



Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet-8: PHP - Form Proccessing

Mata Kuliah Desain dan Pemrograman Web

Pengampu: Tim Ajar Desain dan Pemrograman Web

Oktober 2024

Topik

- Konsep Form Proccessing dengan PHP

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 <code>dasarWeb</code> , 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/ isse</p> <p>Apa yang anda pahami dari penggunaan isset pada file tersebut. Catat di bawah ini pemahaman (soal no 1.1)</p> <p>Jawab : karena variabel umur belum di beri nilai, maka akan muncul hasil seperti berikut :</p> <pre> Anda belum dewasa atau variabel 'umur' tidak ditemukan. </pre>
5	<p>Tambahkan isi dari file isset.php tersebut dengan kode di bawah ini.</p> <pre> \$data = array("nama" => "Jane", "usia" => 25); if (isset(\$data["nama"])) { echo "Nama: " . \$data["nama"]; } else { echo "Variabel 'nama' tidak ditemukan dalam array."; } </pre>
6	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/isset.php buat hasil tidak menjadi 1 baris, jadi tampilan dari echo harus terpisah. Apa yang anda pahami dari penggunaan isset pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 1.2)</p> <p>Jawab : dikarenakan yang dipanggil hanya nama dalam array, maka hasilnya pun akan muncul namanya saja.</p>

Function empty

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

```

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

```

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

Praktikum 2. Function empty

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

1	Buat satu file baru bernama <code>empty.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>empty.php</code>
2	<pre> \$myArray = array(); // Array kosong if (empty(\$myArray)) { echo "Array tidak terdefinisi atau kosong."; } else { echo "Array terdefinisi dan tidak kosong."; } </pre>
3	<p>Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/dasarWeb/empty.php</code>. Apa yang anda pahami dari penggunaan <code>empty</code> pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.1)</p> <p>Jawab : dikarenakan nilai dalam array belum diisi maka hasilnya pun seperti dibawah ini.</p> <pre> Array tidak terdefinisi atau koso ng. </pre> <p>Program tetap dapat dijalankan karena penggunaan <code>empty</code> pada baris program.</p>
4	<p>Tambahkan isi dari file <code>empty.php</code> tersebut dengan kode di bawah ini.</p> <pre> if (empty(\$nonExistentVar)) { echo "Variabel tidak terdefinisi atau kosong."; } else { echo "Variabel terdefinisi dan tidak kosong."; } </pre>
5	<p>Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/dasarWeb/empty.php</code> buat hasil tidak menjadi 1 baris, jadi tampilan dari <code>echo</code> harus terpisah. Apa yang anda pahami dari penggunaan <code>empty</code> pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.2)</p> <p>Jawab : sama seperti sebelumnya, hasil dari output akan seperti ini.</p> <pre> Variabel tidak terdefinisi atau k osong. </pre> <p>Meskipun array tidak di instansiasi, namun dengan adanya fungsi <code>empty</code>, program masih tetap bisa dijalankan.</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>Jawab : setelah mengisi pada kotak yang ada dan meng-submitnya, maka hasil outputnya akan sesuai dengan apa yang telah diisi.</p> <p>Nama: asd Email: aowk@aowk.com</p>
5	<p>Buat satu file baru bernama form_self.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form_self.php</p>

```

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

    // Cek apakah form sudah disubmit
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
      // Validasi nama (contoh: pastikan nama tidak kosong)
      if (empty($_POST["nama"])) {
        $namaErr = "Nama harus diisi!";
      } else {
        $nama = $_POST["nama"];
        echo "Data berhasil disimpan!";
      }
    }
  ?>

  <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
    <label for="nama">Nama:</label>
    <input type="text" name="nama" id="nama" value="<?php echo $nama; ?>">
    <span class="error"><?php echo $namaErr; ?></span><br><br>

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

```

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

Jawab : setelah mengisi data pada textbox, maka akan muncul tanda berhasil setelah submit, dan jika dikosongkan dan disubmit, maka akan muncul text untuk mengisi textbox tersebut.

Form Input PHP

Data berhasil disimpan!

Nama:

HTML Injection

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

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

Contoh HTML Injection:

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

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

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

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

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

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

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

Cara Mengangani HTML Injection

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

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

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

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


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

Praktikum 4 : HTML Injection

Langkah-langkah Praktikum :

Langkah	Keterangan
1	Buat satu file baru bernama <code>html_aman.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>html_aman.php</code>
2	<pre>\$input = \$_POST['input']; \$input = htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8');</pre>
3	Tambahkan script lainnya supaya langkah 2 bisa berjalan normal. Lalu simpan file , kemudian buka browser dan jalankan/refresh <code>localhost/dasarWeb/html_aman.php</code>
4	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1) Jawab : baris kode tersebut berguna untuk menghindari injeksi javascript.
5	Ketikkan kode tambahan pada langkah 6 di dalam <code>html_aman.php</code>
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 <code>localhost/dasarWeb/html_aman.php</code>
8	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.2) Jawab : jika pengisian data pada textbox sesuai, maka data akan berhasil disimpan.

Regular Expression (Regex)

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

Basic Regex Patterns:

- **Karakter Tunggal:** Karakter apa pun akan cocok dengan dirinya sendiri dalam Regex, kecuali karakter khusus yang harus dihindari dengan karakter pelarian \.
Contoh: Pola /a/ akan cocok dengan huruf "a" dalam teks.
- **Karakter Spesifik:** Anda dapat mencocokkan karakter tertentu dengan menyebutkannya dalam pola.
Contoh: Pola /hello/ akan cocok dengan teks "hello" dalam teks.
- **Karakter Khusus:** Beberapa karakter khusus dalam Regex harus dihindari dengan karakter pelarian (\) jika Anda ingin mencocokkannya secara harfiah. Contoh karakter khusus: ., *, +, ?, |, [,], (,), {, }, ^, \$, \.
- **Karakter Set:** Anda dapat mencocokkan karakter dari satu set karakter dengan menggunakan []. Misalnya, /[aeiou]/ akan mencocokkan salah satu huruf vokal.
- **Rentang Karakter:** Anda dapat menggunakan - dalam karakter set untuk menentukan rentang karakter. Misalnya, /[a-z]/ akan mencocokkan huruf kecil apa pun.
- **Kuantifier:** Anda dapat mengatur berapa kali karakter atau kelompok karakter sebelumnya harus muncul. Contoh kuantifier:
 - *: 0 atau lebih kali
 - +: 1 atau lebih kali
 - ?: 0 atau 1 kali
 - {n}: Persis n kali
 - {n,}: Setidaknya n kali
 - {n,m}: Setidaknya n kali, maksimal m kali

Contoh Penggunaan Regex:

1. Mencocokkan pola

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

2. Mencocokkan karakter set


```
$pattern = '/[0-9]/'; // Cocokkan satu digit.
$text = 'There are 3 apples.';
if (preg_match($pattern, $text)) {
    echo "Digit ditemukan!";
} else {
    echo "Tidak ada digit!";
}
```

Praktikum 5 : Penggunaan Regex Pada PHP

Langkah-langkah Praktikum :

Langkah	Keterangan
1	Buat satu file baru bernama <code>regex.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>regex.php</code>
2	<pre><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 <code>localhost/dasarWeb/regex.php</code>
4	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.1) Jawab : setelah program dijalankan, akan memeriksa tiap huruf dalam teks, apakah terdapat huruf kecil didalamnya.
5	Ketikkan kode tambahan pada langkah 6 di dalam <code>regex.php</code>
6	<pre><code>\$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!"; }</code></pre>

7	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php
8	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.2) Jawab : setelah program dijalankan, akan memeriksa apakah terdapat kesamaan pada teks dengan angka.
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	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.3) Jawab : setelah program dijalankan, maka akan memeriksa mana kata apel yang nantinya akan digantikan dengan banana.
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	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.4) Jawab : setelah program dijalankan akan memeriksa mana yang sesuai dengan pattern yang telah disiapkan. Dan karena yang pertama ditemukan kecocokannya adalah god, maka hasil outputnya adalah god.
17	Soal 5.5: Buatlah script langkah 14 dengan variable pattern yang di ubah menggunakan `?` (0 atau 1 kali). Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.5) Jawab : setelah progra dijalankan, maka akan memeriksa adakah kata god pada text.

18	<p>Soal 5.6: Buatlah script langkah 14 dengan variable pattern yang di ubah menggunakan <code>`{n,m}`</code>. Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.6)</p> <p>Jawab : setelah program dijalankan, akan memeriksa adakah huruf n atau m pada text. Dan jika tidak ditemukan maka outputnya akan “tidak ada yang cocok”.</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) Jawab : setelah mengisi form yang ada dan di submit, maka hasilnya akan muncul sesuai dengan pilihan dalam form sebelumnya.</p>

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>
```

6	<p>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)</p> <p>Jawab : setelah form diisi dan disubmit, hasilnya akan segera muncul dalam laman yang sama tanpa berganti.</p>
---	---

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	<p>Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_validasi.php</p>
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.1) Jawab : setelah form diisi dan disubmit, maka data akan berhasil disimpan juga data akan kembali ditampilkan untuk mengkonfirmasi kembali.</p>
5	<p>Kembangkan file bernama form_validasi.php di dalam folder dasarWeb. Ketikkan kode pada langkah 6 di dalam form_validasi.php</p>

6	<pre> <!DOCTYPE html> <html> <head> <title>Form Input dengan Validasi</title> <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script> </head> <body> <h1>Form Input dengan Validasi</h1> <form id="myForm" method="post" action="proses_validasi.php"> <label for="nama">Nama:</label> <input type="text" id="nama" name="nama">
 <label for="email">Email:</label> <input type="text" id="email" name="email">
 <input type="submit" value="Submit"> </form> <script> \$(document).ready(function() { \$("#myForm").submit(function(event) { var nama = \$("#nama").val(); var email = \$("#email").val(); var valid = true; if (nama === "") { \$("#nama-error").text("Nama harus diisi."); valid = false; } else { \$("#nama-error").text(""); } if (email === "") { \$("#email-error").text("Email harus diisi."); valid = false; } else { \$("#email-error").text(""); } if (valid) { (event.preventDefault()); // Menghentikan pengiriman form jika validasi gagal } }); }); </script> </body> </html> </pre>
7	<p>Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_validasi.php</p>
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.2) Jawab : sama seperti sebelumnya, setelah form diisi dan disubmit, maka data akan berhasil disimpan juga data akan kembali ditampilkan untuk mengkonfirmasi kembali. Jika terdapat kesalahan dalam pengisian data, maka akan muncul text peringatan agar mengisi data dengan sesuai.</p>
9	<p>Soal 7.3: Buatlah script langkah 6 dengan menggunakan ajax. Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.3) Jawab : jika form tidak diisi dan langsung dikirim, maka akan mengulang hingga form diisi dengan benar agar bisa dikirim dan disimpan datanya.</p>
10	<p>Soal 7.4: Tambahkan validasi untuk password dengan ketentuan minimal 8 karakter dengan menggunakan jQuery dan PHP tambahkan pada langkah 9. Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.4) Jawab : akan dilakukan pengecekan panjang password yang mana ketika password tidak sesuai yang ditentukan, akan false.</p>

Buatlah sebuah formulir kontak sederhana yang memungkinkan pengunjung website mengirimkan pesan. Formulir ini akan menggunakan PHP untuk memproses pengiriman email, sementara jQuery akan digunakan untuk melakukan validasi input data secara real-time.

Ketentuan :

- **Elemen Form:**
 - Nama lengkap (teks)
 - Alamat email (email)
 - Subjek (teks)
 - Pesan (textarea)
 - Tombol kirim
- **Validasi dengan jQuery:**
 - Nama lengkap minimal 3 karakter
 - Alamat email harus dalam format yang benar
 - Subjek dan pesan tidak boleh kosong
 - Tampilkan pesan error di samping setiap field yang tidak valid
- **Pengiriman Email dengan PHP:**
 - Gunakan fungsi mail() di PHP untuk mengirimkan email.
 - Email berisi data yang diinputkan oleh pengguna.
 - Tampilkan pesan sukses setelah email berhasil dikirim.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Formulir Kontak</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <style>
    .error {
      color: red;
    }
  </style>
</head>

<body>
  <h1>Formulir Kontak</h1>
  <form id="kontak-form">
    <label for="nama">Nama Lengkap:</label>
    <input type="text" id="nama" name="nama">
    <span id="nama-error" class="error"></span><br>

    <label for="email">Alamat Email:</label>
    <input type="email" id="email" name="email">
    <span id="email-error" class="error"></span><br>

    <label for="subjek">Subjek:</label>
    <input type="text" id="subjek" name="subjek">
    <span id="subjek-error" class="error"></span><br>

    <label for="pesan">Pesan:</label>
    <textarea id="pesan" name="pesan"></textarea>
```

```

<span id="pesan-error" class="error"></span><br>

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

<script>
$(document).ready(function() {
    $("#kontak-form").submit(function(event) {
        event.preventDefault();
        var nama = $("#nama").val();
        var email = $("#email").val();
        var subjek = $("#subjek").val();
        var pesan = $("#pesan").val();
        var valid = true;

        // Validasi Nama
        if (nama.length < 3) {
            $("#nama-error").text("Nama harus minimal 3 karakter.");
            valid = false;
        } else {
            $("#nama-error").text("");
        }

        // Validasi Email
        var emailRegex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
        if (!emailRegex.test(email)) {
            $("#email-error").text("Alamat email harus dalam format yang benar.");
            valid = false;
        } else {
            $("#email-error").text("");
        }

        // Validasi Subjek
        if (subjek == "") {
            $("#subjek-error").text("Subjek tidak boleh kosong.");
            valid = false;
        } else {
            $("#subjek-error").text("");
        }

        // Validasi Pesan
        if (pesan == "") {
            $("#pesan-error").text("Pesan tidak boleh kosong.");
            valid = false;
        } else {
            $("#pesan-error").text("");
        }

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

```

```

        url: "proses_kontak.php",
        data: $("#kontak-form").serialize(),
        success: function(data) {
            console.log(data);
            alert("Email berhasil dikirim!");
        }
    });
}
});
</script>
</body>

</html>

```

```

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = $_POST["nama"];
    $email = $_POST["email"];
    $subjek = $_POST["subjek"];
    $pesan = $_POST["pesan"];

    // Kirim email
    $to = "tujuan@email.com"; // Ganti dengan alamat email tujuan
    $subject = $subjek;
    $message = "Nama: $nama\nEmail: $email\nPesan: $pesan";
    $headers = "From: $email\r\n";

    if (mail($to, $subject, $message, $headers)) {
        echo "Email berhasil dikirim!";
    } else {
        echo "Gagal mengirim email.";
    }
}
?>

```

Formulir Kontak

Nama Lengkap:

Alamat Email:

Subjek:

Pesan:

Formulir Kontak

Nama Lengkap:

Alamat Email:

Subjek:

Pesan:

localhost says

Email berhasil dikirim!

OK