



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

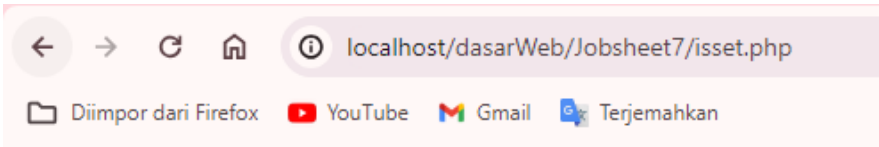
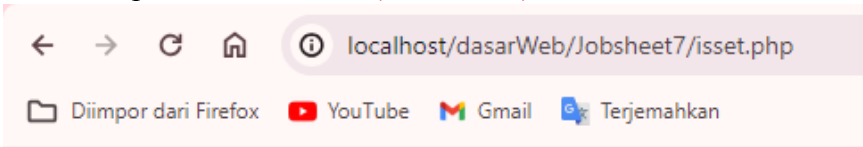
`isset()` adalah sebuah fungsi dalam PHP yang digunakan untuk memeriksa apakah suatu variabel telah didefinisikan (ada) atau tidak. Fungsi ini mengembalikan `true` jika variabel telah didefinisikan dan memiliki nilai, dan `false` jika variabel tidak ada atau bernilai `null`. Berikut adalah contoh script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

Langka h	Keterangan
1	Buatlah satu file baru di dalam direktori dasarWeb, beri nama <code>isset.php</code> .

2	Ketikkan ke dalam file <code>isset.php</code> tersebut kode di bawah ini.
3	<pre> \$umur; if (isset(\$umur) && \$umur >= 18) { echo "Anda sudah dewasa."; } else { echo "Anda belum dewasa atau variabel 'umur' tidak ditemukan."; } </pre>
4	<p>Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/dasarWeb/isset.php</code>. Apa yang anda pahami dari penggunaan <code>isset</code> pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 1.1)</p>  <p>Anda belum dewasa atau variabel umur tidak ditemukan</p> <p>Penggunaan <code>isset</code> pada file tersebut adalah untuk memeriksa apakah variabel tersebut sudah didefinisikan atau belum sehingga pada kode tersebut akan menjalankan perintah <code>echo</code> pada bagian <code>else</code> karena variabel belum didefinisikan.</p>
5	<p>Tambahkan isi dari file <code>isset.php</code> 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 <code>localhost/dasarWeb/isset.php</code> buat hasil tidak menjadi 1 baris, jadi tampilan dari <code>echo</code> harus terpisah. Apa yang anda pahami dari penggunaan <code>isset</code> pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 1.2)</p>  <p>Anda belum dewasa atau variabel umur tidak ditemukan</p> <p>Nama: Jane</p> <p>Penggunaan <code>isset</code> pada kode program tersebut adalah untuk mengecek pada variabel data apakah pada bagian “nama” sudah dideklarasikan atau belum. Jika sudah, maka akan menampilkan nama tersebut.</p>

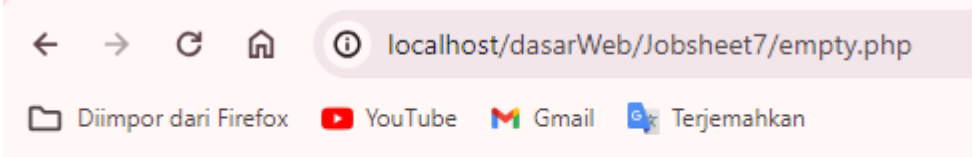
Function empty

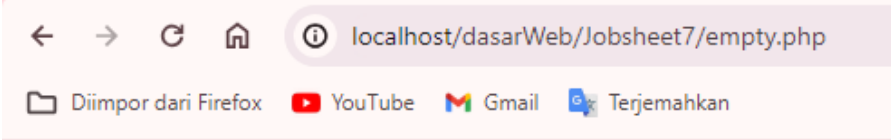
Fungsi `empty()` pada PHP digunakan untuk memeriksa apakah suatu variabel kosong atau tidak terdefinisi. Fungsi ini mengembalikan `true` jika variabel kosong atau tidak terdefinisi, dan `false` jika variabel memiliki nilai atau telah didefinisikan. Berikut adalah penjelasan dan contoh penggunaan `empty()`:

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

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

Praktikum 2. Function empty

Langkah	Keterangan
1	Buat satu file baru bernama <code>empty.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>empty.php</code>
2	<pre><code>\$myArray = array(); // Array kosong if (empty(\$myArray)) { echo "Array tidak terdefinisi atau kosong."; } else { echo "Array terdefinisi dan tidak kosong."; }</code></pre>
3	<p>Simpan file tersebut, kemudian buka browser dan jalankan <code>localhost/dasarWeb/empty.php</code>. Apa yang anda pahami dari penggunaan <code>empty</code> pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.1)</p>  <p>Array tidak terdefinisi atau kosong</p> <p>Penggunaan <code>empty</code> pada kode tersebut adalah untuk mengecek apakah array dari variabel <code>myArray</code> kosong atau tidak. Karena array tersebut kosong, maka akan menjalankan perintah <code>echo</code> yang pertama atau pada bagian <code>if</code>.</p>
4	Tambahkan isi dari file <code>empty.php</code> tersebut dengan kode di bawah ini.

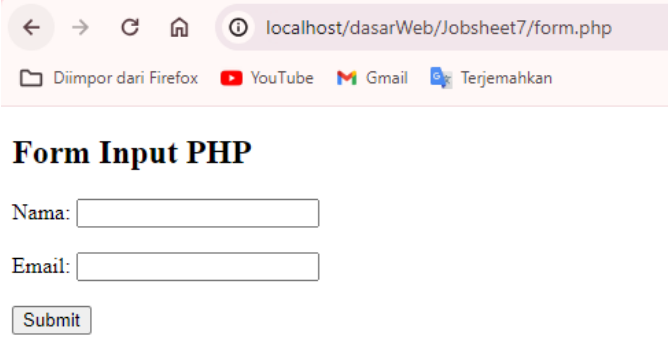
	<pre> if (empty(\$nonExistentVar)) { echo "Variabel tidak terdefinisi atau kosong."; } else { echo "Variabel terdefinisi dan tidak kosong."; } </pre>
5	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/empty.php buat hasil tidak menjadi 1 baris, jadi tampilan dari echo harus terpisah. Apa yang anda pahami dari penggunaan empty pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.2)</p>  <p>Array tidak terdefinisi atau kosong</p> <p>Variabel tidak terdefinisi atau kosong</p> <p>Dari penambahan kode program tersebut, akan muncul tulisan variabel tidak terdefinisi untuk variabel nonExistentVar yang sebelumnya dicek melalui fungsi empty.</p>

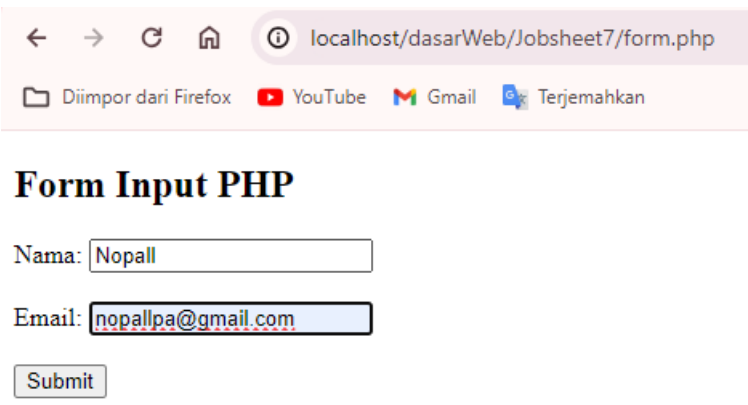
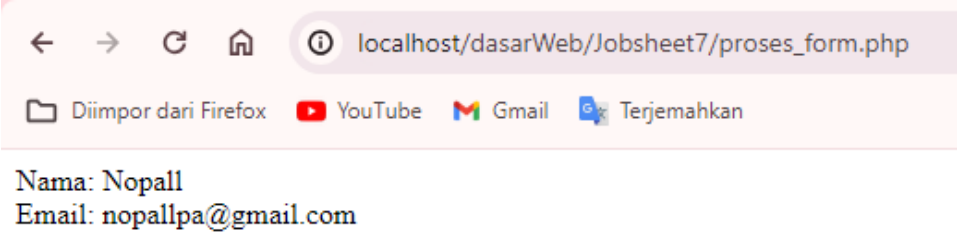
Praktikum Bagian 3 : Form Input PHP

Langka h	Keterangan
1	Buat satu file baru bernama form.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form.php

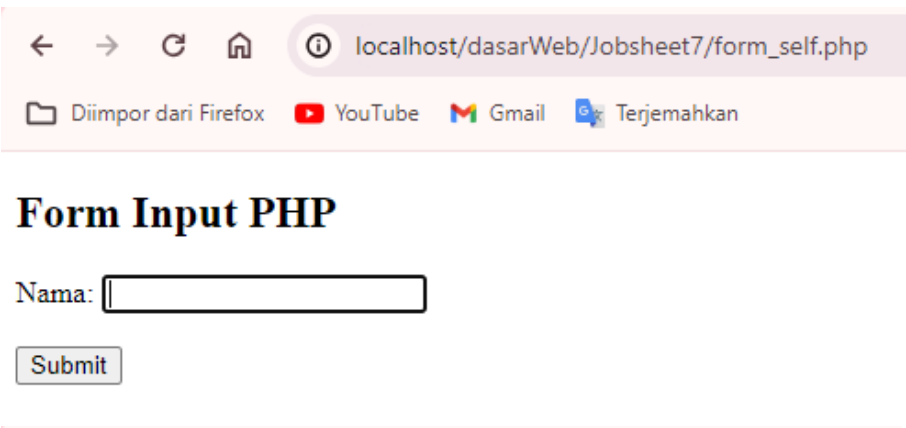
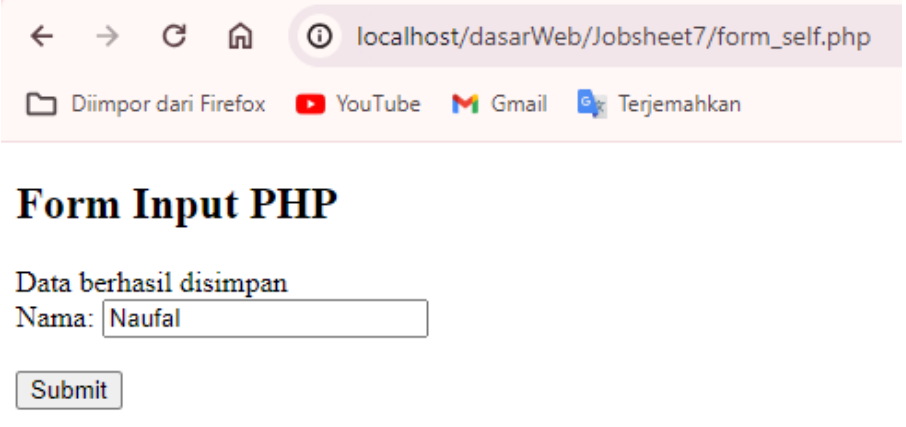
2	<pre> <!DOCTYPE html> <html> <head> <title>Form Input PHP</title> </head> <body> <h2>Form Input PHP</h2> <form method="post" action="proses_form.php"> <label for="nama">Nama:</label> <input type="text" name="nama" id="nama" required>

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

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

	 <p>Form Input PHP</p> <p>Nama: <input type="text" value="Nopall"/></p> <p>Email: <input type="text" value="nopallpa@gmail.com"/></p> <p><input type="submit" value="Submit"/></p>  <p>Nama: Nopall Email: nopallpa@gmail.com</p> <p>Awalnya, akan tampil sebuah form html. Kemudian, kita bisa mengisi nama dan email di kolom tersebut. Setelah klik Submit, maka inputan tersebut akan tampil di halaman proses_form.</p>
5	<p>Buat satu file baru bernama form_self.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form_self.php</p> <pre> <!DOCTYPE html> <html> <head> <title>Form Input PHP</title> </head> <body> <h2>Form Input PHP</h2> <?php // Inisialisasi variabel \$namaErr = ""; \$nama = ""; // Cek apakah form sudah disubmit if (\$_SERVER["REQUEST_METHOD"] == "POST") { // Validasi nama (contoh: pastikan nama tidak kosong) if (empty(\$_POST["nama"])) { \$namaErr = "Nama harus diisi!"; } else { \$nama = \$_POST["nama"]; echo "Data berhasil disimpan!"; } } ?> <form method="post" action="<?php echo htmlspecialchars(\$_SERVER["PHP_SELF"]); ?>"> <label for="nama">Nama:</label> <input type="text" name="nama" id="nama" value="<?php echo \$nama; ?>"> <?php echo \$namaErr; ?>

 <input type="submit" name="submit" value="Submit"> </form> </body> </html> </pre>
6	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/form_self.php. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 3.2)</p>

 
<p>Untuk kode program hampir sama seperti sebelumnya, namun yang menjadi pembeda adalah pada bagian ini hanya menggunakan 1 file saja. Ketika user mengetikkan nama dan klik Submit, data tersebut akan disimpan. Tetapi, jika dalam keadaan kosong atau tidak mengisi nama, maka akan muncul tulisan “Nama harus diisi”.</p>

HTML Injection

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

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

Contoh HTML Injection:

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

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

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

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

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

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

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

Cara Mengangani HTML Injection

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

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

Berikut contoh sederhana dalam PHP untuk mengatasi HTML injection:

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

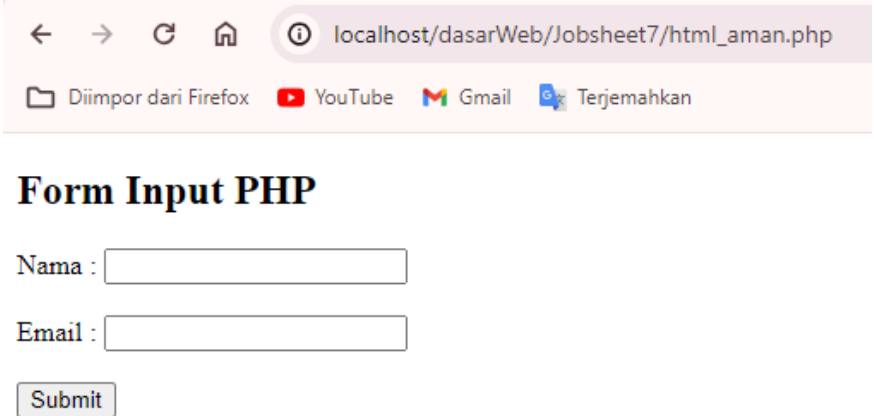
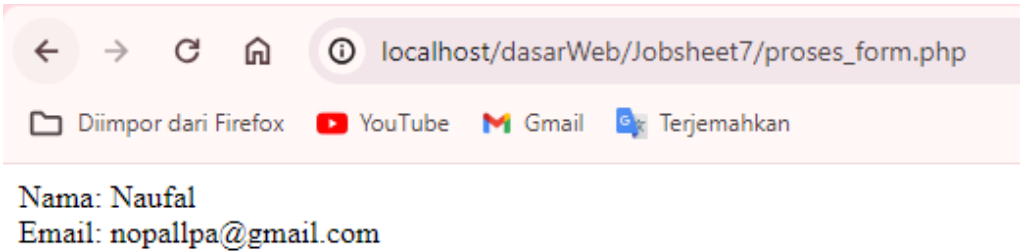
// Menggunakan htmlspecialchars untuk menghindari HTML injection saat menampilkan data
echo '<div>' . htmlspecialchars($user_input, ENT_QUOTES, 'UTF-8') . '</div>';
```

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

Praktikum 4 : HTML Injection

Langkah-langkah Praktikum :

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

3	Tambahkan script lainya supaya langkah 2 bisa berjalan normal. Lalu simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/html_aman.php
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1)</p> <p>Warning: Undefined array key "input" in F:\File xampp\htdocs\dasarWeb\Jobsheet 7\html_aman.php on line 2</p> <p>Muncul error pada hasil yang ditampilkan karena kode diatas mencoba untuk mengakses \$_POST['input'] sebelum memeriksa apakah data post dengan nama "input" telah dikirimkan oleh formular maka dari itu jika halaman pertama kali dimuat sebelum pengiriman formulir ini akan menghasilkan pesan error.</p>
5	Ketikkan kode tambahan pada langkah 6 di dalam html_aman.php
6	<pre>// Periksa apakah input adalah email yang valid \$email = \$_POST['email']; if (filter_var(\$email, FILTER_VALIDATE_EMAIL)) { // Lanjutkan dengan pengolahan email yang aman } else { // Tangani input yang tidak valid }</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>User harus memasukkan nama dan email yang valid. Jika untuk inputan email tidak valid, maka tidak akan bisa disubmit.</p>

Regular Expression (Regex)

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

Basic Regex Patterns:

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

Contoh Penggunaan Regex:

1. Mencocokkan pola

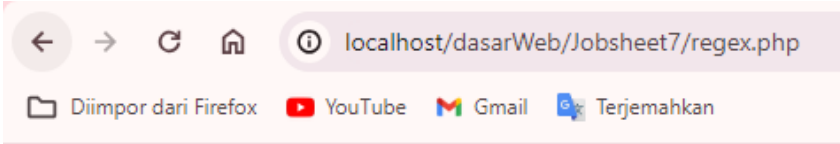
```
$pattern = '/hello/';
$text = 'Hello, world!';
if (preg_match($pattern, $text)) {
    echo "Cocokkan ditemukan!";
} else {
    echo "Tidak ada cocokan!";
}
```

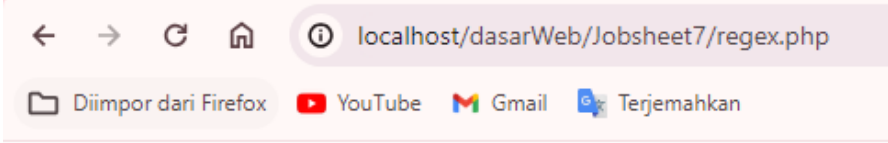
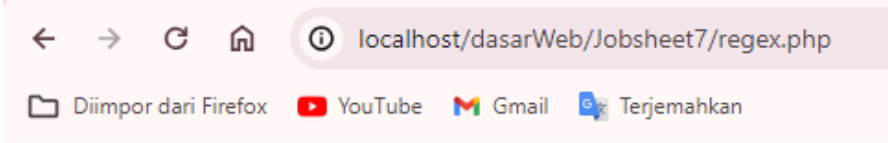
2. Mencocokkan karakter set

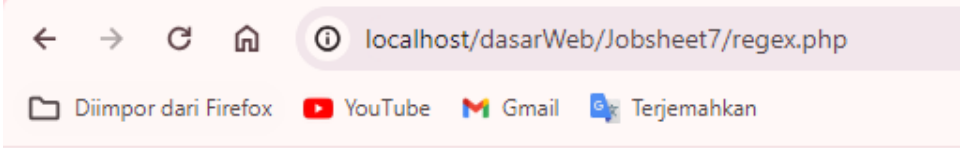
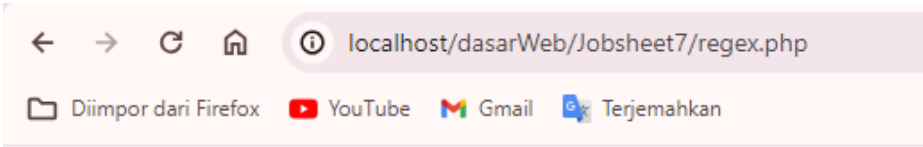
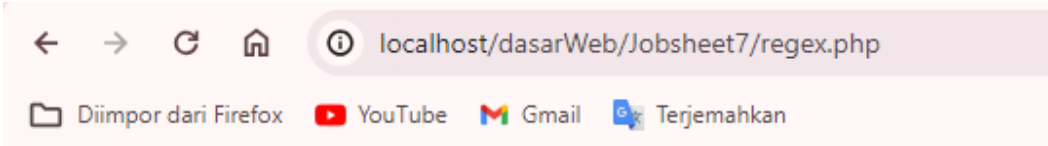
```
$pattern = '/[0-9]/'; // Cocokkan satu digit.
$text = 'There are 3 apples.';
if (preg_match($pattern, $text)) {
    echo "Digit ditemukan!";
} else {
    echo "Tidak ada digit!";
}
```

Praktikum 5 : Penggunaan Regex Pada PHP

Langkah-langkah Praktikum :

Langkah	Keterangan
1	Buat satu file baru bernama <code>regex.php</code> di dalam folder <code>dasarWeb</code> . Ketikkan kode pada langkah 2 di dalam <code>regex.php</code>
2	<pre><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	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.1)</p>  <p>Huruf kecil ditemukan!</p> <p>Hasil yang ditampilkan didapat dari pengecekan isi dari variabel text apakah terdapat huruf kecil atau tidak. Jika iya, maka akan terdeteksi ditemukan. Jika tidak, maka ditampilkan tulisan tidak ada huruf kecil.</p>
5	Ketikkan kode tambahan pada langkah 6 di dalam <code>regex.php</code>

6	<pre> \$pattern = '/[0-9]+/'; // Cocokkan satu atau lebih digit. \$text = 'There are 123 apples.'; if (preg_match(\$pattern, \$text, \$matches)) { echo "Cocokkan: " . \$matches[0]; } else { echo "Tidak ada yang cocok!"; } </pre>
7	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.2)</p>  <p>Huruf kecil ditemukan!Cocokkan: 123</p> <p>Hasil pencocokkan tersebut adalah 123 karena yang dicocokkan adalah angka antara 0 hingga 9 sehingga jika angka tersebut ada di dalam variabel text, maka akan terdeteksi 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	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.3)</p>  <p>Huruf kecil ditemukan!Cocokkan: 123I like banana pie.</p> <p>Hasil yang ditampilkan adalah sebuah kalimat yang awalnya bertuliskan “I like apple pie.” Kemudian berubah menjadi “I like banana pie.”. Hal tersebut terjadi karena adanya fungsi preg_replace yang mengganti nilai dari pattern yaitu apple menjadi nilai dari variabel replacement, yaitu banana sehingga kalimat atau text tersebut berubah menjadi “I like banana pie”.</p>
13	Ketikkan kode tambahan pada langkah 14 di dalam regex.php

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!Cocokkan: 123I like banana pie.Cocokkan: good</p> <p>Hasil yang muncul adalah hasil dari pencocokkan kata yang diawali dengan huruf g, o, dan diakhiri dengan d sehingga hasil yang ditampilkan sesuai, yaitu good.</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!Cocokkan: 123I like banana pie.Cocokkan: god</p> <p>Ketika variabel pattern yang awalnya menggunakan tanda * kemudian diubah menggunakan tanda ?, maka hasil yang ditampilkan akan sama, yaitu god.</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!Cocokkan: 123I like banana pie.Tidak ada yang cocok!</p> <p>Ketika variabel pattern yang awalnya menggunakan tanda * kemudian diubah menggunakan tanda ?, maka hasil yang ditampilkan akan sama, yaitu tidak ada yang cocok</p>

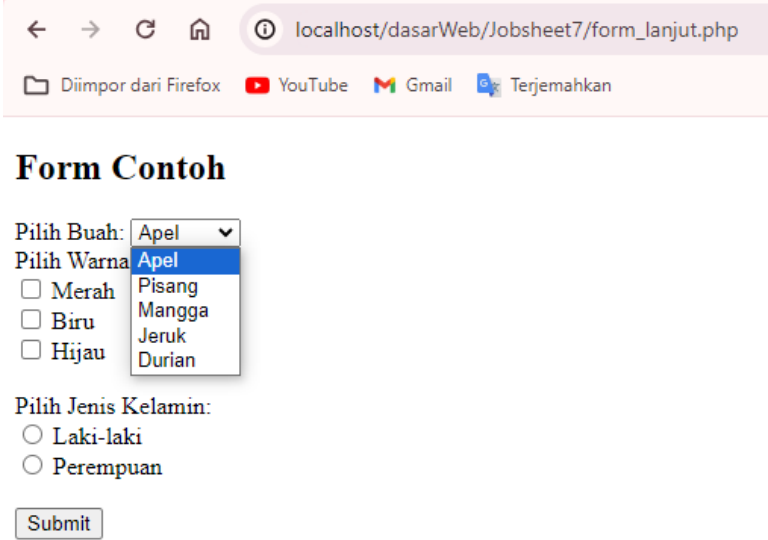
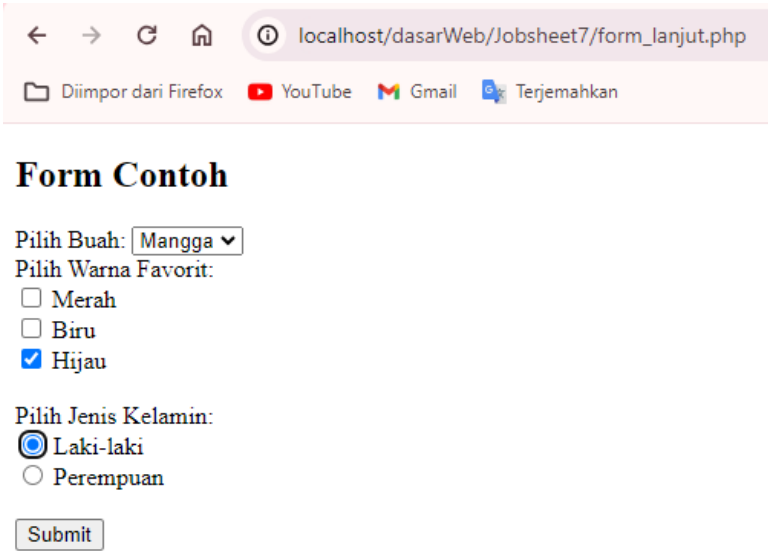
Praktikum 6 : Form Lanjut

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	Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_lanjut.php
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 6.1)</p>   <p>Form Contoh</p> <p>Pilih Buah: Apel ▼ Pilih Warna: Apel ▼ <input type="checkbox"/> Merah <input type="checkbox"/> Biru <input type="checkbox"/> Hijau Pilih Jenis Kelamin: <input type="radio"/> Laki-laki <input type="radio"/> Perempuan <input type="button" value="Submit"/></p> <p>Form Contoh</p> <p>Pilih Buah: Mangga ▼ Pilih Warna Favorit: <input type="checkbox"/> Merah <input type="checkbox"/> Biru <input checked="" type="checkbox"/> Hijau Pilih Jenis Kelamin: <input checked="" type="radio"/> Laki-laki <input type="radio"/> Perempuan <input type="button" value="Submit"/></p> <p>Anda memilih buah: mangga Anda tidak memilih warna favorit. Jenis kelamin Anda: laki-laki</p> <p>Halaman awal yang ditampilkan adalah berupa form. Lalu, user bisa mengisi form tersebut dan mengklik Submit. Setelah itu, akan berlanjut ke halaman proses_lanjut untuk menampilkan inputan atau pilihan dari user.</p>
5	Buat satu file baru bernama form_ajax.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form_ajax.php

```

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

    <br>

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

    <br>

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

    <br>

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

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

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

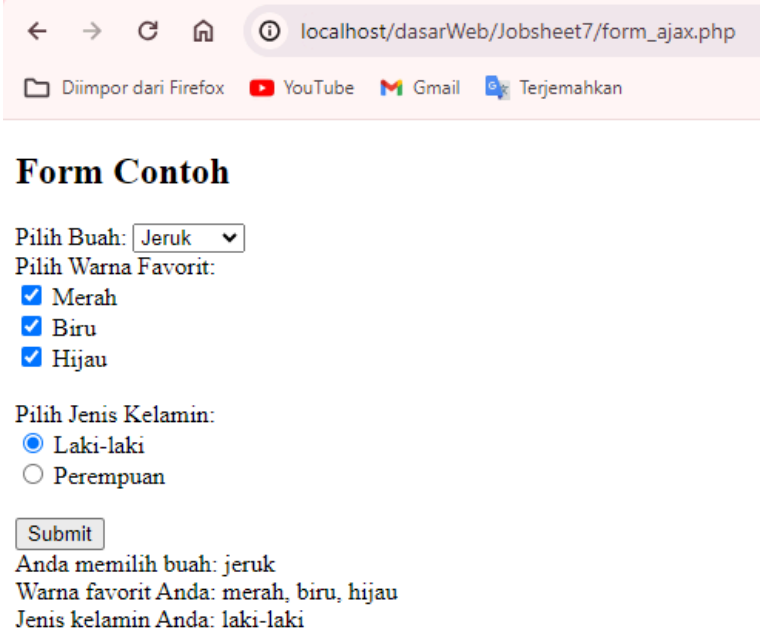
        // Mengumpulkan data form
        var formData = $("#myForm").serialize();

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

```

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)

	 <p>Setelah user mengisi form tersebut, maka akan ditampilkan hasilnya di bawah tombol submit dalam bentuk div yang sebelumnya diteruskan ke file proses_lanjut.php terlebih dahulu.</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>

```

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

    // Validasi Nama
    if (empty($nama)) {
        $errors[] = "Nama harus diisi.";
    }

    // Validasi Email
    if (empty($email)) {
        $errors[] = "Email harus diisi.";
    } elseif (filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $errors[] = "Format email tidak valid.";
    }

    // Jika ada kesalahan validasi
    if (empty($errors)) {
        foreach ($errors as $error) {
            echo $error . "<br>";
        }
    } else {
        // Lanjutkan dengan pemrosesan data jika semua validasi berhasil
        // Misalnya, menyimpan data ke database atau mengirim email
        echo "Data berhasil dikirim: Nama = $nama, Email = $email";
    }
}
?>

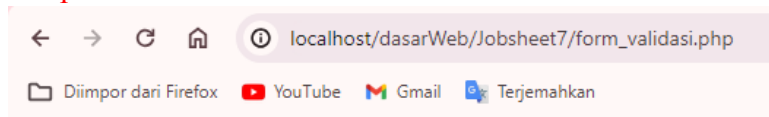
```

3

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

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

Tampilan :

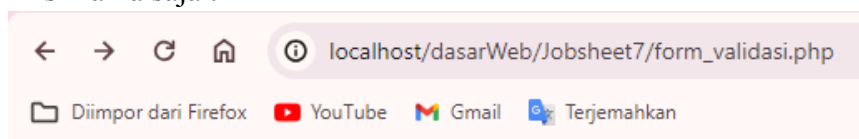


Form Input dengan Validasi

Nama:

Email:

Diisi nama saja :



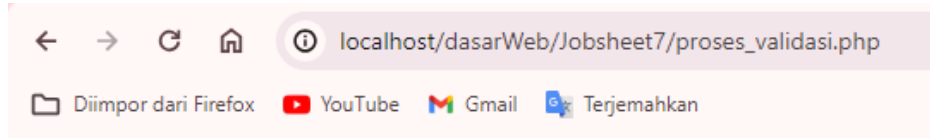
Form Input dengan Validasi

Nama:

Email:

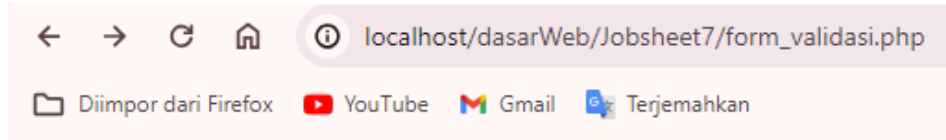
4

Hasil :



Email harus diisi.

Diisi nama saja :

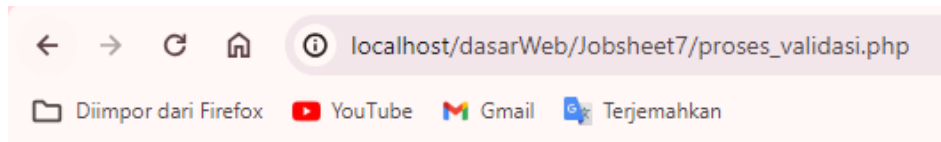


Form Input dengan Validasi

Nama:

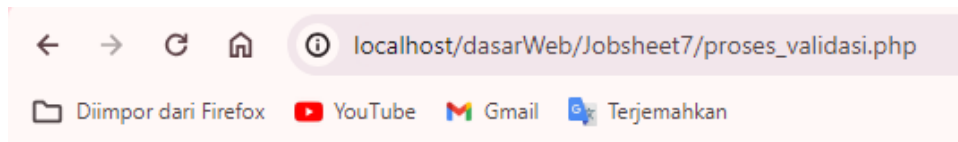
Email:

Hasil :



Nama harus diisi.

Diisi semua :

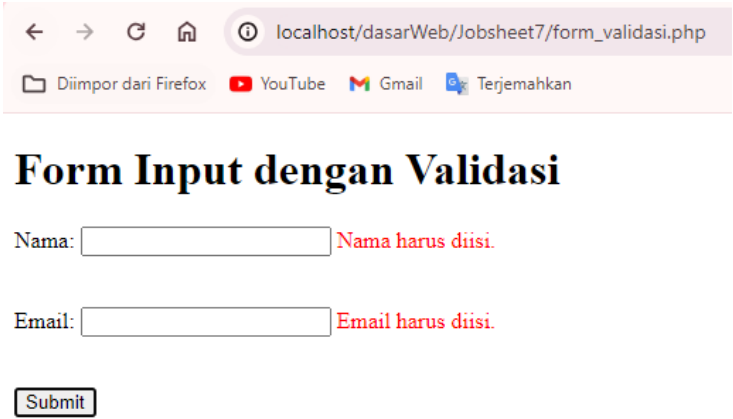


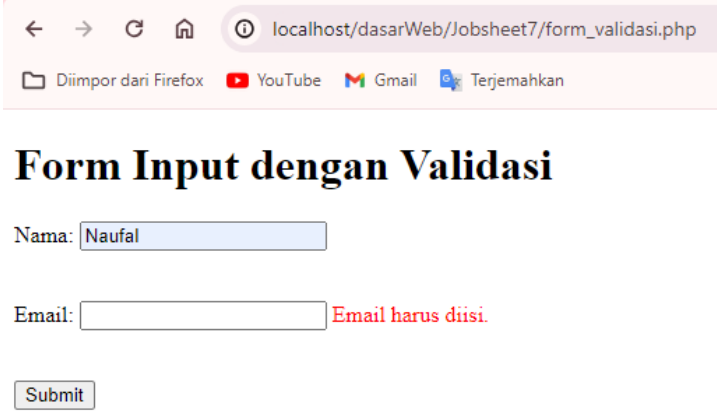
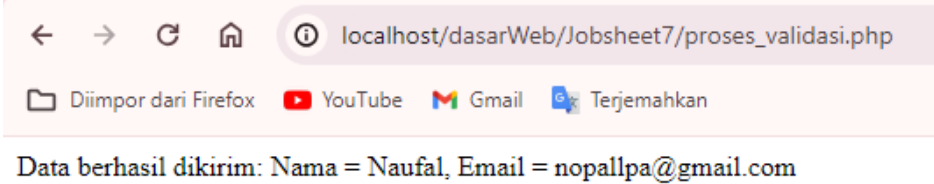

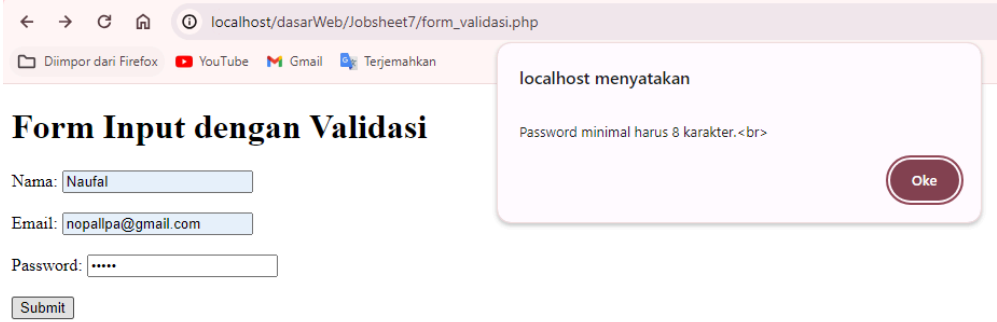
Data berhasil dikirim: Nama = Naufal, Email = nopallpa@gmail.com

Hasil yang ditampilkan setelah melakukan klik Submit berdasarkan inputan yang ada ataupun tidak. Nantinya akan muncul hasil terkait bagian yang kosong atau harus diisi dan email yang tidak valid.

5

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

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) Jika tidak diisi :</p> 

	<p>Jika diisi nama saja :</p>  <p>Jika diisi lengkap :</p> 
9	<p>Soal 7.3: Buatlah script langkah 6 dengan menggunakan ajax. Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.3)</p> 
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)</p> 

The screenshot shows a web browser window with the address bar displaying 'localhost/dasarWeb/Jobsheet7/form_validasi.php'. Below the address bar, there are navigation icons and a search bar. The main content area features a form titled 'Form Input dengan Validasi'. The form contains three input fields: 'Nama:' with the value 'Naufal', 'Email:' with the value 'nopallpa@gmail.com', and 'Password:' with the value '.....'. A 'Submit' button is located below the password field. To the right of the form, a light purple notification box displays the message 'localhost menyatakan' followed by 'Data berhasil dikirim: Nama = Naufal, Email = nopallpa@gmail.com Password = Naufalputra'. An 'Oke' button is positioned at the bottom right of the notification box.

Setelah dilakukan penambahan form, maka akan dilakukan pengecekan apakah Panjang karakter tersebut kurang dari 8 atau tidak. Jika iya, maka akan muncul tulisan seperti di atas, sedangkan jika lebih dari 8 karakter akan menampilkan hasilnya seperti di atas.