|  |  |
| --- | --- |
|  | 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

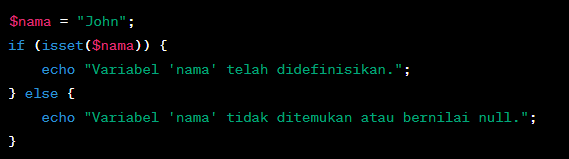
Perhatian

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():



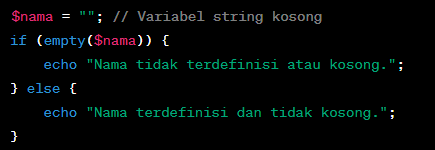
Kode di atas memeriksa apakah variabel $nama telah didefinisikan. Jika ya, maka pesan "Variabel 'nama' telah didefinisikan." akan ditampilkan, jika tidak, maka pesan "Variabel 'nama' tidak ditemukan atau bernilai null." akan ditampilkan.

# 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)   * Variable umur belum diinisialisasi sebelum digunakan di dalam gungsi isset maka output yang keluar adalah pada else. |
| 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)   * Pada penggunaan isset harus inisialisasi dahulu agar output umur keluar. |

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)   * Outputnya adalah *“array tidak terdifinisikan atau kosong”* karena array empty. |
| 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)   * Sama halnya dengan praktikum sebelumnya. |

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)   * Variable seuperglobal $\_POST digunakan untuk mengakses data yang dikirim dari metode POST. Input yang diambil dari dari file form.php |
| 5 | Buat satu file baru bernama form\_self.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form\_self.php |

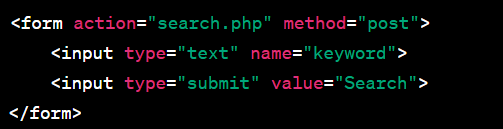
|  |  |
| --- | --- |
|  |  |
| 6 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/ form\_self.php. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 3.2)   * action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>": Formulir mengirimkan data ke halaman yang sama ($\_SERVER["PHP\_SELF"]). Fungsi htmlspecialchars() digunakan untuk memastikan bahwa input di-escape dengan benar dan mencegah potensi serangan XSS (Cross-Site Scripting). * Validasi sederhana agar input nama tidak kosong pada form, yang mana namaErr menampilakan pesan error, nama digunakan untuk pesan benar. |

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:



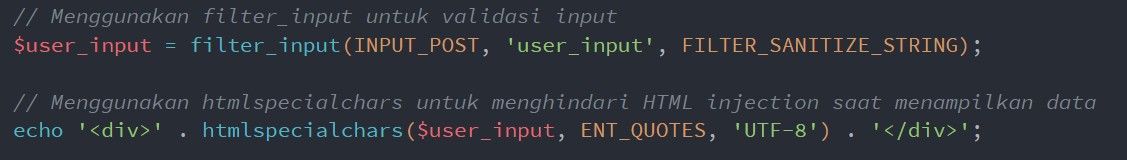
Kode JavaScript berbahaya tersebut akan dieksekusi oleh browser pengguna dan akan memunculkan kotak peringatan "You have been hacked!". Ini adalah contoh HTML injection yang sederhana. Sebagai pengembang web, penting untuk selalu melakukan sanitasi dan validasi input dari pengguna, serta menghindari 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)   * Formulir mengumpulkan nama dan email dari pengguna, lalu ketika formulir dikirim, metode POST mengirimkan dara ke server. Kode PHP memeriksa apakah ada permintaan POST, lalu melakukan sanitasi pada input menggunakan *htmlspecialchras()* untuk mencegah serangan XSS, dan menampilkan data yang telah di-sanitasi di halaman. |

|  |  |
| --- | --- |
|  |  |
| 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)   * Tujuan kode yang saya tambahkan ialah mengecek email sudah dimasukkan atau belum |

# 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.

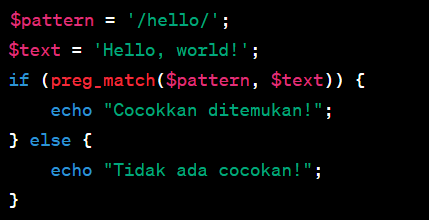
* Rentang Karakter: Anda dapat menggunakan - dalam karakter set untuk menentukan rentang karakter. Misalnya, /[a-z]/ akan mencocokkan huruf kecil apa pun.
* Kuantifier: Anda dapat mengatur berapa kali karakter atau kelompok karakter sebelumnya harus muncul. Contoh kuantifier:
  + \*: 0 atau lebih kali

o +: 1 atau lebih kali

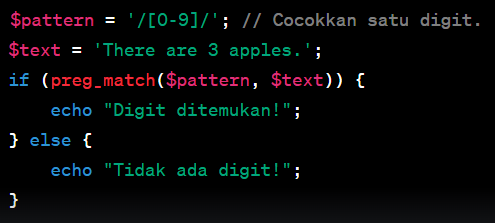
* + ?: 0 atau 1 kali
  + {n}: Persis n kali
  + {n,}: Setidaknya n kali
  + {n,m}: Setidaknya n kali, maksimal m kali

# 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)   * Iterasi kode program di atas menuju if karena pattern ‘/[a-z]/’ yang digunakan untuk mendeteksi 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)   * Pola ‘/[0-9]/’ sama dengan kode sebelumnya namun pedanya pada kode diatas menggunakan preg\_match yang mana fungsinya adalah untuk mencari kecocokan dengan pola tersebut. Jika menemukan angka maka angka pertama akan di tampilkan. |
| 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)   * Output yang dikeluarkan adalah *“I like banana pie”* karena terjadi replacment pada apple. Replacement sendiri digunakan untuk merubah apple menjadi banana pada $newtext. |
| 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)   * Untuk menampilkan god dan good maka preg\_match diubah menjadi preg\_match\_all agar menampilkan semuanya. |
| 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 kode program yang saya buat output yang dihasilkan adalah good. |
| 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)   * Tidak ada yang cocok pada huruf pada kode di atas. |

# 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)   * Form HTML ini memungkinkan pengguna memilih buah, warna, dan jenis kelamin, kemudian data dikirim ke PHP untuk diproses. PHP menampilkan hasil pilihan pengguna, termasuk pengecekan apakah warna dipilih atau tidak. |
| 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)   * Fungsi ajax :   + URL : Mengacu pada file yang akan memproses   + Type : Menggunakan metode POST   + Data: Mengirimkan data form yang sudah dikumpulkan   + Succes: ketika proses berhasil, respon dari server akan dimasukkan ke dala HTML dengan id #hasil. * Keuntungan menggunakan ajax dan jQuery   + Interaksi lebih cepat   + Pengalaman pengguna yang lebih baik   + FLexsibilitas |

# 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)   Pengiriman Form: Data dari form dikirim ke file PHP menggunakan metode POST.   Validasi Server-side:   * PHP mengecek apakah nama dan email sudah diisi. * Jika nama tidak diisi, pesan kesalahan ditampilkan. * Jika email tidak diisi atau formatnya tidak valid, pesan kesalahan juga ditampilkan.   + Jika validasi berhasil, PHP akan menampilkan pesan bahwa data berhasil dikirim beserta nama dan email yang diinputkan. |

|  |  |
| --- | --- |
| 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)   * Validasi nama dan email:   + Jika nama kosong, pesan kesalahan muncul.   + Jika email kosong, pesan kesalahan juga muncul. * Form dikirimkan atau tidak:   + Jika validasi gagal, form tidak akan dikirim ke server, dan pesan kesalahan muncul.   + Jika validasi berhasil (nama dan email terisi), form dikirimkan ke server. |
| 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)   * Ajax digunakan untuk mengirim data form ke server secara asinkron tanpa harus me-reload halaman. |
| 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)   * Mencegah Submit Form Default:   + event.preventDefault(); digunakan untuk mencegah form dikirim secara langsung oleh browser saat tombol "Submit" ditekan. Sebaliknya, proses validasi dilakukan terlebih dahulu. *  Validasi Input di Client-Side:   + Setiap input (nama, email, dan password) divalidasi di sisi klien:   + Nama:   + Jika kosong, pesan error "Nama harus diisi" akan muncul di elemen span dengan ID nama-error. * Email:   + Jika kosong, pesan error "Email harus diisi" muncul.   + Jika format email tidak valid (berdasarkan regex /^[^\s@]+@[^\s@]+\.[^\s@]+$/), pesan error "Format email tidak valid" ditampilkan.   + Password:   + Jika panjang password kurang dari 8 karakter, pesan error "Password harus minimal 8 karakter" muncul. |