**LAPORAN HASIL PRAKTIKUM**

**JOBSHEET 7**

**FORM PROCCESSING**

**DESAIN PEMROGRAMAN WEB**

****

**Disusun Oleh:**

**AGUNG FRADIANSYAH**

**NIM. 2341720045**

**TI-2E / 01**

**D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2024/2025**

Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet-7: PHP - Form Proccessing

Mata Kuliah Desain dan Pemrograman Web

Tim Ajar Desain dan Pemrograman Web

Oktober 2024

**Topik**

- Konsep Form Proccessing dengan PHP dan Jquery

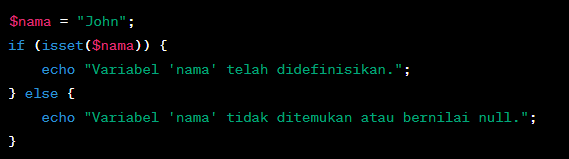
**Tujuan**

Mahasiswa diharapkan dapat:

1. Mahasiswa mampu membuat form menggunakan PHP
2. Mahasiswa mampu membuat form menggunakan jQuery

**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()**:

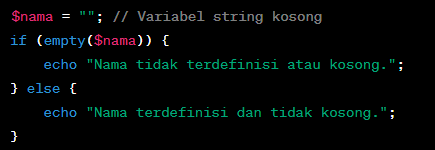


Kode di atas memeriksa apakah variabel **$nama** telah didefinisikan. Jika ya, maka pesan "Variabel 'nama' telah didefinisikan." akan ditampilkan, jika tidak, maka pesan "Variabel 'nama' tidak ditemukan atau bernilai null." akan ditampilkan.

|  |  |
| --- | --- |
| **Praktikum Bagian 1. Function Isset** | |
| **Langkah** | **Keterangan** |
| 1 | Buatlah satu file baru di dalam direktori **minggu7**, beri nama **isset.php.** |
| 2 | Ketikkan ke dalam file **isset.php** kode di bawah ini. |
| 3 |  |
| 4 | Simpan file tersebut, kemudian buka browser dan jalankan **localhost/minggu7/ isset.php**. |

|  |  |
| --- | --- |
|  | Apa yang anda pahami dari penggunaan isset pada file tersebut. Catat di bawah ini pemahaman anda. (Pertanyaan No. 1)  Jawab : fungsi isset() digunakan untuk memeriksa apakah variabel $umur sudah dideklarasikan atau tidak. Pada program isset($umur) mengembalikan nilai true jika variabel $umur sudah diisi jika $umur belum pernah dideklarasikan isset() akan mengembalikan false  Jadi pada program mengecek jika variabel $umur ada isinya atau terdefinisikan dan nilainya lebih atau sama dengan 18, program akan menampilkan “Anda sudah dewasa”, namun jika $umur belum dideklarasikan atau kurang dari 18 akan menampilakan “Anda belum dewasa atau variabel 'umur' tidak ditemukan.", maka hasilnya akan seperti berikut;  Jika tidak menggunakan isset, akan muncul error seperti dibawah ini |
| 5 | Tambahkan isi dari file isset.php tersebut dengan kode di bawah ini. |
| 6 | Simpan file, kemudian buka browser dan jalankan **localhost/minggu7/isset.php**. Buat hasil tidak menjadi 1 baris, jadi tampilan dari echo harus terpisah. Apa yang anda pahami dari penggunaan isset pada file tersebut? Catat pemahaman anda di bawah ini.  (Pertanyaan No. 2)  Jawab : Dalam bagian kedua, isset($data["nama"]) digunakan untuk memeriksa apakah elemen "nama" ada di dalam array $data. Jika elemen "nama" ada, maka program akan menampilkan nilainya ("Jane"). Jika tidak, program akan menampilkan "Variabel 'nama' tidak ditemukan dalam array." Ini berguna untuk memastikan bahwa elemen array tertentu telah terdefinisi sebelum digunakan, sehingga mencegah error saat akses elemen yang tidak ada.  Jika tidak menggunakan isset, lalu kita ganti nama elemennya menjadi nama1 maka akan muncul eror seperti berikut  Jika menggunakan isset dan nama elemennya diganti menjadi nama1, maka tampil seperti berikut, dan tidak muncul eror |

**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 Bagian 2. Function empty

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama **empty.php** di dalam folder minggu7. Ketikkan kode pada langkah 2 di dalam **empty.php** |
| 2 |  |
| 3 | Simpan file tersebut, kemudian buka browser dan jalankan **localhost/minggu7/ empty.php**. Apa yang anda pahami dari penggunaan empty pada file tersebut? Catat pemahaman anda di bawah ini.  (Pertanyaan No. 3)  Jawab : pada program diatas fungsi empty() digunakan untuk memeriksa apakah variabel $myArray kosong atau tidak terdefinisi. Jika variabel kosong tidak didefinisikan fungsi empty() akan mengembalikan true. Sehingga if jika dijalanka akan menampilkan pesan “Array tidak terdefinisi atau kosong” . namun jika $myaaray memiliki setidaknya satu elemen, empty akan mengembalikan false program akan masuk ke bagian else, menampilkan "Array terdefinisi dan tidak kosong." Intinya fungsi empty ini untuk memastikan array tersebut memiliki elemen atau tidak |
| 4 | Tambahkan isi dari file empty.php tersebut dengan kode di bawah ini. |
| 5 | Simpan file tersebut, kemudian buka browser dan jalankan **localhost/minggu7/ 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 pemahaman anda di bawah ini. (Pertanyaan No. 4)  Jawab : Fungsi empty() digunakan untuk memeriksa apakah suatu variabel kosong atau tidak terdefinisi. Pada $myArray, empty() mengembalikan true karena $myArray tidak memiliki elemen (kosong), sehingga menampilkan "Array tidak terdefinisi atau kosong". Kemudia selanjutnya memeriksa variabel $nonExistentVar, yang belum pernah dibuat atau didefinisikan di kode sebelumnya. Karena variabel ini **tidak ada**, maka empty($nonExistentVar) akan memberikan hasil **true** juga, sehingga pesan "Variabel tidak terdefinisi atau kosong" muncul, seperti berikut; |

**Praktikum Bagian 3. Form Input PHP**

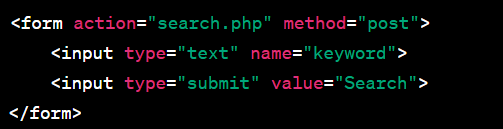
|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama **form.php**. Ketikkan kode pada langkah 2. |
| 2 |  |
| 3 | Buat satu file baru bernama **proses\_form.php**. Ketikkan kode berikut dalam  **proses\_form.php** |
| 4 | Simpan file tersebut, kemudian buka browser dan jalankan **localhost/minggu7/ proses\_form.php**. Apa yang anda pahami dari penggunaan form pada file tersebut? Catat pemahaman anda di bawah ini. Selanjutnya jalankan **localhost/minggu7/form.php**  Apa yang terjadi? Catat pemahaman anda di bawah ini. (Pertanyaan No. 5)  Jawab : saat file form dimuat, akan tampil sebuah inputan nama, email serta tombol submit yang jika diklik halaman ini akan memproses data atau mengirimkan data tersebut ke file proses\_form.php dengan method POST yang menyembunyikan data dari URL. Kemudian pada proses\_form menggunakan kondisi if ($\_SERVER["REQUEST\_METHOD"] == "POST") untuk memeriksa apakah form sudah dikirim. Jika form dikirim, kode akan mengambil data dari input nama dan email, kemudian menampilkannya di halaman yang sama. Hal ini memungkinkan form untuk memproses dan menampilkan data input secara langsung setelah pengguna mengklik Submit.  Jika sudah dikirim atau tampilan di proses\_form |
| 5 | Buat satu file baru bernama **form\_self.php** di dalam folder **minggu7**. Ketikkan kode di bawah ini. |
|  |  |
| 6 | Simpan file tersebut, kemudian buka browser dan jalankan **localhost/minggu7/ form\_self.php**. Apa yang anda pahami dari penggunaan form pada file tersebut? Catat pemahaman anda di bawah ini. (Pertanyaan No. 6)  Jawab : form ini mengecek apakah sudah mengisi kolom nama belum, jika belum maka akan menampilkan pesan dari variabel $namaErr yaitu “Nama harus diisi”. Pada program ini, terdapat dua variabel: $namaErr untuk menyimpan pesan error dan $nama untuk menyimpan nama pengguna. Ketika pengguna menekan tombol Submit, form dikirim menggunakan metode POST, sehingga $\_SERVER["REQUEST\_METHOD"] menjadi POST. Jika kondisi ini true, kode dalam if akan dijalankan untuk memproses data. Jika nama kosong, pesan error disimpan dalam $namaErr "Nama tidak boleh kosong, harus diisi." Jika nama diisi, nilainya disimpan di variabel $nama, dan pesan "Data berhasil disimpan." ditampilkan.  Hasilnya seperti berikut  jika tidak mengisi inputan kolom nama  jika mengisi maka tampilannya akan seperti berikut |

**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:



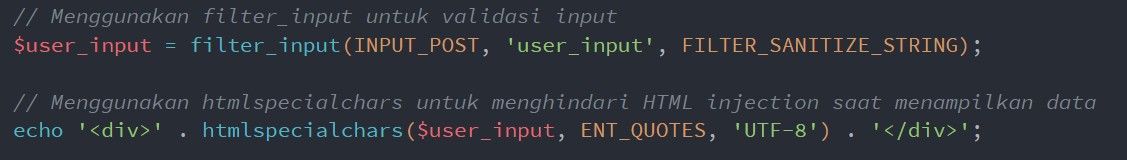
Kode JavaScript berbahaya tersebut akan dieksekusi oleh browser pengguna dan akan memunculkan kotak peringatan "You have been hacked!". Ini adalah contoh HTML injection yang sederhana. Sebagai pengembang web, penting untuk selalu melakukan sanitasi dan validasi input dari pengguna, serta menghindari 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. **Filter Input**: Filter 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 Bagian 4. HTML Injection

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file baru bernama **html\_aman.php** di dalam folder **minggu7**. Ketikkan kode berikut |
| 2 |  |
| 3 | Buat file dengan nama **form1.php**, isi dengan kode untuk membuat form dengan 1 input teks yang akan diproses pada **html\_aman.php**. |
| 4 | Simpan file, buka browser dan jalankan/refresh **localhost/minggu7/form1.php**  Isikan input teks yang memuat tag html. |
| 5 | Selanjutnya hapus kode **$input = htmlspecialchars($input, ENT\_QUOTES, 'UTF-8');** |

|  |  |
| --- | --- |
|  | Simpan file, buka browser dan jalankan/refresh **localhost/minggu7/form1.php**  Isikan kembali input teks yang memuat tag html. |
| 6 | Catat di sini apa yang anda amati, berikan penjelasan anda. (Pertanyaan No. 7)  Jawab : terdapat form yang berisi inputan dan submit yang akan dikirim kehalaman html\_aman.php, jika kita menggunakan htmlspesialchars lalu kita inputkan <b>Agung</b> hasilnya akan tetap sama muncul dengan tag htmlnya  Jika kita mengahpus fungsi htmlspesialchars lalu memasukkan <b>Agung</b>, hasilnya di halaman akan ditampilkan sebagai **Agung**(teks yang ditebalkan), hasilnya akan seperti berikut; |
| 7 | Ketikkan kode di bawah ini pada **html\_aman.php** |
| 8 |  |
| 9 | Lengkapi kode program pada bagian if dan else sehingga dapat dijalankan dengan benar. |
| 10 | Pada **form1.php**, tambahkan kode untuk membuat input betupa email yang akan diproses pada  **html\_aman.php**.  Simpan file, buka browser dan jalankan/refresh **localhost/minggu7/form1.php** |
| 11 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 8)  Jawab : kode program tersebut terdapat fungsi filter\_var() untuk memfilter dan memvalidasi data dalam format email apakah sudah benar atau tidak emailnya. Jika fungsi tersebut mengembalikan nilai true yaitu email valid maka akan menampilkan email, jika tidak maka akan menampilkan “Email tidak valid”  Jika valid  Jika email tidak valid |

**Regular Expression (Regex)**

Regex adalah *tools* 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: **.**, **\***, **+**, **?**, **|**, **[**, **]**, **(**, **)**, **{**, **}**,

**^**, **$**, **\**.

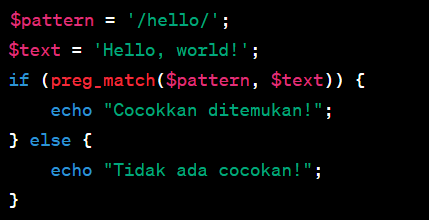
* **Karakter Set**: Anda dapat mencocokkan karakter dari satu set karakter dengan menggunakan **[ ]**. Misalnya, **/[aeiou]/** akan mencocokkan salah satu huruf vokal.
* **Rentang Karakter**: Anda dapat menggunakan **-** dalam karakter set untuk menentukan rentang karakter. Misalnya, **/[a-z]/** akan 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

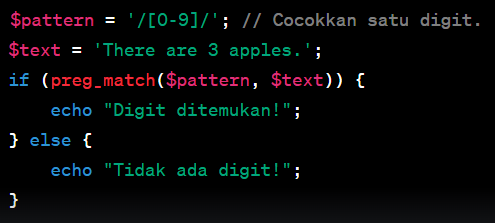
* + **?**: 0 atau 1 kali
  + **{n}**: Persis n kali
  + **{n,}**: Setidaknya n kali
  + **{n,m}**: Setidaknya n kali, maksimal m kali

# Contoh Penggunaan Regex:

1. Mencocokan pola



1. Mencocokan karakter set



# Praktikum Bagian 5 : Penggunaan Regex Pada PHP

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama **regex.php** di dalam folder **minggu7**. Ketikkan kode berikut |
| 2 |  |
| 3 | Simpan file , kemudian buka browser dan jalankan/refresh **localhost/minggu7/regex.php** |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 9)  **Jawab** : program ini untuk memeriksa apakah terdapat huruf kecil dalam variabel $text atau tidak. Kode tersebut pertama akan membuat variabel pattern yang untuk menyimpan pola pencarian huruf kecil a sampai z, kemudian fungsi preg\_match memeriksa apakah pola huruf kecil ada didalam teks jika ada hasilnya true dan menampilkan pesan "Huruf kecil ditemukan!", Jika tidak ada huruf kecil, maka akan muncul pesan "Huruf kecil tidak 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/minggu7/regex.php** |
| 8 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 10)  **Jawab** : pada program $pattern akan menyimpan atau mencari karakter angka dari 0 sampai 9, kemudian tanda + mencari satu atau lebih karakter angka, preg\_match() akan mencari pola dalam $text. Jika ditemukan angka, pesan "Cocokkan: 123" akan ditampilkan (karena "123" adalah angka pertama yang cocok dengan pola dalam $text). Jika tidak ada angka dalam teks, maka "Tidak ada yang cocok" akan ditampilkan .  Maka akan ditampilkan sebagai berikut; |
| 9 | Ketikkan kode tambahan berikut di dalam **regex.php** |
| 10 |  |
| 11 | Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/minggu7/regex.php |
| 12 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 11)  **Jawab** : pertama pattern digunakan untuk mencari kata apel, kemudian $replancement digunakan untuk mendefinisikan teks pengganti dengan ‘banana’ variabel $text untuk menyimpan teks yang akan dimodifikasi lalu fungsi preg\_replace yaitu pertama akan mencari semua kata dalam $text yang cocok dengan kata apple, jika menemukan kata apple fungsi ini akan mengganinya dengan teks $replancement, hasilnya yang sudah dimodifikasi akan disimpan dalam variabel $new\_text. Maka jika dipanggil hasilnya akan seperti berikut |
| 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/minggu7/regex.php** |
| 16 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 12)  **Jawab** : jika halaman tersebut dimuat akan menampilkan seperti berikut;  Yang akan ditampilkan hanya 1 kata pertama jika sudah ada yang cocok, dalam program $pattern digunakan untuk menentukan pola pencarian, yaitu kata yang dimulai dengan huruf “g”, diikuti oleh satu atau lebih huruf “o”, dan diakhiri dengan huruf “d”, seperti "god", "good", "goood", dan sebagainya. $text adalah variabel yang menyimpan kalimat atau string yang akan diperiksa. Fungsi preg\_match digunakan untuk mencari semua kata yang cocok dengan pola dalam variabel $pattern di dalam kalimat yang ada di $text. Untuk menampilkan semua kata yang mengandung "go\*d", gunakan preg\_match\_all. |
| 17 | Pada script di langkah 14, ubah isi variable pattern dari ‘\*’ menjadi ‘?’.  Simpan file , kemudian buka browser dan jalankan/refresh **localhost/minggu7/regex.php**  Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 13)  **Jawab** : Ketika kita menggunakan simbol \* dalam pola pencarian, itu berarti kita mencari kata yang dimulai dengan huruf "g", diikuti oleh nol atau banyak huruf "o", dan diakhiri dengan huruf "d", dengan menggunak go\*d, kita bisa menemukan kata "god", "good", "goood", "gooooood", dan lainnya. Jika kita mengubah pola menggunakan simbol ?, artinya kita mencari kata yang dimulai dengan "g", diikuti oleh satu huruf "o", dan diakhiri dengan "d", dengan ini go?d, hanya kata akan menemukan kata "gd" (tanpa huruf "o") dan "god" (dengan satu huruf "o") yang cocok, sementara "good" dan "goood" tidak cocok karena pola baru ini tidak dapat menampilkan lebih dari satu huruf "o".  Jika mengganti isi dari $text menjadi $text = 'goood is good.'; Maka akan tampil seperti berikut |
| 18 | Ubah script langkah 14 dengan variable pattern **'/[o]{1,3}/'**.  Simpan file , kemudian buka browser dan jalankan/refresh **localhost/minggu7/regex.php**  Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 14)  **Jawab** : akan mencari 1 sampai 3 huruf “o” yang artinya jika setiap ada 1,2 atau 3 huruf o secara berurutan misal “goood” maka akan dianggap cocok  Variabel $text berisi kalimat “goood is good”, lalu fungsi preg\_match akan mencari pola dalam $text apakah ada kata yang menggandung huruf o dan menyimpan kecocokan dalam $matches. Karena hanya menggambil kecocokan pertama maka akan ditemukan “ooo” dalam kata “goood”. Maka hasilnya akan seperti berikut |

**Praktikum Bagian 6 : Form Lanjut**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama **form\_lanjut.php** di dalam folder **minggu7**. Ketikkan kode pada langkah 2 di dalam **form\_lanjut.php** |
| 2 |  |
|  | Buat satu file baru bernama **proses\_lanjut.php** di dalam folder **minggu7**. Ketikkan kode berikut |
| 4 | Simpan file, kemudian buka browser dan jalankan/refresh  **localhost/minggu7/form\_lanjut.php** |
| 5 | Catat di sini apa yang anda amati dari penambahan kode program di atas.  (Pertanyaan No. 15)  Jawab : pertama file form\_lanjut.php dimuat akan tampil sebuah form berisi dropdown buah, checkbox untuk warna favorit, dan radio button untuk jenis kelamin. Tombol submit akan mengirimkan data yang diisi ke server menggunaka method POST, kemudian pada file proses\_lanjut.php Setelah form dikirim, program memeriksa data POST untuk memastikan ada input dari pengguna. Jika ada, program menyimpan pilihan buah, warna favorit (jika dipilih), dan jenis kelamin. Kemudian, hasil pilihan ini ditampilkan kembali di halaman proses\_lanjut.php, sehingga pengguna langsung melihat hasil input mereka setelah submit. Jika tidak ada warna yang dipilih, program menampilkan pesan bahwa tidak ada warna favorit yang dipilih.  Jika tidak memilih warna |
| 6 | Buat satu file baru bernama **form\_ajax.php**. Ketikkan kode berikut. |

|  |  |
| --- | --- |
|  |  |
| 7 | Simpan file, buka browser dan jalankan/refresh **localhost/minggu7/form\_ajax.php**.  Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 16)  **Jawab** : pada program tersebut pertama terdapat form yang berisi beberapa input, yaitu terdapat dropdown untuk memilih buah, checkbox untuk memilih warna favorit, radio button untuk memilih jenis kelamin. Serta tombol submit yang jika diklik , maka AJAX akan mengirim data ke proses\_lanjut.php dengan method POST, tanpa merefresh halaman proses\_lanjut.php akan memproses data yang diterima dan mengembalikan hasilnya. Hasil pemrosesan akan ditampilkan dalam div hasil di halaman yang sama. Maka akan seperti berikut |

# Praktikum Bagian 7 : Validasi Form

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama **form\_validasi.php** di dalam folder **minggu7**. Ketikkan kode berikut. |
| 2 | Buat satu file baru bernama **proses\_validasi.php** di dalam folder **minggu7**. Ketikkan kode berikut. |
| 3 | Simpan file, kemudian buka browser dan jalankan/refresh  **localhost/minggu7/form\_validasi.php** |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 17)  Jawab : Saat form\_validasi.php dimuat, form meminta input "Nama" dan "Email" dengan metode POST. Ketika disubmit, proses\_validasi.php pertama akan memeriksa apakah data dikirim dengan method POST. Data "Nama" dan "Email" diambil, lalu divalidasi jika kosong, pesan "Nama tidak boleh kosong" ditambahkan ke dalam array $errors. empty($email) mengecek apakah "Email" kosong, dan menambahkan pesan "Email tidak boleh kosong" ke array $errors jika benar lalu fungsi filter\_var akan memastikan format valid atau tidak. Terakhir jika terdapar error pesan akan ditampilkan satu persatu menggunakan foreach, jika tidak ada error, data akan ditampilkan dalam pesan sukses dengan menggunakan htmlspecialchars()  Jika inputan tidak diisi    Jika nama tidak disini namun format email tidak valid    Jika benar semua |

|  |  |
| --- | --- |
| 5 | Ubah file **form\_validasi.php** menjadi seperti berikut. |
| 6 |  |
| 7 | Simpan file, buka browser dan jalankan/refresh **localhost/minggu7/form\_validasi.php** |
| 8 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (Pertanyaan No. 18)  Jawab : saat form\_validasi dimuat akan menampilkan sebuah form yang memiliki dua inputan nama dan email serta tombol submit. Setiap input memiliki elemen <span> untuk menampilkan pesan kesalahan jika input tidak sesuai. Ketika form disubmit, fungsi $("#myForm").submit() memeriksa validasi dengan kondisi jika "Nama" kosong, menampilkan pesan kesalahan di <span id="nama-error">. Jika "Email" kosong, menampilkan pesan kesalahan di <span id="email-error">.  Kemudian proses\_validasi.php akan dijalankan jika form dikirim dengan method POST, nama dan email dicek ulang jika tidak sesuai maka akan ditambahkan ke $errors, email akan dicek format kesesuainnya dengan fungsi filter\_var(), jika terdapat kesalah maka akan ditampilkan, jika semua validasi terpenuhi, form akan memproses dan mencetak pesan sukses, menampilkan input yang diterima  Jika nama dan email tidak diisi  Jika semua sesuai |
| 9 | Buatlah script langkah 6 dengan menggunakan ajax. Screenshoot kode yang anda buat. Catat di sini apa yang anda amati dari penambahan kode program.  (Pertanyaan No. 19)  Jawab : Form menampilkan input "Nama" dan "Email" dengan elemen `<span>` untuk menunjukkan pesan kesalahan jika input tidak sesuai. Saat tombol "Submit" diklik, sistem memeriksa apakah kedua kolom terisi. Jika "Nama" atau "Email" kosong, pesan kesalahan akan muncul, dan variabel `valid` menjadi `false`. Setelah semua input valid, data dikirim ke server melalui fungsi `$.ajax()`, dengan URL `proses\_validasi.php` sebagai tujuan dan metode `POST`. Jika pengiriman berhasil, respons dari server ditampilkan di elemen `#hasil`. Jika terjadi kesalahan saat pengiriman, pesan "Terjadi kesalahan saat mengirim data." akan ditampilkan di elemen yang sama. Dengan cara ini, pengguna akan mendapatkan umpan balik langsung tentang status pengisian form mereka.  Kode ; |

|  |  |
| --- | --- |
| 10 | Tambahkan kode untuk validasi password dengan ketentuan minimal 8 karakter dengan menggunakan jQuery dan PHP.  Screen shoot kode dan catat di sini apa yang anda amati dari penambahan kode program tersebut.  (Pertanyaan No. 20)  Jawab : form terdiri dari inputan nama, email dan password yang masing masing terdapan span untuk menampilkan pesan jika input tidak valid, serta tombol submit yang ketika diklik akan mengirim dan memproses diproses\_validasi\_pw.php. bedanya dari sebelumnya program ini akan memeriksa apakah panjangnya minimal 8 karakter. ika semua input valid, data dikirim menggunakan AJAX ke proses\_validasi\_pw.php dengan metode POST, data yang dikirim mencakup nama, email, dan password. Jika pengiriman berhasil, respons dari server ditampilkan di dalam elemen #hasil  Output  Kode form\_validasi.php  <https://github.com/agungfradiansyah12/pemrograman_web/blob/main/minggu7/form_validasi4.php>  Kode validasi\_proses.php  <https://github.com/agungfradiansyah12/pemrograman_web/blob/main/minggu7/proses_validasi_pw.php> |

LINK GITHUB MINGGU 7 AGUNG FRADIANSYAH

<https://github.com/agungfradiansyah12/pemrograman_web/tree/main/minggu7>

<https://github.com/agungfradiansyah12/pemrograman_web/tree/main/minggu7>

<https://github.com/agungfradiansyah12/pemrograman_web/tree/main/minggu7>