



NAME : GANANG ANDIKA KURNIA PUTRA
NIM : 244107060008
CLASS : SIB-2G
MATKUL : PEMPROGRAMAN WEB

Topic

- The Concept of Form Processing 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

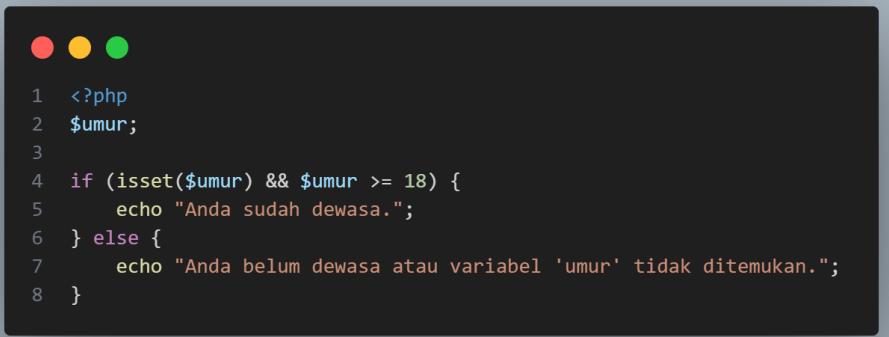
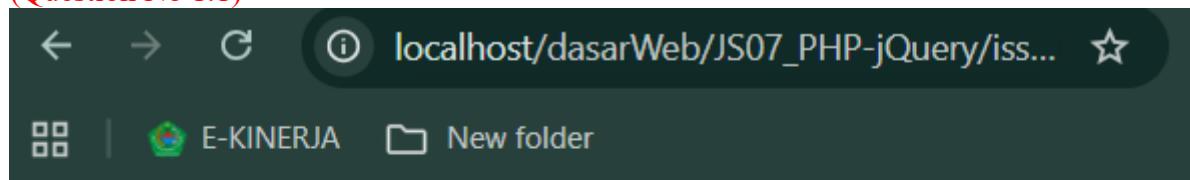
Function isset

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()**:

```
$nama = "John";
if (isset($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.

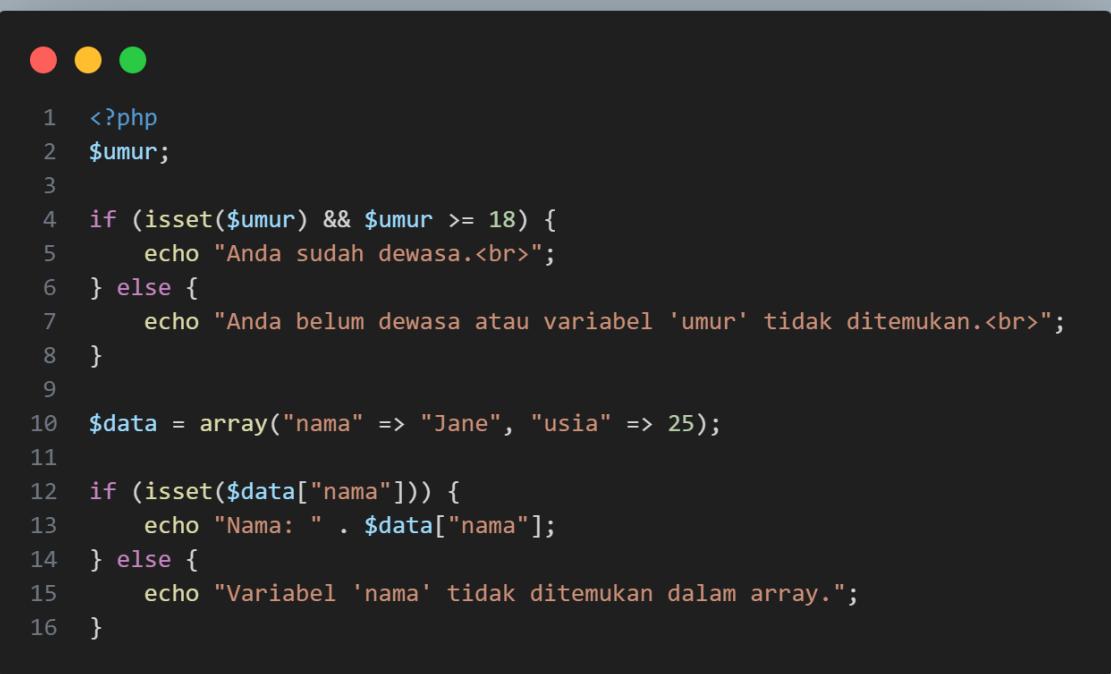
Practical Section 1. Function isset()

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	<pre>\$umur; if (isset(\$umur) && \$umur >= 18) { echo "Anda sudah dewasa."; } else { echo "Anda belum dewasa atau variabel 'umur' tidak ditemukan."; }</pre> 
4	<p>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)</p>  <p>Anda belum dewasa atau variabel 'umur' tidak ditemukan.</p>
5	The isset() function is used to check if a variable has been created and contains a value other than NULL . In this file, since \$umur is not assigned a value, isset(\$umur) returns false, so the program outputs that the variable ' 'umur' ' is not found or that the user is not an adult. It helps prevent errors when using variables that might not exist.
5	Add the contents of the isset.php file with the code below.

```

$data = array("nama" => "Jane", "usia" => 25);
if (isset($data["nama"])) {
    echo "Nama: " . $data["nama"];
} else {
    echo "Variabel 'nama' tidak ditemukan dalam array.";
}

```



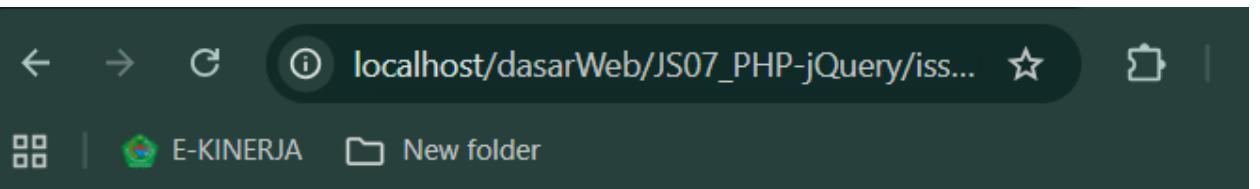
The screenshot shows a terminal window with three colored tabs at the top (red, yellow, green). The code being run is:

```

1 <?php
2 $umur;
3
4 if (isset($umur) && $umur >= 18) {
5     echo "Anda sudah dewasa.<br>";
6 } else {
7     echo "Anda belum dewasa atau variabel 'umur' tidak ditemukan.<br>";
8 }
9
10 $data = array("nama" => "Jane", "usia" => 25);
11
12 if (isset($data["nama"])) {
13     echo "Nama: " . $data["nama"];
14 } else {
15     echo "Variabel 'nama' tidak ditemukan dalam array.";
16 }

```

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)



6

Anda belum dewasa atau variabel 'umur' tidak ditemukan.
Nama: Jane

The `isset()` function is used to check whether a variable or array element has been declared and is not NULL. In this code, `isset($age)` is false because the `$age` variable does not yet have a value, while `isset($data["name"])` is true because the “name” element exists in the `$data` array. With `isset()`, we can avoid errors when checking or using variables that have not been created.

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()`:

```
$nama = ""; // Variabel string kosong
if (empty($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 <code>empty.php</code> inside <code>JS07_PHP-jQuery</code> folder. Write this code into <code>empty.php</code> file
2	<pre>\$myArray = array(); // Array kosong if (empty(\$myArray)) { echo "Array tidak terdefinisi atau kosong."; } else { echo "Array terdefinisi dan tidak kosong."; }</pre> 

Save the file, then open a browser and run localhost/dasarWeb/JS07_PHP-jQuery/empty.php



3

Array tidak terdefinisi atau kosong.

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

The empty() function is used to check whether a variable has no value or is empty. In this code, \$myArray is an empty array, so empty(\$myArray) returns true and the program displays the message "Array is undefined or empty." This function is useful for ensuring that a variable contains data before it is used in a program.

4

Add the contents of the [empty.php](#) file with the code below

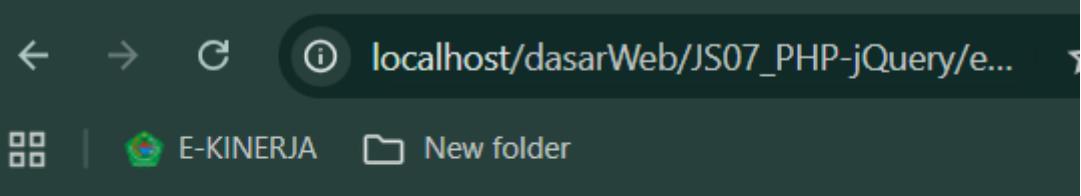
```
if (empty($nonExistentVar)) {  
    echo "Variabel tidak terdefinisi atau kosong.";  
} else {  
    echo "Variabel terdefinisi dan tidak kosong.";  
}
```

```

1 <?php
2 $myArray = array(); // Array kosong
3
4 if (empty($myArray)) {
5     echo "Array tidak terdefinisi atau kosong.<br>";
6 } else {
7     echo "Array terdefinisi dan tidak kosong.<br>";
8 }
9
10 if (empty($nonExistentVar)) {
11     echo "Variabel tidak terdefinisi atau kosong.";
12 } else {
13     echo "Variabel terdefinisi dan tidak kosong.";
14 }

```

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)



5

Array tidak terdefinisi atau kosong.
Variabel tidak terdefinisi atau kosong.

The `empty()` function is used to check whether a variable has not been defined or has no value (`empty`). In this code, the variable `$nonExistentVar` has never been created, so `empty($nonExistentVar)` returns true and displays the message “Variable not defined or empty.” This function helps avoid errors when checking variables that have no value.

Practical Section Part 3: PHP Input Form

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

```

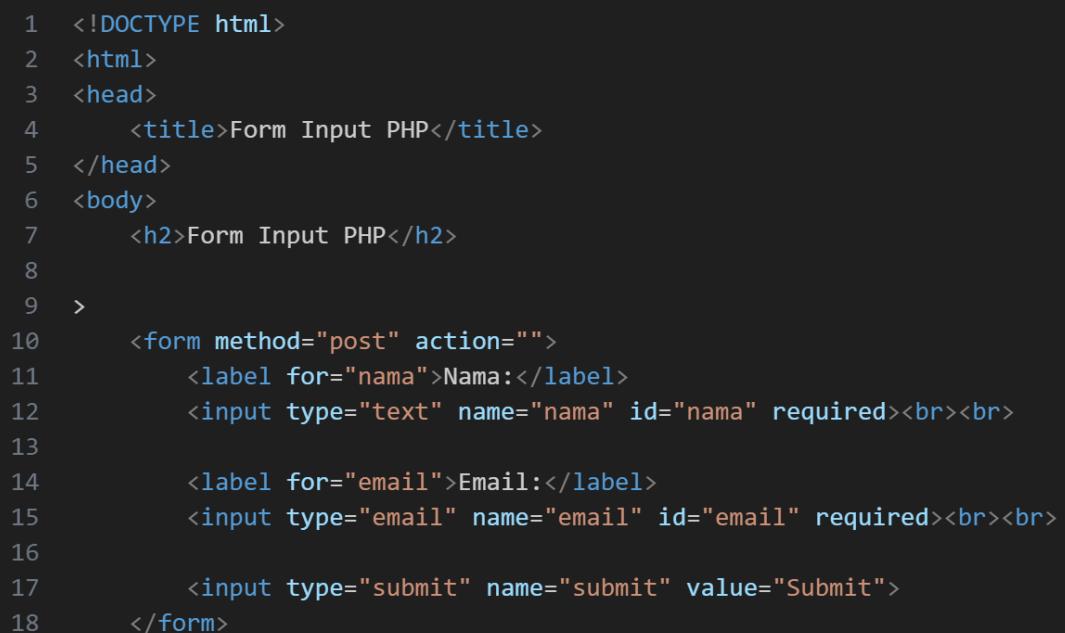
<!DOCTYPE html>
<html>
<head>
    <title>Form Input PHP</title>
</head>
<body>
    <h2>Form Input PHP</h2>
    <form method="post" action="proses_form.php">
        <label for="nama">Nama:</label>
        <input type="text" name="nama" id="nama" required><br><br>

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

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

```

2



The image shows a code editor window with a dark theme. At the top left are three circular icons: red, yellow, and green. The main area contains the following code:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Form Input PHP</title>
5  </head>
6  <body>
7      <h2>Form Input PHP</h2>
8
9  >
10     <form method="post" action="">
11         <label for="nama">Nama:</label>
12         <input type="text" name="nama" id="nama" required><br><br>
13
14         <label for="email">Email:</label>
15         <input type="email" name="email" id="email" required><br><br>
16
17         <input type="submit" name="submit" value="Submit">
18     </form>

```

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`

```

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = $_POST["nama"];
    $email = $_POST["email"];

    echo "Nama: " . $nama . "<br>";
    echo "Email: " . $email;
}

?>

```



```

1  <?php
2
3      if ($_SERVER["REQUEST_METHOD"] == "POST") {
4          $nama = $_POST["nama"];
5          $email = $_POST["email"];
6
7          echo "<h3>Hasil Input:</h3>";
8          echo "Nama: " . $nama . "<br>";
9          echo "Email: " . $email;
10     }
11

```

- 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)

Form Input PHP

Nama:

Email:

Hasil Input:

Nama: ganang

Email: ganangputra939@gmail.com

This file uses a form to send data to the server using the POST method. After the Submit button is pressed, PHP will process the data sent via `$_POST` and display it back on the same page. The `action=""` attribute causes the form to be sent to this file itself. The `$_SERVER["REQUEST_METHOD"]` function ensures that data is only processed when the form is submitted. With a form like this, users can input data and immediately see the results on the same page.

- | | |
|---|---|
| 5 | Create a new file named <code>form_self.php</code> inside <code>JS07_PHP-jQuery</code> folder Type the code below inside <code>form_self.php</code> |
|---|---|

```
<!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>

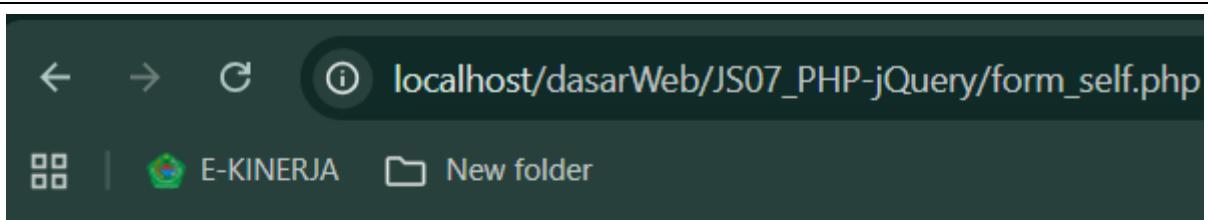
        <input type="submit" name="submit" value="Submit">
    </form>
</body>
</html>
```

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Form Input PHP</title>
5  </head>
6  <body>
7      <h2>Form Input PHP</h2>
8
9      <?php
10     // Inisialisasi variabel
11     $namaErr = "";
12     $nama = "";
13
14     // Cek apakah form sudah disubmit
15     if ($_SERVER["REQUEST_METHOD"] == "POST") {
16         // Validasi nama (contoh: pastikan nama tidak kosong)
17         if (empty($_POST["nama"])) {
18             $namaErr = "Nama harus diisi!";
19         } else {
20             $nama = $_POST["nama"];
21             echo "Data berhasil disimpan!";
22         }
23     }
24     ?>
25
26     <form method="post" action=<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>>>
27         <label for="nama">Nama:</label>
28         <input type="text" name="nama" id="nama" value=<?php echo $nama; ?>>
29         <span class="error"><?php echo $namaErr; ?></span><br><br>
30
31         <input type="submit" name="submit" value="Submit">
32     </form>
33 </body>
34 </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 Input PHP

Data berhasil disimpan!

Nama:

In this file, the form is used to send data to the same page (self-processing form) using `$_SERVER['PHP_SELF']`. This means that when the Submit button is pressed, the form data is not sent to another file, but is processed directly in the same file. The program will perform a simple validation, which is to check whether the name field has been filled in. If it is empty, an error message "Name must be filled in!" will appear, and if it has been filled in, the message "Data successfully saved!" will appear.

In this way, the form can display the input, validation, and results on a single web page without moving to another page.

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:

```
<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:

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

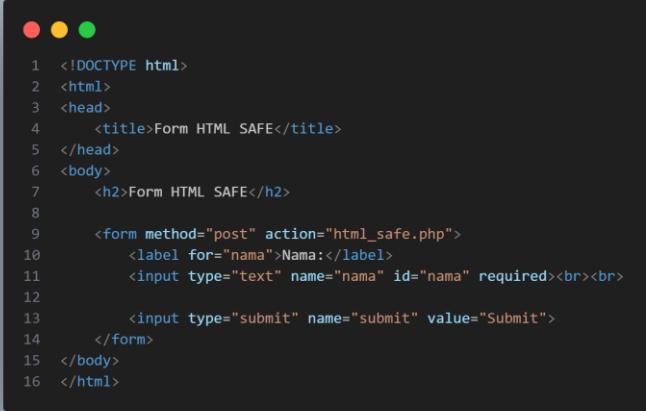
// Menggunakan htmlspecialchars untuk menghindari HTML injection saat menampilkan data
echo '<div>' . htmlspecialchars($user_input, ENT_QUOTES, 'UTF-8') . '</div>';
```

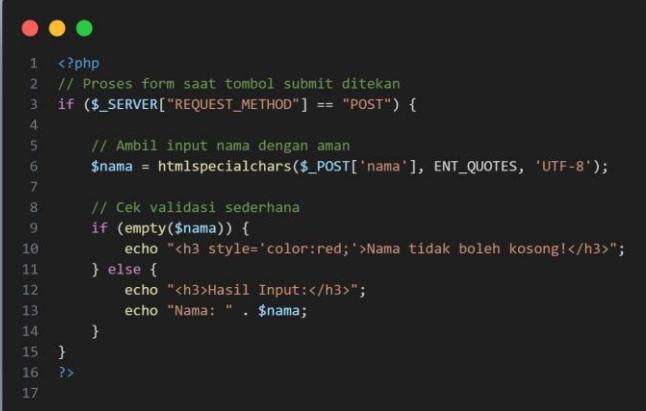
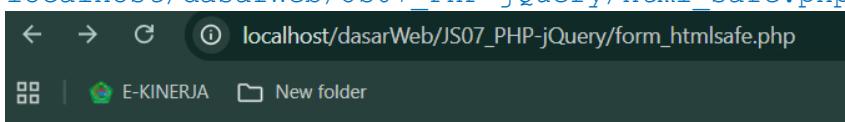
By using these steps, you can mitigate the risk of HTML injection in your web application. It's always important to validate inputs, clean incoming data, and avoid displaying user data without proper escaping.

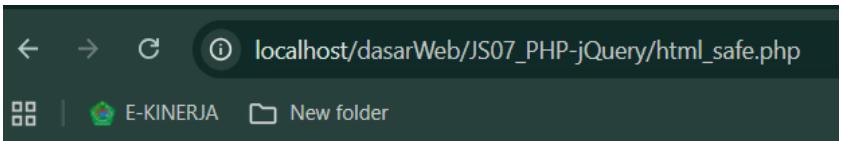
Practical Section 4 : HTML Injection

Practical Section Steps:

Step	Description
------	-------------

1	<p>Create a new file named <code>html_safe.php</code> inside <code>JS07_PHP-jQuery</code> folder. Type the code below inside <code>html_safe.php</code></p>
	<pre>\$input = \$_POST['input']; \$input = htmlspecialchars(\$input, ENT_QUOTES, 'UTF-8');</pre>
2	 <pre>1 <!DOCTYPE html> 2 <html> 3 <head> 4 <title>Form HTML SAFE</title> 5 </head> 6 <body> 7 <h2>Form HTML SAFE</h2> 8 9 <form method="post" action="html_safe.php"> 10 <label for="nama">Nama:</label> 11 <input type="text" name="nama" id="nama" required>

 12 13 <input type="submit" name="submit" value="Submit"> 14 </form> 15 </body> 16 </html></pre>  <pre>1 <?php 2 // Proses form saat tombol submit ditekan 3 if (\$_SERVER["REQUEST_METHOD"] == "POST") { 4 5 // Ambil input nama dengan aman 6 \$nama = htmlspecialchars(\$_POST['nama'], ENT_QUOTES, 'UTF-8'); 7 8 // Cek validasi sederhana 9 if (empty(\$nama)) { 10 echo "<h3 style='color:red;'>Nama tidak boleh kosong!</h3>"; 11 } else { 12 echo "<h3>Hasil Input:</h3>"; 13 echo "Nama: " . \$nama; 14 } 15 } 16 ?> 17</pre>
3	<p>Add another script so that step 2 can run normally, save the file. Then open a browser and run <code>localhost/dasarWeb/JS07_PHP-jQuery/html_safe.php</code></p>  <p>localhost/dasarWeb/JS07_PHP-jQuery/form_htmlsafe.php</p> <p>E-KINERJA New folder</p> <p>Form HTML SAFE</p> <p>Nama: <input type="text" value="ganang"/></p> <p><input type="button" value="Submit"/></p>



Hasil Input:

Nama: ganang

Note here what you observe from the addition of the program code above. (Question No. 4.1)

I noticed that when users fill out the form and press the Submit button, the data entered in the “Name” field is sent to the server using the POST method. The PHP code then processes the input with the htmlspecialchars() function to protect it from dangerous characters such as HTML tags or scripts (preventing XSS). Additionally, there is a simple validation using empty() that ensures the name field cannot be left blank. If the field is empty, the message “Name cannot be empty!” appears, while if it is filled in, the result of the name input entered by the user will be displayed.

5 Type the additional code in step 6 inside the `html_safe.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
}
```

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Form HTML SAFE</title>
5  </head>
6  <body>
7      <h2>Form HTML SAFE</h2>
8
9      <form method="post" action="html_safe.php">
10         <label for="nama">Nama:</label>
11         <input type="text" name="nama" id="nama" required><br><br>
12
13         <label for="email">Email:</label>
14         <input type="email" name="email" id="email" required><br><br>
15
16         <input type="submit" name="submit" value="Submit">
17     </form>
18
19
20     </body>
21 </html>
```

```

1 <?php
2
3 if ($_SERVER["REQUEST_METHOD"] == "POST") {
4
5     $nama = htmlspecialchars($_POST['nama'], ENT_QUOTES, 'UTF-8');
6     $email = htmlspecialchars($_POST['email'], ENT_QUOTES, 'UTF-8');
7
8
9     if (empty($nama)) {
10         echo "<h3 style='color:red;'>Nama tidak boleh kosong!</h3>";
11     }
12
13     elseif (empty($email)) {
14         echo "<h3 style='color:red;'>Email tidak boleh kosong!</h3>";
15     }
16
17     elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
18         echo "<h3 style='color:red;'>Format email tidak valid!</h3>";
19     }
20
21     else {
22         echo "<h3>Hasil Input:</h3>";
23         echo "Nama: " . $nama . "<br>";
24         echo "Email: " . $email . "<br>";
25     }
26 }
27 ?>
28
29

```

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

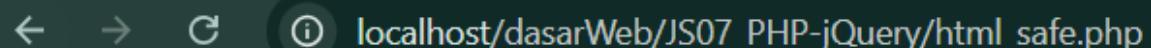
localhost/dasarWeb/JS07_PHP-jQuery/form_htmlsafe.php

 E-KINERJA  New folder

Form HTML SAFE

Nama:

Email:

localhost/dasarWeb/JS07_PHP-jQuery/html_safe.php

 E-KINERJA  New folder

Hasil Input:

Nama: ganang

Email: ganangputra939@gmail.com

7

8 Note here what you observe from the addition of the program code above. (Question No. 4.2)

From the code addition above, I observe that the filter_var() function is used to check whether the email input is in the correct format. If the format is invalid, the program will display an error message, whereas if it is valid, the email data is processed and displayed securely using htmlspecialchars(). In this way, the program becomes more secure and can filter out incorrect input from the outset.

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:
 - o *: 0 or more
 - o +: 1 or more
 - o ?: 0 or 1 time
 - o {n}: Exactly n times
 - o {n,}: At least n times
 - o {n,m}: At least n times, maximum m times
 - o

Examples of Regex Usage:

1. Matching patterns

```

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

```

2. Match set characters

```

$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:

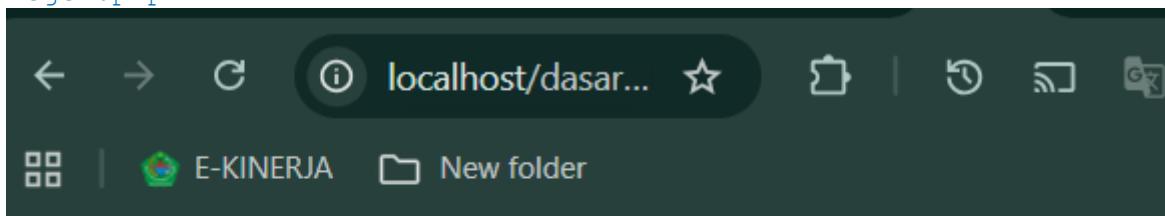
Step	Description
1	Create a new file named <code>regex.php</code> inside <code>JS07_PHP-jQuery</code> folder Type the code in step 2 inside the <code>regex.php</code>
2	<pre> \$pattern = '/[a-z]/'; // Cocokkan huruf kecil. \$text = 'This is a Sample Text.'; if (preg_match(\$pattern, \$text)) { echo "Huruf kecil ditemukan!"; } else { echo "Tidak ada huruf kecil!"; } </pre>

```

1 <?php
2 $pattern = '/[a-z]/'; // Cocokkan huruf kecil.
3 $text = 'This is a Sample Text.';
4
5 if (preg_match($pattern, $text)) {
6     echo "Huruf kecil ditemukan!";
7 } else {
8     echo "Tidak ada huruf kecil!";
9 }

```

Save the file, then open a browser and run /refresh localhost/dasarWeb/JS07_PHP-jQuery/regex.php



Huruf kecil ditemukan!

Note here what you observe from the addition of the program code above. (Question No 5.1)

From the addition of the code above, I understand that the preg_match() function is used to match text patterns using Regular Expressions (regex). The pattern /[a-z]/ means searching for lowercase letters (a to z) in the \$text variable. If lowercase letters are found, the program displays the message “Lowercase letters found!”, while if none are found, it displays “No lowercase letters!”. Thus, this code functions to check for the presence of lowercase letters in a text using regular expressions.

5 Type the additional code in step 6 inside the `regex.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!";
}

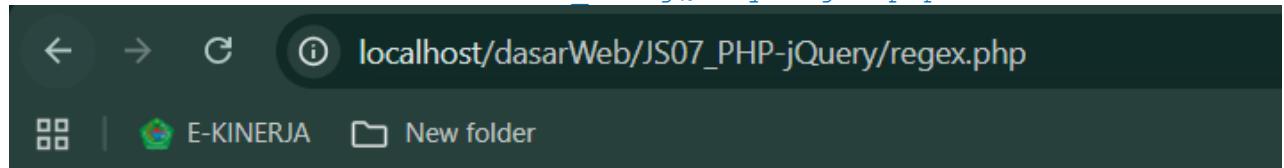
```

```

1 $pattern = '/[a-z]/'; // Cocokkan huruf kecil.
2 $text = 'This is a Sample Text.';
3
4 if (preg_match($pattern, $text)) {
5     echo "Huruf kecil ditemukan!<br>";
6 } else {
7     echo "Tidak ada huruf kecil!<br>";
8 }
9
10 $pattern = '/[0-9]+/'; // Cocokkan satu atau lebih digit
11 $text = 'There are 123 apples.';
12
13 if (preg_match($pattern, $text, $matches)) {
14     echo "Cocokkan: " . $matches[0] . "<br>";
15 } else {
16     echo "Tidak ada yang cocok!<br>";

```

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



7	Huruf kecil ditemukan! Cocokkan: 123
---	---

8	Note here what you observe from the addition of the program code above. (Question No. 5.2) Regex /[0-9]+/ searches for one or more digits in the text. If numbers are found, the program displays them (for example, “123”). This shows that regex can be used to recognize number patterns in a string.
---	--

9	Type the additional code in step 10 inside the <code>regex.php</code>
---	---

10	
----	--

```

$pattern = '/apple/';
$replacement = 'banana';
$text = 'I like apple pie.';
$new_text = preg_replace($pattern, $replacement, $text);
echo $new_text; // Output: "I like banana pie."

```



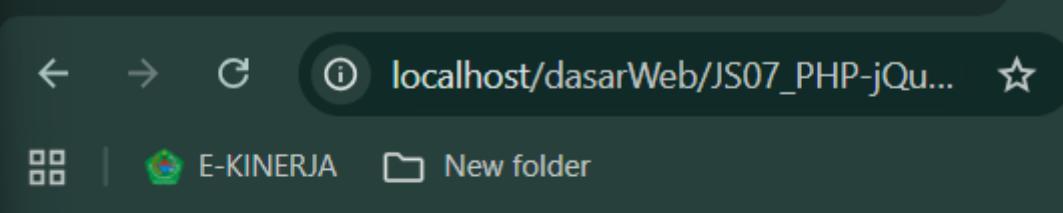
```

1 $pattern = '/apple/';
2 $replacement = 'banana';
3 $text = 'I like apple pie.';
4 $new_text = preg_replace($pattern, $replacement, $text);
5 echo $new_text . "<br><br>"; // Output: "I like banana pie."

```

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

11



Huruf kecil ditemukan!

Cocokkan: 123

I like banana pie.

12

Note here what you observe from the addition of the program code above. (Question No 5.3)

Regex /[0-9]+/ searches for one or more digits in the text. If numbers are found, the program displays them (for example, “123”). This shows that regex can be used to recognize number patterns in a string.

13

Type the additional code in step 14 inside the [regex.php](#)

```

$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!";
}

```

14



```

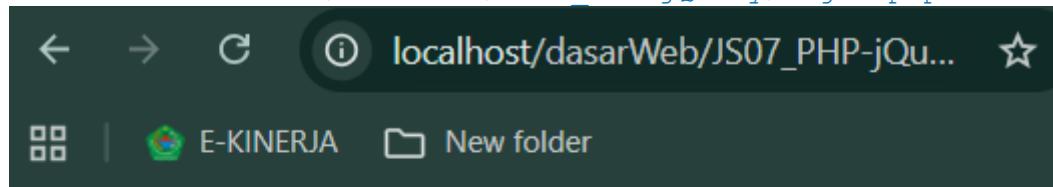
1 $pattern = '/go*d/'; // Cocokkan "god", "good", "goood", dll.
2 $text = 'god is good.';

3

4 if (preg_match($pattern, $text, $matches)) {
5     echo "Cocokkan: " . $matches[0] . "<br>";
6 } else {
7     echo "Tidak ada yang cocok!<br>";
8 }
9 ?>
10

```

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



15

Huruf kecil ditemukan!

Cocokkan: 123

I like banana pie.

Cocokkan: god

16

Note here what you observe from the addition of the program code above. (Question No 5.4)
The regex /apple/ is used to search for the word “apple” within a string. If found, the result displays that word. This shows how regex can be used to search for specific words in text.

From the code above, I understand that the preg_replace() function is used to replace text that matches a specific pattern with new text. In the example, the pattern /apple/ searches for the word “apple” and replaces it with ‘banana’, so the result becomes “I like banana pie.”. Additionally, the pattern /go*d/ is used to match words that have the letter “o” appearing one or more times, such as “god,” “good,” or “goood.” This regex demonstrates how we can detect and manipulate specific text patterns within a string using regular expressions in PHP.

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](#))****

```
● ● ●
1 $pattern = '/apples?/'; // "s" boleh muncul 0 atau 1 kali
2 $text = 'There is one apple and some apples.';
3
4 if (preg_match_all($pattern, $text, $matches)) {
5     echo "Kecocokan ditemukan: " . implode(", ", $matches[0]) . "<br>";
6 } else {
7     echo "Tidak ada yang cocok!<br>";
```

17

Kecocokan ditemukan: apple, apples

From the addition of the code above, I understand that the ? symbol in regex is used to indicate that the previous character or pattern is optional (may appear once or not at all). In the example /apples?/, this pattern matches both the words “apple” and “apples.” Thus, this regex allows for more flexible searches for variations of words that have optional endings.

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](#))

Kecocokan ditemukan: aa, aaa, aaaa, aaaa

```
● ● ●
1 $pattern = '/a{2,4}/'; // Huruf 'a' muncul 2 sampai 4 kali berturut-turut
2 $text = 'a aa aaa aaaa aaaaa';
3
4 if (preg_match_all($pattern, $text, $matches)) {
5     echo "Kecocokan ditemukan: " . implode(", ", $matches[0]) . "<br>";
6 } else {
7     echo "Tidak ada yang cocok!<br>";
8 }
```

18

From the addition of the code above, I understand that the {n,m} sign in regex is used to determine the number of occurrences of a particular character or pattern. In the example /a{2,4}/, this pattern matches the letter “a” that appears between 2 and 4 times consecutively, so the results display “aa”,

‘aaa’, and “aaaa”. This regex is useful for limiting or specifying the length of the pattern to be matched.

Practical Section 6 : Advanced Form

Practical Section Steps:

Step	Description
1	Create a new file named <code>form_next.php</code> inside <code>JS07_PHP-jQuery</code> folder site. Type the code in step 2 inside the <code>form_next.php</code>
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></pre>

```

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;
}

```



The screenshot shows a code editor window with a dark theme. At the top, there are three circular icons: red, yellow, and green. The main area contains the following code:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Contoh Form dengan PHP</title>
5  </head>
6  <body>
7      <h2>Form Contoh</h2>
8      <form method="POST" action="html_next.php">
9          <label for="buah">Pilih Buah:</label>
10         <select name="buah" id="buah">
11             <option value="apel">Apel</option>
12             <option value="pisang">Pisang</option>
13             <option value="mangga">Mangga</option>
14             <option value="jeruk">Jeruk</option>
15         </select>
16
17
18         <br>
19         <label>Pilih Warna Favorit:</label><br>
20         <input type="checkbox" name="warna[]" value="merah"> Merah<br>
21         <input type="checkbox" name="warna[]" value="biru"> Biru<br>
22         <input type="checkbox" name="warna[]" value="hijau"> Hijau<br>
23         <br>
24
25         <br>
26         <label>Pilih Jenis Kelamin:</label><br>
27         <input type="radio" name="jenis_kelamin" value="laki-laki"> Laki-laki<br>
28         <input type="radio" name="jenis_kelamin" value="perempuan"> Perempuan<br>
29         <br>
30
31         <br>
32         <input type="submit" value="Submit">
33
34     </form>
35 </body>
36 </html>
37
38
39
40

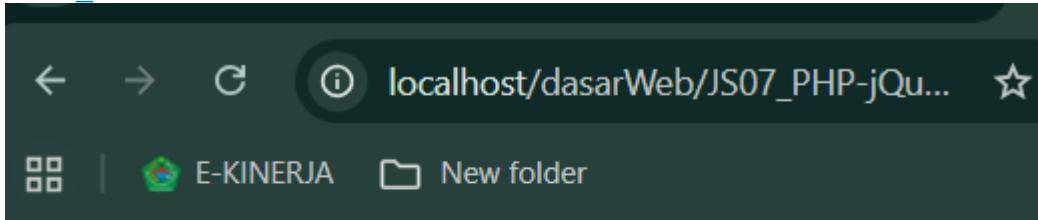
```

```

1  <?php
2  if ($_SERVER["REQUEST_METHOD"] == "POST") {
3      $selectedBuah = $_POST['buah'];
4
5      if (isset($_POST['warna'])) {
6          $selectedWarna = $_POST['warna'];
7      } else {
8          $selectedWarna = [];
9      }
10
11     $selectedJenisKelamin = $_POST['jenis_kelamin'];
12
13     echo "Anda memilih buah: " . $selectedBuah . "<br>";
14
15     if (!empty($selectedWarna)) {
16         echo "Warna favorit Anda: " . implode(", ", $selectedWarna) . "<br>";
17     } else {
18         echo "Anda tidak memilih warna favorit.<br>";
19     }
20
21     echo "Jenis kelamin Anda: " . $selectedJenisKelamin;
22 }
23 ?>

```

Save the file, then open a browser and run /refresh localhost/dasarWeb/JS07_PHP-jQuery/form_next.php



3

Anda memilih buah: apel
Warna favorit Anda: merah
Jenis kelamin Anda: perempuan

Note here what you observe from the addition of the program code above. (Question No 6.1)

4 The program displays the selected results from various input forms (select, checkbox, radio) using the POST method, as well as checking and displaying the data entered by the user dynamically on the same page.

Create a new file named `form_ajax.php` inside `JS07_PHP-jQuery` folder Type the code in step 2 inside the `form_ajax.php`

```
<!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();

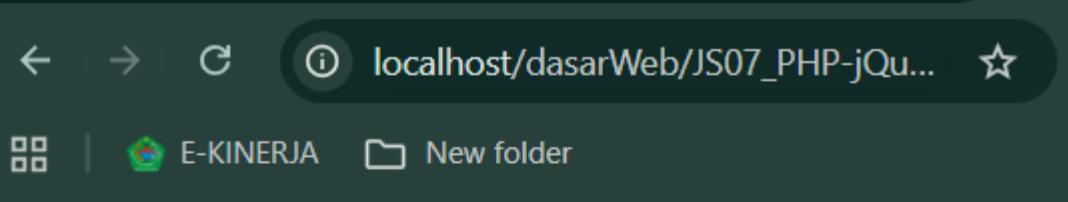
                // Kirim data ke server PHP
                $.ajax({
                    url: "proses_lanjut.php", // Ganti dengan nama file PHP yang sesuai
                    type: "POST",
                    data: formData,
                    success: function (response) {
                        // Tampilkan hasil dari server di div "hasil"
                        $("#hasil").html(response);
                    }
                });
            });
        });
    </script>
</body>
</html>
```

5



```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Contoh Form dengan PHP dan jQuery</title>
5  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
6  </head>
7  <body>
8  <h2>Form Contoh</h2>
9  <form id="myForm">
10     <label for="buah">Pilih Buah:</label>
11     <select name="buah" id="buah">
12         <option value="apel">Apel</option>
13         <option value="pisang">Pisang</option>
14         <option value="mangga">Mangga</option>
15         <option value="jeruk">Jeruk</option>
16     </select>
17
18     <br>
19
20     <label>Pilih Warna Favorit:</label><br>
21     <input type="checkbox" name="warna[]" value="merah"> Merah<br>
22     <input type="checkbox" name="warna[]" value="biru"> Biru<br>
23     <input type="checkbox" name="warna[]" value="hijau"> Hijau<br>
24
25     <br>
26
27     <label>Pilih Jenis Kelamin:</label><br>
28     <input type="radio" name="jenis_kelamin" value="laki-laki"> Laki-laki<br>
29     <input type="radio" name="jenis_kelamin" value="perempuan"> Perempuan<br>
30
31     <br>
32
33     <input type="submit" value="Submit">
34 </form>
35
36 <div id="hasil">
37     </div>
38
39 <script>
40 $(document).ready(function () {
41     $("#myForm").submit(function (e) {
42         e.preventDefault(); // Mencegah pengiriman form secara default
43
44         // Mengumpulkan data form
45         var formData = $("#myForm").serialize();
46
47         // Kirim data ke server PHP
48         $.ajax({
49             url: "html_next.php", // Ganti dengan nama file PHP yang sesuai
50             type: "POST",
51             data: formData,
52             success: function (response) {
53                 // Tampilkan hasil dari server di div "hasil"
54                 $("#hasil").html(response);
55             }
56         });
57     });
58 });
59 </script>
60 </body>
61 </html>
```

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)



6

Form Contoh

Pilih Buah:

Pilih Warna Favorit:

- Merah
- Biru
- Hijau

Pilih Jenis Kelamin:

- Laki-laki
- Perempuan

Anda memilih buah: apel

Warna favorit Anda: merah

Jenis kelamin Anda: perempuan

From the code above, I understand that using jQuery and AJAX allows the form to send data to the server without reloading the page. The data is sent asynchronously to a PHP file, and the response is then displayed directly on the web page via the `<div id="result">` element. This method makes user interaction faster and more dynamic than a regular form.

Practical Section 7 : Form Validation

Practical Section Steps:

Step	Description
1	Create a new file named <code>form_validation.php</code> inside <code>JS07_PHP-jQuery</code> folder Type the code below inside <code>form_validation.php</code>

2

```
<!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>

<?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.";
    }

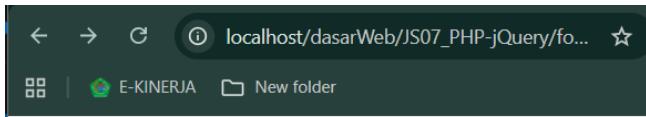
    // Jika ada kesalahan validasi
    if (!empty($errors)) {
        foreach ($errors as $error) {
            echo $error . "<br>";
        }
    } else {
        // Lanjutkan dengan pemrosesan data jika semua validasi berhasil
        // Misalnya, menyimpan data ke database atau mengirim email
        echo "Data berhasil dikirim: Nama = $nama, Email = $email";
    }
}
?>
```

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Form Input dengan Validasi</title>
5  </head>
6  <body>
7  <h1>Form Input dengan Validasi</h1>
8  <form method="post" action="proses_validasi.php">
9    <label for="nama">Nama:</label>
10   <input type="text" id="nama" name="nama">
11   <br>
12
13   <label for="email">Email:</label>
14   <input type="text" id="email" name="email">
15   <br>
16
17   <input type="submit" value="Submit">
18 </form>
19 </body>
20 </html>
21
22
23 <?php
24 if ($_SERVER["REQUEST_METHOD"] == "POST") {
25   $nama = $_POST["nama"];
26   $email = $_POST["email"];
27   $errors = array();
28
29   // Validasi Nama
30   if (empty($nama)) {
31     $errors[] = "Nama harus diisi.";
32   }
33
34   // Validasi Email
35   if (empty($email)) {
36     $errors[] = "Email harus diisi.";
37   } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
38     $errors[] = "Format email tidak valid.";
39   }
40
41   // Jika ada kesalahan validasi
42   if (!empty($errors)) {
43     foreach ($errors as $error) {
44       echo $error . "<br>";
45     }
46   } else {
47     // Lanjutkan dengan pemrosesan data jika semua validasi berhasil
48     // Misalnya, menyimpan data ke database atau mengirim email
49     echo "Data berhasil dikirim: Nama = $nama, Email = $email";
50   }
51 }
52 ?>

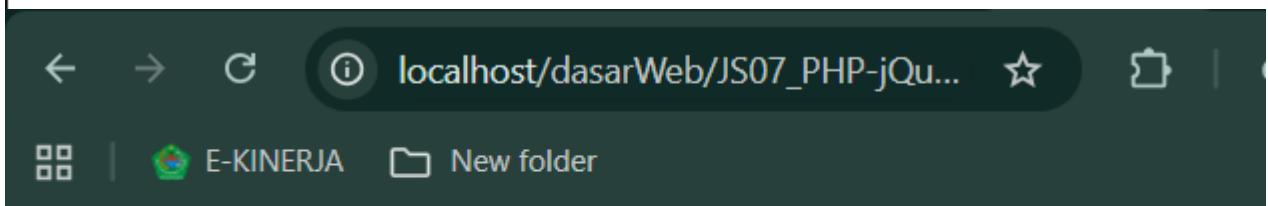
```

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



Form Input dengan Validasi

Nama:
Email:



Note here what you observe from the addition of the program code above. (Question No 7.1)

- | | |
|---|--|
| 4 | From the addition of the program code above, it can be observed that the form now has validation using **jQuery** before being sent to the server, so that empty inputs can be prevented and users are given immediate warnings without having to reload the page. |
| 5 | Develop a file named <code>form_validation.php</code> inside <code>JS07_PHP-jQuery</code> folder site. Type the code in step 6 inside the <code>form_validation.php</code> |

```
<!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;"><br>

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

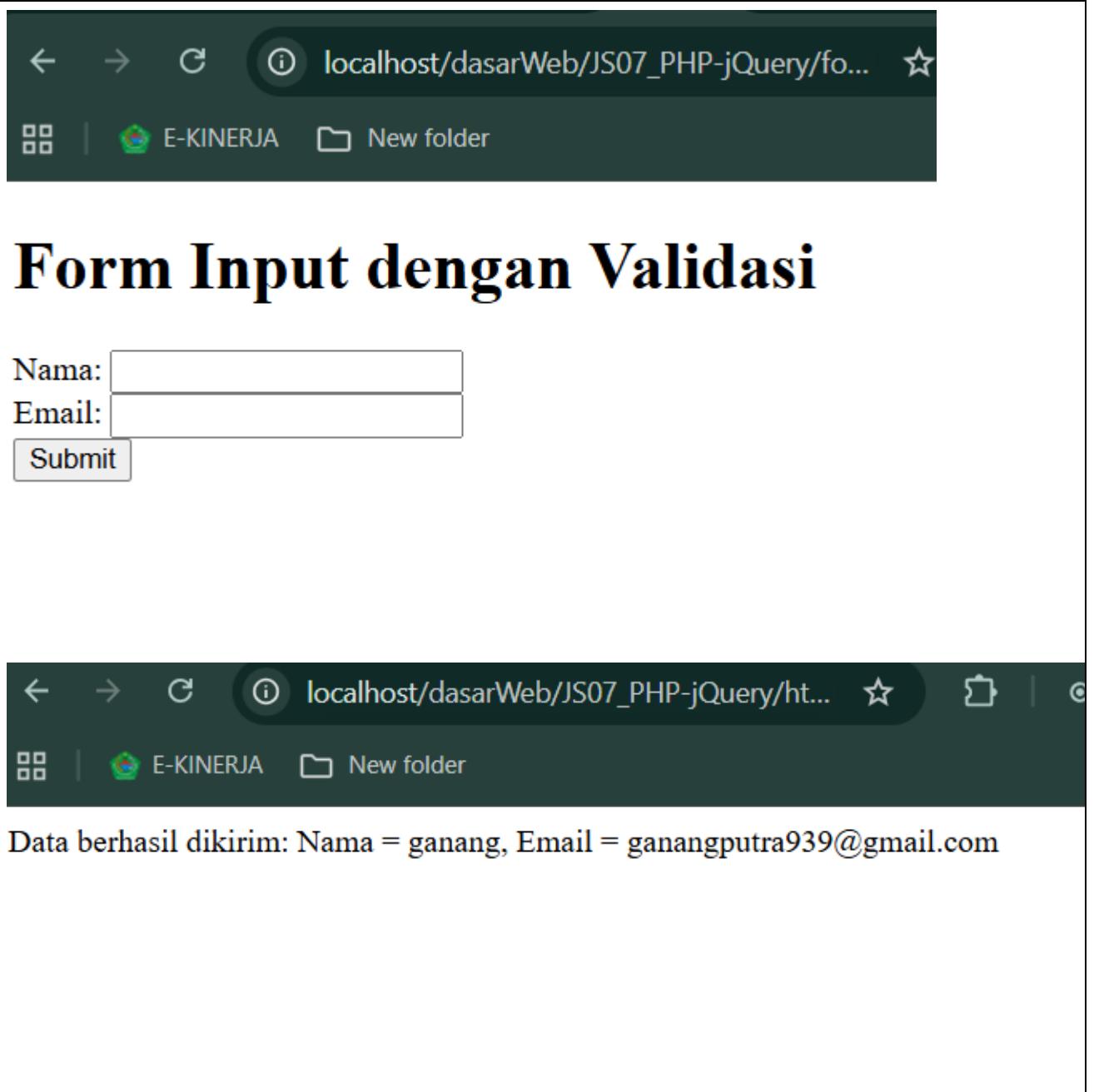
                if (valid) {
                    event.preventDefault();
                    // Menghentikan pengiriman form jika validasi gagal
                }
            });
        });
    </script>
</body>
</html>
```

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Form Input dengan Validasi</title>
5  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
6  </head>
7  <body>
8  <h1>Form Input dengan Validasi</h1>
9  <form id="myForm" method="post" action="proses_validasi.php">
10     <label for="nama">Nama:</label>
11     <input type="text" id="nama" name="nama">
12     <span id="nama-error" style="color: red;"></span><br>
13
14     <label for="email">Email:</label>
15     <input type="text" id="email" name="email">
16     <span id="email-error" style="color: red;"></span><br>
17
18     <input type="submit" value="Submit">
19 </form>
20
21 <script>
22 $(document).ready(function() {
23     $("#myForm").submit(function(event) {
24         var nama = $("#nama").val();
25         var email = $("#email").val();
26         var valid = true;
27
28         if (nama === "") {
29             $("#nama-error").text("Nama harus diisi.");
30             valid = false;
31         } else {
32             $("#nama-error").text("");
33         }
34
35         if (email === "") {
36             $("#email-error").text("Email harus diisi.");
37             valid = false;
38         } else {
39             $("#email-error").text("");
40         }
41
42         if (!valid) {
43             event.preventDefault(); // Menghentikan pengiriman form jika validasi gagal
44         }
45     });
46 });
47 </script>
48 </body>
49 </html>

```

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



The screenshot shows a web browser window with the URL `localhost/dasarWeb/JS07_PHP-jQuery/fo...`. The page title is "E-KINERJA". The main content is a form titled "Form Input dengan Validasi" containing fields for "Nama" and "Email", and a "Submit" button.

Form Input dengan Validasi

Nama:

Email:

Submit

	<p>Note here what you observe from the addition of the program code above. (Question No. 7.2)</p> <p>From the code above, I understand that this program adds form validation on the client side using jQuery before the data is sent to the server. Validation is performed immediately when the submit button is pressed. If the “Name” or “Email” fields are still empty, an error message is displayed without reloading the page. This method makes the validation process faster, more interactive, and provides immediate feedback to the user.</p>
8	<p>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)</p>

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Form Input dengan Validasi dan AJAX</title>
5      <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
6  </head>
7  <body>
8      <h1>Form Input dengan Validasi dan AJAX</h1>
9      <form id="myForm" method="post">
10         <label for="nama">Nama:</label>
11         <input type="text" id="nama" name="nama">
12         <span id="nama-error" style="color:red;"></span><br>
13
14         <label for="email">Email:</label>
15         <input type="text" id="email" name="email">
16         <span id="email-error" style="color:red;"></span><br>
17
18         <input type="submit" value="Submit">
19     </form>
20
21     <div id="result"></div> <!-- Tempat menampilkan hasil dari PHP -->
22
23     <script>
24     $(document).ready(function() {
25         $("#myForm").submit(function(event) {
26             event.preventDefault(); // Hentikan pengiriman form biasa
27
28             var nama = $("#nama").val();
29             var email = $("#email").val();
30             var valid = true;
31
32             // Validasi sederhana
33             if (nama === "") {
34                 $("#nama-error").text("Nama harus diisi.");
35                 valid = false;
36             } else {
37                 $("#nama-error").text("");
38             }
39
40             if (email === "") {
41                 $("#email-error").text("Email harus diisi.");
42                 valid = false;
43             } else {
44                 $("#email-error").text("");
45             }
46
47             if (valid) {
48                 // Kirim data ke PHP menggunakan AJAX
49                 $.ajax({
50                     url: "form_validationajax.php",
51                     type: "POST",
52                     data: { nama: nama, email: email },
53                     success: function(response) {
54                         $("#result").html(response); // Tampilkan hasil dari PHP
55                     },
56                     error: function() {
57                         $("#result").html("<p style='color:red;'>Terjadi kesalahan saat mengirim data.</p>");
58                     }
59                 });
60             }
61         });
62     });
63     </script>
64 </body>
65 </html>
66

```

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Form Input dengan Validasi</title>
5  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
6  </head>
7  <body>
8  <h1>Form Input dengan Validasi</h1>
9  <form id="myForm" method="post" action="html_validation.php">
10 <label for="nama">Nama:</label>
11 <input type="text" id="nama" name="nama">
12 <span id="nama-error" style="color: red;"></span><br>
13
14 <label for="email">Email:</label>
15 <input type="text" id="email" name="email">
16 <span id="email-error" style="color: red;"></span><br>
17
18 <input type="submit" value="Submit">
19 </form>
20
21 <script>
22 $(document).ready(function() {
23     $("#myForm").submit(function(event) {
24         var nama = $("#nama").val();
25         var email = $("#email").val();
26         var valid = true;
27
28         // Validasi Nama
29         if (nama === "") {
30             $("#nama-error").text("Nama harus diisi.");
31             valid = false;
32         } else {
33             $("#nama-error").text("");
34         }
35
36         // Validasi Email
37         if (email === "") {
38             $("#email-error").text("Email harus diisi.");
39             valid = false;
40         } else {
41             $("#email-error").text("");
42         }
43
44         if (!valid) {
45             event.preventDefault(); // Menghentikan pengiriman form jika validasi gagal
46         }
47     });
48 });
49 </script>
50 </body>
51 </html>
```

The screenshot shows a web browser window with the URL `localhost/dasarWeb/JS07_PHP-jQuery/fo...`. The page title is "Form Input dengan Validasi dan AJA". Below the title, there is a form with two input fields: "Nama" containing "ganang" and "Email" containing "myinwork14@gmail.com". A "Submit" button is also present. Below the form, a green message says "Data berhasil dikirim!". Underneath this message, the submitted data is displayed again: "Nama: ganang" and "Email: myinwork14@gmail.com".

Form Input dengan Validasi dan AJA

Nama:

Email:

Data berhasil dikirim!

Nama: ganang
Email: myinwork14@gmail.com

With the addition of AJAX code, forms can send data to the server without reloading the page. The validation process continues, and the results from the PHP file are displayed immediately on the same page. This makes user interaction faster, more dynamic, and more efficient than sending regular forms.

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)
----	--

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Form Validasi Password</title>
5      <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
6  </head>
7  <body>
8  <h1>Form Input dengan Validasi Password</h1>
9
10 <form id="myForm">
11     <label for="nama">Nama:</label>
12     <input type="text" id="nama" name="nama">
13     <span id="nama-error" style="color: red;"></span><br>
14
15     <label for="email">Email:</label>
16     <input type="text" id="email" name="email">
17     <span id="email-error" style="color: red;"></span><br>
18
19     <label for="password">Password:</label>
20     <input type="password" id="password" name="password">
21     <span id="password-error" style="color: red;"></span><br><br>
22
23     <input type="submit" value="Submit">
24 </form>
25
26 <hr>
27 <div id="hasil"></div>
28
29 <script>
30 $(document).ready(function() {
31     $("#myForm").submit(function(e) {
32         e.preventDefault();
33
34         var nama = $("#nama").val();
35         var email = $("#email").val();
36         var password = $("#password").val();
37         var valid = true;
38
39         // Validasi Nama
40         if (nama === "") {
41             $("#nama-error").text("Nama harus diisi.");
42             valid = false;
43         } else {
44             $("#nama-error").text("");
45         }
46
47         // Validasi Email
48         if (email === "") {
49             $("#email-error").text("Email harus diisi.");
50             valid = false;
51         } else {
52             $("#email-error").text("");
53         }
54
55         // Validasi Password (minimal 8 karakter)
56         if (password.length < 8) {
57             $("#password-error").text("Password minimal 8 karakter.");
58             valid = false;
59         } else {
60             $("#password-error").text("");
61         }
62
63         // Kirim data ke server jika semua validasi berhasil
64         if (valid) {
65             $.ajax({
66                 url: "html_validationpassword.php",
67                 type: "POST",
68                 data: $(this).serialize(),
69                 success: function(response) {
70                     $("#hasil").html(response);
71                 }
72             });
73         }
74     });
75 });
76 </script>
77 </body>
78 </html>
79
80
81
```

```
1 <?php
2 if ($_SERVER["REQUEST_METHOD"] == "POST") {
3     $nama = $_POST["nama"];
4     $email = $_POST["email"];
5     $password = $_POST["password"];
6     $errors = array();
7
8     // Validasi Nama
9     if (empty($nama)) {
10         $errors[] = "Nama harus diisi.";
11     }
12
13     // Validasi Email
14     if (empty($email)) {
15         $errors[] = "Email harus diisi.";
16     } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
17         $errors[] = "Format email tidak valid.";
18     }
19
20     // Validasi Password minimal 8 karakter
21     if (strlen($password) < 8) {
22         $errors[] = "Password minimal 8 karakter.";
23     }
24
25     // Jika ada kesalahan validasi
26     if (!empty($errors)) {
27         echo "<h3 style='color:red;'>Kesalahan Input:</h3>";
28         foreach ($errors as $error) {
29             echo "<p style='color:red;'>$error</p>";
30         }
31         echo '<br><a href="html_validation.html">Kembali ke Form</a>';
32     } else {
33         echo "<h3>Data berhasil dikirim:</h3>";
34         echo "Nama: $nama<br>";
35         echo "Email: $email<br>";
36         echo "Password: (disembunyikan demi keamanan)<br>";
37         echo '<br><a href="html_validation.html">Kembali ke Form</a>';
38     }
39 }
40 ?>
41
```

The screenshot shows a web browser window with the URL `localhost/dasarWeb/JS07_PHP-jQuery/fo...`. The page title is "E-KINERJA". The main content is a form titled "Form Input dengan Validasi Password". It contains three input fields: "Nama: ganang", "Email: ganangputra939@gmail.com", and "Password:". Below the form is a "Submit" button.

Form Input dengan Validasi Password

Nama:

Email:

Password:

Data berhasil dikirim:

Nama: ganang
Email: ganangputra939@gmail.com
Password: (disembunyikan demi keamanan)

[Kembali ke Form](#)