



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()**:

```
$nama = "John";  
if (isset($nama)) {  
    echo "Variabel 'nama' telah didefinisikan.";  
} else {  
    echo "Variabel 'nama' tidak ditemukan atau bernilai null.";  
}
```

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> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Variabel 'umur' tidak ditemukan.</p> </div> <p>Jika variabel \$umur tidak ada, tampilkan pesan bahwa variabel tersebut 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> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Variabel 'umur' tidak ditemukan. Nama: Jane</p> </div> <ul style="list-style-type: none"> • Variabel \$umur dideklarasikan tanpa nilai, sehingga nilainya adalah null. • isset(\$umur) digunakan untuk memeriksa apakah \$umur di-set. • Jika \$umur tidak di-set, program mencetak "Variabel 'umur' tidak ditemukan." • echo "
"; menambahkan baris baru di output. • Array \$data berisi dua kunci: "nama" dengan nilai "Jane" dan "usia" dengan nilai 25. • isset(\$data["nama"]) memeriksa apakah kunci "nama" ada dalam array. • Jika ada, program mencetak "Nama: Jane."

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	<pre> \$myArray = array(); // Array kosong if (empty(\$myArray)) { echo "Array tidak terdefinisi atau kosong."; } else { echo "Array terdefinisi dan tidak kosong."; } </pre>
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> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Array tidak terdefinisi atau kosong.</p> </div> <p>Output dari kode di atas adalah "Array tidak terdefinisi atau kosong." karena <code>\$myArray</code> adalah array kosong.</p>
4	<p>Tambahkan isi dari file <code>empty.php</code> 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 <code>localhost/dasarWeb/empty.php</code> buat hasil tidak menjadi 1 baris, jadi tampilan dari <code>echo</code> harus terpisah. Apa yang anda pahami dari penggunaan <code>empty</code> pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.2)</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Array tidak terdefinisi atau kosong. Variabel tidak terdefinisi atau kosong.</p> </div> <p>Array <code>\$myArray</code> diperiksa menggunakan <code>empty()</code>. Karena array kosong, hasilnya adalah "Array tidak terdefinisi atau kosong." <code>\$nonExistentVar</code> tidak dideklarasikan, sehingga <code>empty()</code> mengembalikan <code>true</code>, dan mencetak "Variabel tidak terdefinisi atau kosong."</p>

Praktikum Bagian 3 : Form Input PHP

Langka h	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> <!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 proses_form.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam proses_form.php</p> <pre> <?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 localhost/dasarWeb/proses_form.php. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 3.1)</p> <div data-bbox="268 1630 818 1753" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Nama: Naditya Email: aditandino@gmail.com</p> </div> <ul style="list-style-type: none"> • if (\$_SERVER["REQUEST_METHOD"] == "POST"): Memastikan bahwa kode hanya dijalankan jika form dikirim dengan metode POST. • \$nama = \$_POST["nama"];; Mengambil nilai input nama dari form. • \$email = \$_POST["email"];; Mengambil nilai input email dari form. • echo "Nama: " . htmlspecialchars(\$nama) . "
"; Menampilkan nama yang

	<p>dimasukkan, dengan perlindungan terhadap serangan XSS.</p> <ul style="list-style-type: none"> • <code>echo "Email: " . htmlspecialchars(\$email) . "
;";</code> Menampilkan email yang dimasukkan.
5	<p>Buat satu file baru bernama <code>form_self.php</code> di dalam folder <code>dasarWeb</code>. Ketikkan kode pada langkah 2 di dalam <code>form_self.php</code></p>

```

<!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!";
      }
    }
  ?>

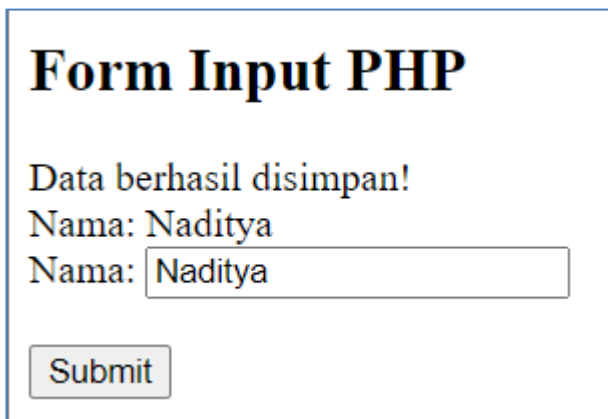
  <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
    <label for="nama">Nama:</label>
    <input type="text" name="nama" id="nama" value="<?php echo $nama; ?>">
    <span class="error"><?php echo $namaErr; ?></span><br><br>

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

```

6

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)



- \$namaErr untuk menyimpan pesan kesalahan jika nama tidak diisi.
- \$nama untuk menyimpan nilai nama yang diinputkan.
- if (\$_SERVER["REQUEST_METHOD"] == "POST"): Memeriksa apakah form disubmit.
- if (empty(\$_POST["nama"])): Memeriksa apakah field nama kosong.
 - Jika kosong, menyimpan pesan kesalahan di \$namaErr.
 - Jika tidak kosong, menyimpan nama yang telah di-HTML encode ke dalam variabel \$nama dan menampilkan pesan "Data berhasil disimpan!".
- Menggunakan metode POST untuk mengirimkan data ke halaman yang sama.
- Menampilkan pesan kesalahan di samping input nama jika ada.

--	--

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>\$input = \$_POST['input']; \$input = htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8');</pre>
3	Tambahkan script lainnya supaya langkah 2 bisa berjalan normal. Lalu simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/html_aman.php</code>
4	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1)

Form Input PHP

Data yang Anda masukkan: Tes

Input:

5	Ketikkan kode tambahan pada langkah 6 di dalam <code>html_aman.php</code>
6	<pre>// Periksa 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> <div data-bbox="279 848 1054 1169" data-label="Form"> <p>Form Input PHP</p> <p>Input: <input type="text" value="Tes"/></p> <p>Email: <input type="text" value="aditandino@gmail.com"/> Email harus diisi.</p> <p><input type="button" value="Submit"/></p> </div> <div data-bbox="279 1279 1078 1702" data-label="Form"> <p>Form Input PHP</p> <p>Data yang Anda masukkan: Tes</p> <p>Email yang Anda masukkan: aditandino@gmail.com</p> <p>Input: <input type="text" value="Tes"/></p> <p>Email: <input type="text" value="aditandino@gmail.com"/></p> <p><input type="button" value="Submit"/></p> </div> <p>Sebelum mengakses <code>\$_POST['email']</code>, tambahkan pengecekan dengan <code>if (isset(\$_POST['email']))</code>. Ini memastikan bahwa hanya jika elemen tersebut ada di dalam array <code>\$_POST</code>, kode akan mencoba untuk menggunakannya. Jika <code>\$_POST['email']</code> tidak diset, maka akan ditampilkan pesan kesalahan yang menyatakan bahwa email harus diisi.</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 \.
 - o Contoh: Pola /a/ akan cocok dengan huruf "a" dalam teks.
- **Karakter Spesifik:** Anda dapat mencocokkan karakter tertentu dengan menyebutkannya dalam pola.
 - o 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

- +: 1 atau lebih kali
- ?: 0 atau 1 kali
- {n}: Persis n kali
- {n,}: Setidaknya n kali
- {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 cocokkan!";
}
```

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><code>\$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!"; }</code></pre>

3	Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.1)</p> <div style="border: 1px solid blue; padding: 5px; margin: 10px 0;">Huruf kecil ditemukan!</div> <p>Pemeriksaan Regex: <code>preg_match(\$pattern, \$text)</code> digunakan untuk mencari apakah ada huruf kecil dalam <code>\$text</code>.</p> <ul style="list-style-type: none"> • Jika huruf kecil ditemukan, program mencetak "Huruf kecil ditemukan!". • Jika tidak ada huruf kecil, program mencetak "Tidak ada huruf kecil!".
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 localhost/dasarWeb/regex.php
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.2)</p> <div style="border: 1px solid blue; padding: 5px; margin: 10px 0;"> <p>Huruf kecil ditemukan!</p> <p>Cocokkan: 123</p> </div> <p>Menggunakan <code>\$patternLowercase</code> dan <code>\$textLowercase</code> untuk pemeriksaan huruf kecil.</p> <p>Menggunakan <code>\$patternDigit</code> dan <code>\$textDigit</code> untuk pemeriksaan digit. Ini meningkatkan keterbacaan dan menghindari kebingungan antara pola dan teks yang berbeda.</p> <p>Menambahkan <code>
</code> di akhir setiap output untuk memastikan bahwa setiap hasil muncul di baris baru pada tampilan web.</p> <p>Mengorganisir kode dalam blok yang terpisah untuk setiap pemeriksaan, sehingga lebih mudah untuk dibaca dan dikelola.</p>
9	Ketikkan kode tambahan pada langkah 10 di dalam <code>regex.php</code>

10	<pre> \$pattern = '/apple/'; \$replacement = 'banana'; \$text = 'I like apple pie.'; \$new_text = preg_replace(\$pattern, \$replacement, \$text); echo \$new_text; // Output: "I like banana pie." </pre>
11	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php
12	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.3)</p> <div style="border: 1px solid blue; padding: 5px; margin: 10px 0;"> <p>Huruf kecil ditemukan! Cocokkan: 123 I like banana pie.</p> </div> <ul style="list-style-type: none"> • \$patternReplace = '/apple/'; mendefinisikan pola untuk mencari kata "apple". • \$replacement = 'banana'; menentukan kata pengganti yang akan digunakan. • \$textReplace = 'I like apple pie.'; adalah string yang akan diperiksa dan diubah.preg_replace(\$patternReplace, \$replacement, \$textReplace); digunakan untuk mengganti semua kemunculan "apple" dengan "banana" dalam teks yang ditentukan. • Hasilnya disimpan dalam variabel \$new_text.echo \$new_text; mencetak hasil penggantian ke layar, yang seharusnya menjadi "I like banana pie."
13	Ketikkan kode tambahan pada langkah 14 di dalam regex.php

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> <div data-bbox="272 730 624 909" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre> Huruf kecil ditemukan! Cocokkan: 123 I like banana pie. Cocokkan: god </pre> </div> <ul style="list-style-type: none"> • Pola Regex untuk Kata "god" dan "good": <ul style="list-style-type: none"> • \$patternGod = '/go*d/'; mendefinisikan pola untuk mencocokkan "god", "good", "goood", dan variasi lainnya. Di sini, o* menunjukkan bahwa huruf 'o' dapat muncul 0 atau lebih kali. • Teks yang Diperiksa: <ul style="list-style-type: none"> • \$textGod = 'god is good.'; adalah string yang akan diperiksa untuk mencari kecocokan dengan pola yang ditentukan. • Pemeriksaan Regex: <ul style="list-style-type: none"> • if (preg_match(\$patternGod, \$textGod, \$matches)) digunakan untuk memeriksa apakah ada kecocokan dengan pola. • Jika ada kecocokan, program mencetak "Cocokkan: " diikuti dengan kata yang cocok (dari array \$matches). • Jika tidak ada kecocokan, program mencetak "Tidak ada yang cocok!".
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> <div data-bbox="272 1832 624 2011" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre> Huruf kecil ditemukan! Cocokkan: 123 I like banana pie. Cocokkan: god </pre> </div>

	<ul style="list-style-type: none"> • Pada bagian <code>\$patternGod</code>, pola regex diubah menjadi <code>'/go?d/'</code>. Ini akan mencocokkan kata "god" (dengan 'o') dan "gd" (tanpa 'o'). • Output: <ul style="list-style-type: none"> • Dengan perubahan ini, program akan mencari "god" atau "gd" dalam teks yang diberikan. Jika ada yang cocok, akan menampilkan hasilnya. Jika tidak ada yang cocok, akan mencetak "Tidak ada yang cocok!".
18	<p>Soal 5.6: Buatlah script langkah 14 dengan variable pattern yang di ubah menggunakan <code>`{n,m}`</code>. Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.6)</p> <div style="border: 1px solid blue; padding: 5px; margin: 10px 0;"> <p>Huruf kecil ditemukan! Cocokkan: 123 I like banana pie. Cocokkan: god</p> </div> <ol style="list-style-type: none"> 1. Pola Regex: <ul style="list-style-type: none"> ○ Pada bagian <code>\$patternGod</code>, pola regex diubah menjadi <code>'/go{0,1}d/'</code>. Ini akan mencocokkan kata "god" (dengan satu huruf 'o') dan "gd" (tanpa huruf 'o'). 2. Output: <ul style="list-style-type: none"> ○ Dengan perubahan ini, program akan mencari "god" atau "gd" dalam teks yang diberikan. Jika ada yang cocok, akan menampilkan hasilnya. Jika tidak ada yang cocok, akan mencetak "Tidak ada yang cocok!".

Praktikum 6 : Form Lanjut

Langkah-langkah Praktikum :

Langkah	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	<p>Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_lanjut.php</p>
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 6.1)</p> <div data-bbox="276 1809 702 1957" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Anda memilih buah: apel Warna favorit Anda: biru Jenis kelamin Anda: laki-laki</p> </div> <ul style="list-style-type: none"> • Cek Metode Request: Memastikan bahwa form disubmit dengan metode POST. • Ambil Data: Mengambil nilai dari input form seperti buah, warna, dan jenis kelamin. • Inisialisasi Warna: Jika tidak ada warna yang dipilih, inisialisasi sebagai array kosong.

	<ul style="list-style-type: none"> • Tampilkan Hasil: Menampilkan pilihan buah, warna favorit (jika ada), dan jenis kelamin yang dipilih pengguna. <div data-bbox="272 194 691 759"> <p>Form Contoh</p> <p>Pilih Buah: <input type="text" value="Apel"/></p> <p>Pilih Warna Favorit:</p> <p><input type="checkbox"/> Merah</p> <p><input checked="" type="checkbox"/> Biru</p> <p><input type="checkbox"/> Hijau</p> <p>Pilih Jenis Kelamin:</p> <p><input checked="" type="radio"/> Laki-laki</p> <p><input type="radio"/> Perempuan</p> <p><input type="button" value="Submit"/></p> </div> <div data-bbox="272 822 681 954"> <p>Anda memilih buah: apel</p> <p>Warna favorit Anda: biru</p> <p>Jenis kelamin Anda: laki-laki</p> </div>
5	<p>Buat satu file baru bernama <code>form_ajax.php</code> di dalam folder <code>dasarWeb</code>. Ketikkan kode pada langkah 2 di dalam <code>form_ajax.php</code></p>

```

<!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)

Form Contoh

Pilih Buah:

Pilih Warna Favorit:

☐ Merah

☒ Biru

☐ Hijau

Pilih Jenis Kelamin:

☒ Laki-laki

☐ Perempuan

Form Contoh

Pilih Buah:

Pilih Warna Favorit:

☐ Merah

☒ Biru

☐ Hijau

Pilih Jenis Kelamin:

☒ Laki-laki

☐ Perempuan

Anda memilih buah: mangga

Warna favorit Anda: biru

Jenis kelamin Anda: laki-laki

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	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.1)

Form Input dengan Validasi

Nama:

Email:

Data berhasil dikirim: Nama = Naditya, Email = aditandino@gmail.com

- **HTML Form:** Kode ini membuat sebuah form input yang meminta pengguna untuk memasukkan nama dan email. Form ini menggunakan metode POST untuk mengirimkan data ke halaman yang sama (`proses_validasi.php`).
- **Validasi Form:** Setelah form disubmit, kode PHP memeriksa apakah metode permintaan adalah POST. Jika iya, ia mengambil nilai dari input nama dan email.
- **Array Kesalahan:** Kode ini membuat array `$errors` untuk menyimpan pesan kesalahan validasi.
- **Validasi Nama dan Email:**
 - **Nama:** Memeriksa apakah input nama kosong. Jika kosong, pesan kesalahan ditambahkan ke array.
 - **Email:** Memeriksa apakah input email kosong atau tidak valid. Jika kosong atau tidak valid, pesan kesalahan ditambahkan ke array.
- **Menampilkan Kesalahan:** Jika ada kesalahan dalam validasi, pesan kesalahan akan ditampilkan. Jika tidak ada kesalahan, data berhasil dikirim dan ditampilkan.
- **Keamanan:** Menggunakan `htmlspecialchars()` untuk menghindari serangan XSS dengan membersihkan input yang dikirimkan.

5	Kembangkan file bernama form_validasi.php di dalam folder dasarWeb. Ketikkan kode pada langkah 6 di dalam form_validasi.php
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	Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_validasi.php
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.2)</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Data berhasil dikirim: Nama = Naditya, Email = nadityaprastia206@gmail.com</p> </div> <p>Output yang ditampilkan berada di halaman yang berbeda</p>
9	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)

Form Input dengan Validasi

Nama:

Email:

Data berhasil dikirim: Nama = Naditya, Email = kpps3507241015-0307@mail.com

```
<!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">
    <span id="nama-error" style="color: red;"></span><br>

    <label for="email">Email:</label>
    <input type="text" id="email" name="email">
    <span id="email-error" style="color: red;"></span><br>

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

  <div id="hasil" style="margin-top: 20px; color: green;"></div>

  <script>
    $(document).ready(function() {
      $("#myForm").submit(function(event) {
        event.preventDefault();

        var nama = $("#nama").val();
        var email = $("#email").val();
        var valid = true;

        // Validasi Nama
        if (nama === "") {
          $("#nama-error").text("Nama harus diisi.");
          valid = false;
        } else {
          $("#nama-error").text("");
        }

        if (email === "") {
          $("#email-error").text("Email harus diisi.");
        }
      });
    });
  </script>
</body>
</html>
```

```

        valid = false;
    } else {
        $("#email-error").text("");
    }

    if (valid) {
        $.ajax({
            url: "proses_validasi.php",
            type: "POST",
            data: { nama: nama, email: email },
            success: function(response) {
                $("#hasil").html(response);
            },
            error: function() {
                $("#hasil").html("Terjadi kesalahan saat mengirim
data.");
            }
        });
    }
});
});
</script>
</body>
</html>

```

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)

Form Input dengan Validasi

Nama:

Email:

Password:

Password harus minimal 8 karakter.

Form Input dengan Validasi

Nama:

Email:

Password:

Data berhasil dikirim: Nama Naditya, Email = aditandino@gmail.com

1. Form Input:

- Form ini meminta pengguna untuk mengisi nama, email, dan password.
- Setiap input memiliki elemen `` untuk menampilkan pesan kesalahan validasi di bawahnya.

2. Validasi dengan jQuery:

- Saat form disubmit, jQuery akan mencegah pengiriman form secara default dan melakukan validasi.
- **Nama:** Memeriksa apakah diisi.
- **Email:** Memeriksa apakah diisi.
- **Password:** Memeriksa apakah panjangnya minimal 8 karakter.
- Jika validasi gagal, pesan kesalahan ditampilkan, dan pengiriman form dihentikan.

3. Pengiriman Data dengan AJAX:

- Jika semua input valid, data form dikirim ke file `proses_validasi.php` menggunakan AJAX.
- Respons dari server ditampilkan di `div #hasil`.

Kode PHP:

1. Penerimaan Data:

- Mengambil data yang dikirim dari form (nama, email, password) dan melakukan validasi tambahan.
- Memeriksa kevalidan nama dan email serta panjang password.

2. Menangani Kesalahan:

- Jika ada kesalahan, pesan kesalahan ditampilkan.
- Jika tidak ada kesalahan, pesan konfirmasi ditampilkan.