|  |  |  |
| --- | --- | --- |
| Mata Kuliah | **:** | Desain dan Pemrograman Web |
| Program Studi | **:** | D4 – Teknik Informatika |
| Semester | **:** | 3 |
| Kelas | **:** | TI-2D |
| NIM | **:** | 244107020086 |
| Nama | **:** | Zacky Rio Orlando |
| Jobsheet Ke- | **:** | 7 |

**Laporan Jobsheet**

**Praktikum 1. Function Isset**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 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)    Pada program isset.php, variabel $umur belum diisi sehingga isset($umur) bernilai false, dan muncul pesan “Anda belum dewasa atau variabel 'umur' tidak ditemukan.” Jadi, isset() berguna untuk memastikan variabel sudah ada sebelum digunakan agar terhindar dari error. |
| 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 program isset.php, elemen $data["nama"] sudah didefinisikan dan berisi “Jane”, sehingga program menampilkan “Nama: Jane”. Fungsi ini berguna untuk memastikan data tersedia sebelum digunakan agar tidak terjadi error, dan hasil ditampilkan terpisah antarbaris dengan menambahkan echo "<br>";. |

**Praktikum 2. Function empty**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 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)    Pada program empty.php, array $myArray tidak memiliki data, sehingga kondisi empty($myArray) bernilai **true** dan menampilkan pesan “Array tidak terdefinisi atau kosong.” Jadi, empty() berguna untuk memastikan suatu variabel atau array berisi nilai sebelum digunakan agar program berjalan dengan benar. |
| 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)    Pada program empty.php, variabel $nonExistentVar belum didefinisikan, sehingga empty($nonExistentVar) bernilai **true** dan menampilkan pesan “Variabel tidak terdefinisi atau kosong.” Jadi, empty() berguna untuk mengecek variabel yang belum ada atau bernilai kosong agar program tidak menimbulkan error. |

**Praktikum Bagian 3 : Form Input PHP**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 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)      Pada program form.php, pengguna mengisi nama dan email, lalu data dikirim ke proses\_form.php menggunakan metode **POST**. File proses\_form.php menerima data tersebut melalui $\_POST["nama"] dan $\_POST["email"], kemudian menampilkannya kembali di halaman web. Jadi, form berfungsi sebagai sarana input data dari pengguna yang kemudian diproses oleh PHP di server. |
| 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)    Pada program form\_self.php, form digunakan untuk mengirim dan memproses data dalam satu file yang sama. Saat pengguna menekan tombol **Submit**, data dikirim ke halaman itu sendiri menggunakan $\_SERVER["PHP\_SELF"]. Program kemudian memeriksa apakah kolom nama kosong menggunakan fungsi empty(). Jika kosong, muncul pesan error “Nama harus diisi!”, dan jika terisi, tampil pesan “Data berhasil disimpan!”. |

**Praktikum 4 : HTML Injection**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 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)  Jawaban:  Pada program html\_aman.php, input dari pengguna diproses menggunakan fungsi htmlspecialchars() agar lebih aman. Saat pengguna mengirimkan data, program akan menampilkan kembali input tersebut tanpa menjalankan tag HTML di dalamnya. |
| 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)  Jawaban:  Pada program html\_aman.php, fungsi filter\_var() digunakan untuk memastikan bahwa input benar-benar berformat email yang valid. Jika format salah, akan muncul pesan error. |

**Praktikum 5 : Penggunaan Regex Pada PHP**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 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 ini menunjukkan cara menggunakan preg\_match() dengan regex [a-z] untuk mendeteksi huruf kecil dalam teks. Jika ditemukan, muncul pesan "Huruf kecil ditemukan!", jika tidak, "Tidak ada huruf kecil!". |
| 5 | Ketikkan kode tambahan pada langkah 6 di dalam regex.php |
| 6 |  |
| 7 | Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php |
| 8 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.2)  Jawaban:  Kode tambahan ini menunjukkan cara menggunakan preg\_match() dengan pola regex [0-9]+ untuk menemukan satu atau lebih digit dalam sebuah string. Jika cocok, angka pertama yang ditemukan ditampilkan melalui $matches[0]; jika tidak, muncul pesan "Tidak ada yang cocok!". |
| 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 menunjukkan cara menggunakan preg\_replace() untuk mengganti kata dalam string berdasarkan pola regex. Dalam contoh tersebut, kata "apple" diganti dengan "banana", menghasilkan output "I like banana pie." |
| 13 | Ketikkan kode tambahan pada langkah 14 di dalam regex.php |
| 14 |  |
| 15 | Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php |
| 16 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.4)  Jawaban:  Kode ini menunjukkan penggunaan regex /go\*d/ dalam preg\_match() untuk mencocokkan kata seperti "god", "good", atau "goooood" dalam string. Karakter \* memungkinkan huruf "o" muncul nol atau lebih kali di antara "g" dan "d". Hasil cocok ditampilkan melalui $matches[0], berguna untuk mengenali pola kata yang fleksibel dalam teks. |
| 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 regex /go?d/ untuk mencocokkan kata "god" atau "gd", karena tanda ? membuat huruf "o" bersifat opsional (boleh ada satu atau tidak sama sekali). Hasil cocok ditampilkan melalui $matches[0], berguna untuk mengenali variasi kata dengan karakter opsional. |
| 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 regex /go{2,4}d/ untuk mencocokkan kata yang memiliki huruf "o" sebanyak 2 hingga 4 kali di antara "g" dan "d", seperti "good", "goood", atau "goooood". Ini berguna untuk mengenali pola dengan jumlah karakter yang dikontrol secara spesifik dalam teks |

**Praktikum 6 : Form Lanjut**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 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)  Jawaban:  Pada program **form\_lanjut.php** dan **proses\_lanjut.php**, data yang dikirim dari form berhasil diproses dan ditampilkan kembali menggunakan metode POST. Setiap input seperti pilihan buah, warna favorit, dan jenis kelamin dapat ditangkap dan ditampilkan sesuai dengan pilihan pengguna. Program juga menampilkan pesan jika pengguna tidak memilih salah satu opsi. |
| 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)    Jawaban:  form form\_ajax.php dikirim ke server tanpa me-refresh halaman. Saat tombol submit ditekan, jQuery mencegah pengiriman default, mengumpulkan data form, lalu mengirimkannya ke proses\_lanjut.php menggunakan AJAX. Hasil respon dari server langsung ditampilkan di elemen , membuat interaksi lebih cepat dan responsif. Ini meningkatkan pengalaman pengguna karena tidak perlu reload halaman untuk melihat hasil input. |

**Praktikum 7 : Validasi Form**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 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)  Jawaban:      Kode PHP yang ditambahkan berfungsi untuk memvalidasi input dari form sebelum diproses lebih lanjut. Saat pengguna mengirim data, script memeriksa apakah kolom nama dan email sudah diisi, serta memastikan format email benar menggunakan fungsi bawaan PHP. Jika ada kesalahan, pesan akan ditampilkan langsung di halaman; jika tidak ada, data dianggap valid dan ditampilkan sebagai konfirmasi. |
| 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)  Jawaban:  Kode yang ditambahkan berfungsi untuk memvalidasi input form secara langsung di browser sebelum data dikirim ke server. Dengan bantuan jQuery, sistem akan mengecek apakah kolom nama dan email sudah diisi. Jika ada yang kosong, pesan kesalahan akan muncul di bawah kolom tersebut dan pengiriman form akan dibatalkan. |
| 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)  Jawaban:      Penambahan kode menggunakan AJAX membuat proses pengiriman form lebih interaktif dan efisien karena tidak perlu reload halaman. Validasi tetap dilakukan sebelum pengiriman, dan jika lolos, data dikirim ke server |
| 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:          Penambahan kode validasi password di sisi server memastikan bahwa data yang dikirim dari form benar-benar memenuhi syarat keamanan, yaitu password minimal 8 karakter. Validasi dilakukan bersama dengan pengecekan nama dan email, sehingga jika ada kesalahan, semua pesan ditampilkan sekaligus. Ini membantu pengguna memperbaiki input dengan cepat dan mencegah data tidak valid masuk ke sistem. |