



## 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

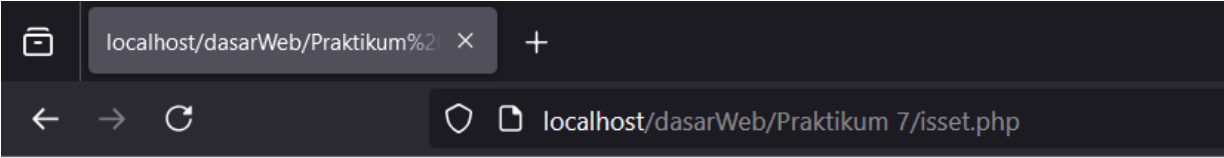
**isset()** adalah sebuah fungsi dalam PHP yang digunakan untuk memeriksa apakah suatu variabel telah didefinisikan (ada) atau tidak. Fungsi ini mengembalikan **true** jika variabel telah didefinisikan dan memiliki nilai, dan **false** jika variabel tidak ada atau bernilai **null**. Berikut adalah contoh script dan penjelasan minimal mengenai **isset()**:

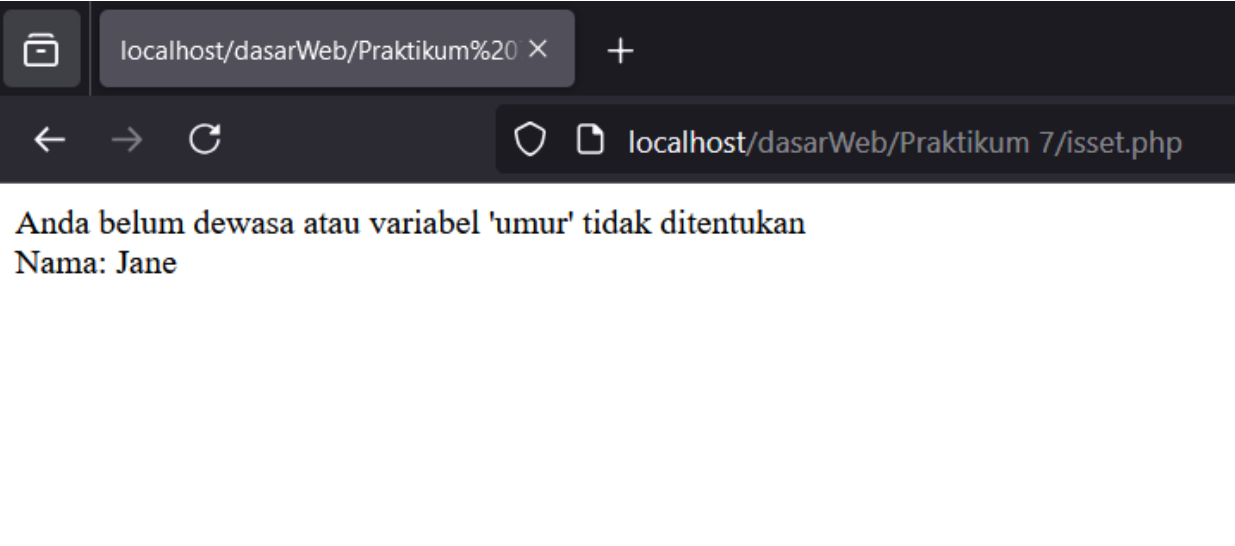
```
$nama = "John";  
if (isset($nama)) {  
    echo "Variabel 'nama' telah didefinisikan.";  
} else {  
    echo "Variabel 'nama' tidak ditemukan atau bernilai null.";  
}
```

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 <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) &amp;&amp; \$umur &gt;= 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 diitentukan</p> <p>Menggunakan variabel \$umur untuk menentukan apakah seseorang sudah dewasa atau belum. Jika, variabel \$umur dideklarasikan, tetapi belum diinisialisasi dengan nilai. Artinya, \$umur tidak memiliki nilai.</p> <p>isset(\$umur) memeriksa apakah variabel \$umur sudah didefinisikan (yaitu, tidak null). Jika variabel sudah didefinisikan dan nilainya lebih besar atau sama dengan 18, maka kondisi dianggap benar.</p> <p>Jika \$umur tidak diatur atau nilainya kurang dari 18, maka program akan memberikan informasi bahwa seseorang belum dewasa atau variabel tersebut belum diatur.</p>
5	<p>Tambahkan isi dari file isset.php tersebut dengan kode di bawah ini.</p> <pre> \$data = array("nama" =&gt; "Jane", "usia" =&gt; 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 ditentukan Nama: Jane</p> <p>memeriksa keberadaan elemen dalam array \$data dan menampilkan informasi berdasarkan hasil pemeriksaan tersebut.</p> <p>array asosiatif \$data dideklarasikan dengan dua elemen: nama yang memiliki nilai "Jane" dan umur yang memiliki nilai 25. Array ini menggunakan kunci untuk mengakses nilai-nilai tersebut. <code>isset(\$data["nama"])</code> digunakan untuk memeriksa apakah elemen dengan kunci nama ada dalam array \$data. Fungsi <code>isset()</code> mengembalikan true jika elemen tersebut ada dan tidak null.</p>
--	--

### Function empty

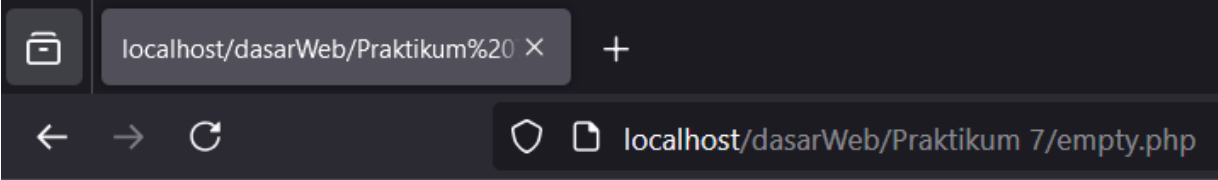
Fungsi **empty()** pada PHP digunakan untuk memeriksa apakah suatu variabel kosong atau tidak terdefinisi. Fungsi ini mengembalikan **true** jika variabel kosong atau tidak terdefinisi, dan **false** jika variabel memiliki nilai atau telah didefinisikan. Berikut adalah penjelasan dan contoh penggunaan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

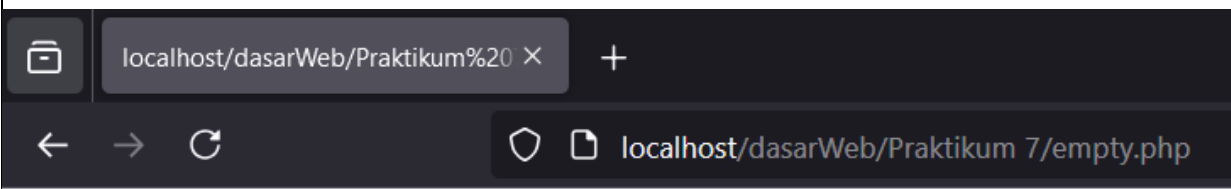
```
$nama = ""; // Variabel string kosong
if (empty($nama)) {
    echo "Nama tidak terdefinisi atau kosong.";
} else {
    echo "Nama terdefinisi dan tidak kosong.";
}
```

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

### Praktikum 2. Function empty

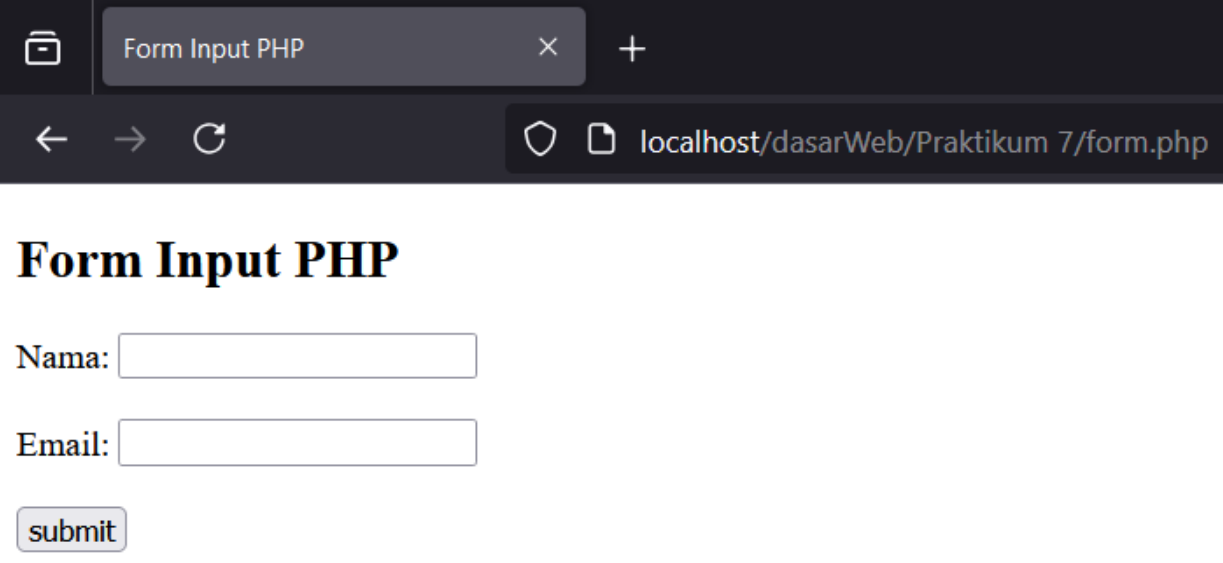
Langkah	Keterangan
---------	------------

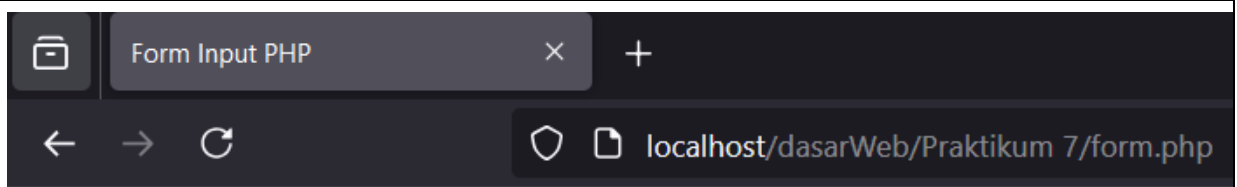
1	Buat satu file baru bernama <code>empty.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>empty.php</code>
2	<pre> \$myArray = array(); // Array kosong if (empty(\$myArray)) {     echo "Array tidak terdefinisi atau kosong."; } else {     echo "Array terdefinisi 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>Array tidak terdefinisi atau kosong.</p> <p>Memeriksa keberadaan dan keadaan variabel <code>\$nama</code> sebelum melakukan operasi lebih lanjut.</p> <p>fungsi <code>empty(\$nama)</code> digunakan untuk memeriksa apakah variabel <code>\$nama</code> tidak ada, atau jika ada, apakah nilainya adalah salah satu dari yang dianggap kosong dalam PHP (seperti <code>null</code>, <code>false</code>, <code>""</code>, <code>0</code>, atau array kosong). Jika <code>\$nama</code> kosong, kondisi akan dianggap benar.</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>Array tidak terdefinisi atau kosong. Variabel tidak terdefinisi atau kosong.</p> <p>Memeriksa apakah variabel \$nonExistentVar ada dan memiliki nilai. Fungsi empty(\$nonExistentVar) digunakan untuk memeriksa keadaan variabel \$nonExistentVar. Fungsi empty() mengembalikan true jika variabel tidak ada, bernilai null, adalah string kosong (""), adalah angka nol (0), adalah array kosong, atau adalah boolean false. Karena \$nonExistentVar tidak terdefinisi, output yang dihasilkan akan selalu "Variabel tidak terdefinisi atau kosong." Ini menunjukkan bahwa fungsi empty() dapat menangani variabel yang tidak dideklarasikan tanpa menyebabkan kesalahan.</p>
--	---

### Praktikum Bagian 3 : Form Input PHP

Langka h	Keterangan
1	Buat satu file baru bernama form.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form.php
2	<pre> &lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt;   &lt;title&gt;Form Input PHP&lt;/title&gt; &lt;/head&gt; &lt;body&gt;   &lt;h2&gt;Form Input PHP&lt;/h2&gt;   &lt;form method="post" action="proses_form.php"&gt;     &lt;label for="nama"&gt;Nama:&lt;/label&gt;     &lt;input type="text" name="nama" id="nama" required&gt;&lt;br&gt;&lt;br&gt;      &lt;label for="email"&gt;Email:&lt;/label&gt;     &lt;input type="email" name="email" id="email" required&gt;&lt;br&gt;&lt;br&gt;      &lt;input type="submit" name="submit" value="Submit"&gt;   &lt;/form&gt; &lt;/body&gt; &lt;/html&gt; </pre>

3	<p>Buat satu file baru bernama <code>proses_form.php</code> di dalam folder <code>dasarWeb</code>. Ketikkan kode pada langkah 2 di dalam <code>proses_form.php</code></p> <pre data-bbox="268 185 1233 663">&lt;?php if (\$_SERVER["REQUEST_METHOD"] == "POST") {     \$nama = \$_POST["nama"];     \$email = \$_POST["email"];      echo "Nama: " . \$nama . "&lt;br&gt;";     echo "Email: " . \$email; } ?&gt;</pre>
4	<p>Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/dasarWeb/proses_form.php</code>. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 3.1)</p> <p>Tampilan Awal:</p>  <p>Input nama dan Email:</p>

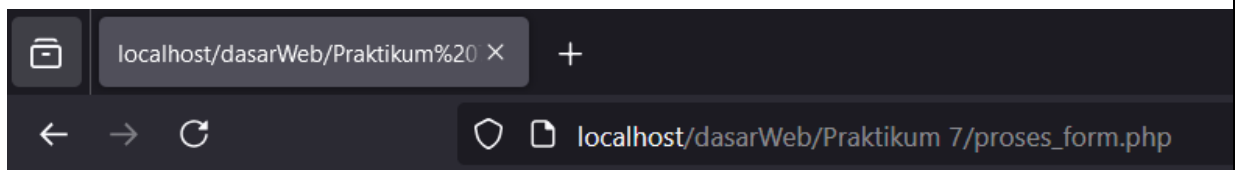


## Form Input PHP

Nama:

Email:

Setelah di klik submit:



Nama: danen

Email: danen@mail.com

Merupakan halaman HTML yang berisi form untuk menginput nama dan email, serta kode PHP untuk memproses data yang dikirimkan melalui form tersebut.

Untuk metode yang digunakan adalah POST, yang menandakan bahwa form telah disubmit.

Jika ya, maka data yang diinputkan pengguna (nama dan email) diambil dari array superglobal `$_POST`. Dengan menggunakan metode POST, data tidak terlihat di URL.

Nilai nama dan email kemudian ditampilkan di halaman dengan menggunakan echo.

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>

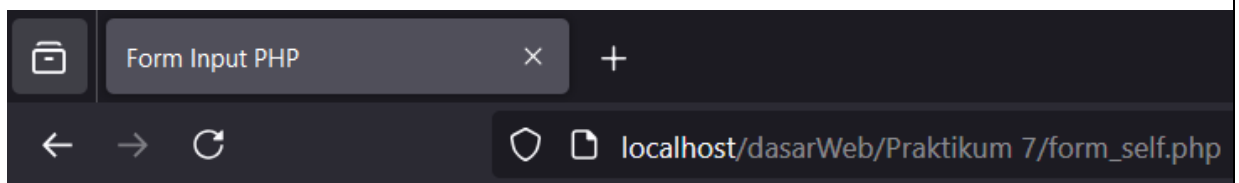
        <input type="submit" name="submit" value="Submit">
    </form>
</body>
</html>

```

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)

Tampilan Awal:

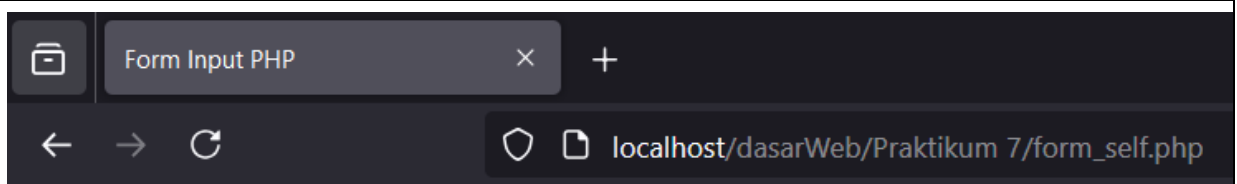


## Form Input PHP

Nama:

Nama di input:

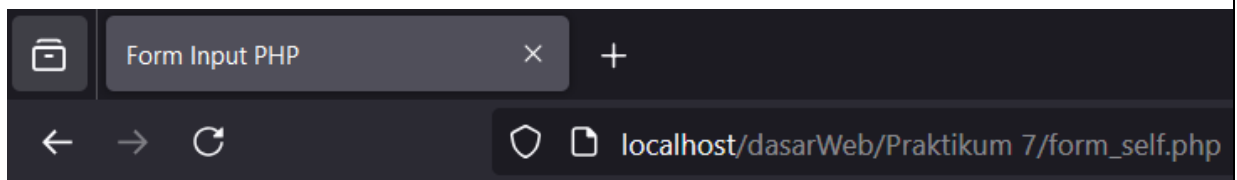




## Form Input PHP

Nama:

Ketika Klik Submit:

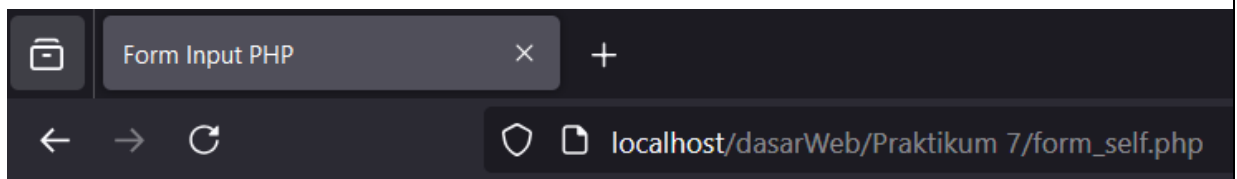


## Form Input PHP

Data berhasil disimpan

Nama:

Jika form kosong dan belum di submit:



## Form Input PHP

Nama:  Nama harus diisi

	<p>Membuat halaman HTML yang berisi formulir untuk menginput nama dengan validasi menggunakan PHP.</p> <ul style="list-style-type: none"> <li>• Mendeklarasikan dua variabel: \$namaErr untuk menyimpan pesan kesalahan dan \$nama untuk menyimpan input nama.</li> <li>• Memeriksa apakah metode permintaan adalah POST, yang menunjukkan bahwa formulir telah disubmit.</li> <li>• Jika field nama kosong, pesan kesalahan "Nama harus diisi" disimpan dalam \$namaErr.</li> <li>• Jika nama diisi, nilai disimpan dalam variabel \$nama, dan pesan "Data berhasil disimpan" ditampilkan.</li> <li>• Formulir dibuat dengan metode POST dan mengarah ke halaman yang sama, menggunakan htmlspecialchars(\$_SERVER["PHP_SELF"]) untuk menghindari serangan XSS dengan mengonversi karakter khusus ke entitas HTML.</li> <li>• Input field untuk nama menampilkan nilai sebelumnya jika ada (agar pengguna tidak perlu mengisi ulang).</li> <li>• Pesan kesalahan ditampilkan dalam elemen &lt;span&gt; dengan kelas error.</li> <li>• Tombol submit untuk mengirimkan data.</li> <li>• Jika pengguna tidak mengisi nama, akan muncul pesan kesalahan. Jika nama diisi, pesan "Data berhasil disimpan" ditampilkan.</li> <li>• Penggunaan htmlspecialchars() membantu melindungi aplikasi dari potensi serangan keamanan dengan memvalidasi input pengguna.</li> </ul>
--	--

## 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);

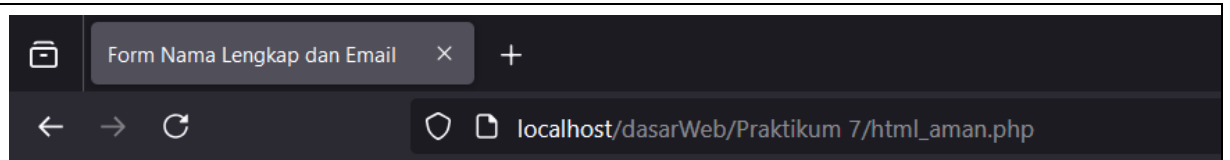
// Menggunakan htmlspecialchars untuk menghindari HTML injection saat menampilkan data
echo '<div>' . htmlspecialchars($user_input, ENT_QUOTES, 'UTF-8') . '</div>';
```

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 :

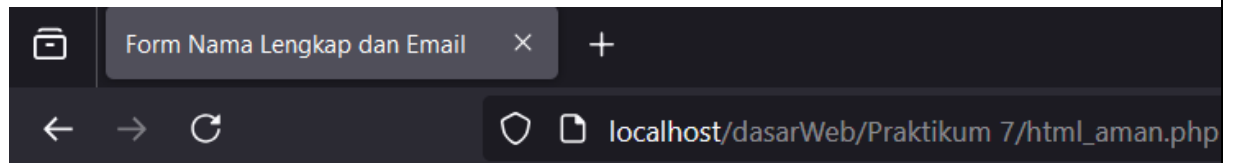
Langkah	Keterangan
1	Buat satu file baru bernama <code>html_aman.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>html_aman.php</code>
2	<pre>\$input = \$_POST['input']; \$input = htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8');</pre>
3	Tambahkan script lainnya 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)  Tampilan Awal:



### Form Input PHP

Nama:

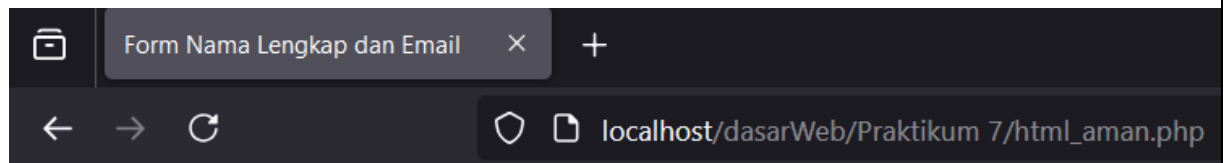
Form di isi kemudian klik submit:



### Form Input PHP

Nama:

Hasil:



### Form Input PHP

Nama:

Nama : Danen

Jika Form kosong:

Form Nama Lengkap dan Email

localhost/dasarWeb/Praktikum 7/html\_aman.php

### Form Input PHP

Nama:

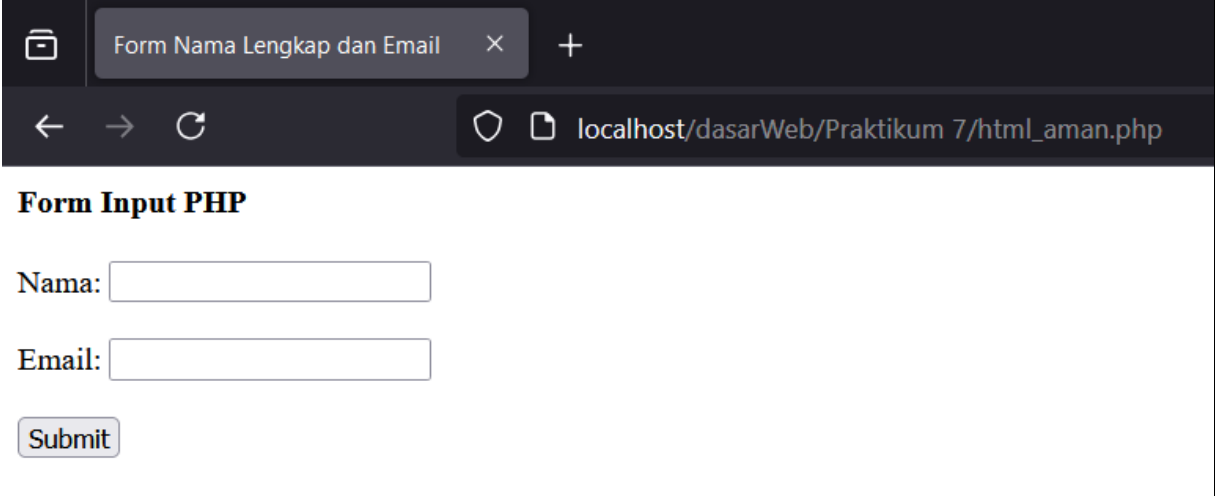
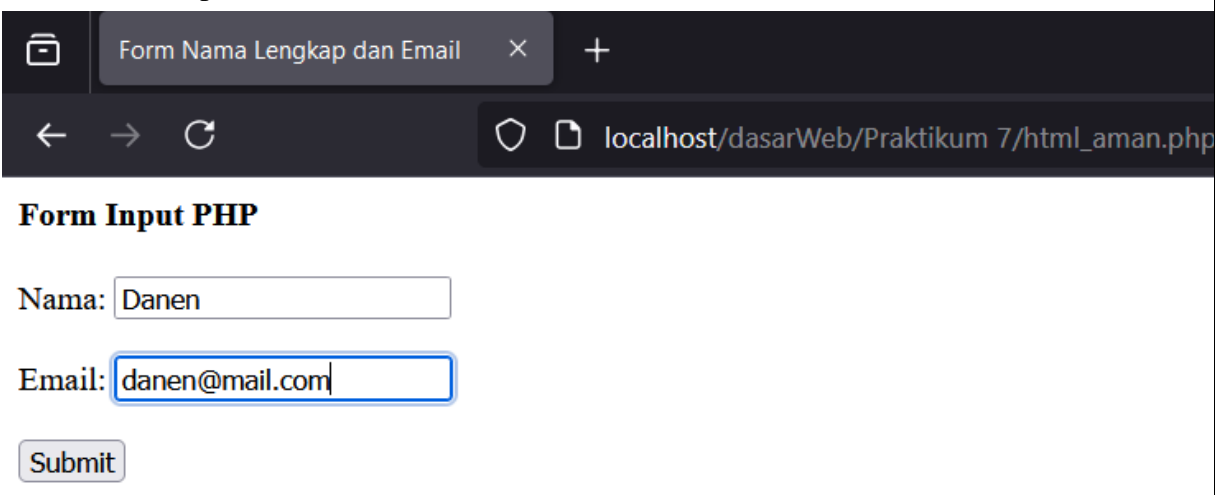
Submit Isi isian ini.

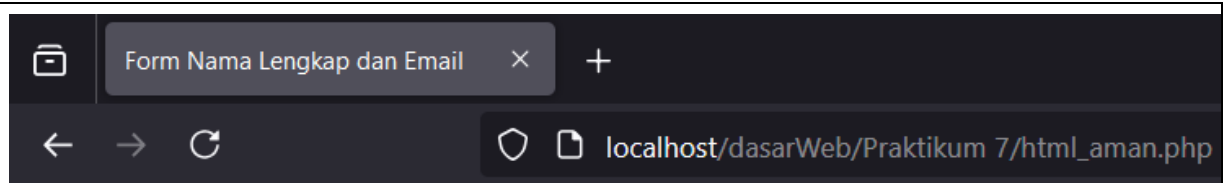
Membuat skrip PHP yang membuat form input untuk mengambil nama lengkap dari pengguna dan menampilkannya setelah pengguna mengirimkan form.

- if (\$\_SERVER["REQUEST\_METHOD"] == "POST"): Mengecek apakah form telah dikirim menggunakan metode POST.
- \$input = \$\_POST['input']; Mengambil nilai dari input form dengan nama input.
- htmlspecialchars(\$input, ENT\_QUOTES, 'UTF-8'): Mengamankan input dari karakter berbahaya (XSS) dengan mengkonversi karakter khusus menjadi entitas HTML.
- \$output = "Nama : " . \$input; Menyusun string output yang akan ditampilkan setelah form dikirim.
- <form method="post" action="">: Membuat form dengan metode POST dan tidak mengarah ke URL lain (action kosong).
- <input type="text" name="input" id="input" value="<?php echo \$input; ?>" required>: Field input untuk nama, dengan nilai awal yang diisi jika form telah dikirim.
- if (!empty(\$output)): Mengecek apakah output tidak kosong.
- echo "<div class='output'>" . \$output . "</div>"; Menampilkan output yang berisi nama yang telah diinputkan pengguna.

Membuat form yang mengumpulkan nama lengkap dari pengguna dan menampilkan nama tersebut setelah form disubmit, dengan keamanan dasar untuk mencegah serangan XSS.

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>Tampilan Awal:</p>  <p>Saat Form di input dan klik submit:</p>  <p>Hasil:</p>



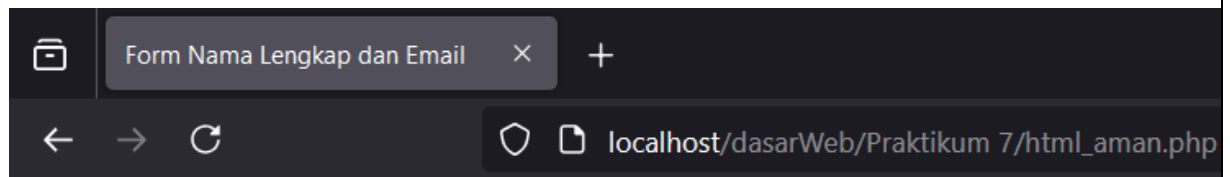
### Form Input PHP

Nama:

Email:

Nama: Danen  
Email: danen@mail.com

Jika tidak memasukkan email sesuai format:



### Form Input PHP

Nama:

Email:

Email tidak valid, silakan masukkan email yang benar.

Membuat skrip PHP yang membuat form untuk menginput nama lengkap dan alamat email dengan validasi email.

- `if ($_SERVER["REQUEST_METHOD"] == "POST")`: Mengecek apakah form telah disubmit dengan metode POST.
- `htmlspecialchars(...)`: Mengamankan kedua input dari karakter berbahaya untuk mencegah serangan XSS.
- `if (filter_var($email, FILTER_VALIDATE_EMAIL))`: Memeriksa apakah email yang diinput valid.
- Jika valid, hasilnya disusun dalam \$output: Nama: ... <br>Email: ....
- Jika tidak valid, \$outputEmail diisi dengan pesan kesalahan.
- `if (!empty($output))`: Mengecek apakah ada output yang valid untuk ditampilkan.
- `echo "<div class='output'>" . $output . "</div>";`: Menampilkan nama dan email jika keduanya valid.

	<ul style="list-style-type: none"> <li>❑ <code>if (!empty(\$outputEmail))</code>: Mengecek apakah ada pesan kesalahan.</li> <li>❑ <code>echo "&lt;div class='error'&gt;" . \$outputEmail . "&lt;/div&gt;";</code> Menampilkan pesan kesalahan jika email tidak valid.</li> </ul> <p>Membuat form untuk mengumpulkan nama dan email, melakukan validasi email, dan menampilkan hasilnya setelah form disubmit. Jika email yang dimasukkan tidak valid, maka akan diberi tahu untuk memasukkan email yang benar. Kode ini juga mengimplementasikan keamanan dasar untuk mencegah XSS.</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



- +: 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!";
}
```

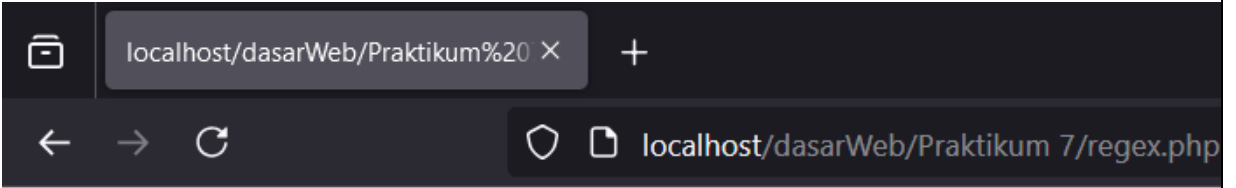
##### 2. Mencocokkan karakter set

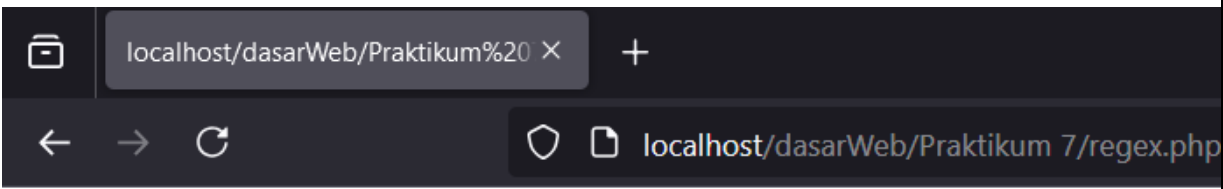
```
$pattern = '/[0-9]/'; // Cocokkan satu digit.
$text = 'There are 3 apples.';
if (preg_match($pattern, $text)) {
    echo "Digit ditemukan!";
} else {
    echo "Tidak ada digit!";
}
```

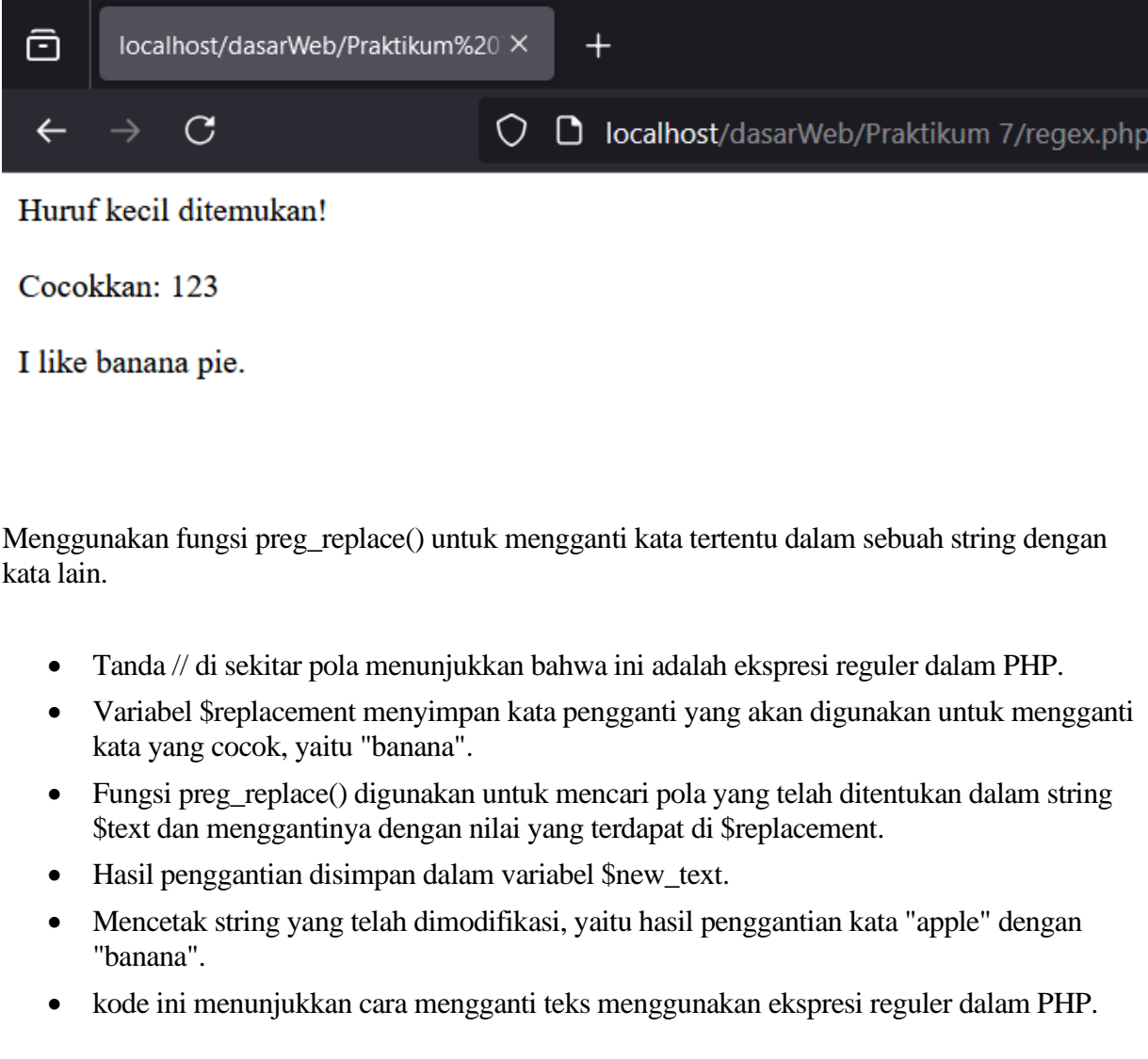
### Praktikum 5 : Penggunaan Regex Pada PHP

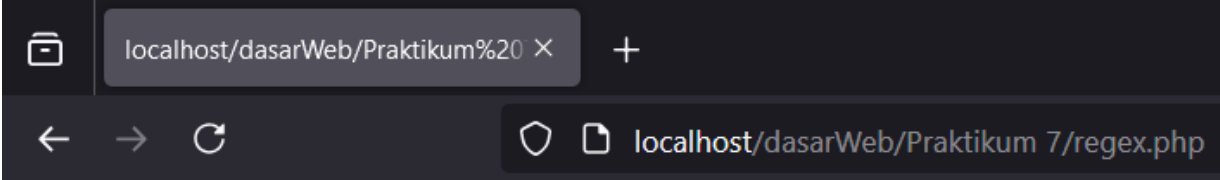
Langkah-langkah Praktikum :

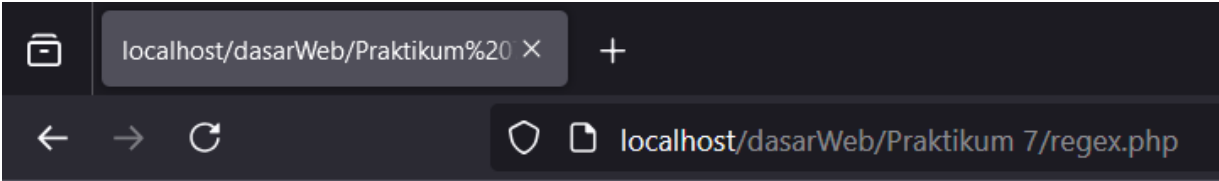
Langkah	Keterangan
1	Buat satu file baru bernama <code>regex.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>regex.php</code>
2	<pre><code>\$pattern = '/[a-z]/'; // Cocokkan huruf kecil. \$text = 'This is a Sample Text.'; if (preg_match(\$pattern, \$text)) {     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>  <p><b>Huruf kecil ditemukan!</b></p> <p>Menggunakan ekspresi reguler untuk memeriksa apakah ada huruf kecil (a-z) dalam sebuah string.</p> <ul style="list-style-type: none"> <li>• Tanda // di sekitar pola menunjukkan bahwa ini adalah ekspresi reguler dalam PHP.</li> <li>• Fungsi preg_match() digunakan untuk mencari pola yang telah ditentukan dalam string \$text.</li> <li>• Jika ada kecocokan (artinya, jika ada huruf kecil), fungsi ini mengembalikan nilai true.</li> <li>• Jika preg_match() mengembalikan true, maka akan mencetak "Huruf kecil ditemukan!".</li> </ul> <p>memeriksa apakah ada huruf kecil dalam string yang diberikan. Dalam contoh ini, string "This is a Sample text." mengandung huruf kecil (misalnya 's', 'a', 'm', 'p', 'l', dan 'e'), sehingga output yang dihasilkan adalah "Huruf kecil ditemukan!". Jika string tersebut tidak memiliki huruf kecil, maka output akan menjadi "Tidak ada huruf kecil!".</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	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.2)

	 <p>Huruf kecil ditemukan!</p> <p>Cocokkan: 123</p> <p>Menggunakan ekspresi reguler untuk mencari angka dalam sebuah string.</p> <ul style="list-style-type: none"> <li>• Tanda // di sekitar pola menunjukkan bahwa ini adalah ekspresi reguler dalam PHP.</li> <li>• Tanda + setelah [0-9] menunjukkan bahwa kita mencari satu atau lebih karakter yang cocok.</li> <li>• Fungsi preg_match() digunakan untuk mencari pola yang telah ditentukan dalam string \$text.</li> <li>• Jika ada kecocokan (artinya, jika ada angka), fungsi ini mengembalikan nilai true.</li> <li>• Parameter ketiga, \$matches, adalah array yang akan menyimpan hasil kecocokan.</li> <li>• Jika preg_match() mengembalikan true, maka kode ini mencetak "Cocokkan: " diikuti oleh angka yang ditemukan, yang disimpan di \$matches[0].</li> </ul> <p>Kode ini digunakan untuk mencari dan mencetak angka yang terdapat dalam string yang diberikan. Dalam contoh ini, string "There are 123 apples." mengandung angka "123", sehingga output yang dihasilkan adalah "Cocokkan: 123". Jika string tersebut tidak memiliki angka, maka output akan menjadi "Tidak ada yang cocok!".</p>
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)

	 <p>Huruf kecil ditemukan!</p> <p>Cocokkan: 123</p> <p>I like banana pie.</p> <p>Menggunakan fungsi preg_replace() untuk mengganti kata tertentu dalam sebuah string dengan kata lain.</p> <ul style="list-style-type: none"> <li>• Tanda // di sekitar pola menunjukkan bahwa ini adalah ekspresi reguler dalam PHP.</li> <li>• Variabel \$replacement menyimpan kata pengganti yang akan digunakan untuk mengganti kata yang cocok, yaitu "banana".</li> <li>• Fungsi preg_replace() digunakan untuk mencari pola yang telah ditentukan dalam string \$text dan menggantinya dengan nilai yang terdapat di \$replacement.</li> <li>• Hasil penggantian disimpan dalam variabel \$new_text.</li> <li>• Mencetak string yang telah dimodifikasi, yaitu hasil penggantian kata "apple" dengan "banana".</li> <li>• kode ini menunjukkan cara mengganti teks menggunakan ekspresi reguler dalam PHP.</li> </ul>
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	<p>Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php</p>
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>Menggunakan ekspresi reguler untuk mencari pola tertentu dalam sebuah string.</p> <ul style="list-style-type: none"> <li>• pola ekspresi reguler yang mencari kata yang dimulai dengan "g", diikuti oleh nol atau lebih huruf "o", dan diakhiri dengan huruf "d".</li> <li>• Tanda // di sekitar pola menunjukkan bahwa ini adalah ekspresi reguler dalam PHP.</li> <li>• Tanda * setelah o berarti bahwa karakter o bisa muncul nol kali atau lebih.</li> <li>• Fungsi preg_match() digunakan untuk mencari pola yang telah ditentukan dalam string \$text.</li> <li>• Jika ada kecocokan (artinya, jika ada kata yang sesuai dengan pola), fungsi ini mengembalikan nilai true.</li> <li>• Parameter ketiga, \$matches, adalah array yang akan menyimpan hasil kecocokan.</li> <li>• Jika preg_match() mengembalikan true, maka kode ini mencetak "Cocokkan: " diikuti oleh kata yang ditemukan yang disimpan di \$matches[0].</li> </ul> <p>string "god is good." mengandung kata "god", yang cocok dengan pola yang ditentukan (karena "g" diikuti oleh nol huruf "o" dan diakhiri dengan "d").</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: god</p> <p><code>"/go?d/"</code></p> <p>Pola ini mencari huruf "g", diikuti oleh nol atau satu huruf "o", dan diakhiri dengan huruf "d".</p> <p>Dengan pola kedua (go?d), kata "god" juga cocok, tetapi pola ini lebih fleksibel dalam hal memungkinkan "g" langsung diikuti oleh "d" tanpa "o".</p> <p>Hasil: Pola ini juga akan mencocokkan "god", sehingga hasil yang dicetak tetap sama.</p> <p>pola menunjukkan cara yang berbeda untuk menemukan kecocokan berdasarkan jumlah karakter yang diizinkan di antara "g" dan "d". akan memberi fleksibilitas dalam mencari pola teks yang dapat bervariasi dalam representasinya.</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: god</p> <p>Cocokkan: god</p> <p><b>"/go{0,1}d/"</b></p> <ul style="list-style-type: none"> <li>• Pola ini mencari huruf "g", diikuti oleh nol atau satu huruf "o", dan diakhiri dengan huruf "d".</li> <li>• Pola "go{0,1}d" lebih spesifik karena hanya mengizinkan nol atau satu huruf "o".</li> <li>• <b>Hasil:</b> Pola ini juga akan mencocokkan "god", sehingga hasil yang dicetak tetap sama.</li> </ul> <p>Pola ini menunjukkan cara yang berbeda untuk menemukan kecocokan berdasarkan jumlah karakter yang diizinkan di antara "g" dan "d". Dengan menggunakan {0,1}, memungkinkan mendapatkan kontrol yang lebih eksplisit atas jumlah kemunculan karakter yang dicari, tetapi hasilnya tetap sama.</p>
--	---

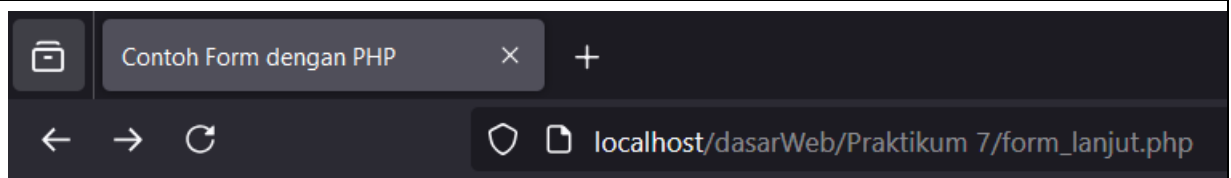
## Praktikum 6 : Form Lanjut

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> &lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt;     &lt;title&gt;Contoh Form dengan PHP&lt;/title&gt; &lt;/head&gt; &lt;body&gt;     &lt;h2&gt;Form Contoh&lt;/h2&gt;     &lt;form method="POST" action="proses_lanjut.php"&gt;         &lt;label for="buah"&gt;Pilih Buah:&lt;/label&gt;         &lt;select name="buah" id="buah"&gt;             &lt;option value="apel"&gt;Apel&lt;/option&gt;             &lt;option value="pisang"&gt;Pisang&lt;/option&gt;             &lt;option value="mangga"&gt;Mangga&lt;/option&gt;             &lt;option value="jeruk"&gt;Jeruk&lt;/option&gt;         &lt;/select&gt;          &lt;br&gt;          &lt;label&gt;Pilih Warna Favorit:&lt;/label&gt;&lt;br&gt;         &lt;input type="checkbox" name="warna[]" value="merah"&gt; Merah&lt;br&gt;         &lt;input type="checkbox" name="warna[]" value="biru"&gt; Biru&lt;br&gt;         &lt;input type="checkbox" name="warna[]" value="hijau"&gt; Hijau&lt;br&gt;          &lt;br&gt;          &lt;label&gt;Pilih Jenis Kelamin:&lt;/label&gt;&lt;br&gt;         &lt;input type="radio" name="jenis_kelamin" value="laki-laki"&gt; Laki-laki&lt;br&gt;         &lt;input type="radio" name="jenis_kelamin" value="perempuan"&gt; Perempuan&lt;br&gt;          &lt;br&gt;          &lt;input type="submit" value="Submit"&gt;     &lt;/form&gt; &lt;/body&gt; &lt;/html&gt;     &lt;pre&gt;     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 . "&lt;br&gt;";          if (!empty(\$selectedWarna)) {             echo "Warna favorit Anda: " . implode(", ", \$selectedWarna) . "&lt;br&gt;";         } else {             echo "Anda tidak memilih warna favorit.&lt;br&gt;";         }          echo "Jenis kelamin Anda: " . \$selectedJenisKelamin;     }     &lt;/pre&gt; </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>Tampilan Awal Form</p>





## Form Contoh

Pilih Buah:

Pilih Warna Favorit:

☐ Merah

☐ Biru

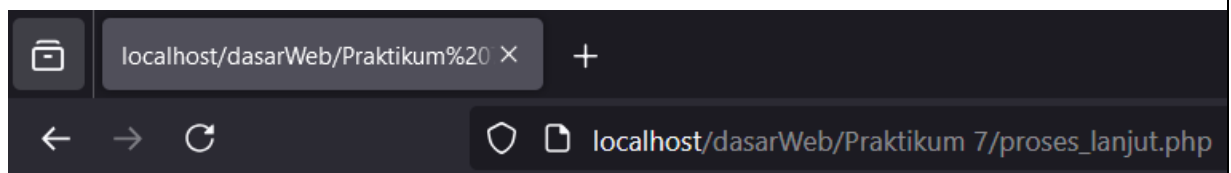
☐ Hijau

Pilih Jenis Kelamin:

☐ Laki-laki

☐ Perempuan

Saat Form Ter isi semua:

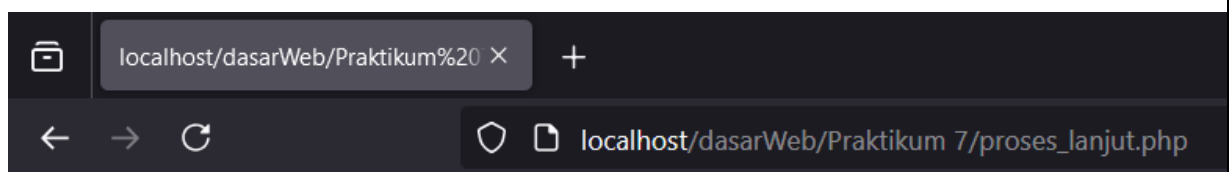


Anda memilih buah: pisang

Warna favorit Anda: biru

Jenis kelamin Anda: Laki-laki

Ketika Foem Tidak di isi:

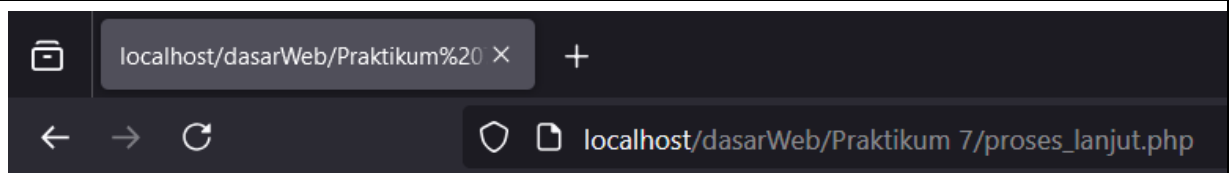


Anda memilih buah: apel

Anda tidak memilih warna favorit.

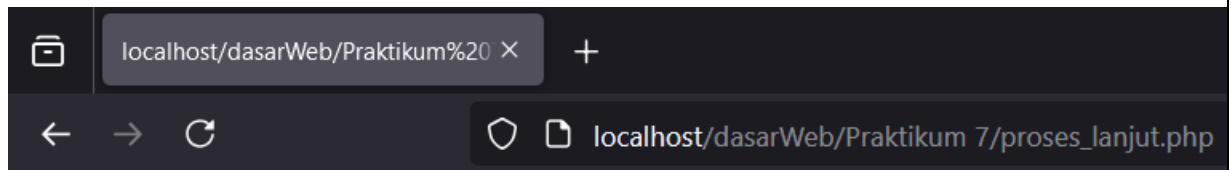
Jenis kelamin Anda: Tidak dipilih

Ketika Jenis kelamin tidak dipilih:



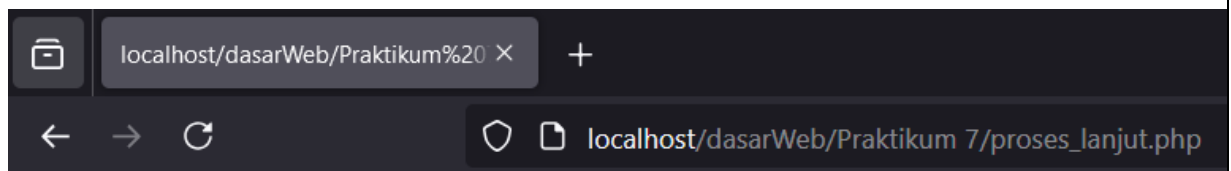
Anda memilih buah: apel  
Warna favorit Anda: biru  
Jenis kelamin Anda: Tidak dipilih

Ketika warna tidak dipilih:



Anda memilih buah: pisang  
Anda tidak memilih warna favorit.  
Jenis kelamin Anda: Laki-laki

Ketika warna di pilih semua:



Anda memilih buah: pisang  
Warna favorit Anda: merah, biru, hijau  
Jenis kelamin Anda: Laki-laki

Membuat Form HTML yang diolah dengan PHP

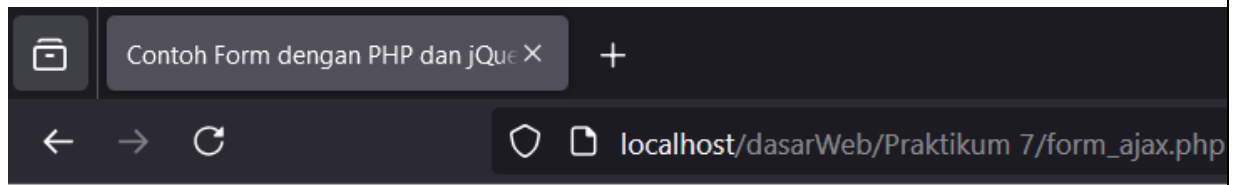
- `<form>` mengatur metode pengiriman data dengan `method="post"` dan tujuan pengolahan data dengan `action="proses_lanjut.php"`.
- `if ($_SERVER["REQUEST_METHOD"] == "POST")` memeriksa apakah form telah dikirim dengan metode POST.
- `$selectedBuah = $_POST['buah'];` mengambil nilai buah yang dipilih.
- Untuk warna, `$selectedWarna` diisi dengan data dari checkbox, jika ada yang dipilih, jika tidak, diatur menjadi array kosong.
- Jenis kelamin diperiksa dengan `isset`, dan jika tidak dipilih, diatur menjadi "Tidak dipilih".
- Jika warna favorit dipilih, ditampilkan dengan `implode` untuk menggabungkan array menjadi string. Jika tidak, ditampilkan pesan bahwa tidak ada warna yang dipilih.

	membuat form sederhana yang mengumpulkan data dari pengguna dan mengolahnya dengan PHP untuk ditampilkan kembali di halaman.
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>
	<pre> &lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt;   &lt;title&gt;Contoh Form dengan PHP dan jQuery&lt;/title&gt;   &lt;script src="https://code.jquery.com/jquery-3.6.0.min.js"&gt;&lt;/script&gt; &lt;/head&gt; &lt;body&gt;   &lt;h2&gt;Form Contoh&lt;/h2&gt;   &lt;form id="myForm"&gt;     &lt;label for="buah"&gt;Pilih Buah:&lt;/label&gt;     &lt;select name="buah" id="buah"&gt;       &lt;option value="apel"&gt;Apel&lt;/option&gt;       &lt;option value="pisang"&gt;Pisang&lt;/option&gt;       &lt;option value="mangga"&gt;Mangga&lt;/option&gt;       &lt;option value="jeruk"&gt;Jeruk&lt;/option&gt;     &lt;/select&gt;      &lt;br&gt;      &lt;label&gt;Pilih Warna Favorit:&lt;/label&gt;&lt;br&gt;     &lt;input type="checkbox" name="warna[]" value="merah"&gt; Merah&lt;br&gt;     &lt;input type="checkbox" name="warna[]" value="biru"&gt; Biru&lt;br&gt;     &lt;input type="checkbox" name="warna[]" value="hijau"&gt; Hijau&lt;br&gt;      &lt;br&gt;      &lt;label&gt;Pilih Jenis Kelamin:&lt;/label&gt;&lt;br&gt;     &lt;input type="radio" name="jenis_kelamin" value="laki-laki"&gt; Laki-laki&lt;br&gt;     &lt;input type="radio" name="jenis_kelamin" value="perempuan"&gt; Perempuan&lt;br&gt;      &lt;br&gt;      &lt;input type="submit" value="Submit"&gt;   &lt;/form&gt;    &lt;div id="hasil"&gt;     &lt;!-- Hasil akan ditampilkan di sini --&gt;   &lt;/div&gt;    &lt;script&gt;     \$(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);           }         });       });     });   &lt;/script&gt; &lt;/body&gt; &lt;/html&gt; </pre>

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)

Tampilan Awal:



## Form Contoh

Pilih Buah:

Pilih Warna Favorit:

☐ Merah

☐ Biru

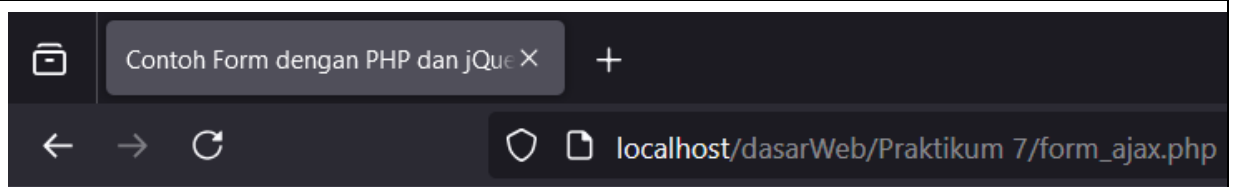
☐ Hijau

Pilih Jenis Kelamin:

☐ Laki-laki

☐ Perempuan

Ketika form di isi semua:



## Form Contoh

Pilih Buah:

Pilih Warna Favorit:

☐ Merah

☒ Biru

☐ Hijau

Pilih Jenis Kelamin:

☒ Laki-laki

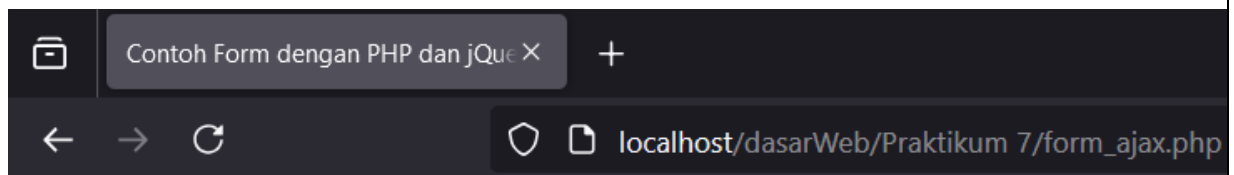
☐ Perempuan

Anda memilih buah: pisang

Warna favorit Anda: biru

Jenis kelamin Anda: Laki-laki

Ketika warna tidak dipilih:



## Form Contoh

Pilih Buah:

Pilih Warna Favorit:

☐ Merah

☐ Biru

☐ Hijau

Pilih Jenis Kelamin:

☒ Laki-laki

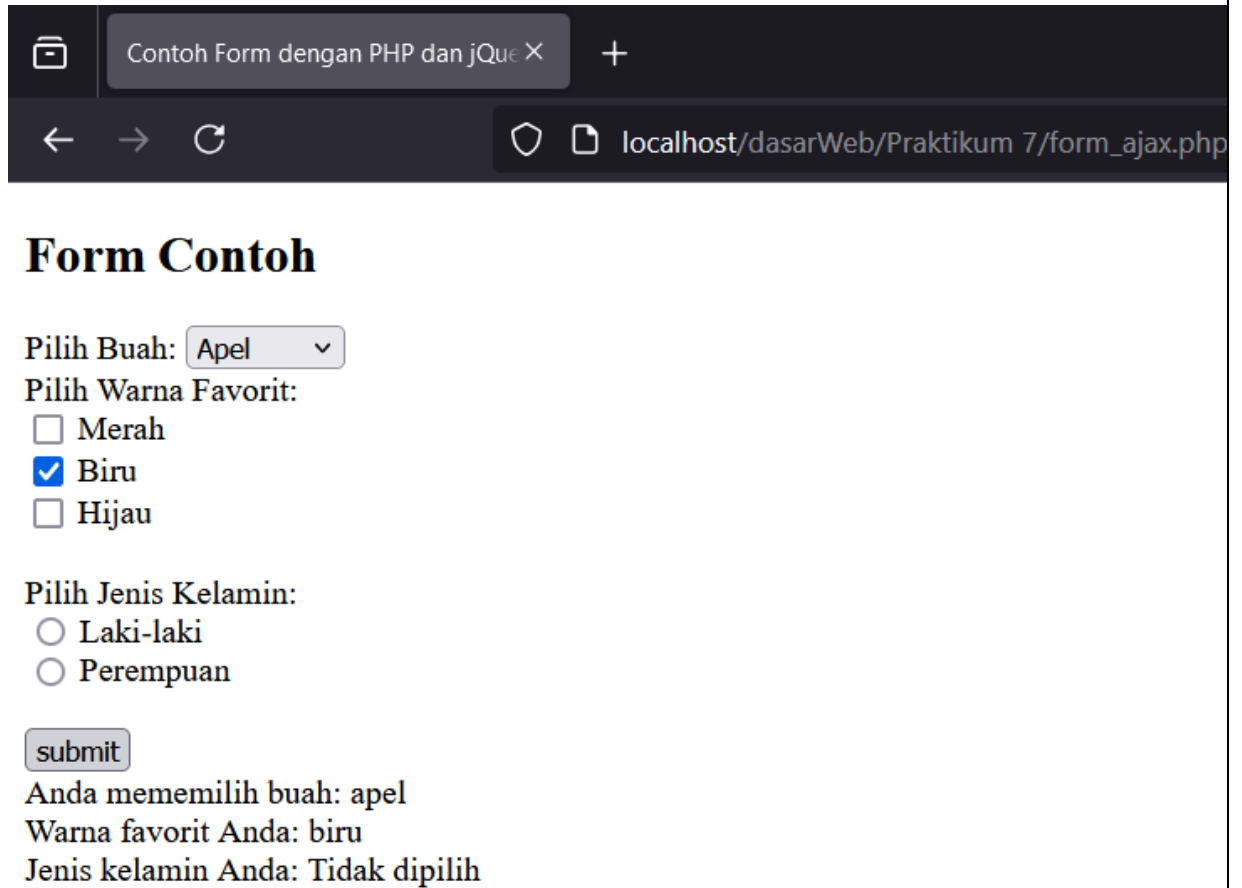
☐ Perempuan

Anda memilih buah: pisang

Anda tidak memilih warna favorit.

Jenis kelamin Anda: Laki-laki

Ketika jenis kelamin tidak dipilih:



Contoh Form dengan PHP dan jQuery X +

localhost/dasarWeb/Praktikum 7/form\_ajax.php

## Form Contoh

Pilih Buah: Apel ▾

Pilih Warna Favorit:

☐ Merah

☒ Biru

☐ Hijau

Pilih Jenis Kelamin:

☐ Laki-laki

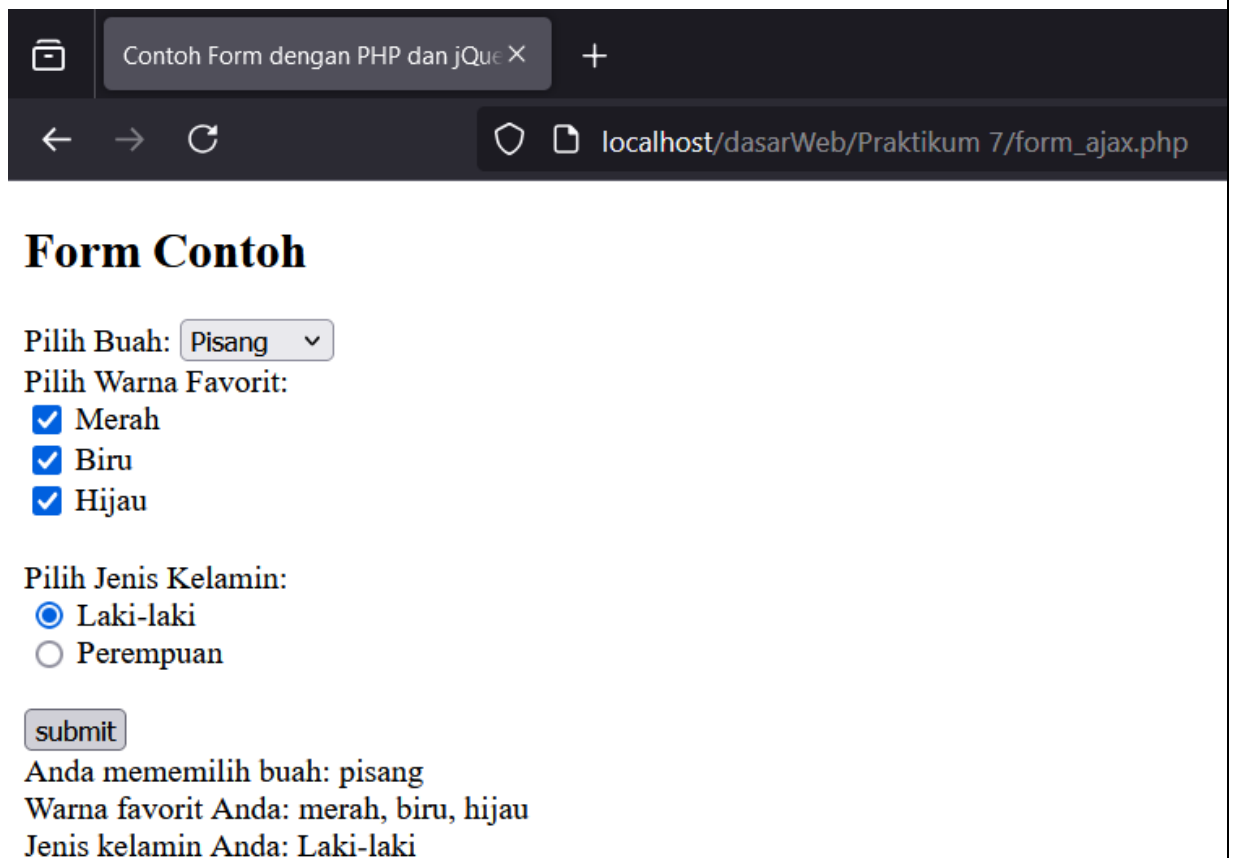
☐ Perempuan

Anda memilih buah: apel

Warna favorit Anda: biru

Jenis kelamin Anda: Tidak dipilih

Ketika warna dipilih semua:



Contoh Form dengan PHP dan jQuery X +

localhost/dasarWeb/Praktikum 7/form\_ajax.php

## Form Contoh

Pilih Buah: Pisang ▾

Pilih Warna Favorit:

☒ Merah

☒ Biru

☒ Hijau

Pilih Jenis Kelamin:

☒ Laki-laki

☐ Perempuan

Anda memilih buah: pisang

Warna favorit Anda: merah, biru, hijau

Jenis kelamin Anda: Laki-laki

Membuat form HTML yang menggunakan PHP dan jQuery untuk mengolah data tanpa memuat ulang halaman.

- `<form id="myForm">` adalah form yang akan diolah dengan jQuery. ID digunakan untuk mengidentifikasi form dalam JavaScript.
- `<div id="hasil">` adalah tempat untuk menampilkan respons dari server setelah form disubmit.
- `$(document).ready(function(){ ... });` memastikan bahwa kode di dalamnya hanya dijalankan setelah DOM siap.
- `$("#myForm").submit(function (e) { ... });` menetapkan fungsi yang akan dijalankan saat form disubmit.
- `e.preventDefault();` mencegah form dari pengiriman default yang akan memuat ulang halaman.
- `var formData = $("#myForm").serialize();` mengumpulkan semua data dari form dalam format yang dapat dikirim ke server.
- `$.ajax({ ... });` melakukan permintaan Ajax untuk mengirim data ke file `proses_lanjut.php`.
- `url: "proses_lanjut.php"` menentukan tujuan pengolahan data.
- `type: "POST"` menunjukkan bahwa data dikirim menggunakan metode POST.
- `data: formData` adalah data yang dikirim ke server.
- `success: function (response) { ... }` adalah callback yang akan dieksekusi jika permintaan berhasil. Respons dari server akan ditampilkan di div dengan ID `hasil`.

menggabungkan HTML, PHP, dan jQuery untuk membuat form yang responsif. Dengan menggunakan Ajax, form dapat mengirimkan data dan menerima respons dari server tanpa memuat ulang halaman.

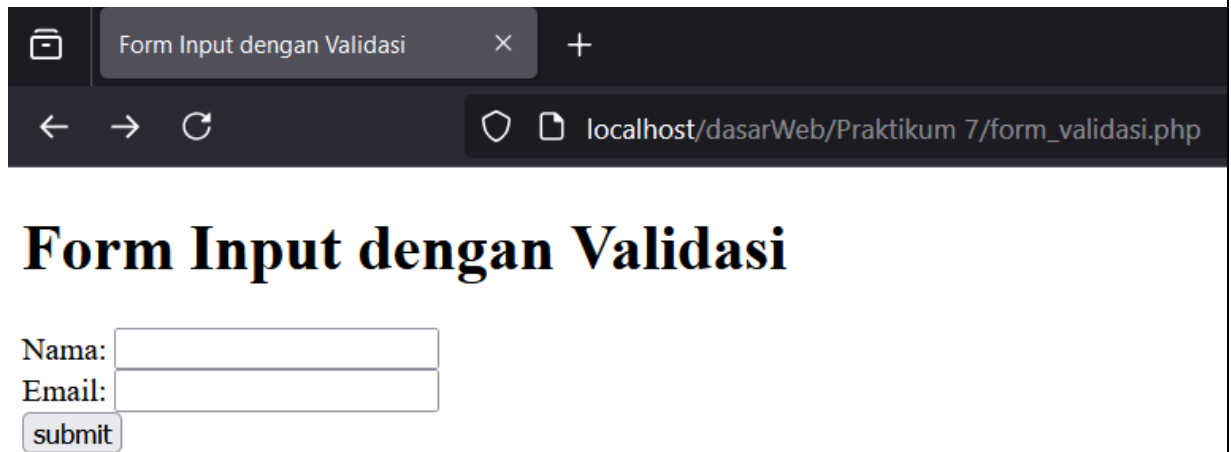
## Praktikum 7 : Validasi Form

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> &lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt;     &lt;title&gt;Form Input dengan Validasi&lt;/title&gt; &lt;/head&gt; &lt;body&gt;     &lt;h1&gt;Form Input dengan Validasi&lt;/h1&gt;     &lt;form method="post" action="proses_validasi.php"&gt;         &lt;label for="nama"&gt;Nama:&lt;/label&gt;         &lt;input type="text" id="nama" name="nama"&gt;         &lt;br&gt;           &lt;label for="email"&gt;Email:&lt;/label&gt;         &lt;input type="text" id="email" name="email"&gt;         &lt;br&gt;           &lt;input type="submit" value="Submit"&gt;     &lt;/form&gt; &lt;/body&gt; &lt;/html&gt; </pre> <pre> &lt;?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 . "&lt;br&gt;";         })     } 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";     } } ?&gt; </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)

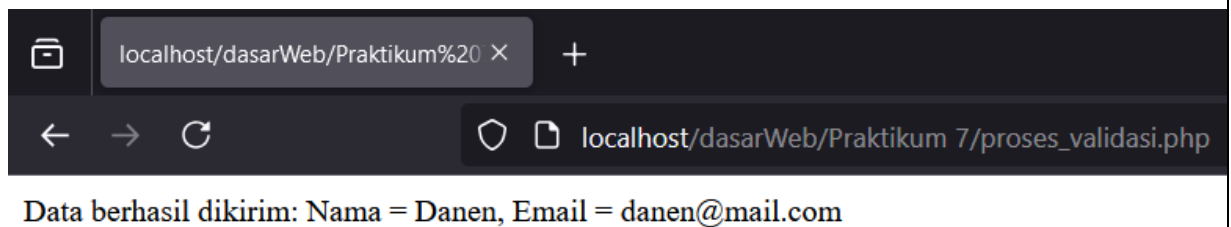


Tampilan Awal:



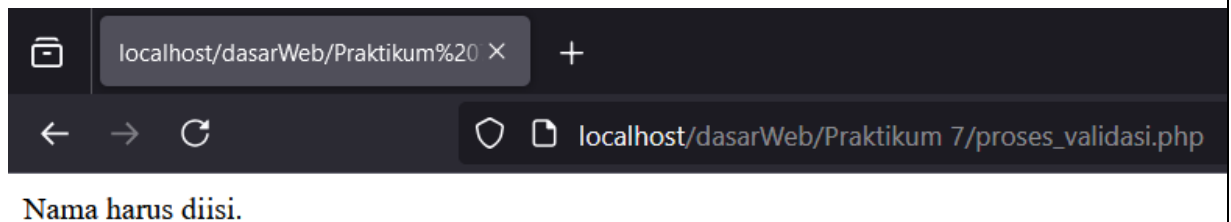
The screenshot shows a web browser window with a single tab titled 'Form Input dengan Validasi'. The address bar displays 'localhost/dasarWeb/Praktikum 7/form\_validasi.php'. The page content features a large heading 'Form Input dengan Validasi' in a bold, black, serif font. Below the heading, there is a form with two input fields: 'Nama:' and 'Email:'. The 'Nama:' field contains the text 'Danen'. Below these fields is a 'submit' button.

Ketika Form diisi dan submit:



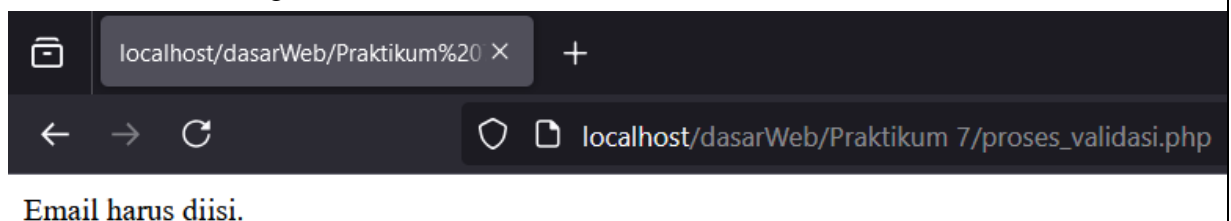
The screenshot shows a web browser window with a single tab titled 'localhost/dasarWeb/Praktikum%20'. The address bar displays 'localhost/dasarWeb/Praktikum 7/proses\_validasi.php'. The page content displays a success message: 'Data berhasil dikirim: Nama = Danen, Email = danen@mail.com'.

Ketika Nama Kosong:



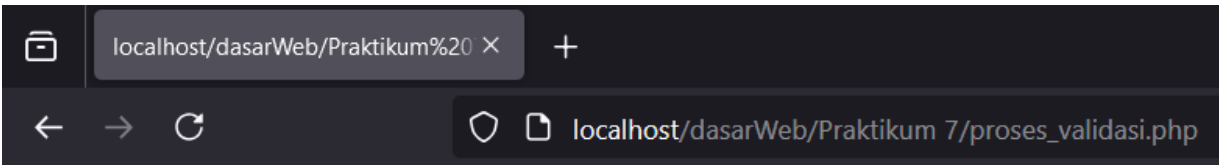
The screenshot shows a web browser window with a single tab titled 'localhost/dasarWeb/Praktikum%20'. The address bar displays 'localhost/dasarWeb/Praktikum 7/proses\_validasi.php'. The page content displays an error message: 'Nama harus diisi.'

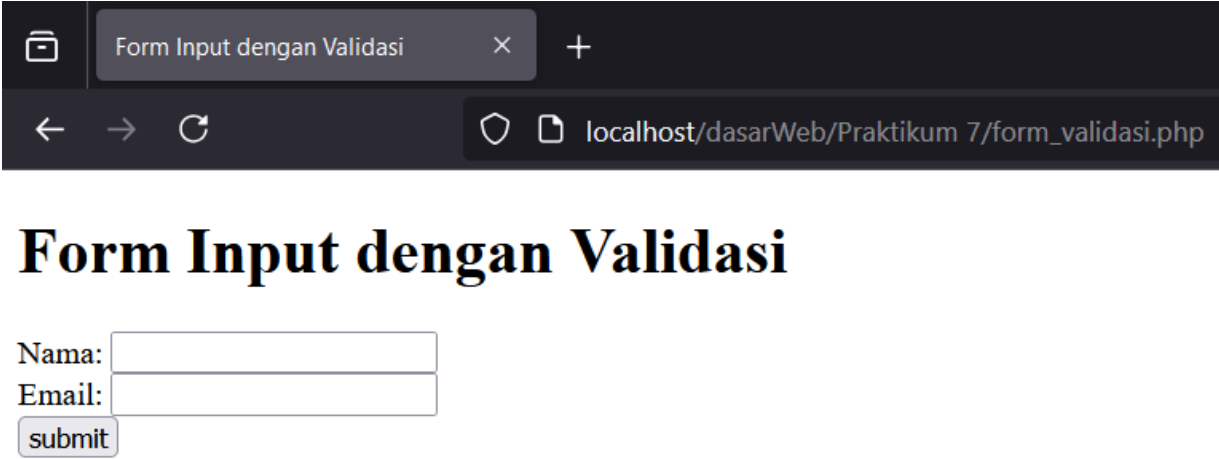
Ketika Email Kosong:



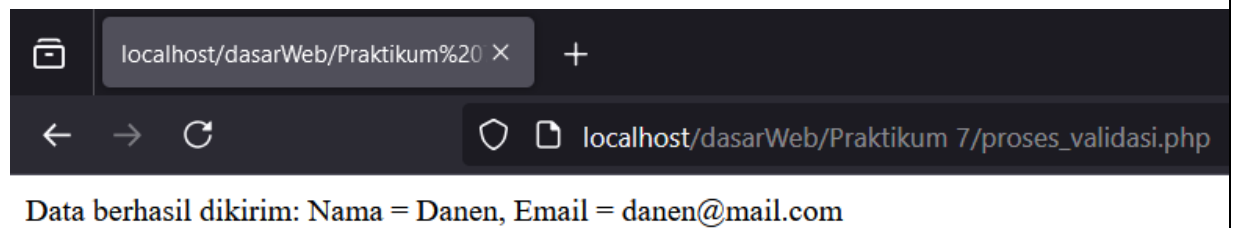
The screenshot shows a web browser window with a single tab titled 'localhost/dasarWeb/Praktikum%20'. The address bar displays 'localhost/dasarWeb/Praktikum 7/proses\_validasi.php'. The page content displays an error message: 'Email harus diisi.'

Ketika Email tidak sesuai:

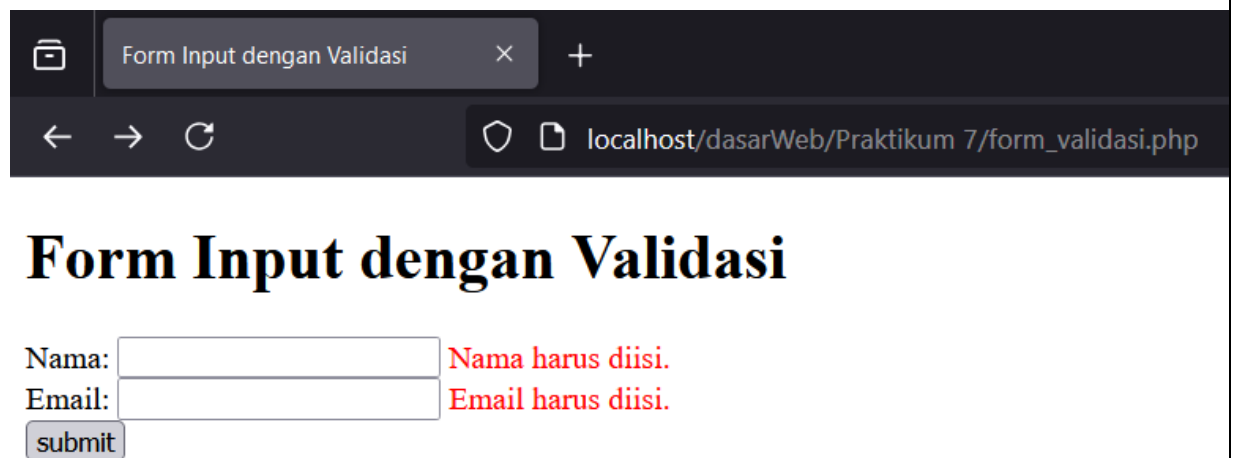
	 <p>Format email tidak valid.</p> <p>Membuat contoh form input sederhana yang dilengkapi dengan validasi menggunakan PHP.</p> <ul style="list-style-type: none"> <li>• <code>&lt;form&gt;</code> dengan <code>method="post"</code> dan <code>action="proses_validasi.php"</code> menunjukkan bahwa data form akan dikirim menggunakan metode POST ke file <code>proses_validasi.php</code>.</li> <li>• <code>if (\$_SERVER['REQUEST_METHOD'] == 'POST') { ... }</code> memeriksa apakah form telah disubmit dengan metode POST.</li> <li>• <code>\$nama = \$_POST['nama'];</code> dan <code>\$email = \$_POST['email'];</code> mengambil data dari input form.</li> <li>• <code>\$errors = array();</code> membuat array untuk menyimpan pesan kesalahan.</li> <li>• <code>if (empty(\$nama)) { \$errors[] = 'Nama harus diisi.'; }</code> memeriksa apakah nama kosong dan menambahkan pesan kesalahan jika iya.</li> <li>• <code>if (empty(\$email)) { \$errors[] = 'Email harus diisi.'; }</code> memeriksa apakah email kosong.</li> <li>• <code>elseif (!filter_var(\$email, FILTER_VALIDATE_EMAIL)) { \$errors[] = 'Format email tidak valid.'; }</code> menggunakan fungsi <code>filter_var</code> untuk memvalidasi format email.</li> <li>• <code>if (!empty(\$errors)) { ... }</code> memeriksa apakah ada kesalahan. Jika ada, setiap pesan kesalahan ditampilkan dengan <code>foreach</code>.</li> <li>• Jika tidak ada kesalahan, menampilkan pesan yang mengonfirmasi bahwa data berhasil dikirim, termasuk nama dan email.</li> </ul> <p>membuat form input dengan validasi dasar di PHP. Validasi dilakukan untuk memastikan bahwa nama dan email diisi dengan benar sebelum mengolah data lebih lanjut. Jika ada kesalahan, pesan kesalahan ditampilkan, sehingga dapat memperbaiki input untuk memastikan data yang diterima adalah valid dan dapat diproses.</p>
5	<p>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></p>

6	<pre> &lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt;   &lt;title&gt;Form Input dengan Validasi&lt;/title&gt;   &lt;script src="https://code.jquery.com/jquery-3.6.0.min.js"&gt;&lt;/script&gt; &lt;/head&gt; &lt;body&gt;   &lt;h1&gt;Form Input dengan Validasi&lt;/h1&gt;   &lt;form id="myForm" method="post" action="proses_validasi.php"&gt;     &lt;label for="nama"&gt;Nama:&lt;/label&gt;     &lt;input type="text" id="nama" name="nama"&gt;     &lt;span id="nama-error" style="color: red;"&gt;&lt;/span&gt;&lt;br&gt;      &lt;label for="email"&gt;Email:&lt;/label&gt;     &lt;input type="text" id="email" name="email"&gt;     &lt;span id="email-error" style="color: red;"&gt;&lt;/span&gt;&lt;br&gt;      &lt;input type="submit" value="Submit"&gt;   &lt;/form&gt;    &lt;script&gt;     \$(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         }       });     });   &lt;/script&gt; &lt;/body&gt; &lt;/html&gt; </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)</p> <p>Tampilan Awal:</p> 

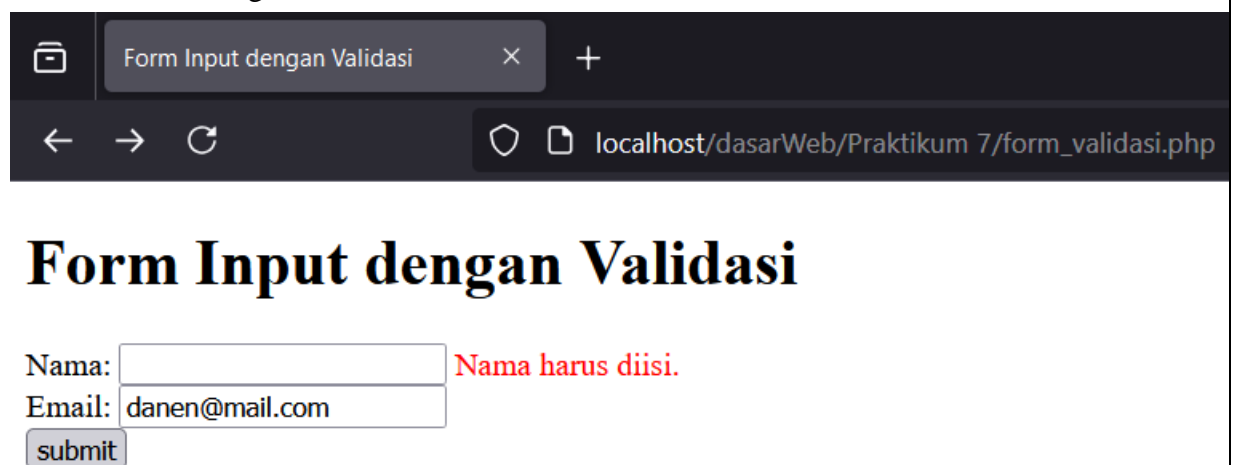
Ketika nama dan email di isi:



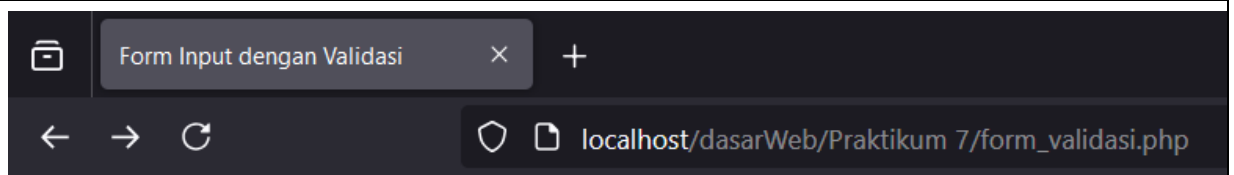
Ketika Form kosong:



Ketika nama kosong:



Ketika email tidak diisi:

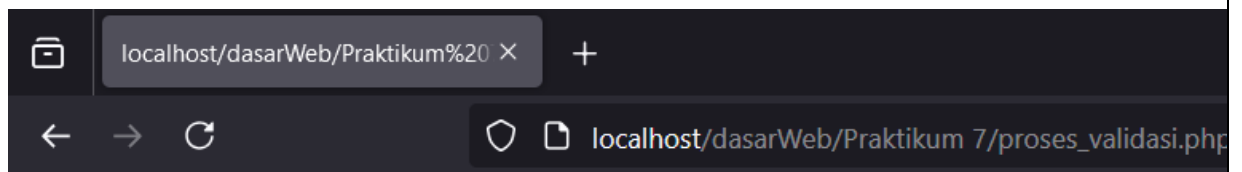


## Form Input dengan Validasi

Nama:

Email:  Email harus diisi.

Ketika format email salah:



Format email tidak valid.

### 1. Validasi Real-time

- Kode melakukan validasi pada saat mencoba untuk mengirimkan form, sebelum data dikirim ke server untuk memberikan umpan balik instan.

### 2. Menampilkan Pesan Kesalahan

- Setiap input (nama dan email) memiliki elemen `<span>` di sampingnya (`#nama-error` dan `#email-error`) yang digunakan untuk menampilkan pesan kesalahan. Pesan ini berwarna merah untuk menandakan bahwa input tersebut tidak valid.

### 3. Pemeriksaan Input

- Sebelum mengirim form, kode memeriksa apakah nama dan email kosong:
  - Jika nama kosong, pesan "Nama harus diisi." ditampilkan.
  - Jika email kosong, pesan "Email harus diisi." ditampilkan.
- Jika semua input valid, form akan disubmit; jika tidak, pengiriman form dicegah dengan `event.preventDefault()`.

### 4. Keterhubungan antara HTML dan JavaScript

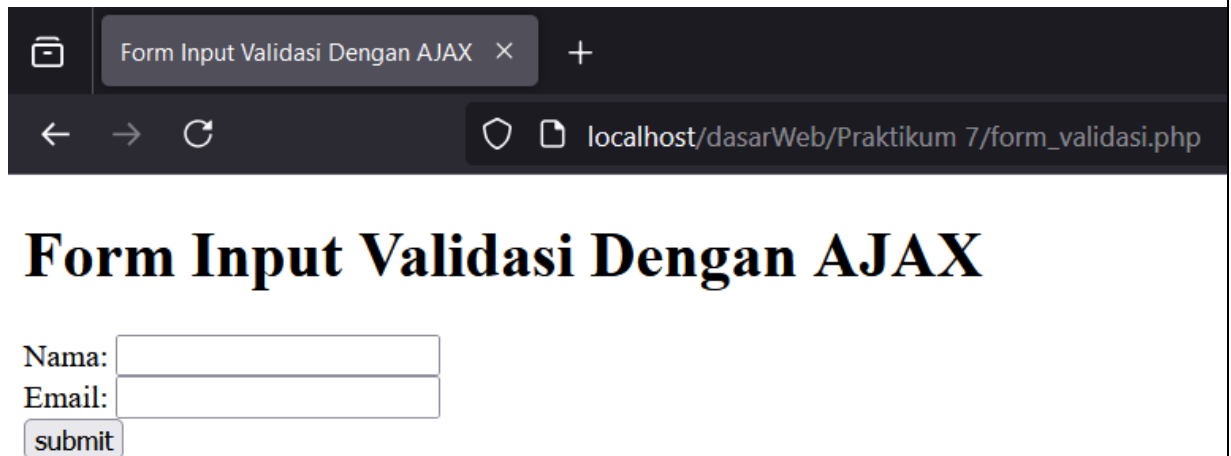
- Penggunaan jQuery untuk mengambil nilai input (`$("#nama").val()` dan `$("#email").val()`) menunjukkan interaksi antara HTML dan JavaScript, membuat kode validasi lebih dinamis.

Penambahan jQuery untuk validasi front-end membuat form lebih interaktif dan responsif. Ini memberikan umpan balik langsung kepada pengguna, memperbaiki pengalaman pengguna secara keseluruhan, dan mengurangi kesalahan pengiriman data ke server. Validasi ini juga bekerja sama dengan validasi sisi server yang ada, meningkatkan keamanan dan keandalan aplikasi.

9

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)

Tampilan Awal:



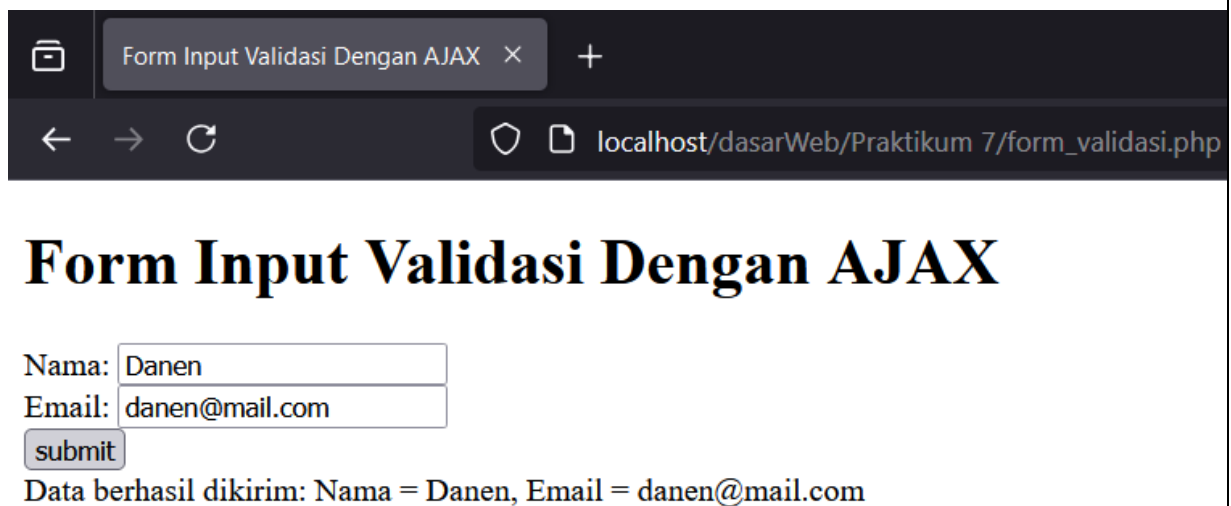
Form Input Validasi Dengan AJAX

Nama:

Email:

submit

Jika Form di input kemudian klik submit:



Form Input Validasi Dengan AJAX

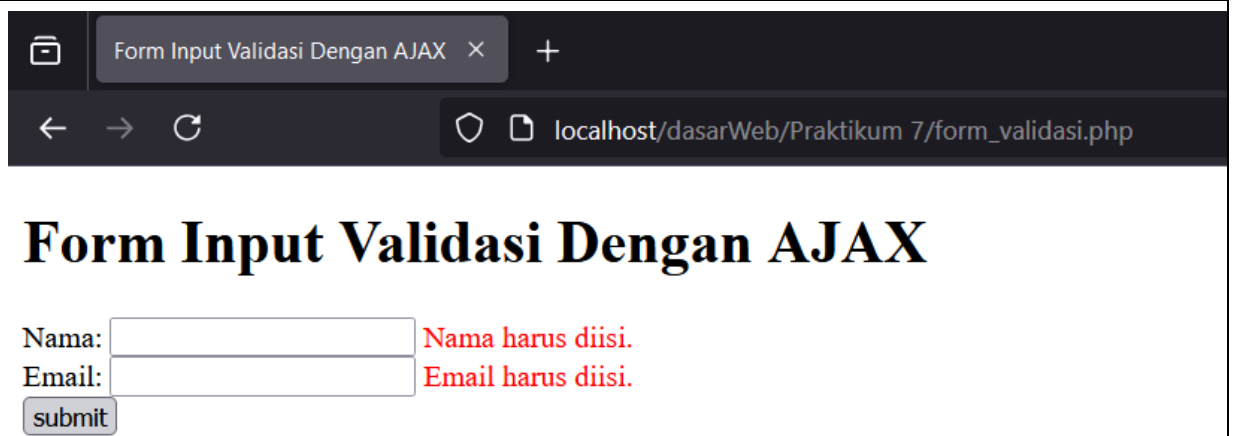
Nama:

Email:

submit

Data berhasil dikirim: Nama = Danen, Email = danen@mail.com

Jika Form Kosong kemudian klik submit:

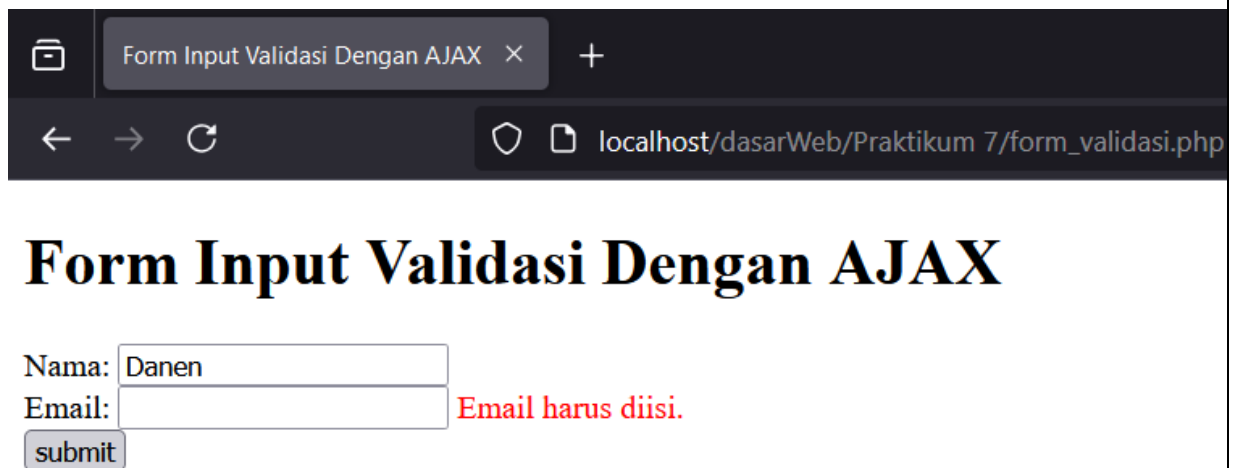


Form Input Validasi Dengan AJAX

Nama:  Nama harus diisi.

Email:  Email harus diisi.

Jika email kosong:

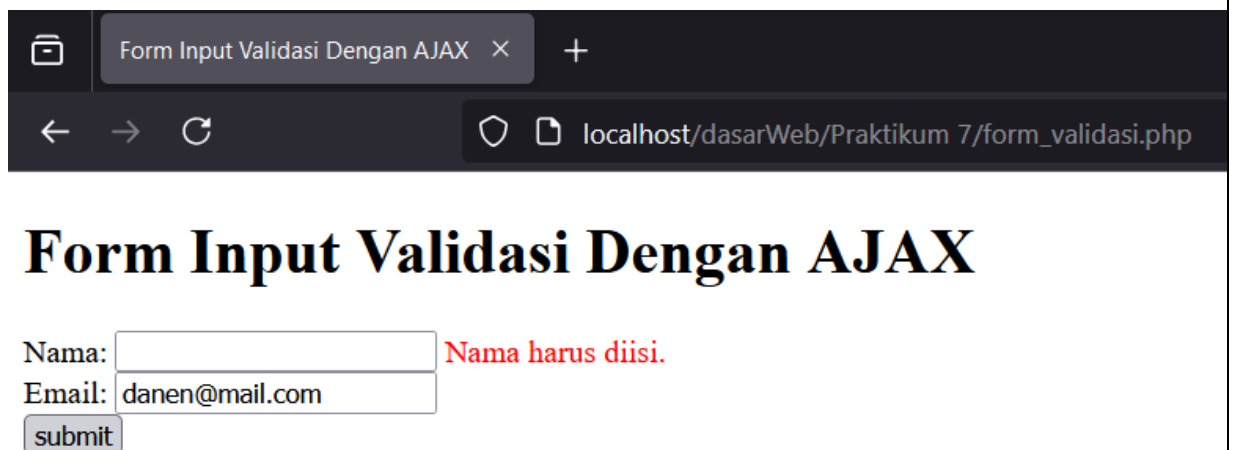


Form Input Validasi Dengan AJAX

Nama:

Email:  Email harus diisi.

Jika Nama Kosong:

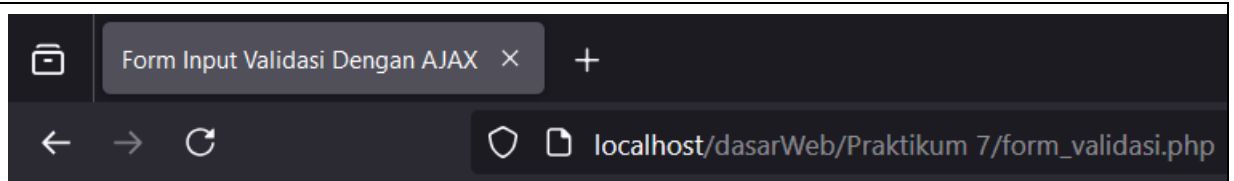


Form Input Validasi Dengan AJAX

Nama:  Nama harus diisi.

Email:

Jika email tidak sesuai format:



## Form Input Validasi Dengan AJAX

Nama:

Email:

Format email tidak valid.

Dengan AJAX:

**1. Pengiriman Data Tanpa Memuat Ulang Halaman:**

- Dengan menambahkan AJAX, data form sekarang dapat dikirim ke server tanpa memuat ulang halaman.

**2. Respons Langsung dari Server:**

- Setelah data berhasil dikirim, respons dari server (misalnya, pesan konfirmasi atau kesalahan) dapat langsung ditampilkan di elemen `<div id="result">`.

**3. Penggunaan `event.preventDefault()`:**

- Mencegah pengiriman form default sangat penting agar pengiriman dilakukan melalui AJAX. Ini memungkinkan kontrol penuh atas pengiriman data dan penanganan respons.

**4. Pengurangan Beban Server:**

- Dengan validasi dilakukan di sisi klien, hanya input yang valid yang dikirim ke server. Ini mengurangi jumlah permintaan yang tidak perlu, menghemat sumber daya server.

**5. Penanganan Kesalahan AJAX:**

- Menambahkan penanganan kesalahan pada permintaan AJAX memberikan umpan balik jika terjadi masalah saat mengirim data, seperti koneksi jaringan yang gagal.

**6. Keterbacaan dan Pemisahan Logika:**

- Kode jQuery menjadi lebih terorganisir dengan logika yang jelas untuk validasi dan pengiriman data. Ini memudahkan pengembangan dan pemeliharaan di masa depan.

### Kesimpulan

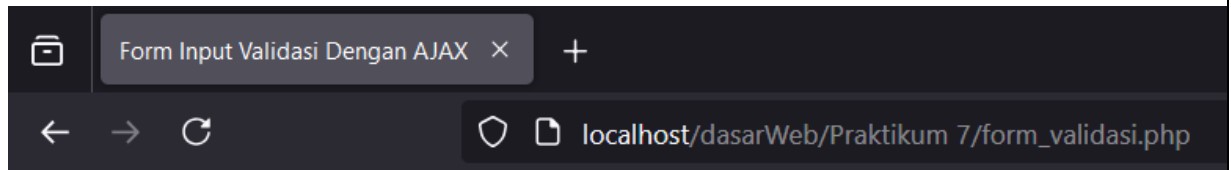
Dengan Penambahan AJAX ke dalam form input dengan validasi ini membawa manfaat signifikan dalam hal interaktivitas dan efisiensi. Sehingga dapat merasakan kecepatan dan responsivitas yang lebih baik, serta meminimalkan gangguan saat mengisi form.



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)

Tampilan Awal:



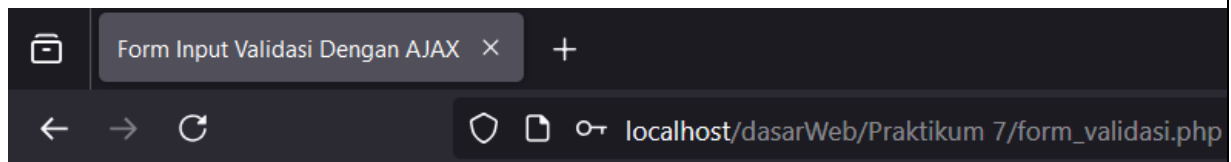
## Form Input Validasi Dengan AJAX

Nama:

Email:

Password:

Ketika submit Berhasil:



## Form Input Validasi Dengan AJAX

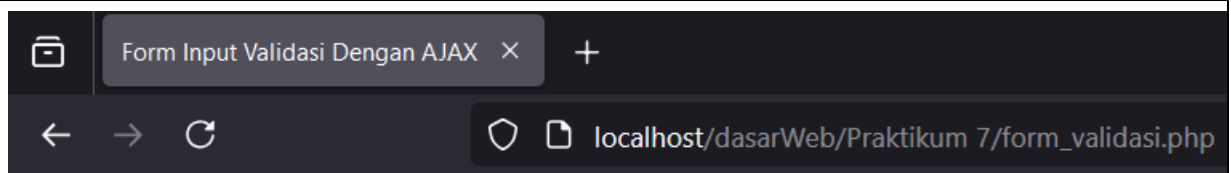
Nama:

Email:

Password:

Data berhasil dikirim: Nama = Danen, Email = danen@mail.com

Ketika Password kosong:



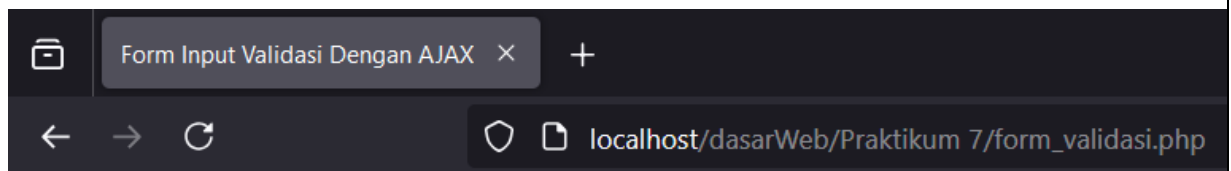
## Form Input Validasi Dengan AJAX

Nama:

Email:

Password:  Password harus diisi.

Ketika Password Kurang dari 8 karakter:



## Form Input Validasi Dengan AJAX

Nama:

Email:

Password:  Password harus minimal 8 karakter.

### 1. Penambahan Input untuk Password:

- Kode baru menambahkan input untuk password dengan label dan elemen span untuk menampilkan pesan kesalahan jika diperlukan.

### 2. Validasi Password di Sisi Klien:

- Validasi password dilakukan dengan memeriksa panjangnya. Jika kurang dari 8 karakter, pesan kesalahan ditampilkan. Ini memastikan bahwa pengguna memahami persyaratan sebelum mengirimkan form.

### 3. Menggunakan AJAX untuk Mengirimkan Data:

- Dengan menambahkan password ke dalam data yang dikirim ke server melalui AJAX, aplikasi sekarang menangani input yang lebih lengkap. Ini memungkinkan backend untuk melakukan validasi lebih lanjut.

### 4. Peningkatan Keamanan:

- Meskipun validasi sisi klien ditambahkan, penting untuk tetap melakukan validasi dan sanitasi di sisi server (dalam file proses\_validasi.php). Ini menghindari potensi risiko keamanan, seperti pengiriman data berbahaya.

	<b>Kesimpulan</b>
--	-------------------

	Penambahan validasi untuk password meningkatkan keamanan dan keandalan form input. Validasi ini, bersama dengan validasi sisi server, memberikan lapisan perlindungan tambahan terhadap data yang tidak valid atau berbahaya.
--	---