



Nama : Dyah Nanda Ayu P
No/NIM : 08 / 2241760017
Kelas : SIB 2C

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> <p>Jawab:</p> <p>Penggunaan isset dalam contoh kode tersebut bertujuan untuk memeriksa apakah variabel \$umur telah diinisialisasi atau tidak sebelum digunakan. Jika \$umur sudah diinput dengan nilai >= 18, maka akan menampilkan “Anda sudah dewasa”. Jika tidak, maka menampilkan “Anda belum dewasa atau variabel ‘umur’ tidak ditemukan.”. karena di kode tersebut belum diinputkan nilai variabel umur, maka di website outputnya seperti berikut:</p> <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> <p>Jawab:</p> <p>Kode itu merupakan proses pendefinisian array yang memiliki 2 nilai yaitu nama dan usia. Setelah itu diperiksa apakah “Nama” telah diinisialisasi. jika sudah diinisialisasi, menampilkan "nama" ditambah nama. Jika belum, menampilkan "Variabel 'nama' tidak ditemukan dalam array". Karena dalam kode diatas variabel \$nama sudah diinisialisasi, maka output nya sebagai berikut:</p> <p>Ditambahkan
 supaya berada pada baris yang berbeda.</p> <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	<pre><code>\$myArray = array(); // Array kosong if (empty(\$myArray)) { echo "Array tidak terdefinisi atau kosong."; } else { echo "Array terdefinisi dan tidak kosong."; }</code></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>Jawab:</p> <p>Kode diatas mendeklarasikan variabel <code>\$myArray</code> dengan nilai kosong (tidak terdefinisi). Lalu dilakukan pemeriksaan apakah array kosong. Fungsi <code>empty()</code> digunakan untuk memeriksa apakah sebuah variabel kosong. Jika array kosong, maka menampilkan "Array tidak terdefinisi atau kosong." Jika array tidak kosong, maka menampilkan "Array terdefinisi dan tidak kosong.". Karena dalam kode diatas, array kosong. Maka output dalam website sebagai berikut:</p> <p>Array tidak terdefinisi atau kosong.</p>
4	<p>Tambahkan isi dari file <code>empty.php</code> tersebut dengan kode di bawah ini.</p> <pre><code>if (empty(\$nonExistentVar)) { echo "Variabel tidak terdefinisi atau kosong."; } else { echo "Variabel terdefinisi dan tidak kosong."; }</code></pre>

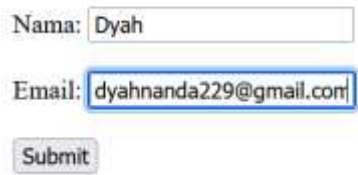

5	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/empty.php buat hasil tidak menjadi 1 baris, jadi tampilan dari echo harus terpisah. Apa yang anda pahami dari penggunaan empty pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.2)</p> <p>Jawab:</p> <p>Tambahan kode diatas digunakan untuk mengecek, apakah variabel \$nonExistentVar sudah diinisialisasi atau tidak. Jika sudah diinisialisasi, maka menampilkan "Variabel tidak terdefinisi atau kosong.". Jika tidak diinisialisasi, maka menampilkan "Variabel terdefinisi dan tidak kosong.". Karena pada kode belum diinisialisasi, maka tampilan pada website sebagai berikut: Ditambahkan
 supaya berada pada baris yang berbeda.</p> <pre> Array tidak terdefinisi atau kosong. Variabel tidak terdefinisi atau kosong. </pre>

Praktikum Bagian 3 : Form Input PHP

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

	<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>Jawab:</p> <p>File form digunakan untuk mengumpulkan data dari pengguna, yaitu nama dan email. Pengguna harus mengisi nama dan email. Setelah pengguna mengisi, data tersebut akan dikirim ke file 'proses_form.php' untuk di proses. Kedua input itu memiliki atribut 'required' yang berarti wajib diisi sebelum form dikirim. Setelah itu data ditampilkan kembali di website sebagai konfirmasi bahwa data berhasil dikirim.</p> <p>Form Input PHP</p>  <p>Tampilan setelah data di submit.</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>

```

<!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)

Jawab:

Halaman web form_self.php menyediakan formulir input PHP untuk memasukkan nama. Formulir ini memvalidasi input nama dan menampilkan pesan error jika kosong. Jika input valid, pesan "Data berhasil disimpan!" akan muncul. Penggunaan htmlspecialchars(\$_SERVER["PHP_SELF"]) pada atribut action memungkinkan formulir diproses di halaman yang sama, menampilkan pesan error atau hasil tanpa memuat ulang halaman.

Tampilan pada website:

Form Input PHP

6

Nama:

Setelah submit Query

Form Input PHP

Data berhasil disimpan!

Nama:

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) Jawab: Tambahan kode sebagai berikut:</p> <pre><?DOCTYPE html> <html> <head> <title>Form Input PHP</title> </head> <body> <h2>Form Input PHP</h2> <?php \$input = ""; // Memproses formulir saat disubmit if (\$_SERVER["REQUEST_METHOD"] == "POST") { // Mendapatkan nilai input dari formulir dan melindunginya dari serangan XSS \$input = \$_POST['input']; \$input = htmlspecialchars(\$input, ENT_QUOTES, "UTF-8"); } <!-- Form untuk memasukkan input --> <form method="post" action="<?php echo htmlspecialchars(\$_SERVER["PHP_SELF"]); ?>"> Input: <input type="text" name="input" value="<?php echo \$input; ?>"> <input type="submit" name="submit" value="submit"> </form> <!-- Hasil input yang telah dimasukkan --> <?php if (\$_SERVER["REQUEST_METHOD"] == "POST") { echo "<h2>Input yang dimasukkan:</h2>"; echo "<p> . \$input . "</p>"; } ?> </body> </html></pre>

	<p>Tampilan pada website:</p> <p>Form Input PHP</p> <p>Input: <input type="text" value="Halo, saya Dyah."/> <input type="submit" value="Submit"/></p> <p>Input yang Dimasukkan:</p> <p>Halo, saya Dyah.</p> <p>Penjelasan: Dalam kode di atas, saya membuat sebuah formulir dengan satu kolom input. Ketika formulir disubmit, nilai input akan ditangkap oleh PHP, dan kemudian diolah menggunakan htmlspecialchars(). Setelah itu, nilai input akan ditampilkan kembali di halaman web.</p>
5	Ketikkan kode tambahan pada langkah 6 di dalam html_aman.php
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 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> <p>Jawab:</p> <p>Penambahan Kode:</p> <pre><!DOCTYPE html> <html> <head> <title>Form Input PHP</title> </head> <body> <h2>Form Input PHP</h2> <?php \$input = ""; \$email = ""; // Memproses formulir saat disubmit if (\$_SERVER["REQUEST_METHOD"] == "POST") { // Mendapatkan nilai input dari formulir dan melindunginya dari serangan XSS \$input = \$_POST['input']; \$input = htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8'); // Mendapatkan nilai email dari formulir \$email = \$_POST['email']; \$email = htmlspecialchars(\$email, ENT_QUOTES, 'UTF-8'); } <?> <!-- Formulir untuk memasukkan input --> <form method="post" action="<?php echo htmlspecialchars(\$_SERVER["PHP_SELF"]); ?>"> Input: <input type="text" name="input" value="<?php echo \$input; ?>"> Email: <input type="text" name="email" value="<?php echo \$email; ?>"> <input type="submit" name="submit" value="Submit"> </form></pre>

```

<!-- Hasil input yang telah dimasukkan -->
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    echo "<h2>Input yang Dimasukkan:</h2>";
    echo "<p>Input: " . $input . "</p>";
    echo "<p>Email: " . $email . "</p>";
}

// Memeriksa apakah input adalah email yg valid
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
        echo "<p>Alamat email yang dimasukkan: " . $email . "</p>";
        // Lanjutkan dengan pengolahan email yang aman
    } else {
        echo "<p>Alamat email tidak valid.</p>";
        // Tangani input yang tidak valid
    }
}
?>
</body>
</html>

```

Tampilan pada website:

Form Input PHP

Input: Email:

Input yang Dimasukkan:

Input: Halo, saya Dyah.

Email: dyahnanda229@gmail.com

Alamat email yang dimasukkan: dyahnanda229@gmail.com

Penjelasan:

Beberapa hal yang perlu ditambahkan sebagai berikut:

- Menambahkan field input untuk email di dalam formulir HTML.
- Mengambil nilai dari input email dan melindunginya dari serangan XSS dengan menggunakan htmlspecialchars.
- Menampilkan nilai email yang dimasukkan oleh pengguna.
- Memeriksa apakah email yang dimasukkan valid menggunakan filter_var dengan FILTER_VALIDATE_EMAIL.

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:
 - *: 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 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	Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/regex.php</code>
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.1)</p> <p>Jawab:</p> <p>Kode PHP yang diberikan digunakan untuk mencari pola tertentu dalam string dan menampilkan pesan berdasarkan apakah pola tersebut ditemukan atau tidak. Karena pola ditemukan, maka output nya sebagai berikut:</p> <hr/> <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> <p>Jawab:</p> <p>Kode ini menggunakan regex untuk mencari semua digit angka dalam string. Jika pola ditemukan, kode akan menampilkan string yang cocok dengan regex. Jika pola tidak ditemukan, kode akan menampilkan string "Tidak ada yang cocok!".</p> <p>Tampilan pada website:</p> <hr/> <p>Huruf kecil ditemukan! Cocokkan: 123</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 <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> <p>Jawab:</p> <p>Kode ini digunakan untuk mengganti kata "apple" dalam string \$text dengan kata "banana". Fungsi <code>preg_replace()</code> mencari pola yang ditentukan oleh \$pattern dalam string \$text dan menggantinya dengan teks pengganti yang ditentukan oleh \$replacement. Hasil penggantian disimpan dalam variabel \$new_text, yang kemudian dicetak. Tampilan pada website sebagai berikut</p> <p>Huruf kecil ditemukan! Cocokkan: 123 I like banana pie.</p>
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	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/regex.php</code>
16	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.4)</p> <p>Jawab:</p> <p>Kode ini mencari pola ekspresi reguler yang mencocokkan kata "god" dengan variasi seperti "good", "goood", dll., dalam string \$text yang berisi "god is good.". Fungsi <code>preg_match()</code> digunakan untuk mencari pola ini, dan jika ditemukan, menampilkan kata yang cocok. Seperti output dibawah ini:</p> <p>Huruf kecil ditemukan! Cocokkan: 123 I like banana pie. Cocokkan: god</p>

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> <p>Jawab:</p> <p>Jika mengubah \$pattern menjadi \$pattern = '/go?d/'; dengan menambahkan ? setelah o, ini berarti kita mengubah pola mencari "god" dengan variasi seperti "god" (0 atau 1 kali 'o')</p> <p>Kode Program:</p> <pre>// soal 5.5 \$pattern = '/go?d/'; // cocokkan "god" dan bukan "good" "goood" karena o tidak boleh muncul lebih dr 1x \$text = 'god is good.'; // mendeklarasikan \$text dengan nilai tersebut if (preg_match(\$pattern, \$text, \$matches)){ // menggunakan fungsi preg_match untuk mencari pola echo "Cocokkan: " . \$matches[0] . "
"; // jika ditemukan } else { echo "Tidak ada yang cocok
"; // jika tidak ditemukan }</pre> <p>Untuk Output seperti ini:</p> <p>Huruf kecil ditemukan!</p> <p>Cocokkan: 123</p> <p>I like banana pie.</p> <p>Cocokkan: god</p> <p>Cocokkan: god</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> <p>Jawab:</p> <p>mengubah \$pattern menjadi \$pattern = '/go{1,3}d/';. Penambahan {1,3} setelah o dalam ekspresi reguler mengubah pola mencari kata "god" dengan variasi seperti "god" (1 hingga 3 kali 'o')</p> <p>Kode Program:</p> <pre>// soal 5.6 \$pattern = '/go{1,3}d/'; // cocokkan "god" dan bukan "good" "goood" karena o tidak boleh muncul lebih dr 1x \$text = 'god is good.'; // mendeklarasikan \$text dengan nilai tersebut if (preg_match(\$pattern, \$text, \$matches)){ // menggunakan fungsi preg_match untuk mencari pola echo "Cocokkan: " . \$matches[0] . "
"; // jika ditemukan } else { echo "Tidak ada yang cocok
"; // jika tidak ditemukan }</pre> <p>Tampilan pada website:</p> <p>Huruf kecil ditemukan!</p> <p>Cocokkan: 123</p> <p>I like banana pie.</p> <p>Cocokkan: god</p> <p>Cocokkan: god</p> <p>Cocokkan: god</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	<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> <p>Jawab:</p> <p>Kode PHP di atas mengambil data dari form yang dikirimkan melalui metode POST, memeriksa apakah ada input dengan nama 'warna', dan menampilkan data yang diterima. Jika input 'warna' ada, maka akan menampilkan warna favorit yang dipilih. Jika tidak, maka akan menampilkan pesan bahwa tidak ada warna favorit yang dipilih.</p> <p>Tampilan pada website:</p>

	<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><input type="button" value="submit"/></p> <p>Setelah submit:</p> <p>Anda memilih buah: mangga</p> <p>Warna favorit anda: biru</p> <p>Jenis kemalin Anda: perempuan</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)
Jawab:

	<p>Dari kode program yang diberikan, penambahan utama adalah penggunaan jQuery untuk mengirimkan data form melalui AJAX, yang memungkinkan pengiriman data tanpa perlu memuat ulang halaman.</p> <p>Tampilan pada website:</p> <p>Form Contoh</p> <p>Pilih Buah: <input type="text" value="Apel"/></p> <p>Pilih Warna Favorit:</p> <p><input checked="" type="checkbox"/> Merah <input type="checkbox"/> Biru <input type="checkbox"/> Hijau</p> <p>Pilih Jenis Kelamin:</p> <p><input type="radio"/> Laki-laki <input checked="" type="radio"/> Perempuan</p> <p><input type="button" value="submit"/></p> <p>Tampilan setelah di submit:</p> <p>Anda memilih buah: apel Warna favorit anda: merah Jenis kemalin Anda: perempuan</p>
--	---

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)) { // Lanjutkan dengan pemrosesan data jika semua validasi berhasil // Misalnya, menyimpan data ke database atau mengirim email echo "Data berhasil dikirim: Nama = \$nama, Email = \$email"; } else { // Tampilkan pesan kesalahan foreach (\$errors as \$error) { echo \$error . "
"; } } } </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>Jawab:</p> <p>Kode tersebut merupakan formulir sederhana dengan dua input teks untuk mengumpulkan nama dan alamat email dari pengguna. Setelah pengguna mengisi formulir, data akan diproses oleh server menggunakan file proses_validasi.php untuk validasi sebelum dikirimkan. Formulir menggunakan metode POST untuk mengirimkan data ke server saat tombol "Submit" ditekan.</p> <p>Tampilan pada website:</p> <h3>Form Input dengan Validasi</h3>  <p>Setelah di submit:</p> <p>Data berhasil dikirim: Nama = Dyah, Email = dyahnanda229@gmail.com</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) Jawab: Script JavaScript yang ditambahkan bertujuan untuk melakukan validasi pada formulir sebelum pengirimannya. Ketika formulir disubmit, script ini akan mendapatkan nilai dari input nama dan email. Tampilan pada website:</p> <h2 style="text-align: center;">Form Input dengan Validasi</h2> <p>Nama: <input type="text" value="dyah"/></p> <p>Email: <input type="text" value="dyahnanda229@gmail.com"/></p> <p><input type="button" value="Submit"/></p> <p>Setelah di submit: Data berhasil dikirim: Nama = dyah, Email = dyahnanda229@gmail.com</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) Jawab:</p>

Tampilan Kode:

```
$(document).ready(function () {
    $("#myForm").submit(function (event) { // Ketika formulir disubmit
        event.preventDefault(); // Mencegah pengiriman formulir default

        var nama = $("#nama").val(); // Mendapatkan nilai input nama
        var email = $("#email").val(); // Mendapatkan nilai input email
        var valid = true; // Menetapkan variabel validitas awal

        if (nama == "") { // Jika input nama kosong
            $("#nama-error").text("Nama harus diisi!"); // Menampilkan pesan kesalahan
            valid = false; // Menetapkan valid = false
        } else {
            $("#nama-error").text(""); // Mengosongkan pesan kesalahan jika nama telah diisi
        }

        if (email == "") { // Jika input email kosong
            $("#email-error").text("Email harus diisi!"); // Menampilkan pesan kesalahan
            valid = false; // Menetapkan valid = false
        } else {
            $("#email-error").text(""); // Mengosongkan pesan kesalahan jika email telah diisi
        }

        if (!valid) { // Jika valid = false (ada kesalahan)
            return false; // Menghentikan eksekusi
        }

        // Mengirim data melalui AJAX
        $.ajax({
            type: 'POST',
            url: 'proses_validasi.php',
            data: $(this).serialize(),
            success: function(response) {
                // Menampilkan respon dari server
                alert(response);
            },
            error: function(jqXHR, textStatus, errorThrown) {
                // Menampilkan pesan error jika terjadi kesalahan
                alert('Terjadi kesalahan: ' + textStatus);
            }
        });
    });
});
```

Output setelah submit:

Form Input dengan Validasi

Nama:

Email:

🌐 localhost

Data berhasil dikirim: Nama = Dyah, Email =
dyahnanda229@gmail.com

Penjelasan:

Setelah diberi AJAX, perubahan utama dalam kode tersebut adalah cara data formulir dikirimkan ke server dan bagaimana respon dari server ditangani. Sebelumnya, formulir akan mengirimkan data ke server dengan cara tradisional, yaitu dengan me-refresh halaman. Namun, dengan AJAX, data dikirimkan tanpa perlu me-refresh halaman.

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)

Jawab:

Untuk menambahkan password, tambahkan kode berikut:

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

```
var nama = $("#nama").val(); // Mendapatkan nilai input nama
var email = $("#email").val(); // Mendapatkan nilai input email
var password = $("#password").val(); // mendapatkan nilai input password
var valid = true; // Menetapkan variabel validitas awal
```

```
if (password.length < 8) { // jika panjang password kurang dari 8
    $("#password-error").text("Password harus minimal 8 karakter!"); // output
    valid = false;
} else {
    $("#password-error").text("");
}
```

```
<!--
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = $_POST["nama"];
    $email = $_POST["email"];
    $password = $_POST["password"];
    $errors = array();

    // validasi nama
    if (empty($nama)) {
        $errors[] = "Nama harus diisi.";
    }

    // validasi email
    if (empty($email)) {
        $errors[] = "Email harus diisi.";
    } else if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $errors[] = "Format email tidak valid.";
    }

    // jika ada kesalahan validasi
    if (count($errors) > 0) {
        foreach ($errors as $error) {
            echo $error . "<br>";
        }
    }

    // lanjutkan dengan pemrosesan data jika semua validasi berhasil
    // simpan data ke database atau kirim email
    echo "Data berhasil dikirim: Nama = $nama, Email = $email";

    // validasi password
    if (strlen($password) < 8 || !preg_match('/[a-z]/', $password) || !preg_match('/[0-9]/', $password) || !preg_match('/[!@#$%^&*]/', $password)) {
        echo "Password harus minimal 8 karakter dan mengandung huruf kecil, huruf besar, angka, dan karakter khusus.";
    } else {
        // semua data yang valid
        echo "Data berhasil dikirim.";
    }
}
```

Tampilan pada website:

	<p>Penjelasan:</p> <p>Dengan kode ini, formulir sekarang memvalidasi password dengan ketentuan minimal 8 karakter dan memeriksa apakah password mengandung huruf besar, huruf kecil, angka, dan karakter khusus.</p>
--	--