



Topik

- Konsep Form Processing dengan PHP dan JQuery

Link Repository : https://github.com/athfizh/AthaulaHafizh_DESPROG_TI2A_2025

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>Kode ini menggunakan fungsi isset() untuk memeriksa apakah variabel \$umur sudah didefinisikan atau belum. Karena variabel \$umur tidak pernah dibuat, isset(\$umur) akan mengembalikan false, sehingga output yang ditampilkan adalah "Anda belum dewasa atau variabel 'umur' tidak ditemukan.". Ini menunjukkan fungsi isset() berguna untuk mencegah error saat mencoba menggunakan variabel yang belum ada.</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>Anda belum dewasa atau variabel 'umur' tidak ditemukan.</p> <p>Nama: Jane</p> <p>isset() digunakan untuk memeriksa keberadaan <i>key</i> "nama" di dalam array \$data. Karena <i>key</i> tersebut ada di dalam array, isset(\$data["nama"]) mengembalikan true, dan program menampilkan "Nama: Jane". Ini menunjukkan bahwa isset() tidak hanya untuk variabel, tetapi juga sangat efektif untuk memeriksa apakah sebuah <i>key</i> ada di dalam array.</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>Array tidak terdefinisi atau kosong.</p> <p>Fungsi <code>empty()</code> digunakan untuk memeriksa apakah sebuah variabel dianggap "kosong". Variabel yang dianggap kosong antara lain string "", angka 0, null, atau array kosong. Dalam kode ini, <code>\$myArray</code> adalah sebuah array kosong, sehingga <code>empty(\$myArray)</code> mengembalikan true dan outputnya adalah "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 <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>Array tidak terdefinisi atau kosong.</p> <p>Variabel tidak terdefinisi atau kosong.</p> <p>Kode ini menunjukkan kelebihan empty() yaitu dapat digunakan pada variabel yang belum didefinisikan (\$nonExistentVar) tanpa menyebabkan error. Karena variabel tersebut tidak ada, empty() menganggapnya kosong dan mengembalikan true, sehingga outputnya adalah "Variabel tidak terdefinisi atau kosong."</p>
--	--

Praktikum Bagian 3 : Form Input PHP

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

2	<pre> <!DOCTYPE html> <html> <head> <title>Form Input PHP</title> </head> <body> <h2>Form Input PHP</h2> <form method="post" action="proses_form.php"> <label for="nama">Nama:</label> <input type="text" name="nama" id="nama" required>

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

 <input type="submit" name="submit" value="Submit"> </form> </body> </html> </pre>
3	<p>Buat satu file baru bernama proses_form.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam proses_form.php</p> <pre> <?php if (\$_SERVER["REQUEST_METHOD"] == "POST") { \$nama = \$_POST["nama"]; \$email = \$_POST["email"]; echo "Nama: " . \$nama . "
"; echo "Email: " . \$email; } ?> </pre>
4	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/proses_form.php. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 3.1)</p> <p>Form Input PHP</p> <p>Nama: <input type="text" value="Hafizh"/></p> <p>Email: <input type="text" value="athafhafizh@gmail.com"/></p> <p><input type="submit" value="Submit"/></p> <p>Submit di form Input</p> <p>Nama: Hafizh Email: athafhafizh@gmail.com</p> <p>Hasil dari submit form diatas</p>

	File form.php membuat sebuah form HTML yang datanya akan dikirim ke proses_form.php menggunakan metode POST. File proses_form.php kemudian mengambil data yang dikirim (nama dan email) dari variabel global \$_POST dan menampilkannya kembali ke browser.
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>

```

6

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

Form Input PHP

Nama:

Form Input PHP

Data berhasil disimpan!

Nama:

Kode ini menunjukkan *self-processing form*, di mana halaman form dan logika pemrosesannya berada dalam satu file. Atribut action pada form diatur ke `$_SERVER["PHP_SELF"]`, yang berarti data dikirim ke halaman itu sendiri. PHP di bagian atas file akan memeriksa apakah ada data POST yang masuk, melakukan validasi sederhana (memeriksa apakah nama kosong), lalu menampilkan pesan error atau sukses.

HTML Injection

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

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

Contoh HTML Injection:

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

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


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

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

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

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

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

Cara Mengangani HTML Injection

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

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

Berikut contoh sederhana dalam PHP untuk mengatasi HTML injection:

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

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

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

Praktikum 4 : HTML Injection

Langkah-langkah Praktikum :

Langkah	Keterangan
---------	------------

1	Buat satu file baru bernama <code>html_aman.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>html_aman.php</code>
2	<pre><code>\$input = \$_POST['input']; \$input = htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8');</code></pre>
3	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>Input Berbahaya: <input type="text"/> <input type="button" value="Submit"/></p> <p>Output Aman:</p> <p>Hafizh</p> <p>Kode ini mendemonstrasikan cara mencegah serangan HTML Injection. Fungsi <code>htmlspecialchars()</code> mengubah karakter spesial HTML (seperti <code><</code> menjadi <code>&lt;</code> dan <code>></code> menjadi <code>&gt;</code>). Jika pengguna memasukkan kode seperti <code><script>alert('tes')</script></code>, kode tersebut tidak akan dieksekusi oleh browser, melainkan hanya ditampilkan sebagai teks biasa.</p>
5	Ketikkan kode tambahan pada langkah 6 di dalam <code>html_aman.php</code>
6	<pre><code>// 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 }</code></pre>
7	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/html_aman.php</code>
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.2)</p> <p>Input Email: <input type="text"/> <input type="button" value="Validate Email"/></p> <p>Hasil Validasi:</p> <p>Format email 'hafizh' tidak valid.</p> <p>Input Email: <input type="text"/> <input type="button" value="Validate Email"/></p> <p>Hasil Validasi:</p> <p>Format email 'athafhafizh@gmail.com' valid.</p> <p>Kode ini menggunakan fungsi <code>filter_var()</code> dengan filter <code>FILTER_VALIDATE_EMAIL</code> untuk memvalidasi format email. Ini adalah cara yang lebih baik dan lebih aman untuk memastikan input sesuai dengan format yang diharapkan daripada membuat aturan sendiri. Jika format email benar, program akan menampilkan pesan valid, jika tidak, akan menampilkan pesan tidak valid.</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 <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>Huruf kecil ditemukan!</p> <p>Fungsi <code>preg_match()</code> digunakan untuk mencari kecocokan pola regex dalam sebuah string. Pola <code>/[a-z]/</code> mencari setidaknya satu huruf kecil. Karena string <code>\$text</code> mengandung huruf kecil, fungsi ini mengembalikan <code>true</code> dan menampilkan "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>Huruf kecil ditemukan!</p> <p>Cocokkan: 123</p> <p>Pola <code>/[0-9]+/</code> mencari satu atau lebih digit secara berurutan. Saat <code>preg_match()</code> menemukan kecocokan ("123"), ia menyimpannya ke dalam array <code>\$matches</code>. Elemen <code>\$matches[0]</code> selalu berisi string lengkap yang cocok dengan pola, sehingga outputnya adalah "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 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>Huruf kecil ditemukan!</p> <p>Cocokkan: 123</p> <p>I like banana pie.</p> <p>Fungsi preg_replace() mencari string yang cocok dengan pola (/apple/) dan menggantinya dengan string lain (banana). Hasilnya adalah string baru "I like banana pie."</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>Huruf kecil ditemukan!</p> <p>Cocokkan: 123</p> <p>I like banana pie.</p> <p>Cocokkan: god</p> <p>Kuantifier * berarti "nol atau lebih" dari karakter sebelumnya. Pola /go*d/ akan cocok dengan "gd", "god", "good", dst. Dalam string 'god is good.', preg_match menemukan kecocokan pertama yaitu "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>Huruf kecil ditemukan!</p> <p>Cocokkan: 123</p> <p>I like banana pie.</p> <p>Cocokkan: god</p> <p>Cocokkan (dengan?): god</p> <p>Kuantifier ? berarti "nol atau satu" dari karakter sebelumnya. Pola /go?d/ hanya akan cocok dengan "gd" atau "god". Karena preg_match menemukan "god" terlebih dahulu, itulah yang ditampilkan. Pola ini tidak akan cocok dengan "good".</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>Huruf kecil ditemukan!</p> <p>Cocokkan: 123</p> <p>I like banana pie.</p> <p>Cocokkan: god</p> <p>Cocokkan (dengan?): god</p> <p>Cocokkan (dengan {1,2}): god</p> <p>Kuantifier {n,m} berarti "minimal n, maksimal m" dari karakter sebelumnya. Pola /go{1,2}d/ akan cocok dengan 'g' diikuti oleh satu atau dua 'o', lalu 'd'. Ini cocok untuk "god" dan "good". Dalam teks, kecocokan pertama yang ditemukan adalah "god".</p>

Praktikum 6 : Form Lanjut

Langkah-langkah Praktikum :

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

2	<pre> <!DOCTYPE html> <html> <head> <title>Contoh Form dengan PHP</title> </head> <body> <h2>Form Contoh</h2> <form method="POST" action="proses_lanjut.php"> <label for="buah">Pilih Buah:</label> <select name="buah" id="buah"> <option value="apel">Apel</option> <option value="pisang">Pisang</option> <option value="mangga">Mangga</option> <option value="jeruk">Jeruk</option> </select>
 <label>Pilih Warna Favorit:</label>
 <input type="checkbox" name="warna[]" value="merah"> Merah
 <input type="checkbox" name="warna[]" value="biru"> Biru
 <input type="checkbox" name="warna[]" value="hijau"> Hijau

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

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

	<h2>Form Contoh</h2> <p>Pilih Buah: <input type="text" value="Apel"/> ▼</p> <p>Pilih Warna Favorit:</p> <p><input checked="" type="checkbox"/> Merah</p> <p><input type="checkbox"/> Biru</p> <p><input type="checkbox"/> Hijau</p> <p>Pilih Jenis Kelamin:</p> <p><input checked="" type="radio"/> Laki-laki</p> <p><input type="radio"/> Perempuan</p> <p><input type="button" value="Submit"/></p> <p>Anda memilih buah: apel Warna favorit Anda: merah Jenis kelamin Anda: laki-laki</p> <p>Kode ini menangani input form yang lebih kompleks: dropdown (select), checkbox, dan radio button. Hal penting yang diamati adalah penggunaan <code>name="warna[]"</code> pada checkbox. Tanda <code>[]</code> membuat PHP mengumpulkan semua nilai checkbox yang dicentang ke dalam sebuah array. Script <code>proses_lanjut.php</code> kemudian menggunakan <code>implode()</code> untuk menggabungkan semua pilihan warna menjadi satu string yang mudah dibaca.</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 via AJAX

Pilih Buah:

Pilih Warna Favorit:

- ☒ Merah
☐ Biru

Pilih Jenis Kelamin:

- ☒ Laki-laki
☐ Perempuan

Hasil:

Anda memilih buah: apel
Warna favorit Anda: merah
Jenis kelamin Anda: laki-laki

Form ini dikirim menggunakan **AJAX** dengan bantuan jQuery. Saat tombol submit ditekan, `e.preventDefault()` mencegah browser me-refresh halaman. Sebagai gantinya, `$.ajax()` mengirim data form ke `proses_lanjut.php` di latar belakang. Respon dari server kemudian diterima dan ditampilkan di dalam `<div id="hasil">` tanpa perlu memuat ulang seluruh halaman, memberikan pengalaman pengguna yang lebih mulus.

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
2	<pre> <!DOCTYPE html> <html> <head> <title>Form Input dengan Validasi</title> </head> <body> <h1>Form Input dengan Validasi</h1> <form method="post" action="proses_validasi.php"> <label for="nama">Nama:</label> <input type="text" id="nama" name="nama">
 <label for="email">Email:</label> <input type="text" id="email" name="email">
 <input type="submit" value="Submit"> </form> </body> </html> </pre> <pre> <?php if (\$_SERVER["REQUEST_METHOD"] == "POST") { \$nama = \$_POST["nama"]; \$email = \$_POST["email"]; \$errors = array(); // Validasi Nama if (empty(\$nama)) { \$errors[] = "Nama harus diisi."; } // Validasi Email if (empty(\$email)) { \$errors[] = "Email harus diisi."; } elseif (filter_var(\$email, FILTER_VALIDATE_EMAIL)) { \$errors[] = "Format email tidak valid."; } // Jika ada kesalahan validasi if (empty(\$errors)) { (foreach (\$errors as \$error) { echo \$error . "
"; }) } else { // Lanjutkan dengan pemrosesan data jika semua validasi berhasil // Misalnya, menyimpan data ke database atau mengirim email echo "Data berhasil dikirim: Nama = \$nama, Email = \$email"; } } ?> </pre>
3	Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_validasi.php
4	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.1)

	<h2>Form Input dengan Validasi</h2> <div> Nama: <input type="text" value="Hafizh"/> Email: <input type="text" value="athafhafizh@gmail.com"/> <input type="button" value="Submit"/> </div> <hr/> <p>Data berhasil dikirim: Nama = Hafizh, Email = athafhafizh@gmail.com</p> <p>Kode ini menunjukkan validasi di sisi server. Saat tombol "Submit" ditekan, halaman akan memuat ulang dan data dikirim ke server. Skrip PHP kemudian memeriksa apakah input "nama" dan "email" valid. Jika tidak, pesan kesalahan akan ditampilkan di bagian atas halaman. Proses ini terjadi sepenuhnya di server setelah pengguna mengirimkan data.</p>
5	Kembangkan file bernama <code>form_validasi.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 6 di dalam <code>form_validasi.php</code>
6	<pre> <!DOCTYPE html> <html> <head> <title>Form Input dengan Validasi</title> <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script> </head> <body> <h1>Form Input dengan Validasi</h1> <form id="myForm" method="post" action="proses_validasi.php"> <label for="nama">Nama:</label> <input type="text" id="nama" name="nama">
 <label for="email">Email:</label> <input type="text" id="email" name="email">
 <input type="submit" value="Submit"> </form> <script> \$(document).ready(function() { \$("#myForm").submit(function(event) { var nama = \$("#nama").val(); var email = \$("#email").val(); var valid = true; if (nama == "") { \$("#nama-error").text("Nama harus diisi."); valid = false; } else { \$("#nama-error").text(""); } if (email == "") { \$("#email-error").text("Email harus diisi."); valid = false; } else { \$("#email-error").text(""); } if (valid) { (!event.preventDefault()); // Menghentikan pengiriman form jika validasi gagal } }); }); </script> </body> </html> </pre>
7	Simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/form_validasi.php</code>
8	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.2)


```

<!DOCTYPE html>
<html>
<head>
  <title>Form Validasi dengan AJAX</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
  <h1>Form Validasi dengan AJAX</h1>
  <div id="hasil"></div>

  <form id="myForm">
    <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 submit form standar

    // Validasi client-side (sama seperti soal 7.2)
    var nama = $("#nama").val();
    var email = $("#email").val();
    var isValid = true;
    if (nama === "") { $("#nama-error").text("Nama harus diisi."); isValid = false; }
    else { $("#nama-error").text(""); }
    if (email === "") { $("#email-error").text("Email harus diisi."); isValid = false; }
    else { $("#email-error").text(""); }

    // Jika valid, kirim data ke server menggunakan AJAX
    if (isValid) {
      $.ajax({
        url: "proses_validasi.php", // File PHP tujuan
        type: "POST",
        data: $(this).serialize(), // Ambil semua data form
        success: function(response) {
          // Tampilkan respon dari server di div #hasil
          $("#hasil").html(response);
        }
      });
    }
  });
});
</script>
</body>
</html>

```

Output :

Form Validasi dengan AJAX

Nama:
Email:

Form Validasi dengan AJAX

Data berhasil dikirim: Nama = Hafizh, Email = athafhafizh@gmail.com

Nama:
Email:

Nama: Nama harus diisi.
Email: Email harus diisi.

Skrip ini menggabungkan validasi **client-side** dengan pengiriman data **AJAX**. Setelah validasi di browser berhasil, data dikirim ke proses_validasi.php di latar belakang tanpa me-refresh halaman. Respon dari server (baik pesan sukses atau error dari validasi server) kemudian ditampilkan di dalam div#hasil. Ini memberikan pengalaman pengguna yang sangat baik karena semuanya terjadi secara instan di satu halaman.

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 form_lengkap.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Form Validasi Lengkap</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
  <h1>Form Validasi Lengkap</h1>
  <div id="hasil"></div>
  <form id="myForm">
    <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>

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

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

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

    // Validasi Nama dan Email (diasumsikan sudah ada)

    // Validasi Password di client-side
    var password = $("#password").val();
    if (password.length < 8) {
      $("#password-error").text("Password minimal 8 karakter.");
      isValid = false;
    } else {
      $("#password-error").text("");
    }

    if (isValid) {
      $.ajax({
        url: "proses_validasi_lengkap.php",
        type: "POST",
        data: $(this).serialize(),
        success: function(response) {
          $("#hasil").html(response);
        }
      });
    }
  });
});
</script>
</body>
</html>
```

Kode proses_validasi_lengkap.php

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = $_POST['nama'] ?? '';
    $email = $_POST['email'] ?? '';
    $password = $_POST['password'] ?? '';
    $errors = [];

    // Validasi server-side
    if (empty($nama)) { $errors[] = "Nama wajib diisi."; }
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) { $errors[] = "Email tidak valid."; }
    if (strlen($password) < 8) { $errors[] = "Password minimal 8 karakter."; }

    if (!empty($errors)) {
        foreach ($errors as $error) {
            echo "<p style='color:red;'>$error</p>";
        }
    } else {
        echo "<p style='color:green;'>Data valid dan berhasil diterima!</p>";
    }
}
?>
```

Output :

Form Validasi Lengkap

Data valid dan berhasil diterima!

Nama:

Email:

Password:

Validasi password diterapkan di **dua level**. Pertama di **client-side** (jQuery) untuk memberikan respons cepat kepada pengguna bahwa password kurang dari 8 karakter. Kedua, validasi yang sama (`strlen() < 8`) juga dilakukan di **server-side** (PHP) sebagai lapisan keamanan utama. Ini adalah praktik terbaik untuk memastikan aturan bisnis selalu ditegakkan, bahkan jika JavaScript di browser pengguna dimatikan.