|  |  |
| --- | --- |
|  | 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 2023* |

**NAMA : APRILIA PUTRI ANGGRAENI**

**KELAS : SIB-2C**

**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

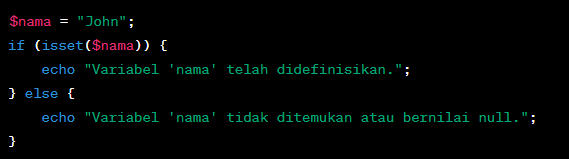
**Perhatian**

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

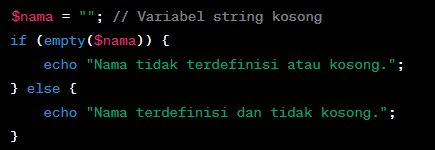
bernilai null." akan ditampilkan.

**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)    **Jawaban :** isset() digunakan untuk memeriksa apakah variabel $umur sudah didefinisikan dan tidak bernilai null. Jika variabel tersebut ada dan bernilai 18 atau lebih, maka ditampilkan pesan "Anda sudah dewasa", jika tidak, pesan "Anda belum dewasa atau variabel 'umur' tidak ditemukan" akan ditampilkan. |
| 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)    **Jawaban :** isset() memeriksa apakah elemen "nama" ada dalam array $data dan tidak null. Jika ada, ditampilkan nilai "nama", jika tidak, ditampilkan pesan bahwa elemen "nama" tidak ditemukan. |

**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)    **Jawaban :** empty() memeriksa apakah array $myArray kosong atau tidak terdefinisi. Jika kosong, ditampilkan pesan "Array tidak terdefinisi atau kosong", jika tidak, ditampilkan pesan bahwa array terdefinisi dan tidak 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)    **Jawaban :** empty() memeriksa apakah variabel $nonExistentVar tidak terdefinisi atau kosong. Jika kosong, ditampilkan pesan "Variabel tidak terdefinisi atau kosong", jika tidak, ditampilkan pesan bahwa variabel terdefinisi dan tidak kosong. |

**Praktikum Bagian 3 : Form Input PHP**

|  |  |
| --- | --- |
| **Langkah** | **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)    **Sesudah di Submit**    **Jawaban :** Form pada file ini digunakan untuk mengumpulkan input dari pengguna, khususnya nama dan email. Pengguna diharuskan mengisi kedua kolom tersebut sebelum mengirimkan data. Setelah form disubmit, data akan diproses oleh kode PHP yang berada di bagian bawah file. Kode tersebut memeriksa apakah permintaan berasal dari metode POST, lalu mengambil nilai dari input yang dikirimkan, dan menampilkannya di halaman sebagai output. Ini menunjukkan cara dasar untuk menangani input pengguna di PHP dan memberikan respons yang sesuai. |
| 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)    **Sesudah di Submit**    **Jawaban** : Form ini digunakan untuk mengumpulkan input nama dari pengguna. Setelah disubmit, form memvalidasi apakah kolom nama diisi. Jika kosong, pesan kesalahan "Nama harus diisi!" ditampilkan. Jika terisi, nilai nama disimpan dan pesan "Data berhasil disimpan!" ditampilkan. Form menggunakan metode POST dan mengarah ke halaman yang sama untuk memproses data, serta menggunakan htmlspecialchars() untuk mencegah XSS. Nilai input tetap ditampilkan setelah submit agar pengguna tidak perlu mengisi ulang. |

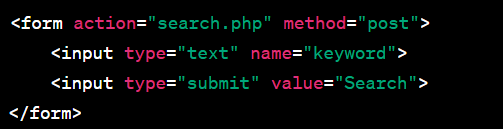
**HTML Injection**

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:



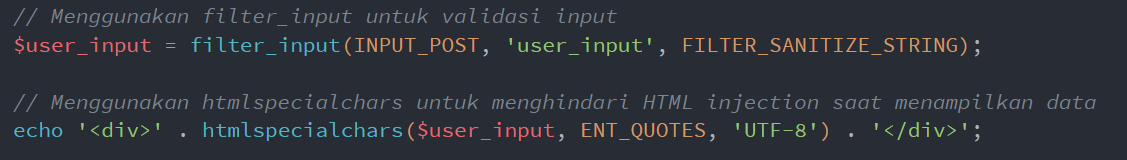
Kode JavaScript berbahaya tersebut akan dieksekusi oleh browser pengguna dan akan memunculkan kotak peringatan "You have been hacked!". Ini adalah contoh HTML injection yang sederhana. Sebagai pengembang web, penting untuk selalu melakukan sanitasi dan validasi input dari pengguna, serta menghindari 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    **Sesudah di submit** |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1)  **Jawaban :** Kode ini menyediakan formulir untuk input teks dengan metode POST. Input disanitasi menggunakan htmlspecialchars() untuk mencegah XSS dan divalidasi agar tidak kosong. Jika input valid, data ditampilkan; jika kosong, muncul pesan kesalahan. Kode ini menunjukkan cara membuat formulir yang aman dan responsif. |
| 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    **Sesudah di Submit** |
| 8 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.2)  **Jawaban :** Penambahan kode program memungkinkan pengguna untuk memasukkan data teks dan email. Input disanitasi menggunakan htmlspecialchars() untuk mencegah XSS, dan email divalidasi dengan filter\_var() untuk memastikan format yang benar. Pesan umpan balik jelas disediakan untuk input valid dan tidak valid, serta pengisian email dijadikan wajib, meningkatkan keamanan dan pengalaman pengguna. |

**Regular Expression (Regex)**

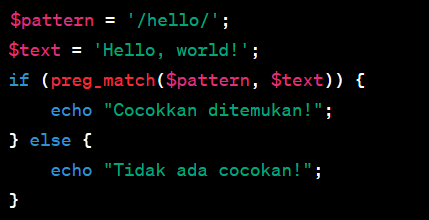
Regex adalah alat yang kuat untuk mencari, mencocokkan, atau memanipulasi teks berdasarkan pola tertentu. Anda dapat menggunakannya untuk berbagai tujuan seperti validasi input, pencarian teks, penggantian teks, dan banyak lagi.

**Basic Regex Patterns:**

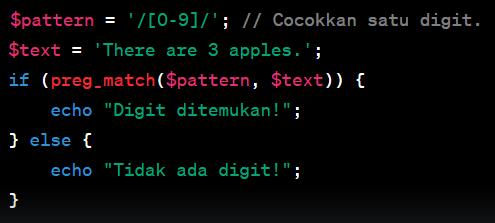
* **Karakter Tunggal**: Karakter apa pun akan cocok dengan dirinya sendiri dalam Regex, kecuali karakter khusus yang harus dihindari dengan karakter pelarian **\**.
  + Contoh: Pola **/a/** akan cocok dengan huruf "a" dalam teks.
* **Karakter Spesifik**: Anda dapat mencocokkan karakter tertentu dengan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

**Contoh Penggunaan Regex:**

1. 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) **Jawaban :** Kode di atas menggunakan ekspresi reguler untuk mencari huruf kecil dalam string $text. Fungsi preg\_match() memeriksa apakah ada huruf kecil. Jika ditemukan, outputnya "Huruf kecil ditemukan!", jika tidak, outputnya "Tidak ada huruf kecil!". Pencocokan bersifat sensitif terhadap huruf. |
| 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)  **Jawaban** : Kode ini mencari angka dalam string $text menggunakan pola '/[0-9]+/'. Jika ditemukan, outputnya "Cocokkan: " diikuti oleh angka yang ditemukan. Jika tidak ada angka, outputnya "Tidak ada yang cocok!". Angka "123" akan dicocokkan dan ditampilkan. |
| 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)  **Jawaban :** Kode ini mengganti kata "apple" dalam string $text dengan "banana" menggunakan fungsi preg\_replace(). Hasilnya, $new\_text menjadi "I like banana pie." ketika ditampilkan. |
| 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)  **Jawaban :** Kode ini mencocokkan kata "god" atau "good" dalam string $text menggunakan pola '/go\*d/'. Jika ditemukan, outputnya "Cocokkan: god". Jika tidak ada kecocokan, outputnya "Tidak ada yang cocok!". Dalam kasus ini, "god" dicocokkan dan ditampilkan. |
| 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)    **Jawaban :** Kode ini menggunakan pola '/go?d/', yang mencocokkan "gd" (tanpa huruf 'o') atau "god" (dengan satu huruf 'o'). Dalam string 'god is good.', "god" akan dicocokkan, dan outputnya adalah "Cocokkan: god". Kata "good" tidak cocok dengan pola ini. |
| 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)    **Jawaban :**  Kode ini menggunakan pola '/go{n,m}d/', yang mencocokkan kata dengan 'g', diikuti oleh huruf 'o' sebanyak antara n hingga m kali, dan diakhiri dengan 'd'. Dalam konteks ini, jika kita set n=1 dan m=2, pola ini akan mencocokkan:   * "god" (1 huruf 'o') * "good" (2 huruf 'o')   Jika digunakan dalam string 'god is good.', outputnya akan menjadi "Cocokkan: god" untuk "god" yang dicocokkan terlebih dahulu. Pola ini tidak akan mencocokkan "gd" (tanpa huruf 'o') atau kata-kata dengan lebih dari 2 huruf 'o'. |

**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    **Sesudah di submit :** |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 6.1)  **Jawaban :** Kode program di atas membuat form yang mengumpulkan input pilihan buah, warna favorit, dan jenis kelamin. Data diproses menggunakan metode POST, menampilkan hasil pilihan pengguna setelah form disubmit. Jika warna tidak dipilih, ditampilkan pesan "Anda tidak memilih warna favorit." |
| 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)    **Sesudah di Submit**    **Jawaban :** Dari penambahan kode program di atas, saya mengamati bahwa form menggunakan jQuery untuk mengirim data ke server secara asynchronous menggunakan AJAX, tanpa me-refresh halaman. Setelah pengguna mengisi form dan mengklik submit, data dikumpulkan dengan fungsi serialize() dan dikirim ke file PHP (proses\_lanjut.php) melalui metode POST.  Hasil respons dari server akan ditampilkan di elemen <div id="hasil">. Hal ini memungkinkan pengalaman pengguna yang lebih interaktif karena halaman tidak perlu dimuat ulang saat form dikirim, dan hasil bisa langsung terlihat di bagian yang telah ditentukan. |

**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    **Sesudah di Submit** |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.1)  **Jawaban :** Dari penambahan kode program di atas, saya mengamati bahwa form input dilengkapi dengan validasi server-side menggunakan PHP. Setelah form dikirim, PHP memeriksa apakah kolom "Nama" dan "Email" telah diisi, serta memvalidasi format email menggunakan filter\_var(). Jika ada kesalahan, pesan error ditampilkan; jika tidak, data diproses lebih lanjut (misalnya ditampilkan atau disimpan). Validasi ini memastikan bahwa data yang diterima valid sebelum dilakukan pemrosesan lebih lanjut. |
| 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  **Jika isi kosong atau tidak ada inputan**    **Sesudah di Submit**    **Jika from di isi**    **Sesudah di Submit** |
| 8 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.2)  **Jawaban :** Kode di atas menambahkan validasi di sisi klien menggunakan jQuery. Jika input "nama" atau "email" kosong, pesan error akan muncul dan pengiriman form dicegah. Ini meningkatkan pengalaman pengguna dengan memberikan umpan balik langsung tanpa perlu memuat ulang halaman. |
| 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)  **Jika from tidak diisi**      **Sesudah di submit**    **Jawaban :**   * Form meminta input *nama* dan *email*, divalidasi menggunakan jQuery. * Jika form valid, data dikirim via AJAX tanpa reload halaman. * Ada kesalahan, variabel password tidak diperlukan karena inputnya tidak ada. * PHP memproses data, memvalidasi, dan mengembalikan respon. |
| 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)  **Jawaban :** Ketika password pada form diisi kurang dari 8 karakter akan diperi peringatan “Password minimal 8 karakter”    **Password 8 karakter**    **Jawaban :**   * *Password* ditambahkan dan divalidasi di jQuery serta PHP. * Jquery memeriksa jika *password* kurang dari 8 karakter dan menampilkan error. * PHP memvalidasi ulang *password* setelah menerima input, memastikan memenuhi aturan minimal 8 karakter. * Validasi password di jQuery dan PHP memastikan input *password* minimal 8 karakter sebelum memproses lebih lanjut. |