

JOBSHEET 7

PHP – FORM PROCESSING



Dibuat oleh: Silmy Maulia Dewi
Kelas: SIB 2E
NIM: 2241760090

**D4 Sistem Informasi Bisnis
Teknologi Informasi
Politeknik Negeri Malang
2024**



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> <p>Anda belum dewasa atau variabel 'umur' tidak ditemukan.</p> <p>Hasil:</p> <p>Jawab: Penggunaan isset() pada file tersebut bertujuan untuk memeriksa apakah variabel \$umur sudah didefinisikan dan memiliki nilai atau belum sebelum mencoba mengakses nilainya. Jika variabel \$umur sudah didefinisikan dan nilainya lebih besar atau sama dengan 18, maka pesan "Anda sudah dewasa." akan ditampilkan. Jika variabel \$umur belum didefinisikan atau tidak memiliki nilai, maka pesan "Anda belum dewasa atau variabel 'umur' tidak ditemukan." akan ditampilkan.</p>
5	<p>Tambahkan isi dari file isset.php tersebut dengan kode di bawah ini.</p> <pre><code>\$data = array("nama" => "Jane", "usia" => 25); if (isset(\$data["nama"])) { echo "Nama: " . \$data["nama"]; } else { echo "Variabel 'nama' tidak ditemukan dalam array."; }</code></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> <p>Nama: Jane</p> <p>Hasil:</p> <p>Jawab: Setelah variabel diisi dengan array dan dipanggil menggunakan echo, maka nilai dalam variabel akan muncul. Penggunaan isset() pada file tersebut bertujuan untuk memeriksa apakah kunci "nama" telah ada di dalam array \$data sebelum mencoba mengakses nilainya</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	<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> <p>Hasil: Array tidak terdefinisi atau kosong.</p> <p>Jawab: <code>empty()</code> digunakan untuk memeriksa apakah sebuah variabel kosong atau tidak terdefinisi. Karena array tidak didefinisikan, maka pesan yang tercetak adalah pada kondisi <code>empty</code></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> <p>Hasil: Array tidak terdefinisi atau kosong.</p> <p>Hasil: Variabel tidak terdefinisi atau kosong.</p> <p>Jawab: Pemahaman saya tentang penggunaan <code>empty()</code> pada file tersebut adalah sebagai berikut:</p> <ul style="list-style-type: none"> • Pengecekan Array Kosong atau Tidak Terdefinisi: Pada langkah kedua (// langkah 2), <code>empty(\$myArray)</code> digunakan untuk memeriksa apakah array <code>\$myArray</code> kosong atau tidak terdefinisi. Jika <code>\$myArray</code> kosong atau tidak terdefinisi, pesan "Array tidak terdefinisi atau kosong." akan ditampilkan. • Pengecekan Variabel Tidak Terdefinisi atau Kosong: Pada langkah keempat (// langkah 4), <code>empty(\$nonExistentVar)</code> digunakan untuk memeriksa apakah variabel <code>\$nonExistentVar</code> tidak terdefinisi atau kosong.

	Karena \$nonExistentVar tidak pernah didefinisikan sebelumnya, maka kondisi di dalam blok if akan dievaluasi sebagai benar.

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> <!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> <?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> <p>Form Input PHP</p> <p>Nama: <input type="text" value="Silmy Maulia Dewi"/></p> <p>Email: <input type="text" value="silmy.smd@gmail.com"/></p> <p>Hasil: <input type="button" value="Submit"/></p> <p>Nama: Silmy Maulia Dewi Email: silmy.smd@gmail.com</p> <p>Jawab: Form menggunakan metode POST untuk mengirimkan data ke server, karena kondisi <code>if (\$_SERVER["REQUEST_METHOD"] == "POST")</code> mengecek apakah data dikirimkan melalui metode POST</p>
5	<p>Buat satu file baru bernama form_self.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form_self.php</p> <pre> <!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; ?>"> <?php echo \$namaErr; ?>

 <input type="submit" name="submit" value="Submit"> </form> </body> </html> </pre>
6	<p>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)</p> <p>Form Input PHP</p> <p>Data berhasil disimpan!</p> <p>Nama: <input type="text" value="Silmy Maulia Dewi"/></p> <p>Hasil: <input type="button" value="Submit"/></p> <p>Jawab: Kode tersebut merupakan contoh implementasi form input sederhana yang digunakan untuk meminta pengguna memasukkan nama. Setelah form disubmit, data yang dimasukkan akan divalidasi oleh kode PHP. Jika nama tidak diisi, pesan kesalahan akan ditampilkan di sebelah input field, sedangkan jika nama telah diisi, pesan sukses akan muncul. Form menggunakan metode POST untuk pengiriman data ke server, dengan aksi yang ditujukan ke halaman itu sendiri</p>

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	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1)</p> <p>Hasil input: silmy maulia</p> <p>Form Input PHP</p> <p>Hasil: Input: <input type="text"/> <input type="submit" value="Submit"/></p> <p>Jawab: Penanganan formulir tetap terjaga dengan baik, dengan menggunakan metode POST dan menghindari serangan XSS dengan menggunakan fungsi <code>htmlspecialchars()</code>. Dalam kode tersebut, meskipun <code>htmlspecialchars()</code> digunakan untuk menghindari serangan XSS, potensi serangan HTML injection masih ada jika input pengguna tidak disaring dengan benar sebelum ditampilkan kembali di halaman web. HTML injection terjadi ketika penyerang menyisipkan kode HTML atau JavaScript berbahaya yang dieksekusi oleh browser pengguna saat ditampilkan kembali</p>
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 localhost/dasarWeb/html_aman.php
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.2)</p> <div> <div> <p>Hasil input: silmy Email valid: silmy.smd@gmail.com</p> <p>Form Input PHP</p> <p>Input: <input type="text"/></p> <p>Email: <input type="text"/></p> <p>Hasil: <input type="submit" value="Submit"/></p> </div> <div> <p>Hasil input: silmy Email tidak valid</p> <p>Form Input PHP</p> <p>Input: <input type="text"/></p> <p>Email: <input type="text"/></p> <p>Hasil: <input type="submit" value="Submit"/></p> </div> </div> <p>Jawab: Dengan menambahkan input untuk field "email" pada form HTML, pengguna sekarang dapat memasukkan alamat email mereka. Kemudian, saat form disubmit, nilai dari input email akan dikirimkan melalui metode POST dan dapat diakses di dalam PHP menggunakan \$_POST['email']. Dengan demikian, kita dapat melakukan validasi terhadap alamat email yang dimasukkan oleh pengguna dan melakukan pengolahan data lebih lanjut sesuai kebutuhan.</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
 - 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 regex.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam regex.php

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	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> <p>Hasil: Huruf kecil ditemukan</p> <p>Jawab: Kode tersebut menggunakan regular expression (regex) untuk mencocokkan apakah terdapat huruf kecil dalam variabel \$text. Pada baris pertama, kita mendefinisikan pola regex menggunakan '/[a-z]/', yang berarti mencocokkan setiap karakter huruf kecil dari a hingga z. Kemudian, kita menggunakan fungsi preg_match() untuk mencocokkan pola tersebut dengan teks yang diberikan (\$text). Jika terdapat setidaknya satu huruf kecil dalam teks, maka preg_match() akan mengembalikan nilai true, dan pesan "Huruf kecil ditemukan" akan ditampilkan. Jika tidak ada huruf kecil yang ditemukan, pesan "Tidak ada huruf kecil!" akan ditampilkan</p>
5	Ketikkan kode tambahan pada langkah 6 di dalam regex.php
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> <p>Hasil: Cocokkan: 123</p> <p>Jawab: Kode tersebut menggunakan regular expression (regex) untuk mencocokkan apakah terdapat satu atau lebih digit dalam variabel \$text. Pada baris pertama, kita mendefinisikan pola regex menggunakan '/[0-9]+/', yang berarti mencocokkan setiap digit dari 0 hingga 9 dan memungkinkan satu atau lebih kemunculan. Kemudian, kita menggunakan fungsi preg_match() untuk mencocokkan pola tersebut dengan teks yang diberikan (\$text). Jika terdapat setidaknya satu digit dalam teks, preg_match() akan mengembalikan nilai true, dan angka yang cocok akan disimpan dalam array \$matches. Pesan "Cocokkan: " akan ditampilkan bersama dengan angka yang cocok (dalam kasus ini, "123"). Jika tidak ada digit yang cocok, pesan "Tidak ada yang cocok!" akan ditampilkan</p>
9	Ketikkan kode tambahan pada langkah 10 di dalam regex.php

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> <p>Hasil: I like banana pie.</p> <p>Jawab: Kode tersebut menggunakan fungsi preg_replace() untuk mengganti setiap kemunculan pola regex yang cocok dalam variabel \$text dengan string pengganti yang diberikan dalam variabel \$replacement. Dalam contoh ini, pola regex yang digunakan adalah '/apple/', yang mencocokkan kata "apple". Kemudian, kita menetapkan string pengganti menjadi 'banana'. Jika pola regex ditemukan dalam teks, fungsi preg_replace() akan mengganti setiap kemunculan pola tersebut dengan string pengganti. Hasilnya, setiap kemunculan "apple" dalam teks akan diganti dengan "banana". Kemudian, hasilnya akan ditampilkan dengan menggunakan perintah echo.</p>
13	Ketikkan kode tambahan pada langkah 14 di dalam regex.php
14	<pre> \$pattern = '/go*d/'; // Cocokkan "god", "good", "gooooo", dll. \$text = 'god is good.'; if (preg_match(\$pattern, \$text, \$matches)) { echo "Cocokkan: " . \$matches[0]; } else { echo "Tidak ada yang cocok!"; } </pre>
15	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php
16	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.4)</p> <p>Hasil: Cocokkan: god</p> <p>Jawab: Pola regex dalam kode di atas adalah '/god/', yang mencocokkan string yang terdiri dari karakter "g", diikuti oleh satu atau lebih karakter "o", dan diakhiri dengan karakter "d". Tanda "" menunjukkan bahwa karakter "o" dapat muncul nol kali atau lebih. Dalam teks "god is good.", pola regex ini cocok dengan kata "god" dan "good". Hasilnya, jika pola regex ditemukan dalam teks, maka akan mencetak kata yang cocok, yaitu "god". Jika tidak ada yang cocok, akan mencetak pesan "Tidak ada yang cocok!"</p>
17	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)

	<pre> \$pattern = '/go?d/'; //cocokkan "god", "good" \$text = 'god is good.'; if(preg_match(\$pattern, \$text, \$matches)){ echo "Cocokkan: ", \$matches[0]; }else{ echo "Tidak ada yang cocok!"; } </pre> <p>Kode program:</p> <p>Hasil: Cocokkan: god</p> <p>Jawab: Penggunaan regex dengan ? memiliki arti bahwa karakter sebelumnya dapat muncul 0 atau 1 kali dalam teks yang dicocokkan. Ini berbeda dengan penggunaan *, di mana karakter sebelumnya dapat muncul 0 atau lebih kali dalam teks yang dicocokkan. Dengan kata lain, penggunaan ? membatasi karakter sebelumnya untuk hanya muncul sekali atau tidak sama sekali, sedangkan penggunaan * memungkinkannya muncul berkali-kali atau tidak sama sekali. Misalnya, /go?d/ akan cocok dengan "god" dan "good", sedangkan /go*d/ akan cocok dengan "god", "good", "goood", dan seterusnya.</p>
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> <pre> //langkah 18 \$pattern = '/go{1,2}d/'; // Cocokkan "god", "good", "go \$text = 'god is good.'; if(preg_match(\$pattern, \$text, \$matches)){ echo "Cocokkan: ", \$matches[0]; } else { echo "Tidak ada yang cocok!"; } </pre> <p>Kode program:</p> <p>Hasil: Cocokkan: god</p> <p>Jawab: Pada kode tersebut, {1,2} menunjukkan bahwa karakter sebelumnya (yaitu "o") dapat muncul antara 1 hingga 2 kali. Sehingga, regex akan cocok dengan kata-kata seperti "god", "good", "goood", dan sejenisnya</p>

Praktikum 6 : Form Lanjut

Langkah-langkah Praktikum :

Langkah	Keterangan
1	Buat satu file baru bernama form_lanjut.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form_lanjut.php

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 localhost/dasarWeb/form_lanjut.php
4	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 6.1)

	<p>Form Contoh</p> <p>Pilih Buah: <input type="text" value="Mangga"/></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 type="radio"/> Laki-laki</p> <p><input checked="" type="radio"/> Perempuan</p> <p>Hasil: <input type="button" value="Submit"/></p> <p>Anda memilih buah: mangga</p> <p>Warna favorit Anda: biru</p> <p>Jenis kelamin Anda: perempuan</p> <p>Jawab: Kode tersebut merupakan contoh form HTML yang mengizinkan pengguna untuk memilih buah, warna favorit, dan jenis kelamin mereka. Setelah formulir dikirim, PHP digunakan untuk menangani data yang dikirimkan melalui metode POST. Data buah, warna favorit, dan jenis kelamin yang dipilih oleh pengguna ditampilkan kembali sebagai respons di halaman yang sama setelah formulir dikirimkan</p>
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

Anda memilih buah: mangga

Warna favorit Anda: hijau

Jenis kelamin Anda: perempuan

Hasil:

Jawab: Kesimpulan dari kode tersebut adalah bahwa halaman HTML tersebut berisi sebuah form yang menggunakan jQuery untuk mengirim data secara asynchronous (tanpa perlu memuat ulang halaman). Ketika form tersebut disubmit, data form dikumpulkan menggunakan metode `serialize()` jQuery, dan kemudian dikirim ke server menggunakan AJAX dengan bantuan fungsi `$.ajax()`. Hasil dari proses tersebut akan ditampilkan di dalam sebuah div dengan id "hasil" tanpa memuat ulang halaman secara keseluruhan. Pada sisi server, file "proses_lanjut.php" mengelola data yang diterima dan memberikan respons yang sesuai.

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	<p>Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_validasi.php</p>
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.1)</p> <p>Hasil: Data berhasil dikirim: Nama = Silmy Maulia Dewi, Email = silmy.smd@gmail.com</p> <p>Jawab: Kode tersebut merupakan contoh validasi formulir dengan PHP. Saat formulir dikirim, PHP memeriksa apakah nama dan email telah diisi. Jika tidak, pesan kesalahan akan ditampilkan. Jika email yang dimasukkan tidak valid, pesan kesalahan lain akan ditampilkan. Jika tidak ada kesalahan validasi, pesan sukses akan ditampilkan bersama dengan data yang dimasukkan. Formulir itu sendiri sederhana, terdiri dari field nama dan email, dan saat di-submit, akan memanggil halaman yang sama untuk validasi dan pemrosesan data.</p>
5	<p>Kembangkan file bernama form_validasi.php di dalam folder dasarWeb. Ketikkan kode pada langkah 6 di dalam form_validasi.php</p>

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) Hasil: Data berhasil dikirim: Nama = Silmy Maulia Dewi, Email = silmy.smd@gmail.com Jawab: Kode tersebut merupakan sebuah form HTML yang menggunakan JavaScript, khususnya jQuery, untuk melakukan validasi input sebelum pengiriman. Ketika pengguna mencoba mengirimkan formulir, JavaScript memeriksa apakah kedua input (nama dan email) telah diisi. Jika ada input yang kosong, pesan kesalahan akan ditampilkan tepat di bawah input yang bersangkutan, dengan warna teks merah untuk menyoroti kesalahan. Selain itu, jika formulir diserahkan dan ada input yang tidak valid, pengiriman formulir akan dicegah. Ini membantu memastikan bahwa data yang dikirimkan sesuai dengan format yang diharapkan sebelum diproses lebih lanjut.</p>
9	<p>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) Kode program:</p>

```

<!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>

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

        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("");
        }

        // Validasi email
        if (email === ""){
          $("#email-error").text("Email harus diisi.");
          valid = false;
        }else{
          if (valid) {
            $.ajax({
              url: "proses_validasi_ajax.php",
              type: "POST",
              data: $("#myForm").serialize(), // Mengirim data form menggunakan
              success: function(response){
                // Menampilkan hasil dari server
                alert(response);
              }
            });
          }
        }
      });
    });
  </script>
</body>
</html>

```

Jawab: Dalam kode tersebut, ketika formulir disubmit, script JavaScript mencegah pengiriman formulir secara default dengan menggunakan `event.preventDefault()`. Kemudian, data formulir diambil menggunakan metode `serialize()` dan dikirim ke server menggunakan fungsi `$.ajax()`. Setelah menerima respons dari server, respons tersebut ditampilkan ke pengguna, dalam hal ini menggunakan fungsi `alert()`. Dengan menggunakan Ajax, proses validasi dapat dilakukan secara dinamis tanpa perlu me-refresh halaman, meningkatkan pengalaman pengguna.

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 program:

```

<!-- penambahan input field password -->
<label for="password">Password:</label>
<input type="password" id="password" name="password">
<span id="password-error" style="color: red;"></span>
<br>

// Validasi password
var password = $("#password").val();
if (password.length < 8) {
    $("#password-error").text("Password harus memiliki minimal 8 karakter.");
    valid = false;
} else {
    $("#password-error").text("");
}

//validasi password
$password = $_POST["password"];
$errors = array();
if (strlen($password) < 8) {
    $errors[] = "Password harus memiliki minimal 8 karakter.";
}

```

Form Input dengan Validasi

Nama:

Email:

Password:

Hasil:

Data berhasil dikirim: Nama = Silmy Maulia Dewi, Email = silmy.smd@gmail.com

Jawab: Penambahan kode tersebut memberikan validasi pada input password untuk memastikan bahwa pengguna memasukkan password dengan panjang minimal 8 karakter. Validasi dilakukan baik di sisi klien (menggunakan jQuery) maupun di sisi server (menggunakan PHP). Pada sisi klien, pesan kesalahan akan ditampilkan di bawah input password jika panjang password kurang dari 8 karakter. Sedangkan pada sisi server, pesan kesalahan akan disimpan dalam array jika panjang password kurang dari 8 karakter. Dengan demikian, pengguna akan mendapatkan pemberitahuan jika password yang mereka masukkan tidak memenuhi persyaratan minimal.