

**JOBSHEET 7
PEMROGRAMAN WEB**



NAFISAH ALIYAH KHUMAINI

**244107060066
SIB 2C**

**PROGRAM STUDI SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

2025



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> <hr/> <p>Anda belum dewasa atau variabel 'umur' tidak ditemukan.</p> <p>Penggunaan isset() dalam PHP berfungsi untuk memeriksa apakah suatu variabel telah didefinisikan dan tidak bernilai null.</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> <hr/> <p>Anda belum dewasa atau variabel 'umur' tidak ditemukan. Nama: Jane</p> <p>Penggunaan isset(\$data["nama"]) itu untuk memeriksa apakah elemen array dengan key "nama" sudah ada dan tidak bernilai null. Ini mencegah error saat mencoba mengakses elemen array yang belum didefinisikan. Karena variabel nama sudah terisi dan bernilai Jane jadi outputnya 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

```

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

```

nilai atau telah didefinisikan. Berikut adalah penjelasan dan contoh penggunaan **empty()**:
Fungsi **empty()** dapat digunakan untuk memeriksa apakah suatu string kosong atau tidak.

Praktikum 2. Function empty

Langkah	Keterangan
1	Buat satu file baru bernama <code>empty.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>empty.php</code>
2	<pre> \$myArray = array(); // Array kosong if (empty(\$myArray)) { echo "Array tidak terdefinisi atau kosong."; } else { echo "Array terdefinisi dan tidak kosong."; } </pre>
3	<p>Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/dasarWeb/empty.php</code>. Apa yang anda pahami dari penggunaan <code>empty</code> pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.1)</p> <hr/> <p>Array tidak terdefinisi atau kosong.</p> <p>Penggunaan <code>empty(\$myArray)</code> itu untuk memeriksa apakah array kosong atau belum terisi data. Variabel akan mengembalikan <code>true</code> jika variabel belum didefinisikan atau variabel bernilai <code>null</code>, <code>false</code>, <code>0</code>, <code>""</code>, atau array kosong.</p>
4	<p>Tambahkan isi dari file <code>empty.php</code> tersebut dengan kode di bawah ini.</p> <pre> if (empty(\$nonExistentVar)) { echo "Variabel tidak terdefinisi atau kosong."; } else { echo "Variabel terdefinisi dan tidak kosong."; } </pre>
5	<p>Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/dasarWeb/empty.php</code> buat hasil tidak menjadi 1 baris, jadi tampilan dari <code>echo</code> harus terpisah. Apa yang anda pahami dari penggunaan <code>empty</code> pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.2)</p> <hr/> <p>Variabel tidak terdefinisi atau kosong.</p> <p>Karena variabel <code>\$nonExistentVar</code> belum pernah dibuat, <code>empty()</code> menganggapnya sebagai kosong. Maka output yang didapatkan adalah variabel tidak terdefinisi atau kosong.</p>

Praktikum Bagian 3 : Form Input PHP

Langka h	Keterangan
-------------	------------

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

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

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

 <input type="submit" name="submit" value="Submit"> </form> </body> </html> </pre>
3	<p>Buat satu file baru bernama proses_form.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam proses_form.php</p> <pre> <?php if (\$_SERVER["REQUEST_METHOD"] == "POST") { \$nama = \$_POST["nama"]; \$email = \$_POST["email"]; echo "Nama: " . \$nama . "
"; echo "Email: " . \$email; } ?> </pre>
4	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/proses_form.php. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 3.1)</p> <h3>Form Input PHP</h3> <p>Nama: <input type="text" value="picha"/></p> <p>Email: <input type="text" value="nafisahaliyah01@gmail.com"/></p> <p><input type="submit" value="Submit"/></p> <p>Di form.php ini menyediakan form HTML dengan method POST dan mengirim data ke proses_form.php</p>

	<p>Nama: picha</p> <p>Email: nafisahaliyah01@gmail.com</p> <p>Di proses_form.php ini menerima data dari form melalui \$_POST. Kemudian menampilkan kembali data nama dan email yang dikirim.</p>
5	<p>Buat satu file baru bernama form_self.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form_self.php</p>

	<pre> <!DOCTYPE html> <html> <head> <title>Form Input PHP</title> </head> <body> <h2>Form Input PHP</h2> <?php // Inisialisasi variabel \$namaErr = ""; \$nama = ""; // Cek apakah form sudah disubmit if (\$_SERVER["REQUEST_METHOD"] == "POST") { // Validasi nama (contoh: pastikan nama tidak kosong) if (empty(\$_POST["nama"])) { \$namaErr = "Nama harus diisi!"; } else { \$nama = \$_POST["nama"]; echo "Data berhasil disimpan!"; } } ?> <form method="post" action="<?php echo htmlspecialchars(\$_SERVER["PHP_SELF"]); ?>"> <label for="nama">Nama:</label> <input type="text" name="nama" id="nama" value="<?php echo \$nama; ?>"> <?php echo \$namaErr; ?>

 <input type="submit" name="submit" value="Submit"> </form> </body> </html> </pre>
6	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/form_self.php. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 3.2)</p> <p>Ada validasi input nama menggunakan empty() untuk memastikan data tidak kosong. Menampilkan pesan error di bawah input dengan span error. Mengisi ulang field nama setelah submit agar pengguna tidak perlu mengetik ulang. Menggunakan htmlspecialchars(\$_SERVER["PHP_SELF"]) untuk keamanan XSS saat form mengirim ke dirinya sendiri</p>

HTML Injection

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

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

Contoh HTML Injection:

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

```

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

```

Sekarang, jika aplikasi web tidak memvalidasi input dengan benar, seorang penyerang dapat

memasukkan input berbahaya seperti ini:

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

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

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

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

Cara Mengangani HTML Injection

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

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

Berikut contoh sederhana dalam PHP untuk mengatasi HTML injection:

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

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


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

Praktikum 4 : HTML Injection

Langkah-langkah Praktikum :

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

	<pre> <?php if (\$_SERVER["REQUEST_METHOD"] == "POST") { // Langkah 1: Ambil input dan amankan dari XSS \$input = \$_POST['input']; \$input = htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8'); // Langkah 2: Proses atau tampilkan input echo "<h3>Hasil Input Anda:</h3>"; echo "<p>" . \$input . "</p>"; } ?> <!-- Form input --> <form method="POST" action=""> <label>Masukkan teks:</label>
 <input type="text" name="input" required> <button type="submit">Kirim</button> </form> </pre> <p>Hasil Input Anda:</p> <p>ppppp</p> <p>Masukkan teks:</p> <input type="text"/> <input type="button" value="Kirim"/> <p>Jadi fungsi htmlspecialchars berfungsi untuk mencegah karakter berbahaya seperti < dan > tidak dieksekusi.</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>Masukkan email:</p> <p><input type="text" value="ppp"/></p> <div>  Please include an '@' in the email address. 'ppp' is missing an '@'. </div>

	Validasi input ditingkatkan. Dengan menambahkan <code>filter_var(\$email, FILTER_VALIDATE_EMAIL)</code> , program kini memeriksa apakah input memiliki format yang benar
--	--

Regular Expression (Regex)

Regex adalah alat yang kuat untuk mencari, mencocokkan, atau memanipulasi teks berdasarkan pola tertentu. Anda dapat menggunakannya untuk berbagai tujuan seperti validasi input, pencarian teks, penggantian teks, dan banyak lagi.

Basic Regex Patterns:

- **Karakter Tunggal:** Karakter apa pun akan cocok dengan dirinya sendiri dalam Regex, kecuali karakter khusus yang harus dihindari dengan karakter pelarian `\`.
 - o Contoh: Pola `/a/` akan cocok dengan huruf "a" dalam teks.
- **Karakter Spesifik:** Anda dapat mencocokkan karakter tertentu dengan menyebutkannya dalam pola.
 - o Contoh: Pola `/hello/` akan cocok dengan teks "hello" dalam teks.
- **Karakter Khusus:** Beberapa karakter khusus dalam Regex harus dihindari dengan karakter pelarian (`\`) jika Anda ingin mencocokkannya secara harfiah. Contoh karakter khusus: `.`, `*`, `+`, `?`, `|`, `[`, `]`, `(`, `)`, `{`, `}`, `^`, `$`, `\`.
- **Karakter Set:** Anda dapat mencocokkan karakter dari satu set karakter dengan menggunakan `[]`. Misalnya, `/[aeiou]/` akan mencocokkan salah satu huruf vokal.
- **Rentang Karakter:** Anda dapat menggunakan `-` dalam karakter set untuk menentukan rentang karakter. Misalnya, `/[a-z]/` akan mencocokkan huruf kecil apa pun.
- **Kuantifier:** Anda dapat mengatur berapa kali karakter atau kelompok karakter sebelumnya harus muncul. Contoh kuantifier:
 - o `*`: 0 atau lebih kali

- +: 1 atau lebih kali
- ?: 0 atau 1 kali
- {n}: Persis n kali
- {n,}: Setidaknya n kali
- {n,m}: Setidaknya n kali, maksimal m kali

Contoh Penggunaan Regex:

1. Mencocokkan pola

```
$pattern = '/hello/';
$text = 'Hello, world!';
if (preg_match($pattern, $text)) {
    echo "Cocokkan ditemukan!";
} else {
    echo "Tidak ada 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><code>\$pattern = '/[a-z]/'; // Cocokkan huruf kecil. \$text = 'This is a Sample Text.'; if (preg_match(\$pattern, \$text)) { echo "Huruf kecil ditemukan!"; } else { echo "Tidak ada huruf kecil!"; }</code></pre>

3	Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.1)</p> <div>Huruf kecil ditemukan!</div> <p>Mengecek apakah string \$text mengandung huruf kecil (a-z) menggunakan regular expression (regex) dan fungsi preg_match(). Dari text 'this is a sample'ada huruf kecilnya sehingga keluar output hurud 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> <div>Cocokkan: 123</div> <p>Ini adalah pola regular expression:</p> <ul style="list-style-type: none"> - [0-9] berarti karakter angka dari 0-9 - + berarti satu atau lebih digit berturut-turut. - Preg_match() mencari pola dalam \$text - Jika ditemukan, hasil pertama disimpan di array \$matches - \$matches[0] berisi angka pertama yang cocok, yaitu 123.
9	Ketikkan kode tambahan pada langkah 10 di dalam regex.php
10	<pre>\$pattern = '/apple/'; \$replacement = 'banana'; \$text = 'I like apple pie.'; \$new_text = preg_replace(\$pattern, \$replacement, \$text); echo \$new_text; // Output: "I like banana pie."</pre>
11	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php
12	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.3)</p> <div>I like banana pie.</div> <ul style="list-style-type: none"> - Pola regex mencari kata apple

	<ul style="list-style-type: none"> - Kemudian ada pengganti yang akan digunakan jika pola ditemukan. - Fungsi <code>preg_replace()</code> mencari pola dalam <code>\$text</code> dan mengganti dengan <code>\$replacement</code>
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 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>Cocokkan: god</p> <p>Menggunakan regez dengan karakter kuantifikasi untuk mencocokkan pola tertentu. Mencari kata dalam text yang cocok dengan pola /go*d/. G karakter tetap, o* nol atau lebih huruf 'o', dan d karakter tetap. Karena minta yang indeks ke 0 maka yang idtampilkan adalah 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>Cocokkan: god</p> <p>Kalau pakai ? itu o nya Cuma boleh muncul 0 atau sekali. Yang muncul god karena fungsi preg_match() mengambil pencocokan pertama yang paling panjang dan paling awal.</p>
18	<p>Soal 5.6: Buatlah script langkah 14 te Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.6)</p> <p>Cocokkan: god</p> <ul style="list-style-type: none"> - G nya wajib - O {1,2} itu huruf 'o' harus muncul minimal 1 kali dan maksimal 2 kali - D nya wajib <p>Karena god muncul terlebih dahulu maka yang ditampilkan adalah 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> <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> </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>Anda memilih buah: apel Warna favorit Anda: merah Jenis kelamin Anda: perempuan</p> <ul style="list-style-type: none"> - Mengecek apakah form dikirim dengan metode POST - Menyimpan pilihan buah ke variabel

	<ul style="list-style-type: none"> - Mengecek pemilihan warna, klo ga dipilih dikosongi - Menyimpan jenis kelamin - Menampilkan buah yang dipilih
5	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>

```

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

        <br>

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

        <br>

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

        <br>

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

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

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

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

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

```

6

Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_ajax.php. Catat di sini apa yang anda amati dari penambahan kode program di atas.
(soal no 6.2)

Form Contoh

Pilih Buah!

Pilih Warna Favorit:

☒ Merah

☐ Biru

☐ Hijau

Pilih Jenis Kelamin:

☐ Laki-laki

☒ Perempuan

Anda memilih buah: apel

Warna favorit Anda: merah

Jenis kelamin Anda: perempuan

Serialize() otomatis menangkap semua input form dan menyusunnya dalam format key = value. Ajax mengirim data ke proses_lanjut.php tanpa record halaman. Respons dari php langsung ditampilkan di div hasil.

Praktikum 7 : Validasi Form

Langkah-langkah Praktikum :

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

Form Input dengan Validasi

Nama:
Email:

Data berhasil dikirim: Nama = picha, Email = nafisahaliyah01@gmail.com

Email tidak valid

Form Input dengan Validasi

Nama:
Email:

Format email tidak valid.

Menyimpan input dari form ke variabel, ada validasi nama dan validasi email, ada menampilkan hasil.

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
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.2)</p> <p>Form Input dengan Validasi</p> <div> Nama: <input type="text" value="picha"/> Email: <input type="text" value="nafisahaliyah01@gmail.com"/> <input type="submit" value="Submit"/> </div> <p>Data berhasil dikirim: Nama = picha, Email = nafisahaliyah01@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> <pre> <!DOCTYPE html> <html> <head> <title>Form Input dengan Validasi AJAX</title> <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script> </head> <body> <h1>Form Input dengan Validasi AJAX</h1> <form id="myForm"> <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> <div id="response-message" style="margin-top: 20px;"></div> <script> \$(document).ready(function(){ \$("#myForm").submit(function(event){ event.preventDefault(); // cegah submit biasa var nama = \$("#nama").val().trim(); var email = \$("#email").val().trim(); var valid = true; // Reset error \$("#nama-error").text(""); \$("#email-error").text(""); \$("#response-message").text(""); }); }); </script> </pre>


```

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

if (email === "") {
    $("#email-error").text("Email harus diisi.");
    valid = false;
}

if (valid) {
    $.ajax({
        url: "proses_validasi.php",
        type: "POST",
        data: { nama: nama, email: email },
        success: function(response){
            $("#response-message").html("<span style='color: green;'>" + response + "</span>");
            $("#myForm")[0].reset(); // reset form
        },
        error: function(){
            $("#response-message").html("<span style='color: red;'>Terjadi kesalahan saat mengirim data.</span>");
        }
    });
}
});
</script>
</body>
</html>

```

Form Input dengan Validasi AJAX

Nama:

Email:

Data berhasil dikirim: Nama = picha, Email = nafisah.aliyah2006@gmail.com

Form yang sebelumnya menggunakan metode submit langsung ke proses_validasi.php, kini menggunakan **AJAX** untuk mengirim data tanpa me-refresh 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)

```

<!DOCTYPE html>
<html>
<head>
    <title>Form Input dengan Validasi AJAX</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
</head>

```

```

<body>
  <h1>Form Input dengan Validasi AJAX</h1>
  <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>

    <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>

  <div id="response-message" style="margin-top: 20px;"></div>

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

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

    $("#nama-error, #email-error, #password-error").text("");
    $("#response-message").text("");

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

    if (email === "") {
      $("#email-error").text("Email harus diisi.");
      valid = false;
    }

    if (password.length < 8) {
      $("#password-error").text("Password minimal 8 karakter.");
      valid = false;
    }

    if (valid) {
      $.ajax({
        url: "proses_validasi.php",
        type: "POST",

```

```

        data: { nama: nama, email: email, password: password },
        success: function(response){
            $("#response-message").html("<span style='color: green;'>" +
response + "</span>");
            $("#myForm")[0].reset();
        },
        error: function(){
            $("#response-message").html("<span style='color:
red;'>Terjadi kesalahan saat mengirim data.</span>");
        }
    });
}
});
});
</script>
</body>
</html>

```

Nama:

Email:

Password: Password minimal 8 karakter.

- Input baru untuk **password** ditambahkan ke form HTML.
- Validasi dilakukan di dua sisi:
- **Client-side (jQuery):** Mengecek apakah panjang password minimal 8 karakter sebelum data dikirim.
- **Server-side (PHP):** Mengecek ulang panjang password agar tidak hanya bergantung pada validasi di browser.