



Mata Kuliah : Desain Pemrograman Website
Program Studi : D4 – Teknik Informatika
Semester : 3

Kelas : 2D
NIM : 244107020150
Nama : Gaduh Prakoso
Jobsheet Ke- : 7

Laporan Jobsheet

Praktikum Ke-1

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	<pre><?php \$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>isset() adalah fungsi bawaan PHP yang berfungsi untuk memeriksa apakah sebuah variabel sudah diset (telah memiliki nilai) dan apakah nilainya bukan NULL</p>



5	<pre><code>\$data = array("nama" => "Jane", "usia" => 25); if (isset(\$data["nama"])) { // Kunci "nama" ada dan diset nilainya. echo "Nama: " . \$data["nama"]; } else { // ... }</code></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>isset() adalah fungsi pengecek di PHP yang memastikan sebuah variabel atau elemen array telah diset (diberi nilai) dan nilainya bukan NULL. Dalam kode di atas, untuk \$umur; yang tidak diberi nilai, isset() mengembalikan FALSE sehingga eksekusi masuk ke blok else dan mencetak pesan "variabel 'umur' tidak ditemukan." Sementara itu, pada array \$data, isset(\$data["nama"]) mengembalikan TRUE karena kunci "nama" ada dan memiliki nilai ("Jane"), sehingga kode aman untuk mencetak nama tersebut.</p>

Praktikum Ke-2

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	<pre><code>\$MyArray = array(); if (empty(\$MyArray)) { echo "Array tidak terdefinisi atau kosong.
"; } else { echo "Array terdefinisi dan tidak kosong.
"; }</code></pre>
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)



	<p>Array tidak terdefinisi atau kosong.</p> <p>\$MyArray diinisialisasi sebagai array kosong (array()). Ketika empty(\$MyArray) dijalankan, fungsi ini mengembalikan TRUE karena array tersebut tidak memiliki elemen di dalamnya. Hasilnya, kode masuk ke blok if dan mencetak "Array tidak terdefinisi atau kosong."</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>\$NonExistentVar belum pernah dideklarasikan, PHP menganggapnya sebagai NULL. Nilai NULL dianggap "kosong" oleh empty(), sehingga fungsi ini mengembalikan TRUE. Hasilnya, kode masuk ke blok if dan mencetak "Variabel tidak terdefinisi atau kosong."</p>

Praktikum Ke-3

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	<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	<p>Buat satu file baru bernama proses_form.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam proses_form.php</p> <pre><?php if (\$_SERVER['REQUEST_METHOD'] == "POST") { \$nama = \$_POST["nama"]; \$email = \$_POST["email"]; echo "Nama: " . \$nama . "
"; echo "Email: " . \$email . "
"; } ?></pre>
4	<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>orm HTML mengumpulkan data dan mengirimkannya melalui metode POST, dan script PHP di proses_form.php menerima data tersebut melalui array superglobal \$_POST untuk kemudian ditampilkan atau diproses.</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 \$namaError = ""; \$nama = ""; if (\$_SERVER["REQUEST_METHOD"] == "POST") { if (empty(\$_POST["nama"])) { \$namaError = "Nama harus diisi!"; } else { \$nama = \$_POST["nama"]; echo "Data berhasil disimpan!"; } } ?> <form method="post" action="<?php echo htmlspecialchars(string: \$_SERVER["PHP_SELF"]); ?>"> <label for="nama">Nama:</label>
 <input type="text" name="nama" id="nama" value="<?php echo \$nama; ?>"> <?php echo \$namaError; ?>

 <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> <p>Form Input PHP</p> <p>Data berhasil disimpan!</p> <p>Nama:</p> <div><input type="text" value="gadu"/></div> <div>Submit</div> <p>File form_self.php memproses datanya sendiri, memungkinkan validasi input secara real-time dan menampilkan pesan error di form yang sama. Pola ini membuat alur kerja pengguna lebih lancar dan efektif untuk validasi client-side dasar.</p>

Praktikum Ke-4

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	<pre>\$input = \$_POST['input']; \$input = htmlspecialchars(string: \$input, flags: ENT_QUOTES, encoding: '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	Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1) htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8'); bertujuan untuk melakukan sanitasi pada input dari pengguna. Ketika input yang mengandung tag HTML (seperti atau <script>) dimasukkan, kode ini mengubah karakter khusus seperti tanda kurung sudut (< dan >) menjadi entitas HTML (< dan >).
5	Ketikkan kode tambahan pada langkah 6 di dalam html_aman.php
6	<pre>\$email = \$_POST['email']; if (filter_var(value: \$email, filter: FILTER_VALIDATE_EMAIL)) { } else {</pre>
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) menggunakan fungsi filter_var(\$email, FILTER_VALIDATE_EMAIL) berfungsi untuk memastikan bahwa data yang dimasukkan ke kolom email memiliki struktur format email yang valid (contoh: nama@domain.tld)

Praktikum Ke-5

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	<pre>\$pattern = '/[a-z]/'; \$text = 'This is a Sample Text.'; if (preg_match(pattern: \$pattern, subject: \$text)) { echo "Huruf kecil ditemukan!"; } else { echo "Tidak ada huruf kecil!"; }</pre>



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) reg_match() digunakan untuk mencocokkan sebuah pola (\$pattern) di dalam sebuah <i>string</i> (\$text). Fungsi ini mengembalikan nilai true (1) jika pola ditemukan dan false (0) jika pola tidak ditemukan.
5	Ketikkan kode tambahan pada langkah 6 di dalam regex.php
6	<pre>\$pattern = '/[0-9]+/'; \$text = 'There are 123 apples.'; if (preg_match(pattern: \$pattern, subject: \$text, matches: &\$matches)) { echo "Cocokkan: " . \$matches[0]; } else { echo "Tidak ada cocok!"; }</pre>
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) preg_match() yang ini menyertakan argumen ketiga, yaitu variabel \$matches. Selain mengembalikan nilai <i>boolean</i> (true), fungsi ini juga mengisi array \$matches dengan hasil kecocokan. Nilai yang cocok dengan seluruh pola (/[0-9]+)/ dapat diakses melalui elemen array \$matches[0]
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: \$pattern, replacement: \$replacement, subject: \$text); echo \$new_text;</pre>
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) preg_replace() berbeda dari preg_match() karena tidak hanya mencari kecocokan, tetapi juga mengganti teks yang cocok. Fungsi ini mengambil tiga argumen utama: pola (\$pattern), nilai pengganti (\$replacement), dan <i>string</i> target (\$text). Hasilnya adalah string baru (\$new_text) di mana semua kejadian dari pola yang cocok telah diganti dengan <i>string</i> pengganti. Dalam kasus ini, kata "apple" diganti menjadi "banana"



13	Ketikkan kode tambahan pada langkah 14 di dalam regex.php
14	<pre>\$pattern = '/go*d/'; \$text = 'god is good.'; if (preg_match(pattern: \$pattern, subject: \$text, matches: &\$amp;matches)) { echo "Cocokkan: " . \$matches[0] . "
"; } else { echo "Tidak ada yang cocok!"; }</pre>
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) quantifier * pada pola /go*d/ berarti karakter 'o' dapat muncul nol kali atau lebih. Dalam string 'god is good.', pola ini berhasil mencocokkan kata "god" dan juga akan mencocokkan kata "good" jika dicari lagi (menggunakan preg_match_all).
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) quantifier ? pada pola /go?d/ berarti karakter 'o' hanya dapat muncul nol atau satu kali. Dalam string 'god is good.', pola berhasil mencocokkan "god" (satu 'o'). Jika string diubah menjadi 'good', pola ini tidak akan cocok karena karakter 'o' muncul dua 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) quantifier {1,2} pada pola /go{1,2}d/ berarti karakter 'o' harus muncul minimal 1 kali dan maksimal 2 kali. Dalam string 'god is good.', pola berhasil mencocokkan "god" (1 'o') dan juga akan mencocokkan "good" (2 'o'). Namun, jika diuji dengan string 'goood' (3 'o'), pola tersebut gagal mencocokkan karena melebihi batas maksimum yang ditetapkan

Praktikum Ke-6

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	<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> <?php 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(separator: ", ", array: \$selectedWarna) . ".
"; } else { echo "Anda tidak memilih warna favorit.
"; } echo "Jenis kelamin Anda: " . \$selectedJenisKelamin . ".
"; }</pre>
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)



	<p>Form Contoh</p> <p>Pilih Buah: <input type="text" value="Apel"/></p> <p>Pilih warna favorit:</p> <p><input type="checkbox"/> Merah</p> <p><input type="checkbox"/> Biru</p> <p><input type="checkbox"/> Hijau</p> <p>Pilih Jenis Kelamin:</p> <p><input type="radio"/> Laki-laki</p> <p><input type="radio"/> Perempuan</p> <p><input type="button" value="Submit"/></p> <p>kode ini berhasil menangani tiga jenis input formulir yang berbeda: 1) select (drop-down) untuk Buah, yang datanya selalu ada; 2) checkbox untuk Warna, yang datanya dikirim sebagai array (warna[]) dan harus diperiksa keberadaannya menggunakan isset() karena pengguna mungkin tidak memilih apa pun; dan 3) radio button untuk Jenis Kelamin, yang datanya selalu ada jika salah satu opsi diwajibkan. Jika warna dipilih, kode menggunakan fungsi implode() untuk menggabungkan elemen array warna menjadi satu string yang dipisahkan koma, siap untuk ditampilkan kepada pengguna.</p>
5	Buat satu file baru bernama form_ajax.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form_ajax.php



	<pre><!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"> </div> <script> \$(document).ready(function () { \$("#myForm").submit(function (e) { e.preventDefault(); var formData = \$("#myForm").serialize(); \$.ajax({ url: "proses_lanjut.php", type: "POST", data: formData, success: function (response) { \$("#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>ketika tombol <i>Submit</i> ditekan, halaman tidak mengalami <i>reload (refresh)</i> karena adanya perintah <code>e.preventDefault()</code>. Sebaliknya, <i>script</i> jQuery mengambil semua data formulir menggunakan <code>\$("#myForm").serialize()</code> dan mengirimkannya di latar belakang ke file <code>proses_lanjut.php</code> melalui permintaan <code>\$.ajax()</code>. Hasil atau <i>response</i> yang dikirim kembali oleh <code>proses_lanjut.php</code> kemudian disuntikkan langsung ke dalam elemen <code>div</code> dengan ID <code>hasil</code> tanpa meninggalkan halaman utama</p>



Praktikum Ke-7

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	<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(); if (empty(\$nama)) { \$errors[] = "Nama harus diisi."; } if (empty(\$email)) { \$errors[] = "Email harus diisi."; } elseif (!filter_var(\$email, FILTER_VALIDATE_EMAIL)) { \$errors[] = "Format email tidak valid."; } if (!empty(\$errors)) { foreach (\$errors as \$error) { echo \$error . "
"; } } else { echo "Data berhasil dikirim: Nama = \$nama, Email = \$email"; } }</pre>
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) Form Input dengan Validasi Nama: <input type="text"/> Email: <input type="text"/> <input type="submit" value="Submit"/>



	<p>Kode program ini adalah implementasi dasar formulir input dengan validasi yang terbagi menjadi HTML dan PHP. Bagian HTML membuat formulir sederhana untuk memasukkan Nama dan Email yang datanya dikirim menggunakan metode POST. Bagian PHP berfungsi sebagai validator; ia memeriksa apakah kolom Nama dan Email sudah diisi dan memastikan Email memiliki format yang valid (menggunakan fungsi filter_var). Jika ditemukan kesalahan (seperti kolom kosong atau format email salah), program akan menampilkan pesan kesalahan tersebut; namun, jika semua data lolos validasi, program akan menampilkan pesan sukses bahwa data berhasil dikirim, mensimulasikan proses lebih lanjut seperti penyimpanan ke database.</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(); } }); }); </script></pre>



7	Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/form_validasi.php
8	<p>Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.2)</p> <p>program ini mengimplementasikan validasi client-side pada formulir input Nama dan Email menggunakan pustaka jQuery. Setelah formulir dimuat, skrip menencegat event submit formulir (#myForm). Di dalam fungsi submit, skrip mengambil nilai input dan melakukan pengecekan dasar (empty check). Jika input Nama atau Email kosong, skrip menetapkan status validasi menjadi false dan menampilkan pesan kesalahan instan (berwarna merah) pada span terkait tanpa perlu memuat ulang halaman. Kunci operasionalnya adalah penggunaan fungsi event.preventDefault(), yang akan dipanggil jika status validasi adalah false, berfungsi untuk menghentikan pengiriman data ke server (proses_validasi.php) sampai pengguna memperbaiki semua kesalahan yang ada.</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>Untuk membuat script formulir menggunakan AJAX, fungsi submit formulir standar (event.preventDefault()) dihentikan. Kemudian, data formulir diambil menggunakan jQuery (form.serialize()) dan dikirim ke server (proses_validasi.php) secara asinkron (di latar belakang) menggunakan metode \$.ajax().</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>Penambahan validasi <i>password</i> minimal 8 karakter dilakukan di dua tempat: menggunakan jQuery untuk validasi sisi klien dan PHP untuk validasi sisi server. Pengamatan utamanya adalah validasi jQuery memberikan umpan balik instan kepada pengguna <i>tanpa</i> memuat ulang halaman, sehingga <i>user experience</i> (UX) menjadi lebih cepat dan baik. Sementara itu, validasi PHP berfungsi sebagai lapisan keamanan terakhir untuk memastikan integritas data, menolak <i>request</i> yang tidak sah meskipun lolos dari JavaScript. Dengan formulir yang menggunakan AJAX, respons <i>error</i> dari PHP (jika validasi server gagal) dapat ditangkap dan ditampilkan secara dinamis, menunjukkan bahwa kedua lapisan validasi bekerja sama secara efektif</p>