|  |  |
| --- | --- |
|  | Jurusan Teknologi Informasi Politeknik Negeri Malang  **Jobsheet-7: PHP - Form Proccessing**  **Mata Kuliah Desain dan Pemrograman Web** Pengampu: Tim Ajar Desain dan Pemrograman Web *Oktober 2024* |

**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

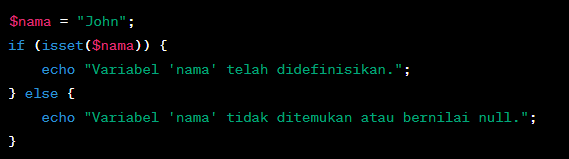
**Perhatian**

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()**:



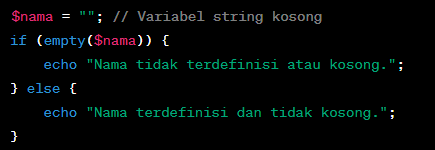
Kode di atas memeriksa apakah variabel **$nama** telah didefinisikan. Jika ya, maka pesan "Variabel 'nama' telah didefinisikan." akan ditampilkan, jika tidak, maka pesan "Variabel 'nama' tidak ditemukan atau bernilai null." akan ditampilkan.

# Praktikum 1. Function Isset

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buatlah satu file baru di dalam direktori dasarWeb, beri nama isset.php. |
| 2 | Ketikkan ke dalam file isset.php tersebut kode di bawah ini. |

|  |  |
| --- | --- |
| 3 |  |
| 4 | 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)    Jika variabel $umur tidak ada, tampilkan pesan bahwa variabel tersebut tidak ditemukan. |
| 5 | Tambahkan isi dari file isset.php tersebut dengan kode di bawah ini. |
| 6 | 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)     * Variabel $umur dideklarasikan tanpa nilai, sehingga nilainya adalah null. * isset($umur) digunakan untuk memeriksa apakah $umur di-set. * Jika $umur tidak di-set, program mencetak **"Variabel 'umur' tidak ditemukan."** * echo "<br>"; menambahkan baris baru di output. * Array $data berisi dua kunci: "nama" dengan nilai "Jane" dan "usia" dengan nilai 25. * isset($data["nama"]) memeriksa apakah kunci "nama" ada dalam array. * Jika ada, program mencetak **"Nama: Jane."** |

**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()**:



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

# Praktikum 2. Function empty

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama empty.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam empty.php |
| 2 |  |
| 3 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/ empty.php. Apa yang anda pahami dari penggunaan empty pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.1)    Output dari kode di atas adalah **"Array tidak terdefinisi atau kosong."** karena $myArray adalah array kosong. |
| 4 | Tambahkan isi dari file empty.php tersebut dengan kode di bawah ini. |
| 5 | 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)    Array $myArray diperiksa menggunakan empty(). Karena array kosong, hasilnya adalah **"Array tidak terdefinisi atau kosong."** $nonExistentVar tidak dideklarasikan, sehingga empty() mengembalikan true, dan mencetak **"Variabel tidak terdefinisi atau kosong."** |

**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 |  |
| 3 | Buat satu file baru bernama proses\_form.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam proses\_form.php |
| 4 | 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)     * if ($\_SERVER["REQUEST\_METHOD"] == "POST"): Memastikan bahwa kode hanya dijalankan jika form dikirim dengan metode POST. * $nama = $\_POST["nama"];: Mengambil nilai input nama dari form. * $email = $\_POST["email"];: Mengambil nilai input email dari form. * echo "Nama: " . htmlspecialchars($nama) . "<br>";: Menampilkan nama yang dimasukkan, dengan perlindungan terhadap serangan XSS. * echo "Email: " . htmlspecialchars($email) . "<br>";: Menampilkan email yang dimasukkan. |
| 5 | Buat satu file baru bernama form\_self.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form\_self.php |

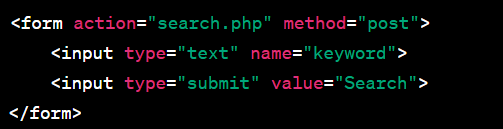
|  |  |
| --- | --- |
|  |  |
| 6 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/ form\_self.php. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 3.2)     * $namaErr untuk menyimpan pesan kesalahan jika nama tidak diisi. * $nama untuk menyimpan nilai nama yang diinputkan. * if ($\_SERVER["REQUEST\_METHOD"] == "POST"): Memeriksa apakah form disubmit. * if (empty($\_POST["nama"])): Memeriksa apakah field nama kosong.   + Jika kosong, menyimpan pesan kesalahan di $namaErr.   + Jika tidak kosong, menyimpan nama yang telah di-HTML encode ke dalam variabel $nama dan menampilkan pesan "Data berhasil disimpan!". * Menggunakan metode POST untuk mengirimkan data ke halaman yang sama. * Menampilkan pesan kesalahan di samping input nama jika ada. |

**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:



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



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



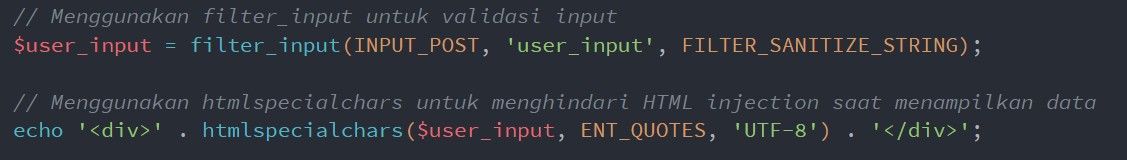
Kode JavaScript berbahaya tersebut akan dieksekusi oleh browser pengguna dan akan memunculkan kotak peringatan "You have been hacked!". Ini adalah contoh HTML injection yang sederhana. Sebagai pengembang web, penting untuk selalu melakukan sanitasi dan validasi input dari pengguna, serta menghindari 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:



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 html\_aman.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam html\_aman.php |
| 2 |  |
| 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 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1) |

|  |  |
| --- | --- |
|  |  |
| 5 | Ketikkan kode tambahan pada langkah 6 di dalam html\_aman.php |
| 6 |  |
| 7 | Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/html\_aman.php |
| 8 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.2)      Sebelum mengakses $\_POST['email'], tambahkan pengecekan dengan if (isset($\_POST['email'])). Ini memastikan bahwa hanya jika elemen tersebut ada di dalam array $\_POST, kode akan mencoba untuk menggunakannya. Jika $\_POST['email'] tidak diset, maka akan ditampilkan pesan kesalahan yang menyatakan bahwa email harus diisi. |

# Regular Expression (Regex)

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

# Basic Regex Patterns:

* **Karakter Tunggal**: Karakter apa pun akan cocok dengan dirinya sendiri dalam Regex, kecuali karakter khusus yang harus dihindari dengan karakter pelarian **\**.
  + Contoh: Pola **/a/** akan cocok dengan huruf "a" dalam teks.
* **Karakter Spesifik**: Anda dapat mencocokkan karakter tertentu dengan menyebutkannya dalam pola.
  + Contoh: Pola **/hello/** akan cocok dengan teks "hello" dalam teks.
* **Karakter Khusus**: Beberapa karakter khusus dalam Regex harus dihindari dengan karakter pelarian (**\**) jika Anda ingin mencocokkannya secara harfiah. Contoh karakter khusus: **.**, **\***, **+**, **?**,

**|**, **[**, **]**, **(**, **)**, **{**, **}**, **^**, **$**, **\**.

* **Karakter Set**: Anda dapat mencocokkan karakter dari satu set karakter dengan menggunakan **[**

**]**. Misalnya, **/[aeiou]/** akan mencocokkan salah satu huruf vokal.

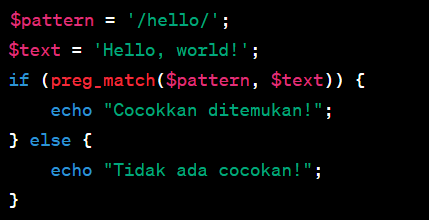
* **Rentang Karakter**: Anda dapat menggunakan **-** dalam karakter set untuk menentukan rentang karakter. Misalnya, **/[a-z]/** akan mencocokkan huruf kecil apa pun.
* **Kuantifier**: Anda dapat mengatur berapa kali karakter atau kelompok karakter sebelumnya harus muncul. Contoh kuantifier:
  + **\***: 0 atau lebih kali

o **+**: 1 atau lebih kali

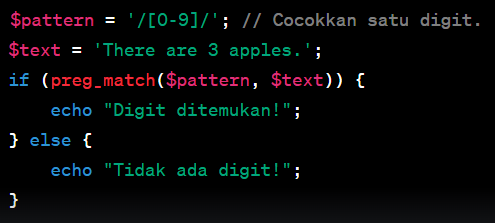
* + **?**: 0 atau 1 kali
  + **{n}**: Persis n kali
  + **{n,}**: Setidaknya n kali
  + **{n,m}**: Setidaknya n kali, maksimal m kali

# Contoh Penggunaan Regex:

1. Mencocokan pola



1. Mencocokan karakter set



# Praktikum 5 : Penggunaan Regex Pada PHP

Langkah-langkah Praktikum :

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama regex.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam regex.php |
| 2 |  |

|  |  |
| --- | --- |
| 3 | Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/ regex.php |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.1)    **Pemeriksaan Regex**: preg\_match($pattern, $text) digunakan untuk mencari apakah ada huruf kecil dalam $text.   * Jika huruf kecil ditemukan, program mencetak "Huruf kecil ditemukan!". * Jika tidak ada huruf kecil, program mencetak "Tidak ada huruf kecil!". |
| 5 | Ketikkan kode tambahan pada langkah 6 di dalam regex.php |
| 6 |  |
| 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)    Menggunakan $patternLowercase dan $textLowercase untuk pemeriksaan huruf kecil.  Menggunakan $patternDigit dan $textDigit untuk pemeriksaan digit. Ini meningkatkan keterbacaan dan menghindari kebingungan antara pola dan teks yang berbeda.  Menambahkan <br> di akhir setiap output untuk memastikan bahwa setiap hasil muncul di baris baru pada tampilan web.  Mengorganisir kode dalam blok yang terpisah untuk setiap pemeriksaan, sehingga lebih mudah untuk dibaca dan dikelola. |
| 9 | Ketikkan kode tambahan pada langkah 10 di dalam regex.php |
| 10 |  |
| 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)     * $patternReplace = '/apple/'; mendefinisikan pola untuk mencari kata "apple". * $replacement = 'banana'; menentukan kata pengganti yang akan digunakan. * $textReplace = 'I like apple pie.'; adalah string yang akan diperiksa dan diubah.preg\_replace($patternReplace, $replacement, $textReplace); digunakan untuk mengganti semua kemunculan "apple" dengan "banana" dalam teks yang ditentukan. * Hasilnya disimpan dalam variabel $new\_text.echo $new\_text; mencetak hasil penggantian ke layar, yang seharusnya menjadi "I like banana pie." |
| 13 | Ketikkan kode tambahan pada langkah 14 di dalam regex.php |

|  |  |
| --- | --- |
| 14 |  |
| 15 | Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php |
| 16 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.4)     **Pola Regex untuk Kata "god" dan "good"**:   * $patternGod = '/go\*d/'; mendefinisikan pola untuk mencocokkan "god", "good", "gooood", dan variasi lainnya. Di sini, o\* menunjukkan bahwa huruf 'o' dapat muncul 0 atau lebih kali.    **Teks yang Diperiksa**:   * $textGod = 'god is good.'; adalah string yang akan diperiksa untuk mencari kecocokan dengan pola yang ditentukan.    **Pemeriksaan Regex**:   * if (preg\_match($patternGod, $textGod, $matches)) digunakan untuk memeriksa apakah ada kecocokan dengan pola. * Jika ada kecocokan, program mencetak "Cocokkan: " diikuti dengan kata yang cocok (dari array $matches). * Jika tidak ada kecocokan, program mencetak "Tidak ada yang cocok!". |
| 17 | Soal 5.5: Buatlah script langkah 14 dengan variable pattern yang di ubah menggunakan `?` (0 atau 1 kali). Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.5)     * Pada bagian $patternGod, pola regex diubah menjadi '/go?d/'. Ini akan mencocokkan kata "god" (dengan 'o') dan "gd" (tanpa 'o').    **Output**:   * Dengan perubahan ini, program akan mencari "god" atau "gd" dalam teks yang diberikan. Jika ada yang cocok, akan menampilkan hasilnya. Jika tidak ada yang cocok, akan mencetak "Tidak ada yang cocok!". |
| 18 | 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)     1. **Pola Regex**:    * Pada bagian $patternGod, pola regex diubah menjadi '/go{0,1}d/'. Ini akan mencocokkan kata "god" (dengan satu huruf 'o') dan "gd" (tanpa huruf 'o'). 2. **Output**:    * Dengan perubahan ini, program akan mencari "god" atau "gd" dalam teks yang diberikan. Jika ada yang cocok, akan menampilkan hasilnya. Jika tidak ada yang cocok, akan mencetak "Tidak ada yang cocok!". |

# Praktikum 6 : Form Lanjut

Langkah-langkah Praktikum :

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama form\_lanjut.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form\_lanjut.php |

|  |  |
| --- | --- |
| 2 |  |
| 3 | Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/ form\_lanjut.php |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 6.1)     **Cek Metode Request**: Memastikan bahwa form disubmit dengan metode POST.   **Ambil Data**: Mengambil nilai dari input form seperti buah, warna, dan jenis kelamin.   **Inisialisasi Warna**: Jika tidak ada warna yang dipilih, inisialisasi sebagai array kosong.   **Tampilkan Hasil**: Menampilkan pilihan buah, warna favorit (jika ada), dan jenis kelamin yang dipilih pengguna. |
| 5 | Buat satu file baru bernama form\_ajax.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form\_ajax.php |

|  |  |
| --- | --- |
|  |  |
| 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) |

# 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 |  |
| 3 | Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/ form\_validasi.php |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.1)     **HTML Form**: Kode ini membuat sebuah form input yang meminta pengguna untuk memasukkan nama dan email. Form ini menggunakan metode POST untuk mengirimkan data ke halaman yang sama (proses\_validasi.php).   **Validasi Form**: Setelah form disubmit, kode PHP memeriksa apakah metode permintaan adalah POST. Jika iya, ia mengambil nilai dari input nama dan email.   **Array Kesalahan**: Kode ini membuat array $errors untuk menyimpan pesan kesalahan validasi.   **Validasi Nama dan Email**:   * **Nama**: Memeriksa apakah input nama kosong. Jika kosong, pesan kesalahan ditambahkan ke array. * **Email**: Memeriksa apakah input email kosong atau tidak valid. Jika kosong atau tidak valid, pesan kesalahan ditambahkan ke array.    **Menampilkan Kesalahan**: Jika ada kesalahan dalam validasi, pesan kesalahan akan ditampilkan. Jika tidak ada kesalahan, data berhasil dikirim dan ditampilkan.   **Keamanan**: Menggunakan htmlspecialchars() untuk menghindari serangan XSS dengan membersihkan input yang dikirimkan. |

|  |  |
| --- | --- |
| 5 | Kembangkan file bernama form\_validasi.php di dalam folder dasarWeb. Ketikkan kode pada langkah 6 di dalam form\_validasi.php |
| 6 |  |
| 7 | Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/ form\_validasi.php |
| 8 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.2)    Output yang ditampilkan berada di halaman yang berbeda |
| 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)    <!DOCTYPE html>  <html>  <head>      <title>Form Input dengan Validasi</title>      <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  </head>  <body>      <h1>Form Input dengan Validasi</h1>      <form id="myForm" method="post" action="proses\_validasi.php">          <label for="nama">Nama:</label>          <input type="text" id="nama" name="nama">          <span id="nama-error" style="color: red;"></span><br>            <label for="email">Email:</label>          <input type="text" id="email" name="email">          <span id="email-error" style="color: red;"></span><br>            <input type="submit" value="Submit">      </form>      <div id="hasil" style="margin-top: 20px; color: green;"></div>      <script>          $(document).ready(*function*() {              $("#myForm").submit(*function*(*event*) {                  event.preventDefault();    *var* nama = $("#nama").val();  *var* email = $("#email").val();  *var* valid = true;                  // Validasi Nama                  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) {                      $.ajax({                          url: "proses\_validasi.php",                          type: "POST",                          data: { nama: nama, email: email },                          success: *function*(*response*) {                              $("#hasil").html(response);                          },                          error: *function*() {                              $("#hasil").html("Terjadi kesalahan saat mengirim data.");                          }                      });                  }              });          });      </script>  </body>  </html> |
| 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)       1. **Form Input**:    * Form ini meminta pengguna untuk mengisi nama, email, dan password.    * Setiap input memiliki elemen <span> untuk menampilkan pesan kesalahan validasi di bawahnya. 2. **Validasi dengan jQuery**:    * Saat form disubmit, jQuery akan mencegah pengiriman form secara default dan melakukan validasi.    * **Nama**: Memeriksa apakah diisi.    * **Email**: Memeriksa apakah diisi.    * **Password**: Memeriksa apakah panjangnya minimal 8 karakter.    * Jika validasi gagal, pesan kesalahan ditampilkan, dan pengiriman form dihentikan. 3. **Pengiriman Data dengan AJAX**:    * Jika semua input valid, data form dikirim ke file proses\_validasi.php menggunakan AJAX.    * Respons dari server ditampilkan di div #hasil.  Kode PHP:  1. **Penerimaan Data**:    * Mengambil data yang dikirim dari form (nama, email, password) dan melakukan validasi tambahan.    * Memeriksa kevalidan nama dan email serta panjang password. 2. **Menangani Kesalahan**:    * Jika ada kesalahan, pesan kesalahan ditampilkan.    * Jika tidak ada kesalahan, pesan konfirmasi ditampilkan. |