan
1	Buat satu file baru bernama <code>form_validasi.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>form_validasi.php</code>
2	<pre> <!DOCTYPE html> <html> <head> <title>Form Input dengan Validasi</title> </head> <body> <h1>Form Input dengan Validasi</h1> <form method="post" action="proses_validasi.php"> <label for="nama">Nama:</label> <input type="text" id="nama" name="nama">
 <label for="email">Email:</label> <input type="text" id="email" name="email">
 <input type="submit" value="Submit"> </form> </body> </html> </pre> <pre> <?php if (\$_SERVER["REQUEST_METHOD"] == "POST") { \$nama = \$_POST["nama"]; \$email = \$_POST["email"]; \$errors = array(); // Validasi Nama if (empty(\$nama)) { \$errors[] = "Nama harus diisi."; } // Validasi Email if (empty(\$email)) { \$errors[] = "Email harus diisi."; } elseif (filter_var(\$email, FILTER_VALIDATE_EMAIL)) { \$errors[] = "Format email tidak valid."; } // Jika ada kesalahan validasi if (empty(\$errors)) { (foreach (\$errors as \$error) { echo \$error . "
"; } } else { // Lanjutkan dengan pemrosesan data jika semua validasi berhasil // Misalnya, menyimpan data ke database atau mengirim email echo "Data berhasil dikirim: Nama = \$nama, Email = \$email"; } } ?> </pre>
3	Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/form_validasi.php</code>
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.1)</p> <ul style="list-style-type: none"> - <code>\$nama</code> dan <code>\$email</code>: Mengambil data yang dimasukkan pengguna dari form menggunakan <code>\$_POST</code>. - <code>\$errors</code>: Membuat array kosong untuk menyimpan pesan kesalahan validasi. - Nama: Memeriksa apakah field nama kosong. Jika kosong, tambahkan pesan kesalahan ke dalam array <code>\$errors</code>.

	<ul style="list-style-type: none"> - Email: Memeriksa apakah field email kosong. Jika kosong, tambahkan pesan kesalahan. Jika tidak kosong, menggunakan filter_var untuk memeriksa apakah format email valid. Jika tidak valid, tambahkan pesan kesalahan. - Secara keseluruhan, kode ini merupakan contoh dari form input sederhana dengan validasi data menggunakan PHP. Pengguna diminta untuk mengisi nama dan email, dan jika ada kesalahan (seperti field kosong atau format email yang tidak valid), pesan kesalahan akan ditampilkan. Jika semua input valid, data yang dimasukkan akan ditampilkan ke pengguna.  
5	Kembangkan file bernama <code>form_validasi.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 6 di dalam <code>form_validasi.php</code>

6	<pre> <!DOCTYPE html> <html> <head> <title>Form Input dengan Validasi</title> <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script> </head> <body> <h1>Form Input dengan Validasi</h1> <form id="myForm" method="post" action="proses_validasi.php"> <label for="nama">Nama:</label> <input type="text" id="nama" name="nama">
 <label for="email">Email:</label> <input type="text" id="email" name="email">
 <input type="submit" value="Submit"> </form> <script> \$(document).ready(function() { \$("#myForm").submit(function(event) { var nama = \$("#nama").val(); var email = \$("#email").val(); var valid = true; if (nama === "") { \$("#nama-error").text("Nama harus diisi."); valid = false; } else { \$("#nama-error").text(""); } if (email === "") { \$("#email-error").text("Email harus diisi."); valid = false; } else { \$("#email-error").text(""); } if (valid) { (!event.preventDefault()); // Menghentikan pengiriman form jika validasi gagal } }); }); </script> </body> </html> </pre>
7	<p>Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_validasi.php</p>
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.2)</p> <ul style="list-style-type: none"> - Meskipun hasil akhir (output) mungkin serupa dengan pendekatan PHP murni, interaksi yang lebih baik dan performa yang ditingkatkan membuatnya lebih menarik dan sesuai untuk aplikasi web modern. 

Data berhasil dikirim:

Nama = rere,

Email = reikaamalia17@gmail.com

Soal 7.3: Buatlah script langkah 6 dengan menggunakan ajax. Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.3)

- Penggunaan AJAX dalam script ini membuat pengalaman pengguna lebih baik dan interaktif. Hal ini juga mempermudah pengelolaan kode dan mempercepat pengembangan fitur baru. Pendekatan ini sangat cocok untuk aplikasi web modern yang mengutamakan interaksi pengguna dan kecepatan.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Form Input dengan AJAX</title>
5    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
6  </head>
7  <body>
8    <h1>Form Input dengan AJAX</h1>
9    <form id="myForm">
10     <label for="nama">Nama:</label>
11     <input type="text" id="nama" name="nama">
12     <span id="nama-error" style="color:red;"></span><br>
13
14     <label for="email">Email:</label>
15     <input type="text" id="email" name="email">
16     <span id="email-error" style="color:red;"></span><br>
17
18     <input type="submit" value="Submit">
19   </form>
20
21   <div id="hasil"></div>
22
23   <script>
24     $(document).ready(function(){
25       $("#myForm").submit(function(event){
26         event.preventDefault(); // Mencegah reload halaman
27
28         var nama = $("#nama").val();
29         var email = $("#email").val();
30         var valid = true;
31
32         // Validasi Nama
33         if (nama === "") {
34           $("#nama-error").text("Nama harus diisi.");
35           valid = false;
36         } else {
37           $("#nama-error").text("");
38         }
39
40         // Validasi Email
41         if (email === "") {
42           $("#email-error").text("Email harus diisi.");
43           valid = false;
44         } else {
45           $("#email-error").text("");
46         }
47
48         // Jika validasi berhasil, kirim data ke server
49         if (valid) {
50           $.ajax({
51             url: "proses_validasi.php",
52             type: "POST",
53             data: {nama: nama, email: email},
54             success: function(response) {
55               $("#hasil").html(response); // Menampilkan hasil di div
56             }
57           });
58         }
59       });
60     });
61   </script>
62 </body>
63 </html>
64

```

```

1  <?php
2  if ($_SERVER["REQUEST_METHOD"] == "POST") {
3      $nama = $_POST["nama"];
4      $email = $_POST["email"];
5      $errors = array();
6
7      // Validasi Nama
8      if (empty($nama)) {
9          $errors[] = "Nama harus diisi.";
10     }
11     // Validasi Email
12     if (empty($email)) {
13         $errors[] = "Email harus diisi.";
14     } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
15         $errors[] = "Format email tidak valid.";
16     }
17
18     // Jika ada kesalahan validasi
19     if (!empty($errors)) {
20         foreach ($errors as $error) {
21             echo $error . "<br>";
22         }
23     } else {
24         // Jika validasi berhasil, tampilkan data yang dikirim
25         echo "Data berhasil dikirim: <br> Nama = $nama, <br> Email = $email";
26     }
27 }

```

← → ↺ ⓘ localhost/dasarWeb/pertemuan%207/for... 🔍 ☆

Form Input dengan AJAX

Nama:

Email:

Data berhasil dikirim:
 Nama = rere,
 Email = reikaamalia17@gmail.com

10

Soal 7.4: Tambahkan validasi untuk password dengan ketentuan minimal 8 karakter dengan menggunakan jQuery dan PHP tambahkan pada langkah 9. Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.4)

Kode di atas terdiri dari dua bagian: HTML dengan JavaScript menggunakan jQuery untuk validasi form dan PHP untuk memproses data. Form ini meminta input dari pengguna untuk nama, email, dan password, dengan validasi yang memastikan bahwa nama tidak kosong, email valid, dan password memiliki minimal 8 karakter. Saat form disubmit, jQuery mencegah pengiriman default, melakukan validasi di sisi klien, dan jika semua input valid, data dikirim ke server menggunakan AJAX tanpa memuat ulang halaman. PHP menerima data, melakukan validasi tambahan, dan menampilkan pesan kesalahan jika ada. Jika semua input valid, PHP mengembalikan pesan sukses yang menunjukkan data yang berhasil dikirim. Pendekatan ini meningkatkan interaktivitas dan pengalaman pengguna dengan memberikan umpan balik instan dan memproses data secara efisien.

localhost/dasarWeb/pertemuan%2...

Form Input dengan Validasi Password

Nama:

Email:

Password:

Data berhasil dikirim:

Nama = rere,

Email = reikaamalia17@gmail.com