

LAPORAN LATIHAN JOBSHEET 7

PEMOGRAMAN WEB

PHP - Form Proccessing



Disusun Oleh:

Najwa Kus Syafira

(244107060034)

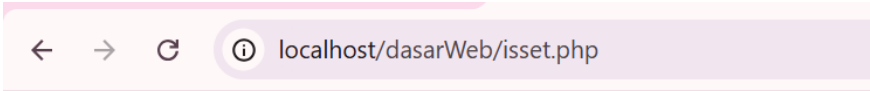
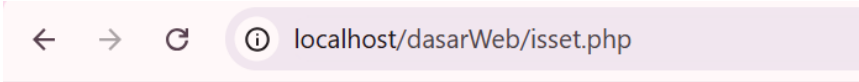
POLITEKNIK NEGERI MALANG

JURUSAN TEKNOLOGI INFORMASI

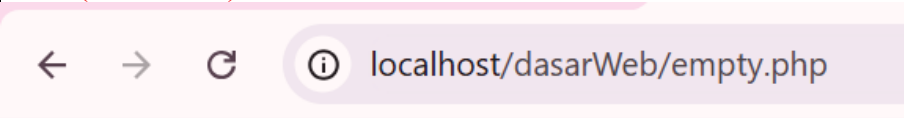
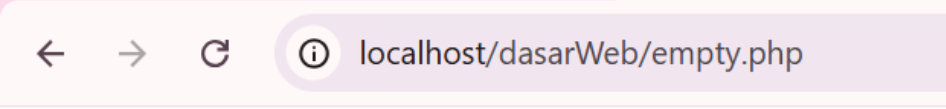
PROGRAM STUDI D4 SISTEM INFOMASI BISNIS

2024

Praktikum Bagian 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	<pre>\$umur; if (isset(\$umur) && \$umur >= 18) { echo "Anda sudah dewasa."; } else { echo "Anda belum dewasa atau variabel 'umur' tidak ditemukan."; }</pre>
4	<p>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)</p>  <p>Anda belum dewasa atau variabel 'umur' tidak ditemukan.</p> <p>Fungsi isset() digunakan untuk mengecek apakah sebuah variabel sudah ada dan berisi nilai. Pada percobaan ini, variabel \$umur belum diberi nilai sehingga isset(\$umur) bernilai false, maka muncul pesan “Anda belum dewasa atau variabel 'umur' tidak ditemukan. Kalau \$umur diisi, misalnya 20, hasilnya akan berubah jadi “Anda sudah dewasa.”</p>
5	<p>Tambahkan isi dari file isset.php tersebut dengan kode di bawah ini.</p> <pre>\$data = array("nama" => "Jane", "usia" => 25); if (isset(\$data["nama"])) { echo "Nama: " . \$data["nama"]; } else { echo "Variabel 'nama' tidak ditemukan dalam array."; }</pre>
6	<p>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)</p>  <p>Anda belum dewasa atau variabel 'umur' tidak ditemukan. Nama: Jane</p> <p>Dari percobaan kedua, saya tahu kalau isset() juga bisa dipakai untuk mengecek apakah suatu elemen ada di dalam array. Selain itu, supaya hasilnya tidak menempel jadi satu baris, saya menambahkan echo "
"; agar tampil di baris baru. Jadi, isset() berguna untuk memastikan data ada sebelum digunakan, sedangkan
 membuat tampilan hasilnya lebih rapi.</p>

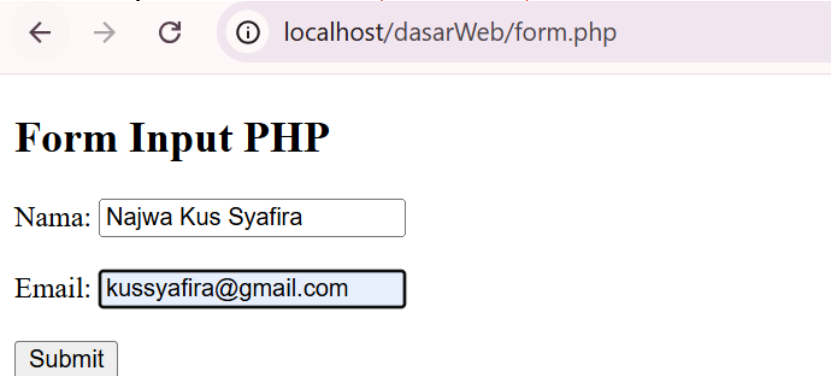
Praktikum 2 : Function empty

Langkah	Keterangan
1	Buat satu file baru bernama empty.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam empty.php
2	<pre>\$myArray = array(); // Array kosong if (empty(\$myArray)) { echo "Array tidak terdefinisi atau kosong."; } else { echo "Array terdefinisi dan tidak kosong."; }</pre>
3	<p>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)</p>  <p>Array tidak terdefinisi atau kosong.</p> <p>empty() mengecek apakah suatu variabel “kosong”. Karena \$myArray adalah array kosong, empty(\$myArray) bernilai true, jadi yang muncul “Array tidak terdefinisi atau kosong.” Intinya, empty() aman dipakai buat ngecek isi tanpa bikin error.</p>
4	<p>Tambahkan isi dari file empty.php tersebut dengan kode di bawah ini.</p> <pre>if (empty(\$nonExistentVar)) { echo "Variabel tidak terdefinisi atau kosong."; } else { echo "Variabel terdefinisi dan tidak kosong."; }</pre>
5	<p>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)</p>  <p>Array tidak terdefinisi atau kosong. Variabel tidak terdefinisi atau kosong.</p> <p>empty(\$nonExistentVar) tetap true meski variabelnya belum pernah dibuat, jadi tidak memicu error dan menampilkan “Variabel tidak terdefinisi atau kosong.” Penambahan
 bikin hasil dua echo tampil di baris terpisah sehingga lebih rapi dibaca.</p>

Praktikum 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	<pre><!DOCTYPE html> <html> <head> <title>Form Input PHP</title> </head> <body> <h2>Form Input PHP</h2> <form method="post" action="proses_form.php"> <label for="nama">Nama:</label> <input type="text" name="nama" id="nama" required>

 <label for="email">Email:</label> <input type="email" name="email" id="email" required>

 <input type="submit" name="submit" value="Submit"> </form> </body> </html></pre>
3	Buat satu file baru bernama proses_form.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam proses_form.php
4	<pre><?php if (\$_SERVER["REQUEST_METHOD"] == "POST") { \$nama = \$_POST["nama"]; \$email = \$_POST["email"]; echo "Nama: " . \$nama . "
"; echo "Email: " . \$email; } ?></pre> <p>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)</p>  <p>The screenshot shows a web browser window with the address bar displaying 'localhost/dasarWeb/form.php'. The page title is 'Form Input PHP'. The form contains two input fields: 'Nama:' with the value 'Najwa Kus Syafira' and 'Email:' with the value 'kussyafira@gmail.com'. Below the fields is a 'Submit' button.</p>

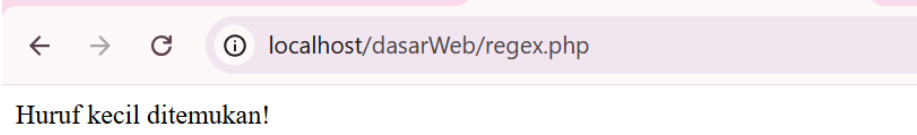
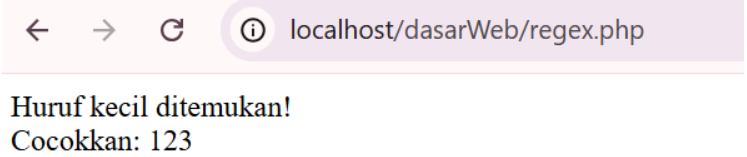
	<div>← → ↻ ⓘ localhost/dasarWeb/proses_form.php</div> <p>Nama: Najwa Kus Syafira Email: kussyafira@gmail.com</p> <p>Form di form.php mengirim data pakai metode POST ke proses_form.php. Di file proses, data diambil lewat \$_POST["nama"] dan \$_POST["email"], lalu dicetak ke layar (dipisah
). Intinya: form → kirim POST → file proses baca \$_POST → tampilkan hasil.</p>
5	<p>Buat satu file baru bernama form_self.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form_self.php</p> <pre><!DOCTYPE html> <html> <head> <title>Form Input PHP</title> </head> <body> <h2>Form Input PHP</h2> <?php // Inisialisasi variabel \$namaErr = ""; \$nama = ""; // Cek apakah form sudah disubmit if (\$_SERVER["REQUEST_METHOD"] == "POST") { // Validasi nama (contoh: pastikan nama tidak kosong) if (empty(\$_POST["nama"])) { \$namaErr = "Nama harus diisi!"; } else { \$nama = \$_POST["nama"]; echo "Data berhasil disimpan!"; } } ?> <form method="post" action="<?php echo htmlspecialchars(\$_SERVER["PHP_SELF"]); ?>"> <label for="nama">Nama:</label> <input type="text" name="nama" id="nama" value="<?php echo \$nama; ?>"> <?php echo \$namaErr; ?>

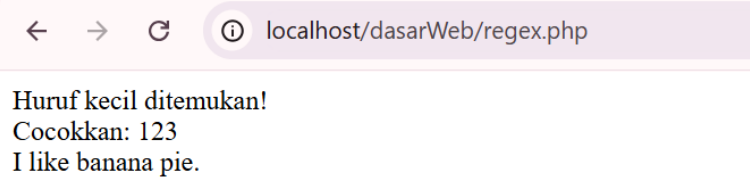
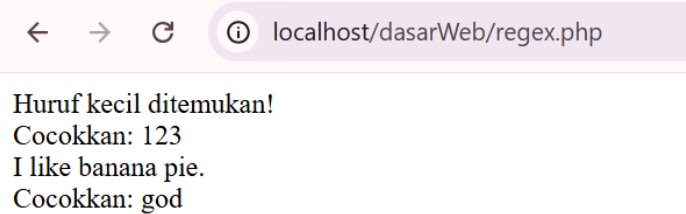
 <input type="submit" name="submit" value="Submit"> </form> </body> </html></pre>
6	<p>Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/ form_self.php. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 3.2)</p> <div>← → ↻ ⓘ localhost/dasarWeb/form_self.php.</div> <h3>Form Input PHP</h3> <p>Nama: <input type="text" value="Najwa Kus Syafira"/></p> <p><input type="button" value="Submit"/></p> <p>Pada form_self.php, form dikirim dan diproses di halaman yang sama (action ke \$_SERVER["PHP_SELF"]). Saat tombol Submit ditekan, blok if (\$_SERVER["REQUEST_METHOD"]=="POST") dijalankan: kalau nama kosong muncul pesan “Nama harus diisi!”, kalau diisi akan mengecho “Data berhasil disimpan!”. Nilai input juga tetap tertulis (sticky), jadi enak buat koreksi tanpa pindah halaman. Intinya: self-submit + validasi sederhana + pesan langsung ditampilkan sesuai hasil.</p>

Praktikum 4: HTML Injection

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	<pre>\$input = \$_POST['input']; \$input = htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8');</pre>
3	Tambahkan script lainnya supaya langkah 2 bisa berjalan normal. Lalu simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/html_aman.php
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1)</p> <pre> 1 <?php 2 if (\$_SERVER['REQUEST_METHOD'] === 'POST') { 3 \$input = \$_POST['input'] ?? ''; 4 \$safe = htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8'); 5 } 6 7 <!DOCTYPE html> 8 <html> 9 <head> 10 <meta charset="utf-8"> 11 <title>Input Aman </title> 12 </head> 13 <body> 14 <h3>Demo Input Aman </h3> 15 16 <form method="post" action=""> 17 <textarea name="input" rows="4" cols="50" placeholder="Coba halo atau <script>alert(1)</script>"></textarea> 18 <button type="submit">Kirim</button> 19 </form> 20 21 <?php if (isset(\$safe)): ?> 22 <p>Output aman: <?php echo \$safe; ?></p> 23 <?php endif; ?> 24 </body> 25 </html> </pre> <p>Pada program, input dari user disimpan ke \$input, lalu disanitasi dengan \$safe = htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8'); sebelum ditampilkan. Akibatnya, karakter khusus seperti <, >, &, dan tanda kutip diubah jadi entity HTML. Jadi kalau saya kirim halo atau <script>alert(1)</script>, yang tampil hanyalah teks literal (tidak dieksekusi). Intinya, output jadi aman dari HTML/JS injection (XSS) karena kita menampilkan versi yang sudah disanitasi, bukan tag aslinya.</p>
5	Ketikkan kode tambahan pada langkah 6 di dalam html_aman.php
6	<pre>// Memeriksa apakah input adalah email yang valid \$email = \$_POST['email']; if (filter_var(\$email, FILTER_VALIDATE_EMAIL)) { // Lanjutkan dengan pengolahan email yang aman } else { // Tangani input yang tidak valid }</pre>
7	Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/html_aman.php
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.2)</p> <p>Pada langkah 5–6, saya menambahkan input email dan memeriksa formatnya memakai filter_var(\$email, FILTER_VALIDATE_EMAIL). Jika format benar (mis. nama@domain.com), program menampilkan “Email valid: ...” (email juga di-escape dengan htmlspecialchars() saat ditampilkan). Jika formatnya salah—termasuk dikosongi—program menampilkan “Email tidak valid!”. Intinya, langkah ini memastikan email berformat sesuai standar sebelum dipakai/ditampilkan, sehingga input lebih rapi dan aman (tidak raw HTML/JS).</p>

Praktikum 5: Penggunaan Regex Pada PHP

Langkah	Keterangan
1	Buat satu file baru bernama regex.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam regex.php
2	<pre> \$pattern = '/[a-z]/'; // Cocokkan huruf kecil. \$text = 'This is a Sample Text.'; if (preg_match(\$pattern, \$text)) { echo "Huruf kecil ditemukan!"; } else { echo "Tidak ada huruf kecil!"; } </pre>
3	<p>Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/ regex.php</p>  <p>Huruf kecil ditemukan!</p>
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.1)</p> <p>Pada langkah ini, pola /[a-z]/ digunakan untuk mencari huruf kecil di dalam teks. Karena pada kalimat “This is a Sample Text.” terdapat huruf kecil, maka program menampilkan pesan “Huruf kecil ditemukan!”. Artinya regex ini berhasil mendeteksi adanya huruf kecil.</p>
5	Ketikkan kode tambahan pada langkah 6 di dalam regex.php
6	<pre> \$pattern = '/[0-9]+/'; // Cocokkan satu atau lebih digit. \$text = 'There are 123 apples.'; if (preg_match(\$pattern, \$text, \$matches)) { echo "Cocokkan: " . \$matches[0]; } else { echo "Tidak ada yang cocok!"; } </pre>
7	<p>Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php</p>  <p>Huruf kecil ditemukan! Cocokkan: 123</p>

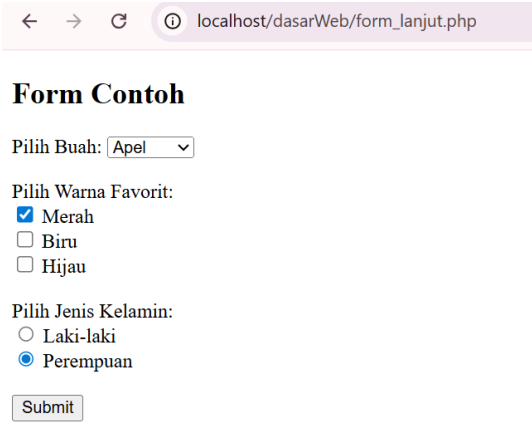
8	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.2) Regex /[0-9]+/ berfungsi mencari satu atau lebih angka di dalam teks. Dalam kalimat “There are 123 apples.” ditemukan angka 123, sehingga muncul hasil “Cocokkan: 123”. Ini menunjukkan bahwa regex bisa digunakan untuk menemukan pola angka dalam teks.
9	Ketikkan kode tambahan pada langkah 10 di dalam regex.php
10	<pre> \$pattern = '/apple/'; \$replacement = 'banana'; \$text = 'I like apple pie.'; \$new_text = preg_replace(\$pattern, \$replacement, \$text); echo \$new_text; // Output: "I like banana pie." </pre>
11	<p>Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php</p>  <p>Huruf kecil ditemukan! Cocokkan: 123 I like banana pie.</p>
12	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.3) Pola /apple/ pada fungsi preg_replace() menggantikan kata <i>apple</i> dengan <i>banana</i>. Jadi, teks “I like apple pie.” berubah menjadi “I like banana pie.”. Dari sini terlihat bahwa regex juga bisa digunakan untuk mengganti kata dalam sebuah kalimat.
13	Ketikkan kode tambahan pada langkah 14 di dalam regex.php
14	<pre> \$pattern = '/go*d/'; // Cocokkan "god", "good", "goood", dll. \$text = 'god is good.'; if (preg_match(\$pattern, \$text, \$matches)) { echo "Cocokkan: " . \$matches[0]; } else { echo "Tidak ada yang cocok!"; } </pre>
15	<p>Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php</p>  <p>Huruf kecil ditemukan! Cocokkan: 123 I like banana pie. Cocokkan: god</p>
16	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.4) Regex /go*d/ berarti huruf o bisa muncul nol kali atau lebih. Jadi cocok dengan kata seperti “gd”, “god”, “good”, atau “goood”. Pada teks “god is good.”, hasil yang cocok pertama adalah “god”.
17	<p>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)</p> <p>Jika polanya diubah menjadi /go?d/, maka o hanya boleh muncul 0 atau 1 kali, jadi cocok untuk “gd” atau “god”, tapi tidak untuk “good”. Hasil pada teks yang sama adalah “god”.</p>
18	<p>Soal 5.6: Buatlah script langkah 14 dengan variable pattern yang di ubah menggunakan `{n,m}`.</p> <p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.6)</p> <p>Regex /go{2,}d/ berarti huruf o harus muncul minimal dua kali. Jadi cocok untuk “good”, “goood”, dan seterusnya. Pada teks “god is good.” hasil yang cocok adalah “good”.</p>

Praktikum 6: Form Lanjut

Langkah	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	<pre> <!DOCTYPE html> <html> <head> <title>Contoh Form dengan PHP</title> </head> <body> <h2>Form Contoh</h2> <form method="POST" action="proses_lanjut.php"> <label for="buah">Pilih Buah:</label> <select name="buah" id="buah"> <option value="apel">Apel</option> <option value="pisang">Pisang</option> <option value="mangga">Mangga</option> <option value="jeruk">Jeruk</option> </select>
 <label>Pilih Warna Favorit:</label>
 <input type="checkbox" name="warna[]" value="merah"> Merah
 <input type="checkbox" name="warna[]" value="biru"> Biru
 <input type="checkbox" name="warna[]" value="hijau"> Hijau

 <label>Pilih Jenis Kelamin:</label>
 <input type="radio" name="jenis_kelamin" value="laki-laki"> Laki-laki
 <input type="radio" name="jenis_kelamin" value="perempuan"> Perempuan

 <input type="submit" value="Submit"> </form> </body> </html> if (\$_SERVER["REQUEST_METHOD"] == "POST") { \$selectedBuah = \$_POST['buah']; if (isset(\$_POST['warna'])) { \$selectedWarna = \$_POST['warna']; } else { \$selectedWarna = []; } \$selectedJenisKelamin = \$_POST['jenis_kelamin']; echo "Anda memilih buah: " . \$selectedBuah . "
"; if (!empty(\$selectedWarna)) { echo "Warna favorit Anda: " . implode(", ", \$selectedWarna) . "
"; } else { echo "Anda tidak memilih warna favorit.
"; } echo "Jenis kelamin Anda: " . \$selectedJenisKelamin; } </pre>

3	<p>Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/ form_lanjut.php</p> 
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 6.1)</p> <p>Form form_lanjut.php mengirim data buah, warna[], dan jenis_kelamin ke proses_lanjut.php. Di server, data dibaca dari \$_POST, checkbox diproses sebagai array, lalu hasil ditampilkan (buah yang dipilih, daftar warna—atau pesan jika kosong—dan jenis kelamin).</p>
5	<p>Buat satu file baru bernama form_ajax.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form ajax.php</p> <pre data-bbox="252 880 1096 1872"><!DOCTYPE html> <html> <head> <title>Contoh Form dengan PHP dan jQuery</title> <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script> </head> <body> <h2>Form Contoh</h2> <form id="myForm"> <label for="buah">Pilih Buah:</label> <select name="buah" id="buah"> <option value="apel">Apel</option> <option value="pisang">Pisang</option> <option value="mangga">Mangga</option> <option value="jeruk">Jeruk</option> </select>
 <label>Pilih Warna Favorit:</label>
 <input type="checkbox" name="warna[]" value="merah"> Merah
 <input type="checkbox" name="warna[]" value="biru"> Biru
 <input type="checkbox" name="warna[]" value="hijau"> Hijau


 <label>Pilih Jenis Kelamin:</label>
 <input type="radio" name="jenis_kelamin" value="laki-laki"> Laki-laki
 <input type="radio" name="jenis_kelamin" value="perempuan"> Perempuan

 <input type="submit" value="Submit"> </form> <div id="hasil"> <!-- Hasil akan ditampilkan di sini --> </div> <script> \$(document).ready(function () { \$("#myForm").submit(function (e) { e.preventDefault(); // Mencegah pengiriman form secara default // Mengumpulkan data form var formData = \$("#myForm").serialize(); // Kirim data ke server PHP \$.ajax({ url: "proses_lanjut.php", // Ganti dengan nama file PHP yang sesuai type: "POST", data: formData, success: function (response) { // Tampilkan hasil dari server di div "hasil" \$("#hasil").html(response); } }); }); }); </script> </body> </html></pre>
6	<p>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)</p> <p>Pada form_ajax.php, pengiriman form dicegat oleh jQuery (preventDefault()), data diserialisasi dan dikirim via AJAX ke proses_lanjut.php. Halaman tidak reload; respons server langsung ditampilkan di <div id="hasil">.</p>

Praktikum 7: Validasi Form

Langkah	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	<pre> <!DOCTYPE html> <html> <head> <title>Form Input dengan Validasi</title> </head> <body> <h1>Form Input dengan Validasi</h1> <form method="post" action="proses_validasi.php"> <label for="nama">Nama:</label> <input type="text" id="nama" name="nama">
 <label for="email">Email:</label> <input type="text" id="email" name="email">
 <input type="submit" value="Submit"> </form> </body> </html> <?php if (\$_SERVER["REQUEST_METHOD"] == "POST") { \$nama = \$_POST["nama"]; \$email = \$_POST["email"]; \$errors = array(); // Validasi Nama if (empty(\$nama)) { \$errors[] = "Nama harus diisi."; } // Validasi Email if (empty(\$email)) { \$errors[] = "Email harus diisi."; } elseif (filter_var(\$email, FILTER_VALIDATE_EMAIL)) { \$errors[] = "Format email tidak valid."; } // Jika ada kesalahan validasi if (empty(\$errors)) { (foreach (\$errors as \$error) { echo \$error . "
"; }) } else { // Lanjutkan dengan pemrosesan data jika semua validasi berhasil // Misalnya, menyimpan data ke database atau mengirim email echo "Data berhasil dikirim: Nama = \$nama, Email = \$email"; } } ?> </pre>
3	Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_validasi.php

	<div> <div> ← → ↻ localhost/dasarWeb/form_validasi.php </div> <div> <h2>Form Input dengan Validasi</h2> <p>Nama: <input type="text" value="Najwa"/></p> <p>Email: <input type="text" value="kussyafira@gmail.com"/></p> <p><input type="button" value="Submit"/></p> </div> </div> <div> <div> ← → ↻ localhost/dasarWeb/proses_validasi.php </div> <p>Data berhasil dikirim: Nama = Najwa, Email = kussyafira@gmail.com</p> </div>
4	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.1)</p> <p>Disini saya membuat file lagi bernama proses_validasi.php. Saat form disubmit, datanya dikirim ke proses_validasi.php. Di sana PHP ngecek \$_POST['nama'] dan \$_POST['email']: kalau nama kosong muncul “Nama harus diisi.”, kalau email kosong “Email harus diisi.”, dan kalau format email nggak valid muncul “Format email tidak valid.”. Kalau semua lolos, tampil “Data berhasil dikirim: Nama =Najwa, Email = Kussyafira@gmail.com”. Jadi, validasi terjadi di sisi server, memastikan data tetap dicek sebelum diproses.</p>
5	<p>Kembangkan file bernama form_validasi.php di dalam folder dasarWeb. Ketikkan kode pada langkah 6 di dalam form_validasi.php</p>
6	<pre> <!DOCTYPE html> <html> <head> <title>Form Input dengan Validasi</title> <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script> </head> <body> <h1>Form Input dengan Validasi</h1> <form id="myForm" method="post" action="proses_validasi.php"> <label for="nama">Nama:</label> <input type="text" id="nama" name="nama">
 <label for="email">Email:</label> <input type="text" id="email" name="email">
 <input type="submit" value="Submit"> </form> <script> \$(document).ready(function() { \$("#myForm").submit(function(event) { var nama = \$("#nama").val(); var email = \$("#email").val(); var valid = true; if (nama === "") { \$("#nama-error").text("Nama harus diisi."); valid = false; } else { \$("#nama-error").text(""); } if (email === "") { \$("#email-error").text("Email harus diisi."); valid = false; } else { \$("#email-error").text(""); } if (valid) { (!event.preventDefault()); // Menghentikan pengiriman form jika validasi gagal } }); }); </script> </body> </html> </pre>

7	<p>Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_validasi.php</p>
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.2)</p> <p>Setelah ditambah jQuery (langkah 5–6), form nge-cek input sebelum dikirim. Kalau nama/email kosong, muncul tulisan merah “harus diisi” di samping field dan submit dibatalkan (event.preventDefault()), jadi halaman nggak reload. Begitu kolom diisi, pesan error hilang dan form baru dikirim ke proses_validasi.php. Intinya: ada validasi client-side yang kasih feedback cepat tanpa pindah halaman, tapi tetap perlu validasi server-side untuk keamanan.</p>
9	<p>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)</p>  <p>Form Input dengan Validasi</p> <p>Nama: <input type="text" value="Najwa"/> Email: <input type="text" value="kussyafira@gmail.com"/> <input type="button" value="Submit"/></p> <p>Data valid. Nama = Najwa, Email = kussyafira@gmail.com</p> <p>Dengan AJAX, form tidak reload. Data dikirim ke proses_validasi.php, dan balasan (error/sukses) langsung ditampilkan di <div id="hasil">.</p>
10	<p>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)</p> <p>Pada langkah 7.4 saya menambah field password dengan aturan minimal 8 karakter. Di sisi klien pakai jQuery: kalau panjangnya kurang dari 8, muncul pesan merah “Password minimal 8 karakter” dan form nggak boleh dikirim. Begitu memenuhi, pesannya hilang dan form bisa lanjut. Di sisi server pakai PHP dicek lagi dengan strlen(\$password) < 8; kalau masih kurang, server balikan error, kalau sudah oke bareng input lain maka tampil pesan sukses. Intinya ada dua lapis validasi: cepat buat pengguna (UX jalan) dan tetap aman karena server tetap memverifikasi kalau ada yang coba bypass.</p>