Topic

-   The Concept of Form Proccessing with PHP and Jquery

Purpose

Students are expected to:
1. Students are able to create forms using PHP
2. Students are able to create forms using jQuery

Attention

This jobsheet must be done step-by-step according to the Practical Section steps that have been given. Questions can be answered directly in the column provided using the PDF Editor.

Introduction

Function isset

isset() is a function in PHP that is used to check whether a variable has been defined (exists) or not. This function returns **true** if the variable has been defined and has a value, and **false** if the variable does not exist or has a null value. Here's an example script and a minimal explanation of **isset()**:

```php
$nama = "John";
if (isset($nama)) {
    echo "Variabel 'nama' telah didefinisikan.";
} else {
    echo "Variabel 'nama' tidak ditemukan atau bernilai null.";
}
```

The above code checks if **$nama** variable has been defined. If yes, then the message "Variable 'name' has been defined." will be displayed, if not, then the message "Variable 'name' not found or null value." will be displayed.

Practical Section 1. Function isset()

Step	Description
1	Create a new folder **JS07_PHP-jQuery** in **dasarWeb** and name it **isset.php**
2	Type into the **isset.php** file the code below.

| 3 | ```php
$umur;
if (isset($umur) && $umur >= 18) {
    echo "Anda sudah dewasa.";
} else {
    echo "Anda belum dewasa atau variabel 'umur' tidak ditemukan.";
}
``` |
| 4 | Save the file, then open a browser and run `localhost/dasarWeb/JS07_PHP-jQuery/isset.php`. What do you understand from using the isset on the file. Give me your answer. (Question No 1.1)

Fungsi isset() dalam PHP digunakan sebagai pemeriksaan untuk memastikan apakah suatu variabel atau elemen array ada dan tidak null sebelum digunakan. |
| 5 | Add the contents of the `isset.php` file with the code below.

```php
$data = array("nama" => "Jane", "usia" => 25);
if (isset($data["nama"])) {
    echo "Nama: " . $data["nama"];
} else {
    echo "Variabel 'nama' tidak ditemukan dalam array.";
}
``` |
| 6 | Save the file, then open the browser and run `localhost/dasarWeb/JS07_PHP-jQuery/isset.php`. Ensure that the output does not appear in a single line; the result from the `echo` should be displayed separately. Explain what you understand from the use of `isset()` in that file. Write your understanding below. (Question No 1.2)

Penggunaan isset() di sini berfungsi untuk memeriksa apakah kunci (key) "**nama**" ada dan telah didefinisikan dalam array asosiatif $data. |

Function `empty()`

The `empty()` function in PHP is used to check whether a variable is empty or undefined. This function returns `true` if the variable is empty or undefined, and `false` if the variable has a value or has been defined. Here is an explanation and example of using `empty()`:

```php
$nama = ""; // Variabel string kosong
if (empty($nama)) {
    echo "Nama tidak terdefinisi atau kosong.";
} else {
    echo "Nama terdefinisi dan tidak kosong.";
}
```

The `empty()` function can be used to check whether a string is empty or not.

Practical Section 2. Function `empty()`

Step	Description
1	Create a new file named `empty.php` inside `JS07_PHP-jQuery` folder. Write this code into `empty.php` file
| 2 | ```php
$myArray = array(); // Array kosong
if (empty($myArray)) {
    echo "Array tidak terdefinisi atau kosong.";
} else {
    echo "Array terdefinisi dan tidak kosong.";
}
``` |
| 3 | Save the file, then open a browser and run `localhost/dasarWeb/JS07_PHP-jQuery/empty.php`

 What do you understand from the use of empty on the file. Write your understanding below. (Question No 2.1)

 empty($myArray) mengembalikan **true** karena $myArray adalah array **kosong** (array()), dan program mencetak pesan array kosong. |
| 4 | Add the contents of  the `empty.php` file  with the code below <br><br> ```php
if (empty($nonExistentVar)) {
    echo "Variabel tidak terdefinisi atau kosong.";
} else {
    echo "Variabel terdefinisi dan tidak kosong.";
}
``` |
| 5 | Save the file, then open the browser and run `localhost/dasarWeb/JS07_PHP-jQuery/empty.php`. Ensure that the output does not appear in a single line; the result from the `echo` should be displayed separately. Explain what you understand from the use of `empty()` in that file. Write your understanding below. (Question No 2.2)

 empty($nonExistentVar) mengembalikan **true** karena variabel tersebut **belum pernah didefinisikan** (undefined). empty() dapat memeriksa variabel undefined tanpa error. |

Practical Section Part 3: PHP Input Form

3

Step	Description
1	Create a new file named **form.php** inside JS07_PHP-jQuery folder. Write this code below

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	Create a new file named form_process.php inside JS07_PHP-jQuery folder Type the code in step 2 inside the form_process.php
	
	<pre><?php
	if ($_SERVER["REQUEST_METHOD"] == "POST") {
	    $nama = $_POST["nama"];
	    $email = $_POST["email"];
	
	    echo "Nama: " . $nama . "<br>";
	    echo "Email: " . $email;
	}
	?></pre>

4	Save the file, then open a browser and run localhost/dasarWeb/JS07_PHP-jQuery/form_process.php. What do you understand from the use of forms in the file. Write your understanding below. (Question No. 3.1) Data dikirim dengan method="post" dari form.php dan diterima di form_process.php melalui variabel superglobal **$_POST**.
5	Create a new file named form_self.php inside **JS07_PHP-jQuery** folder Type the code below inside form_self.php

| | ```html
<!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; ?>">
        <span class="error"><?php echo $namaErr; ?></span><br><br>

        <input type="submit" name="submit" value="Submit">
    </form>
</body>
</html>
``` |
| 6 | Save the file, then open a browser and run `localhost/dasarWeb/JS07_PHP-jQuery/ form_self.php`. What do you understand from the use of forms in the file. Write your understanding below. (Question No 3.2)

Form diproses **oleh file itu sendiri** (self-processing) dengan action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>". Terdapat validasi sederhana dan nilai input dipertahankan. |

HTML Injection

HTML injection (also known as an "HTML injection attack" or "client-side injection") is a security attack that occurs when an attacker injects malicious HTML or JavaScript code into the input received by a web application. The malicious code will be executed by the user's browser viewing the affected web page, which could result in illegal access to data, changes in page view, or other attacks.

An attacker can try to inject malicious HTML or JavaScript code into the input received by the web application. If the web application does not properly sanitize or escape input, then the malicious code will be executed by the user's browser, which can cause security issues.

Example HTML Injection:

For example, we have a simple search form on a website that searches for keywords among user reviews:

```html
<form action="search.php" method="post">
    <input type="text" name="keyword">
    <input type="submit" value="Search">
</form>
```

Now, if a web application doesn't validate the input correctly, an attacker can enter malicious input like this:

```
<script>alert('You have been hacked!');</script>
```

If the web application does not avoid or clean up this input before displaying it on the search results page, then the result will look like this:

```
<div>Hasil pencarian untuk "<script>alert('You have been hacked!');</script>":</div>
```

The malicious JavaScript code will be executed by the user's browser and will bring up a "You have been hacked!" warning box. This is a simple example of HTML injection. As a web developer, it is important to always sanitize and validate input from users, as well as avoid displaying user input directly on web pages without proper escaping or sanitation to avoid these kinds of attacks.

How to Handle HTML Injection

The way to handle HTML injection is to validate, filter, and avoid displaying user input without proper escaping. Here are some steps to protect your app from HTML injection:

1. **Input Validation**: Validate input from users to ensure that only valid data is received. You can use functions such as `filter_input()` or `filter_var()` to perform validation.
2. **Filter Input**: Filter user input to remove or replace potentially harmful characters, such as **<, >, &**, and more. You can use functions like `htmlspecialchars()` or `strip_tags()` for this.
3. **Parameterized Statements (Query):** If you generate SQL queries with user input, use parameterized statements or prepared statements to prevent SQL injection, which can be a form of attack similar to HTML injection.
4. **Content Security Policy (CSP):** Apply a Content Security Policy (CSP) to your HTTP headers to control the resources that can be used within your web pages.
5. **Escape Output**: When you display data on a web page, make sure you avoid JavaScript injection by using `htmlspecialchars()` or similar methods.

Here's a simple example in PHP to solve HTML injection:

```php
// Menggunakan filter_input untuk validasi input
$user_input = filter_input(INPUT_POST, 'user_input', FILTER_SANITIZE_STRING);

// Menggunakan htmlspecialchars untuk menghindari HTML injection saat menampilkan data
echo '<div>' . htmlspecialchars($user_input, ENT_QUOTES, 'UTF-8') . '</div>';
```

By using these steps, you can mitigate the risk of HTML injection in your web application. It's always important to validate inputs, clean incoming data, and avoid displaying user data without proper escaping.

Practical Section 4 : HTML Injection

Practical Section Steps:

Step	Description

1	Create a new file named `html_safe.php` inside **JS07_PHP-jQuery** folder. Type the code below inside `html_safe.php`

2	```php
	$input = $_POST['input'];
	$input = htmlspecialchars($input, ENT_QUOTES, 'UTF-8');
	``

3	Add another script so that step 2 can run normally, save the file. Then open a browser and run `localhost/dasarWeb/JS07_PHP-jQuery/html_safe.php`
4	Note here what you observe from the addition of the program code above. (Question No. 4.1) Penggunaan **htmlspecialchars()** untuk **melarikan diri (escape) output**. Ini mencegah browser mengeksekusi script atau kode HTML berbahaya yang dimasukkan pengguna (mitigasi **XSS**).
5	Type the additional code in step 6 inside the `html_safe.php`

6	```php
	// 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
	}
	``

7	Complete the program code above so that the result is neat. Save the file, then open a browser and run /refresh `localhost/dasarWeb/JS07_PHP-jQuery/html_safe.php`
8	Note here what you observe from the addition of the program code above. (Question No. 4.2) Penggunaan **filter_var()** dengan FILTER_VALIDATE_EMAIL untuk **validasi format input**. Memastikan data yang diterima sesuai dengan format yang diharapkan (email).

Regular Expression (Regex)

Regex is a powerful tool for searching, matching, or manipulating text based on specific patterns. You can use it for various purposes such as input validation, text search, text replacement, and more.

Basic Regex Patterns:
-   **Single Character**: Any character will match itself in Regex, except for special characters that must be avoided with escape characters.
    o   Example: The **/a/** pattern will match the letter "a" in the text.
-   **Specific Characters**: You can match specific characters by mentioning them in a pattern.
    o   Example: **The /hello/** pattern will match the text "hello" in the text.

- **Special Characters**: Some special characters in Regex should be avoided with escape characters (\) if you want to match them literally. Examples of special characters: `., *, +, ?, |, [, ], (, )`, `{, }, ^,` `$, \`.
- **Character Set**: You can match characters from a set of characters by using `[ ]`. For example, `/[aeiou]/` will match one of the vowels.
- **Character Range**: You can use **-** in a character set to specify a character range. For example, `/[a-z]/` will match any lowercase letters.
- **Quantifier**: You can set the number of times a character or group of characters should appear beforehand. Example quantifier:
  - `*`: 0 or more
  - `+`: 1 or more
  - `?`: 0 or 1 time
  - `{n}`: Exactly `n` times
  - `{n,}`: At least `n` times
  - `{n,m}`: At least `n` times, maximum `m` times
  -

Examples of Regex Usage:
1. Matching patterns

```php
$pattern = '/hello/';
$text = 'Hello, world!';
if (preg_match($pattern, $text)) {
    echo "Cocokkan ditemukan!";
} else {
    echo "Tidak ada cocokan!";
}
```

2. Match set characters

```php
$pattern = '/[0-9]/'; // Cocokkan satu digit.
$text = 'There are 3 apples.';
if (preg_match($pattern, $text)) {
    echo "Digit ditemukan!";
} else {
    echo "Tidak ada digit!";
}
```

Practical Section 5 : The Use of Regex in PHP

Practical Section Steps:

Step	Description

1	Create a new file named `regex.php` inside **JS07_PHP-jQuery** folder Type the code in step 2 inside the `regex.php`
| 2 | ```php
$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!";
}
``` |
3	Save the file, then open a browser and run /refresh `localhost/dasarWeb/JS07_PHP-jQuery/regex.php`
4	Note here what you observe from the addition of the program code above. (Question No 5.1) **preg_match()** dengan pola / [a-z] / mencari **satu huruf kecil** pertama. Karena string mengandung huruf kecil, ia mengembalikan true.
5	Type the additional code in step 6 inside the `regex.php`
| 6 | ```php
$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!";
}
``` |
7	Complete the program code above so that the result is neat. Save the file, then open a browser and run /refresh `localhost/dasarWeb/JS07_PHP-jQuery/regex.php`
8	Note here what you observe from the addition of the program code above. (Question No. 5.2) Quantifier + pada / [0-9]+ / mencocokkan **satu atau lebih** digit yang berurutan. Hasilnya adalah **"123"**.
9	Type the additional code in step 10 inside the `regex.php`
| 10 | ```php
$pattern = '/apple/';
$replacement = 'banana';
$text = 'I like apple pie.';
$new_text = preg_replace($pattern, $replacement, $text);
echo $new_text; // Output: "I like banana pie."
``` |
| 11 | Complete the program code above so that the result is neat. Save the file, then open a browser and run /refresh `localhost/dasarWeb/JS07_PHP-jQuery/regex.php` |
| 12 | Note here what you observe from the addition of the program code above. (Question No 5.3) |

	Fungsi **preg_replace()** digunakan untuk **mengganti** pola / apple / dengan **"banana"**. Outputnya adalah **"I like banana pie."**.
13	Type the additional code in step 14 inside the `regex.php`
| 14 | ```php
$pattern = '/go*d/'; // Cocokkan "god", "good", "gooood", dll.
$text = 'god is good.';
if (preg_match($pattern, $text, $matches)) {
    echo "Cocokkan: " . $matches[0];
} else {
    echo "Tidak ada yang cocok!";
}
``` |
15	Complete the program code above so that the result is neat. Save the file, then open a browser and run /refresh `localhost/dasarWeb/JS07_PHP-jQuery/regex.php`
16	Note here what you observe from the addition of the program code above. (Question No 5.4) Quantifier ***** pada / go*d / mencocokkan karakter 'o' **0 kali atau lebih**. Kecocokan pertama yang ditemukan adalah **"god"**.
17	Question 5.5: Create a script for step 14 with a variable pattern modified using `'?'` (0 or 1 time). Note here what you observe from the addition of the program code above. (Question No 5.5) Pola / go?d / hanya mencocokkan 'o' **0 atau 1 kali**. Dalam 'god is good.', kecocokan pertama adalah **"god"**.
18	Question 5.6: Create a script for step 14 with a variable pattern modified using `'{n,m}'`. Note here what you observe from the addition of the program code above. (Question No 5.6) Pola / go{2,3}d / mencocokkan 'o' sebanyak **2 hingga 3 kali**. Dalam 'god is good.', kecocokan adalah **"good"**.

Practical Section 6 : Advanced Form

Practical Section Steps:

Step	Description
1	Create a new file named `form_next.php` inside **JS07_PHP-jQuery** folder site. Type the code in step 2 inside the `form_next.php`

2	```html
<!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 . "<br>";

    if (!empty($selectedWarna)) {
        echo "Warna favorit Anda: " . implode(", ", $selectedWarna) . "<br>";
    } else {
        echo "Anda tidak memilih warna favorit.<br>";
    }

    echo "Jenis kelamin Anda: " . $selectedJenisKelamin;
}
``` |
3	Save the file, then open a browser and run /refresh `localhost/dasarWeb/JS07_PHP-jQuery/form_next.php`
4	Note here what you observe from the addition of the program code above. (Question No 6.1) Mendemonstrasikan cara memproses *input* **<select>**, **radio**, dan **checkbox**. *Checkbox* menggunakan name="warna[]" untuk diterima sebagai **array** di PHP, yang kemudian dicetak dengan implode().
5	Create a new file named `form_ajax.php` inside **JS07_PHP-jQuery** folder Type the code in step 2 inside the `form_ajax.php`

```html
<!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>

        <br>

        <label>Pilih Warna Favorit:</label><br>
        <input type="checkbox" name="warna[]" value="merah"> Merah<br>
        <input type="checkbox" name="warna[]" value="biru"> Biru<br>
        <input type="checkbox" name="warna[]" value="hijau"> Hijau<br>

        <br>

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

        <br>

        <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>
```

6	Save the file, then open a browser and run /refresh `localhost/dasarWeb/JS07_PHP-jQuery/form_ajax.php`. Note here what you observe from the addition of the program code above. (Question No. 6.2)

	Pengiriman form dilakukan secara **Asinkron** menggunakan **AJAX (jQuery)**. e.preventDefault() mencegah page reload, serialize() mengumpulkan data, dan hasil respons ditampilkan di halaman tanpa pindah halaman.

Practical Section 7 : Form Validation

Practical Section Steps:

Step	Description
1	Create a new file named `form_validation.php` inside **JS07_PHP-jQuery** folder Type the code below inside `form_validation.php`
2	

```html
<!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">
        <br>

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

        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

13

```php
<?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 . "<br>";
        }
    } 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";
    }
}
?>
```

3	Save the file, then open a browser and run /refresh `localhost/dasarWeb/JS07_PHP-jQuery/form_validation.php`
4	Note here what you observe from the addition of the program code above. (Question No 7.1) Melakukan **Validasi Sisi Server (PHP)**. Error dikumpulkan dalam array $errors. Menggunakan empty() untuk kolom wajib dan filter_var() untuk validasi format email.
5	Develop a file named `form_validation.php` inside **JS07_PHP-jQuery** folder site. Type the code in step 6 inside the `form_validation.php`

14

| 6 | <pre>

```html
<!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">
        <span id="nama-error" style="color: red;"></span><br>

        <label for="email">Email:</label>
        <input type="text" id="email" name="email">
        <span id="email-error" style="color: red;"></span><br>

        <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 | Save the file, then open a browser and run /refresh `localhost/dasarWeb/JS07_PHP-jQuery/form_validation.php` |

| 8 | Note here what you observe from the addition of the program code above. (Question No. 7.2)

Melakukan **Validasi Sisi Klien (jQuery)**. Memberikan umpan balik instan kepada pengguna. event.preventDefault() digunakan untuk menghentikan pengiriman form jika validasi gagal. |

| 9 | Question 7.3: Create a script for step 6 using ajax. Note here what you observe from the addition of the program code above. (Question No. 7.3)

Menggabungkan **Validasi Klien** (jQuery) dan **Pengiriman Data Asinkron** (AJAX). Data hanya dikirim via AJAX jika client-side valid, dan respons server-side ditampilkan di halaman. |

| 10 | Question 7.4: Add validation for the password with a minimum of 8 characters using jQuery and PHP added in step 9. Note here what you observe from the addition of the program code above. (Question No 7.4) |

Validasi password dilakukan di **kedua sisi**:

- **Klien (jQuery):** Memberikan umpan balik cepat menggunakan password.length < 8.
- **Server (PHP):** Memastikan keamanan menggunakan strlen($password) < 8 sebelum memproses data. Validasi sisi server tidak boleh diabaikan.