



Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet-7: PHP - Form Processing

Mata Kuliah Desain dan Pemrograman Web

Pengampu: Tim Ajar Desain dan Pemrograman Web

Okttober 2023

Topik

- Konsep Form Processing 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 Iset

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

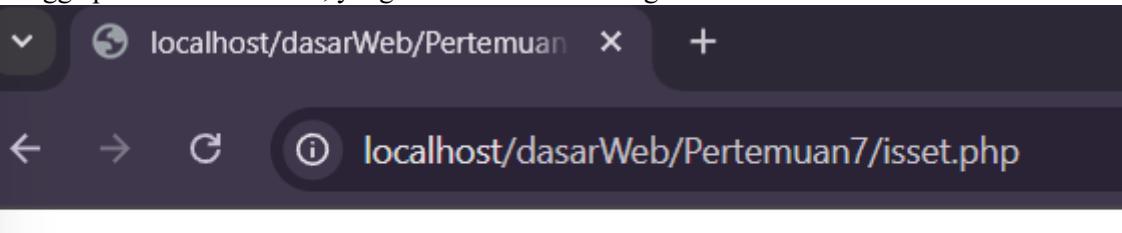
3	<pre>\$umur; if (isset(\$umur) && \$umur >= 18) { echo "Anda sudah dewasa."; } else { echo "Anda belum dewasa atau variabel 'umur' tidak ditemukan."; }</pre>
4	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/ isset.php. Apa yang anda pahami dari penggunaan isset pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 1.1)</p> <p>Jawaban:</p> <p>Ditampilkan pada browser yakni tulisan “Anda belum dewasa atau variabel ‘umur’ belum ditentukan. Hal ini dikarenakan pada kode program hanya dideklarasikan variabel umur tanpa menginisialisasikan nilai dari variabel tersebut. Hal ini menjadikan nilai dari variabel tersebut bernilai null. Sehingga pada blok if-else, yang dieksekusi adalah bagian else.</p> <p>Anda belum dewasa atau variabel 'umur' belum ditentukan.</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>

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)

Jawaban:

Ditampilkan pada browser, yakni nama: Jane. Hal ini dikarenakan pada kode program telah dideklarasikan dan diinisialisasikan nilai dari variabel \$data, yang dimana pada key nama, telah diisikan nilai Jane, sehingga pada kondisi if-else, yang dieksekusi adalah bagian if.

6



Anda belum dewasa atau variabel 'umur' belum ditentukan.

Nama: Jane

Function empty

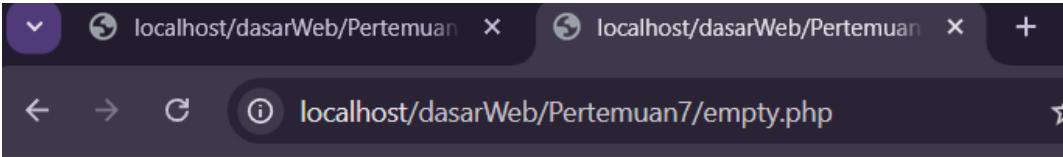
Fungsi **empty()** pada PHP digunakan untuk memeriksa apakah suatu variabel kosong atau tidak terdefinisi. Fungsi ini mengembalikan **true** jika variabel kosong atau tidak terdefinisi, dan **false** jika variabel memiliki nilai atau telah didefinisikan. Berikut adalah penjelasan dan contoh penggunaan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

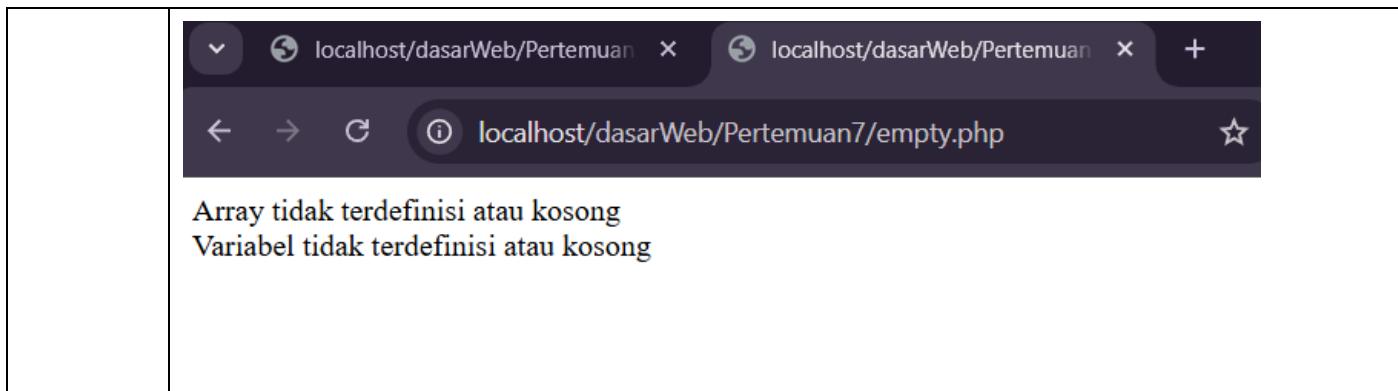
```
$nama = ""; // Variabel string kosong
if (empty($nama)) {
    echo "Nama tidak terdefinisi atau kosong.";
} else {
    echo "Nama terdefinisi dan tidak kosong.";
}
```

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

Praktikum 2. Function empty

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

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 localhost/dasarWeb/empty.php. Apa yang anda pahami dari penggunaan empty pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.1)</p> <p>Jawaban:</p> <p>Ditampilkan pada browser yakni tulisan “Array tidak terdefinisi atau kosong”. Hal ini dikarenakan pada kode program, telah dideklarasikan array kosong dengan nama \$myArray, namun tidak diisikan nilainya. Sehingga ketika dilakukan pengecekan dengan sintak if(empty(\$myArray)), maka nilai dari kondisi tersebut adalah true. Hal ini karena ia mengecek fungsi empty() pada \$myArray, sehingga dikembalikan nilai true.</p>  <p>Array tidak terdefinisi atau kosong</p>
4	<p>Tambahkan isi dari file empty.php tersebut dengan kode di bawah ini.</p> <pre>if (empty(\$nonExistentVar)) { echo "Variabel tidak terdefinisi atau kosong."; } else { echo "Variabel terdefinisi dan tidak kosong."; }</pre>
5	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/empty.php buat hasil tidak menjadi 1 baris, jadi tampilan dari echo harus terpisah. Apa yang anda pahami dari penggunaan empty pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.2)</p> <p>Jawaban:</p> <p>Ditampilkan pada browser, yakni tulisan “Variabel tidak terdefinisi atau kosong”. Hal ini dikarenakan pada kode program, diisikan pengecekan if(empty(\$nonExistentVar)), yang dimana, ia akan mengembalikan nilai true. Hal ini karena fungsi empty() tersebut mengecek apakah variabel nya tidak terdefinisi atau bernilai NULL atau bukan.</p>



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>
---	---

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

3

```
<?php  
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $nama = $_POST["nama"];  
    $email = $_POST["email"];  
  
    echo "Nama: " . $nama . "<br>";  
    echo "Email: " . $email;  
}  
?>
```

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)

Jawaban:

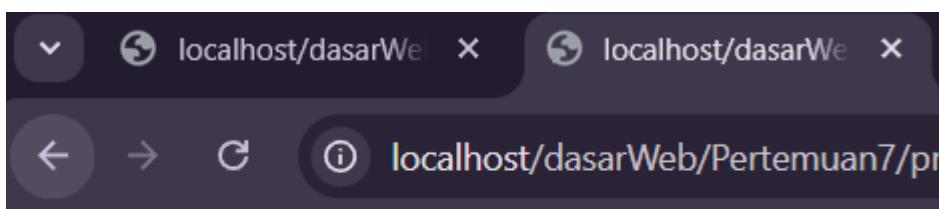
Ditampilkan pada browser, yakni apabila menjalankan pada browser proses.php, ia hanya akan menampilkan lembar kosong. Namun apabila menjalankan form.php, maka ia akan memunculkan sebuah form yang diharuskan untuk mengisikan nama dan email, dan submit. Setelah submit, maka akan muncul nama dan email yang telah diinputkan tadi. Hal ini dikarenakan pada kode program, telah ditambahkan pada form.php, yakni sintaks form untuk mengisikan nilai nama dan email, dan dihubungkan dengan sintaks action proses_form.php, sehingga ketika terdapat submit/post, maka ia akan mengeksekusi sintaks yang ada pada file proses_form.php, karena telah diisikan sintaks REQUEST_METHOD == POST.

Form Input PHP

4

Nama:

Email:



Nama: Indhira

Email: indhirayuantikac1176@gmail.com

5

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

```

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

    // Cek apakah form sudah disubmit
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        // Validasi nama (contoh: pastikan nama tidak kosong)
        if (empty($_POST["nama"])) {
            $namaErr = "Nama harus diisi!";
        } else {
            $nama = $_POST["nama"];
            echo "Data berhasil disimpan!";
        }
    }
    ?>
    <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
        <label for="nama">Nama:</label>
        <input type="text" name="nama" id="nama" value="<?php echo $nama; ?>">
        <span class="error"><?php echo $namaErr; ?></span><br><br>

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

```

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

Jawaban:

Ditampilkan pada browser, yakni apabila menjalankan pada browser form_self.php, maka akan muncul sebuah form yang meminta pengguna untuk mengisikan nama, lalu menekan tombol submit. Ketika tombol submit ditekan, maka halaman yang sama akan memproses data tersebut karena pada sintaks form telah menggunakan action="<?php echo htmlspecialchars(\$_SERVER["PHP_SELF"]);?>", yang berarti form akan memanggil dirinya sendiri. Jika kolom nama tidak diisi, maka akan muncul pesan error "Nama harus diisi". Namun apabila kolom nama telah diisi, maka setelah tombol submit ditekan akan muncul pesan "Data berhasil disimpan." dan nilai nama yang telah diinputkan akan tetap terlihat pada kolom input.

Hal ini dikarenakan pada kode program, form memanfaatkan metode POST dan melakukan pengecekan terhadap \$_SERVER["REQUEST_METHOD"] == "POST", sehingga ketika form dikirim, data langsung diproses oleh file yang sama tanpa memerlukan file eksternal seperti proses_form.php.

The screenshot displays three separate browser windows, likely from a browser extension or a virtual environment, illustrating a form input process.

- Window 1:** Shows a simple form titled "Form Input PHP". It has a label "Nama:" followed by an empty input field and a "Submit" button.
- Window 2:** Shows the same form, but the input field is empty and a validation message "Nama harus diisi" (Name must be filled) is displayed next to it.
- Window 3:** Shows the form again, but this time the input field contains the value "Indhira". Below the form, a success message "Data berhasil disimpan." (Data has been saved successfully) is displayed.

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:

- 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.
- 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.
- 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.
- Content Security Policy (CSP):** Terapkan Content Security Policy (CSP) pada header HTTP Anda untuk mengontrol sumber daya yang dapat digunakan dalam halaman web Anda.
- 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 html_aman.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam html_aman.php
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 localhost/dasarWeb/html_aman.php
4	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1) Jawaban: Ditampilkan pada browser, yakni apabila menjalankan pada browser html_aman.php, maka akan muncul sebuah form untuk mengisikan teks. Setelah teks diinputkan dan tombol kirim ditekan, maka halaman yang sama akan menampilkan hasil input yang telah dimasukkan. Namun, jika mencoba memasukkan tag HTML atau script seperti Halo atau <script>alert('hai')</script>, maka teks tersebut tidak akan dijalankan sebagai kode HTML atau JavaScript, melainkan hanya akan tampil sebagai teks biasa di browser. Hal ini terjadi karena pada kode program digunakan fungsi htmlspecialchars() yang berfungsi untuk mengamankan input dari karakter berbahaya, sehingga mencegah terjadinya serangan seperti XSS (Cross-Site Scripting) dan menjadikan tampilan halaman lebih aman dari eksekusi script berbahaya.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>HMTL aman</title>
7  </head>
8  <body>
9      <h2>HTML aman</h2>
10
11
12      <form action=<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>" method="post">
13          <label for="input">Input:</label>
14          <input type="text" id="input" name="input">
15          <br>
16          <input type="submit" value="Submit">
17      </form>
18
19      <br>
20
21      <?php
22
23      if($_SERVER["REQUEST_METHOD"] == "POST") {
24          $input = $_POST['input'];
25          $input = htmlspecialchars($input, ENT_QUOTES, 'UTF-8');
26          echo "Input: " . $input;
27      }
28
29      ?>
30  </body>
31  </html>
32
33

```

HTML aman

Input:

Input: halo

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	<p>Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/html_aman.php</p>
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.2)</p> <p>Jawaban:</p> <p>Ditampilkan pada browser, yakni ketika menjalankan pada browser html_aman.php, maka akan muncul form untuk mengisikan teks dan email. Setelah diisi dan tombol kirim ditekan, halaman yang sama akan menampilkan hasil input. Apabila pengguna memasukkan alamat email dengan format yang benar, maka akan muncul pesan “Email valid”. Namun, jika formatnya salah, maka akan muncul pesan “Email tidak valid!”. Hal ini terjadi karena pada kode program telah ditambahkan fungsi filter_var(\$email, FILTER_VALIDATE_EMAIL) yang berfungsi untuk memvalidasi format email secara otomatis, sehingga hanya input email dengan format yang benar yang akan diproses lebih lanjut.</p> <p>Dengan begitu, data yang diterima menjadi lebih aman dan terhindar dari kesalahan input maupun potensi penyalahgunaan data.</p>

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>HMTL aman</title>
7  </head>
8  <body>
9      <h2>HTML aman</h2>
10
11
12     <form action=<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>" method="post">
13         <label for="input">Input:</label>
14         <input type="text" id="input" name="input">
15         <br>
16
17         <label for="email">Email:</label>
18         <input type="email" id="email" name="email">
19         <br>
20
21         <input type="submit" value="Submit">
22     </form>
23
24     <br>
25
26     <?php
27
28     if($_SERVER["REQUEST_METHOD"] == "POST") {
29         $input = $_POST['input'];
30         $input = htmlspecialchars($input, ENT_QUOTES, 'UTF-8');
31         echo "Input: " . $input;
32
33         echo "<br>";
34
35         $email = $_POST['email'];
36
37         if(filter_var($email, FILTER_VALIDATE_EMAIL)) {
38             echo "Email valid";
39         } else {
40             echo "Email tidak valid";
41         }
42     }
43
44
45
46
47     ?>
48 </body>
49 </html>
50

```

HTML aman

Input:

Email:

Input: Indhira
 Email valid: indhirayuantikac1176@gmail.com

Regular Expression (Regex)

Regex adalah alat yang kuat untuk mencari, mencocokkan, atau memanipulasi teks berdasarkan pola tertentu. Anda dapat menggunakan 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,m}: Setidaknya n kali, maksimal m kali

Penggunaan Regex:

1. Mencocokan pola

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

2. Mencocokan 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 :

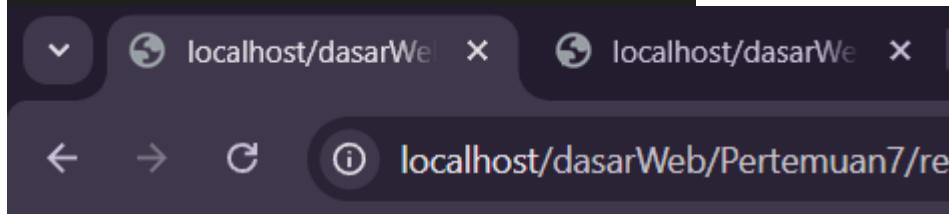
Langkah	Keterangan
1	Buat satu file baru bernama regex.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam regex.php
2	<pre> \$pattern = '/[a-z]/'; // Cocokkan huruf kecil. \$text = 'This is a Sample Text.'; if (preg_match(\$pattern, \$text)) { echo "Huruf kecil ditemukan!"; } else { 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>Jawaban:</p> <p>Ditampilkan pada browser, yakni huruf kecil ditemukan. Hal ini dikarenakan pada kode program, telah ditambahkan variabel \$pattern, yang diinisialisasi dengan nilai [a-z] dan \$text yang diinisialisasi dengan nilai 'This is a example text'. Kemudian, ditambahkan sintaks untuk pemilihan, yakni fungsi preg_match(), yang digunakan untuk mencocokkan dua variabel, sehingga pada kode program ini dikembalikan nilai true, dan mengeksekusi pada bagian if.</p>

	<p>Huruf kecil ditemukan.</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>Jawaban:</p> <p>Ditampilkan pada browser, yakni ketika menjalankan pada browser regex.php, maka akan muncul hasil pencarian pola berupa angka “123” dari teks yang diuji. Hal ini terjadi karena pada kode program digunakan fungsi preg_match() dengan pola /[0-9]+/, yang berfungsi untuk mencocokkan satu atau lebih digit angka di dalam string. Dengan penambahan kode tersebut, PHP dapat mendekripsi dan menampilkan angka yang terdapat di dalam teks, sehingga kita dapat mengenali bagian teks yang sesuai dengan pola tertentu menggunakan regular expression (regex).</p>

```

1 <?php
2
3 $pattern = '/[a-z]/';
4 $text = 'This is a example text.';
5
6 if(preg_match($pattern, $text)) {
7     echo "Huruf kecil ditemukan.";
8 } else {
9     echo "Tidak ada huruf kecil!";
10}
11
12 $pattern = '/[0-9]/';
13 $text = 'There are 123 apples.';
14
15 echo "<br>";
16
17 if(preg_match($pattern, $text, $matches)) {
18     echo "Cocokkan: " . $matches[0];
19 } else {
20     echo "Tidak ada yang cocok!";
21 }
22
23
24
25 ?>

```



Huruf kecil ditemukan.

Cocokkan: 1

9 Ketikkan kode tambahan pada langkah 10 di dalam regex.php

```

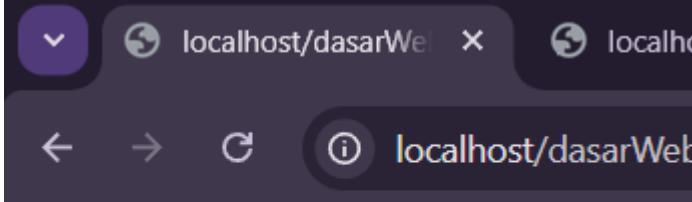
$pattern = '/apple/';
$replacement = 'banana';
$text = 'I like apple pie.';
$new_text = preg_replace($pattern, $replacement, $text);
echo $new_text; // Output: "I like banana pie."

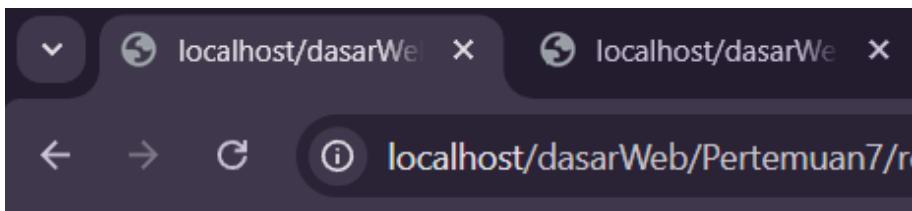
```

11 Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php

12 Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.3)

Jawaban:

	<p>Ditampilkan pada browser, yakni tulisan “I like banana pie”. Hal ini dikarenakan pada kode program telah ditambahkan variabel pattern yang diinisialisasi dengan nilai <code>apple.</code>, replacement yang diisikan dengan nilai banana, dan text yang diinisialisasi dengan nilai ‘I like apple pie’, sehingga ketika dipanggil fungsi preg_replace, maka ia menggantikan nilai pada variabel text, dan dibuat variabel baru yakni newText yang berisi nilai ‘I like banana pie’.</p>  <p>Huruf kecil ditemukan. Cocokkan: 1 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", "gooood", 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>Jawaban:</p> <p>Ditampilkan pada browser, yakni ia mencocokkan kata ‘god’. Hal ini dikarenakan pada kode program telah diisikan nilai dari variabel <code>\$pattern</code> yakni ‘/go*d/’, yang dimana ia akan mencocokkan kata pertama yang sama dengan pattern /go*d/, yakni god. Hal ini dikarenakan pada kode program ditambahkan sintaks <code>preg_match</code> untuk mencocokkan kata dengan pattern.</p>



Huruf kecil ditemukan.

Cocokkan: 1

I like banana pie.

Cocokkan: good

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)

Jawaban:

Ditampilkan pada browser, yakni cocokkan: god. Hal ini dikarenakan pada variabel \$pattern, diganti yang tadinya pattern pecocokannya menggunakan '*' menjadi '?'. Hal ini memiliki arti bahwa karakter sebelumnya hanya boleh ada satu atau 0 kali. Sehingga yang cocok hanyalah god dan gd.

17

```
● ● ●
1 $pattern = '/go?d/';
2 $text = 'god is good.';
3
4 if (preg_match($pattern, $text, $matches)) {
5     echo "Cocokkan: " . $matches[0];
6 } else {
7     echo "Tidak ada yang cocok!";
```

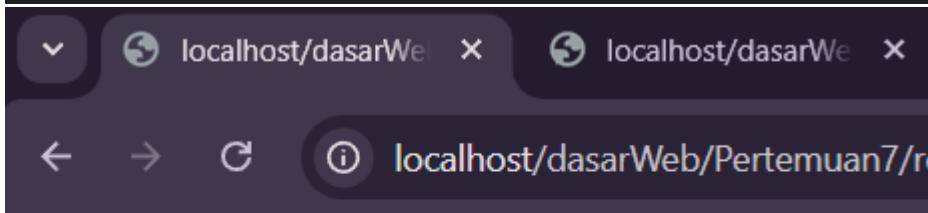
Soal 5.6: Buatlah script langkah 14 dengan variable pattern yang di ubah menggunakan `{n,m}`. Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.6)

18

Jawaban:

Ditampilkan pada browser, yakni cocokkan: good. Hal ini dikarenakan pada kode program telah dimodifikasi pada pattern yakni menggunakan {m,n}. Hal ini memiliki arti bahwa huruf sebelum pattern tersebut harus muncul minimal m hingga n kali. Pada kode program saya, saya isikan nilai m menjadi 2, dan nilai n menjadi 4. Hal ini berarti ia akan mencocokkan dengan kata 'good'.

```
1 $pattern = '/go{2,4}d/';
2 $text = 'god is good.';
3
4 if (preg_match($pattern, $text, $matches)) {
5     echo "Cocokkan: " . $matches[0];
6 } else {
7     echo "Tidak ada yang cocok!";
8 }
```



Huruf kecil ditemukan.

Cocokkan: 1

I like banana pie.

Cocokkan: good

Praktikum 6 : Form Lanjut

Langkah-langkah Praktikum :

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

```

<!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>

        <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>
</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 . "<br>";

    if (!empty($selectedWarna)) {
        echo "Warna favorit Anda: " . implode(", ", $selectedWarna) . "<br>";
    } else {
        echo "Anda tidak memilih warna favorit.<br>";
    }

    echo "Jenis kelamin Anda: " . $selectedJenisKelamin;
}

```

2

3 Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_lanjut.php

Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 6.1)

4

Jawaban:

Ditampilkan pada kode program, yakni form untuk mengisikan buah (opsi pilihan), warna (opsi checkbox), dan jenis kelamin (opsi radio). Di saat berhasil mengisikan ketiga indikator tersebut, maka akan dipanggil action proses_lanjut.php untuk memproses form tersebut.

Dalam proses_lanjut.php, ia memiliki pengecekan apakah REQUEST METHOD == POST, setelah itu, ia akan memproses setiap variabel yang diinputkan oleh pengguna, sehingga ditampilkan echo ketiga indikator yang dipilih tersebut. Pada indikator warna, pengguna bisa memilih lebih dari satu, satu saja, atau tidak sama sekali. Hal ini dikarenakan pada kode program telah diisikan pengecekan untuk mengatasi setiap kemungkinan yang ada, (Ketika pengguna memilih warna maupun tidak).

Form Contoh:

Pilih Buah:

Pilih Warna Favorit:

- Merah
- Biru
- Hijau

Pilih Jenis Kelamin:

- Laki-Laki
- Perempuan

Anda memilih buah: apel

Warna favorit Anda: hijau

Jenis Kelamin Anda: perempuan

5

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

```

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

        <br>

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

        <br>

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

        <br>

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

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

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

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

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

```

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)

Jawaban:

Ditampilkan pada browser, yakni form dapat diisi dan dikirim tanpa melakukan reload halaman. Setelah tombol Submit diklik, data yang diinputkan pada form dikirim ke file proses_lanjut.php melalui proses AJAX menggunakan jQuery. Hasil dari proses tersebut langsung ditampilkan pada bagian <div id="hasil"> tanpa perlu memuat ulang halaman. Hal ini terjadi karena pada kode program terdapat sintaks yang berfungsi untuk mengirim data ke server secara *asynchronous* (tanpa refresh halaman) dan menampilkan respon dari server ke dalam elemen HTML secara langsung.

Form Contoh

Pilih Buah: ▾

6

Pilih Warna Favorit:

- Merah
- Biru
- Hijau

Pilih Jenis Kelamin:

- Laki-Laki
- Perempuan

Anda memilih buah: jeruk

Warna favorit Anda: merah, biru, hijau

Jenis Kelamin Anda: perempuan

Praktikum 7 : Validasi Form

Langkah-langkah Praktikum :

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

```

<!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">
        <br>

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

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

```

2

```

<?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 . "<br>";
        }
    } 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";
    }
}
?>

```

3

Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_validasi.php

4

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

Jawaban:

Ditampilkan pada browser localhost/dasarWeb/form_validasi.php, form meminta input Nama dan Email. Jika tombol Submit ditekan tanpa mengisi salah satu atau kedua field, muncul pesan seperti "*Nama harus diisi.*" atau "*Email harus diisi.*".

Hal ini terjadi karena ditambahkan validasi pada kode PHP yang memeriksa apakah input kosong atau format email tidak sesuai. Jika ada kesalahan, pesan akan ditampilkan dan proses dihentikan sehingga data yang tidak valid tidak diproses.

Form Input dengan Validasi Password

Nama: Indhira

Email: indhirayuantikac1176@gmail.com

Submit

Form berhasil dikirim

Nama: Indhira

Email: indhirayuantikac1176@gmail.com

5

Kembangkan file bernama form_validasi.php di dalam folder dasarWeb. Ketikkan kode pada langkah 6 di dalam form_validasi.php

6

```
<!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) {
                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>
```

7

Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_validasi.php

	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.2)</p> <p>Jawaban:</p> <p>Ditampilkan pada browser, yakni ketika menjalankan localhost/dasarWeb/form_validasi.php, muncul form yang berisi input Nama dan Email. Apabila tombol Submit ditekan tanpa mengisi salah satu atau kedua field tersebut, maka akan muncul pesan kesalahan secara langsung di bawah kolom input, seperti "Nama harus diisi." atau "Email harus diisi.", tanpa perlu memuat ulang halaman. Hal ini dikarenakan pada kode program ditambahkan sintaks JQuery untuk validasi pengecekan dengan event \$("#myForm").submit(function(event), yang digunakan untuk mengecek apakah nilai input kosong atau data sudah terkirim. Apabila validasi gagal, maka ia akan memanggil fungsi event.preventDefault().</p>
8	<h2>Form Input dengan Validasi Password</h2> <p>Nama: <input type="text"/> Nama harus diisi.</p> <p>Email: <input type="text"/> Email harus diisi.</p> <p><input type="button" value="Submit"/></p> <p>Form berhasil dikirim</p> <p>Nama: Indhira</p> <p>Email: indhirayuantikac1176@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)</p> <p>Jawaban:</p> <p>Dari penambahan kode program di atas, terlihat bahwa proses pengiriman form tidak lagi melakukan reload halaman. Ketika tombol Submit ditekan, data dikirim ke file proses_validasi.php menggunakan AJAX. Hasil respon dari server langsung ditampilkan pada elemen <div id="response"> tanpa memuat ulang seluruh halaman.</p>

```

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

```

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)

10

Jawaban:

Ditampilkan pada browser, yakni form yang mengharuskan pengguna untuk memasukkan nama, email, dan password yang panjangnya minimal 8 karakter. Hal ini dikarenakan pada kode program telah dibuat sintaks untuk pengecekan setiap kondisi untuk mengatasi apabila terdapat error ketika pengguna

tidak mengisikan nama maupun email maupun paswword yang panjangnya minimal 8 huruf, dengan menggunakan JQuery. Lalu untuk memproses form yang telah diinputkan, memanggil action untuk menghubungkannya dengan file php, yang dimana pada file php ini dimasukkan pengecekan apakah REQUEST_METHOD == POST, lalu menghandle inputan nama, email dan password.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Form Validasi Password</title>
5      <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
6  </head>
7  <body>
8  <h1>Form Input dengan Validasi Password</h1>
9  <form id="myForm" method="post" action="proses_validasi_password.php">
10     <label for="nama">Nama:</label>
11     <input type="text" id="nama" name="nama">
12     <span id="nama-error" style="color: red;"></span><br><br>
13
14     <label for="email">Email:</label>
15     <input type="text" id="email" name="email">
16     <span id="email-error" style="color: red;"></span><br><br>
17
18     <label for="password">Password:</label>
19     <input type="password" id="password" name="password">
20     <span id="password-error" style="color: red;"></span><br><br>
21
22     <input type="submit" value="Submit">
23 </form>
24
25 <script>
26 $(document).ready(function() {
27     $("#myForm").submit(function(event) {
28         var nama = $("#nama").val();
29         var email = $("#email").val();
30         var password = $("#password").val();
31         var valid = true;
32
33         if (nama === "") {
34             $("#nama-error").text("Nama harus diisi.");
35             valid = false;
36         } else {
37             $("#nama-error").text("");
38         }
39
40         if (email === "") {
41             $("#email-error").text("Email harus diisi.");
42             valid = false;
43         } else {
44             $("#email-error").text("");
45         }
46
47         if (password.length < 8) {
48             $("#password-error").text("Password minimal 8 karakter.");
49             valid = false;
50         } else {
51             $("#password-error").text("");
52         }
53
54         if (!valid) {
55             event.preventDefault();
56         }
57     });
58 });
59 </script>
60 </body>
61 </html>
62
```

```
1 <?php
2 if ($_SERVER["REQUEST_METHOD"] == "POST") {
3     $nama = $_POST["nama"];
4     $email = $_POST["email"];
5     $password = $_POST["password"];
6
7     if (empty($nama) || empty($email) || empty($password)) {
8         echo "Semua field harus diisi!";
9     } elseif (strlen($password) < 8) {
10        echo "Password terlalu pendek! Minimal 8 karakter.";
11    } else {
12        echo "Form berhasil dikirim.<br>";
13        echo "Nama: $nama<br>";
14        echo "Email: $email<br>";
15        echo "Password: $password<br>";
16    }
17}
18?>
19
```

Form Input dengan Validasi Password

Nama:

Email:

Password:

Form Input dengan Validasi Password

Nama:

Email:

Password: Password minimal 8 karakter.

Form berhasil dikirim
Nama: Indhira
Email: indhirayuantikac1176@gmail.com

