|  |  |
| --- | --- |
|  | 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* |

**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

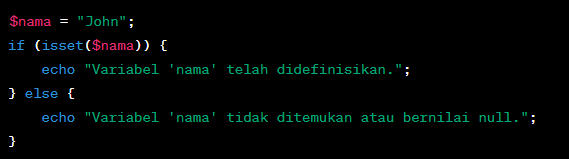
**Perhatian**

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()**:



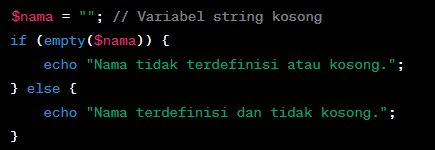
Kode di atas memeriksa apakah variabel **$nama** telah didefinisikan. Jika ya, maka pesan "Variabel 'nama' telah didefinisikan." akan ditampilkan, jika tidak, maka pesan "Variabel 'nama' tidak ditemukan atau bernilai null." akan ditampilkan.

**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)  Jawab :    Pada hasil diatas akan tampil tersebut karena terdapat asset, yang dimana fungsi asset ini mengembalikan true jika variabel telah didefinisikan dan memiliki nilai, dan false jika variabel tidak ada atau bernilai null. |
| 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)    Jawab :  Isset digunakan ketika nilai tersebut ditemukan atau true maka akan mengeluarkan outputnya, ketika false atau salah maka akan tidak mengembalikan nilai. |

**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)  Jawab :    Fungsi empty adalah untuk mengecek variabel kosong atau tidaknya, dan mengembalikan nilai true ketika kosong, dan mengembalikan nilai false ketika tidak kosong atau terdapat isi. |
| 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)  Jawab :    Sama seperti percobaan sebelumnya tetapi pada percobaan yang baru saja dilakukan tidak adanya variabel dan tidak ada nilainya, sehingga ketika dijalankan akan tampil “Variabel tidak terdefinisi atau 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)  Jawab :      Pada hasil diatas dapat dijelaskan bahwa ketika menginputkan nama serta email kemudian di submit maka akan mengalihkan ke halaman selanjutkan untuk menampilkan yang diinputkan tadi. |
| 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)  Jawab :      \*ketika menekan submit tanpa mengisi nama    Penjelasan :  Pada hasil diatas sama seperti percobaan sebelumnya mengisi nama dan menekan tombol submit dan akan dialihkan kehalaman berikutnya dan ketika belum mengisi nama tetapi menekan tombol submit maka akan terdapat perintah “Nama harus diisi” dengan demikian hasil diatas menambahkan percabangan if else untuk memberikan perintah sepeti gambar diatas. |

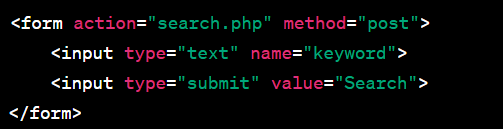
**HTML Injection**

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:



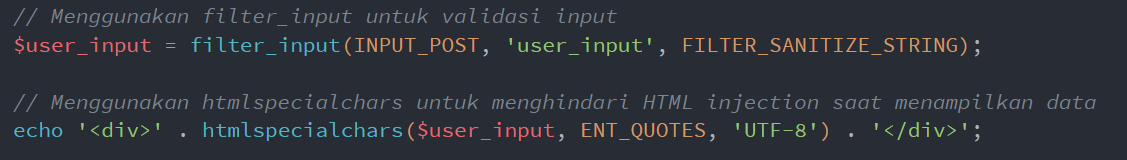
Kode JavaScript berbahaya tersebut akan dieksekusi oleh browser pengguna dan akan memunculkan kotak peringatan "You have been hacked!". Ini adalah contoh HTML injection yang sederhana. Sebagai pengembang web, penting untuk selalu melakukan sanitasi dan validasi input dari pengguna, serta menghindari 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)  Jawab:    Penggunaan kode diatas “htmlspecialchars” digunakan untuk menghapus atau menggantikan karakter yang berpotensi berbahaya, seperti **<**, **>**, **&**, dan lainnya. Anda bisa menggunakan fungsi seperti **htmlspecialchars()** atau **strip\_tags()** untuk ini. |
| 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)  Jawab :    \*ketika tidak mengisi dengan email    \*ketika mengisi dengan email    Penjelasan :  Pada gambar diatas dapat dijelaskan bahwa penggunaan “FILTER\_VALIDATE\_EMAIL” akan membuat form tersebut harus mengisi email dengan tepat dan ketika mengisi form tersebut tanpa email maka akan keluar perintah “Input tidak valid, silakan masukkan email yang benar” |

**Regular Expression (Regex)**

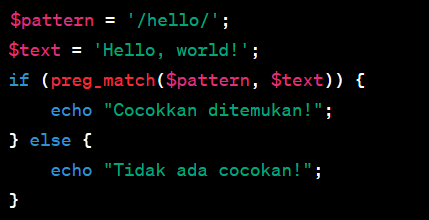
Regex adalah alat yang kuat untuk mencari, mencocokkan, atau memanipulasi teks berdasarkan pola tertentu. Anda dapat menggunakannya untuk berbagai tujuan seperti validasi input, pencarian teks, penggantian teks, dan banyak lagi.

**Basic Regex Patterns:**

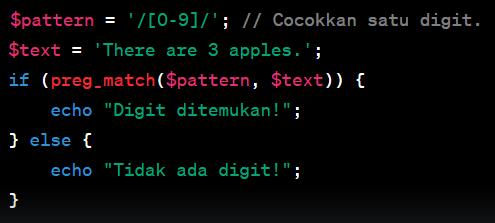
* **Karakter Tunggal**: Karakter apa pun akan cocok dengan dirinya sendiri dalam Regex, kecuali karakter khusus yang harus dihindari dengan karakter pelarian **\**.
  + Contoh: Pola **/a/** akan cocok dengan huruf "a" dalam teks.
* **Karakter Spesifik**: Anda dapat mencocokkan karakter tertentu dengan 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)  Jawab :    Pada hasil diatas dapat dijelaskan bahwa penggunaan regex untuk mencocokkan setiap huruf pada patternnya, misalnya pada percobaan tadi yang dilakukan pattern yang diminta adalah dari huruf a sampai z, dan text nya “This is a sample text” sehingga pattern akan mencocokkan semua huruf pada variabel text untuk menemukan huruf kecil pada text tersebut, jika ada akan muncul pernyataan “Huruf kecil ditemukan!”. |
| 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)  Jawab :    Pada hasil percobaan yang tadi dilakukan sama halnya percobaan yang pertama, cuman bedanya pada percobaan kali ini terdapat tanda “+” yang menandakan bahwa pada pattern yang berisikan “ ’/[0-9]/’+ ” yang akan mencocokkan angka pada text yang bertuliskan “There are 123 apples” yang dimana terdapat angka “123” yang cocok pada pattern tersebut, penggunaan dari “+” ini adalah untuk mencocokan berapa kali dalam text sehinga pada gambar tersebut tampil hasil “Cocokkan: 123”. |
| 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)  Jawab :    Pada percobaan 3 ini penggunaan replacement adalah untuk menggantikan, seperti pada kode program yang baru saja dicoba, untuk pattern nya adalah apple, untuk replacement nya adalah banana dan text nya adalah “I like apple pie” dan pada new\_text ini berikan keyword “preg\_replace” yang dimana fungsinya adalah untuk menggantikan yang berada didalam parameter tersebut. Pada program tersebut apple yang akan menggantikan banana sehingga pada text “I like apple pie” akan mencocokkan kata “banana” ketika sudah ditemukan maka otomatis akan menggantikan kata tersebut dan akan tampil seperti hasil diatas. |
| 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)  Jawab :    Pada hasil diatas akan tampil “Cocokkan : god” dikarenakan pada kode program tersebut pada pattern terdapat “ ‘/do\*d/’ “ yang akan mengeksekusi nilai variabel yang berada di $text kemudian akan dicocokkan menggunakan “preg\_match” matches berfungsi untuk menyimpan nilai yang sudah dicocokkan tadi. |
| 17 | Soal 5.5: Buatlah script langkah 14 dengan variable pattern yang di ubah menggunakan `?` (0 atau 1 kali). Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.5)  Jawab :  Kode Program :    Hasil :    Penjelasan :  Pada kode program diatas pada saat dirubah ke “?” maka tampilannya sama seperti percobaan sebelumnya, hanya saja perbedaan antara 2 tanda tersebut adalah terletak pada fungsi nya, yang dimana “\*” lebih fleksibel itu berrti “o” yang berada di pattern bisa muncul beberapa kali, tetapi penggunaan “?” lebih terbatas karena o di dalam text harus muncul 1 kali. |
| 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)  Jawab :  Kode Program :    Hasil :    Penjelasan :  Pada program diatas penggunaan {n,m} akan meng-cocokkan hasil dari ‘o’ 1 kali terlebih dahulu, ketika dirasa ditemukan, maka akan langsung ditampilkan, ketika tidak ditemukan maka akan berlanjut ke ‘o’ 2 kali yang dimana akan meng-cocokkan o yang berjumlah 2 kali. |

**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)  Jawab :    Pada program diatas adalah contoh form yang bisa memilih buah menggunakan “option” sehigga kita bisa memilih pilihan buah, serta warna. Untuk pilih warna menggunakan “checkbox” yang dimana bisa meng-checklist 1 atau semua warna yang berada dipilihan tersebut. Pada pilihan “jenis kelamin” mengunakan radio yang bentuknya bulat, pada pilihan ini hanya bisa memilih salah satu dari piihan tersebut. |
| 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)  Jawab:    Pada hasil tersebut, saat sudah memilih semua pilihannya, akan dikirim melalui proses\_lanjut.php yang dimana disana lah yang akan menerima datanya, lalu hasilnya akan tampilkan di div id=”hasil”. |

**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)  Jawab :      Pada hasil diatas sama dengan sama halnya dengan percobaan yang tadi, cuman perbedaannya adalah tidak menggunakan ajax untuk menampilkan setelah mengisi form tersebut, melainkan menggunakan percabangan if, kemudian akan menampilkan kesalahan jika didalam form tersebut terdapat kesalahan. |
| 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)  Jawab :  \*menekan tombol submit tanpa mengisi form didalamnya    Pada gambar diatas terdapat peringatan yang berwarna merah ketika tidak mengisi form yang ada didalamnya.  \*ketika menekan tombol submit dengan mengisi form yang ada didalamnya    Sama seperti percobaan sebelumnya, ketika sudah mengisi form tersebut maka, akan dialihkan ke halaman selanjutnya kemudian akan menampilkan nama dengan email yang sudah diketikkan sebelumnya. |
| 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)  Jawab :  Kode program :    Hasil :    Pada program tersebut sudah ditambahkan ajax, sehingga saat dijalankan program tersebut ketika membuat kesalahan atau mengisi dan meng-submit form tersebut tidak pindah ke halaman selanjutnya. |
| 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)  Jawab :  Kode program form\_validasi.php :      Kode program proses\_validasi.php :    Hasil :      Penjelasan :  Pada hasil diatas untuk menambahkan form password dibutuhkan namanya variabel password pada proses\_validasi.php dan form\_validasi.php. untuk memberikan validasi pada peringatan password, maka dibutuhkan percabangan kondisi, dengan menggunakan if yang dimana ketika panjang password tersebut kurang dari 8 maka akan muncul peringatan “password terlalu pendek, mohon isi dengan 8 karakter saja” ketika semua form sudah diisi dengan passwordnya, maka akan beralih ke halaman proses\_validasi yang berisikan form yang sudah diiskan sebelumnya. |