JOBSHEET 7

PHP – Form Processing

Web Design and Programming Courses

Febrian Arka Samudra

2341720066 / 10

STUDY PROGRAM D-IV INFORMATICS ENGINEERING

INFORMATICS TECHNOLOGY MAJOR

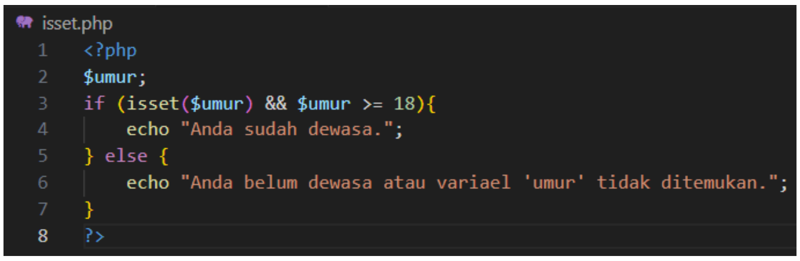
POLITEKNIK NEGERI MALANG

Jl.Soekarno Hatta No.9,Jatimulyo, kec, Lowokwaru, Kota Malang, Jawa Timur 65141

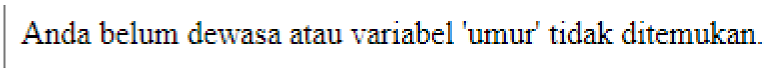
Practical Section 1. Function isset()

What do you understand from using the isset on the file?

(Question No. 1)



```php
isset.php
<?php
$umur;
if (isset($umur) && $umur >= 18){
    echo "Anda sudah dewasa.";
} else {
    echo "Anda belum dewasa atau variael 'umur' tidak ditemukan.";
}
?>
```

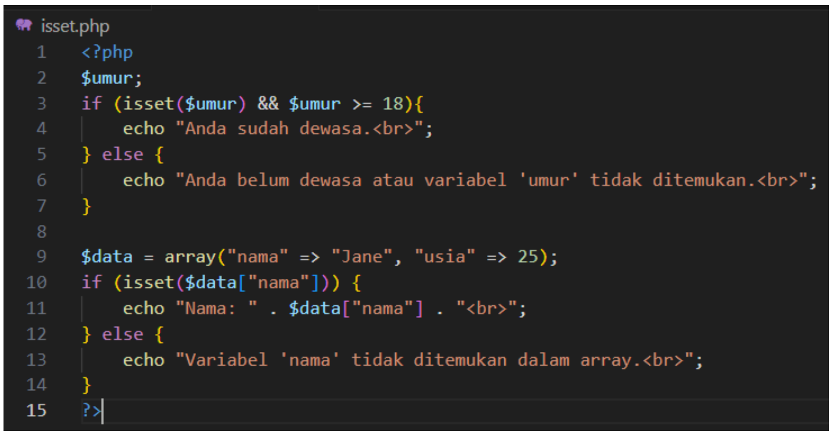


```
Anda belum dewasa atau variabel 'umur' tidak ditemukan.
```

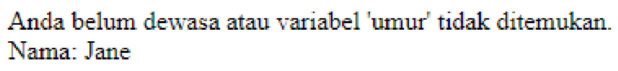
Explanation :

- **isset($umur)** checks whether the variable $umur is set.
- If $umur is set **and** the condition $umur >= 18 is true, the message "Anda sudah dewasa." (You are an adult) will be displayed.
- If $umur is not set or if it is less than 18, the message "Anda belum dewasa atau variabel 'umur' tidak ditemukan." (You are not an adult, or the 'umur' variable was not found) will be shown.

Save the file, then open the browser and run localhost/week7/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. 2)



```php
isset.php
<?php
$umur;
if (isset($umur) && $umur >= 18){
    echo "Anda sudah dewasa.<br>";
} else {
    echo "Anda belum dewasa atau variabel 'umur' tidak ditemukan.<br>";
}

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



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

Explanation :

1. **For $umur:**

- isset($umur) checks if the $umur variable is declared and has a value.
- If $umur is not set or less than 18, it outputs that you are not an adult, or the variable is not found.

2. **For $data["nama"]:**

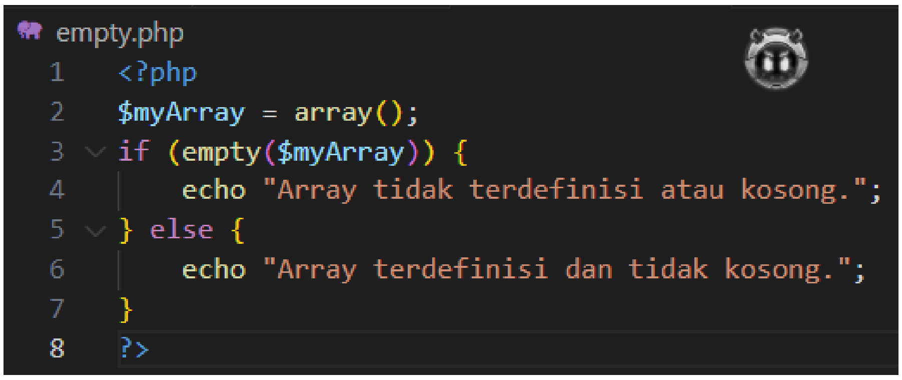
- isset($data["nama"]) checks if the "nama" key exists in the $data array.
- If it exists, it outputs "Nama: Jane", otherwise, it says the "nama" variable is not found in the array.

Practical Section 2. Function empty()

Save the file, then open a browser and run localhost/week7/empty.php

What do you understand from the use of empty on the file? Write your understanding below.

(Question No. 3)



empty.php

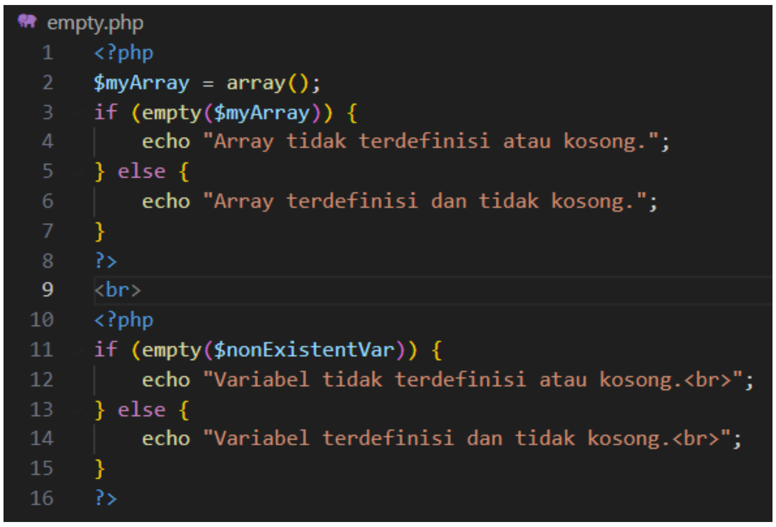
```php
<?php
$myArray = array();
if (empty($myArray)) {
    echo "Array tidak terdefinisi atau kosong.";
} else {
    echo "Array terdefinisi dan tidak kosong.";
}
?>
```

```
Array tidak terdefinisi atau kosong.
```

Explanation :

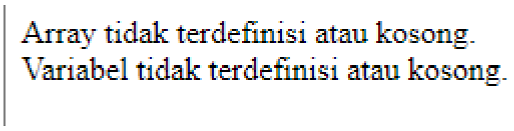
- Since $myArray is initialized as an empty array (array()), empty($myArray) returns true.
- The output is: Array tidak terdefinisi atau kosong.
- **empty()** is useful for checking whether a variable, including arrays, is not set or holds an empty value.
- For arrays, empty() returns true if the array has no elements.

Save the file, then open the browser and run localhost/week7/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. 4)



empty.php

```php
<?php
$myArray = array();
if (empty($myArray)) {
    echo "Array tidak terdefinisi atau kosong.";
} else {
    echo "Array terdefinisi dan tidak kosong.";
}
?>
<br>
<?php
if (empty($nonExistentVar)) {
    echo "Variabel tidak terdefinisi atau kosong.<br>";
} else {
    echo "Variabel terdefinisi dan tidak kosong.<br>";
}
?>
```



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

Explanation :

The empty() function in the provided PHP code is used to check if variables or arrays are not set or contain no meaningful values.

For $myArray:

- The array is initialized as empty (array()), so empty($myArray) returns true.
- The output is:Array tidak terdefinisi atau kosong.

For $nonExistentVar:

- This variable is not defined, and empty($nonExistentVar) also returns true.
- The output is:Variabel tidak terdefinisi atau kosong.

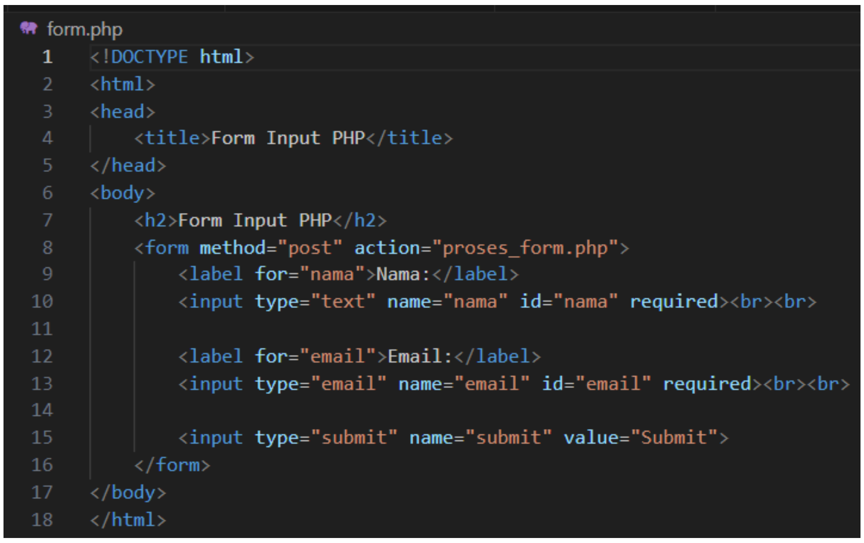
- **empty()** checks both arrays and variables for emptiness, returning true if they are not set or contain no elements.
- It prevents errors by allowing safe checks on variables and arrays without causing runtime errors when trying to access undefined variables.

Practical Section 3. PHP Input Form

Save the file, then open a browser and run localhost/week7/proses_form.php. Explain what happened and write your understanding below.

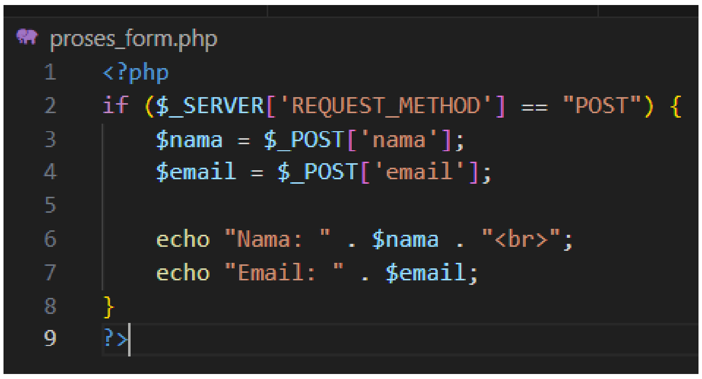
Then run localhost/week7/form.php. Explain what happened and write your understanding below.

(Question No. 5)



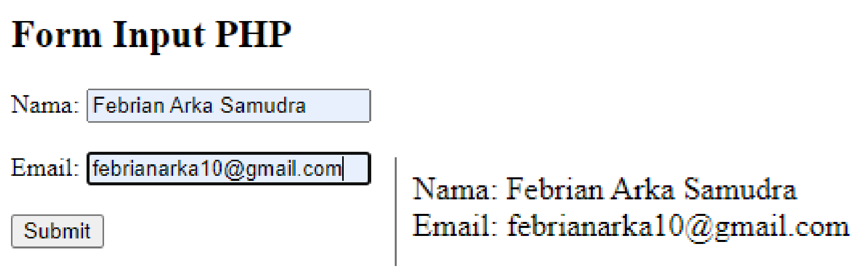
form.php

```html
<!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>
```



proses_form.php

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

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



Form Input PHP

Nama: Febrian Arka Samudra

Email: febrianarka10@gmail.com

Submit

```
Nama: Febrian Arka Samudra
Email: febrianarka10@gmail.com
```

Explanation :

1. **Running localhost/week7/proses_form.php:**

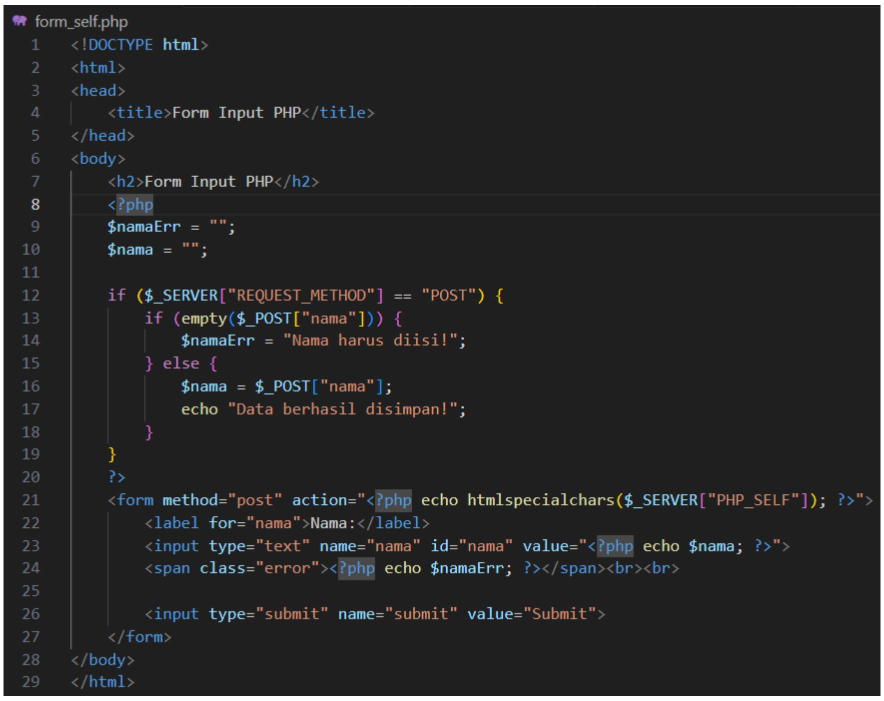
- This file processes data submitted from a form.
- It checks if the request method is POST, retrieves the nama and email values from the $_POST array, and displays them.

- If a user fills out the form and submits it, the output will show the entered name and email in the browser.

2. **Running localhost/week7/form.php:**

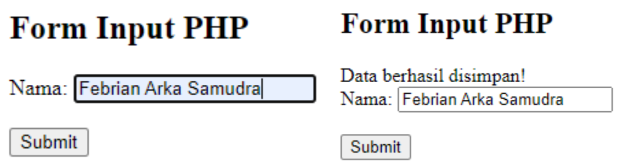
- This file presents an HTML form to users, allowing them to input their name and email.
- The form uses the POST method to send data to proses_form.php when submitted.
- It includes required attributes on input fields to ensure users enter valid data before submission.

Save the file, then open a browser and run localhost/week7/ form_self.php. What do you understand from the use of forms in the file? Write your understanding below. (Question No. 6)



form_self.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Form Input PHP</title>
</head>
<body>
    <h2>Form Input PHP</h2>
    <?php
    $namaErr = "";
    $nama = "";

    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        if (empty($_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>
```



Form Input PHP

Nama: Febrian Arka Samudra

Submit

Form Input PHP

Data berhasil disimpan!
Nama: Febrian Arka Samudra

Submit

Explanation :

1. **Form Structure**:

- The form includes an input field for the user's name and a submit button.
- It uses the POST method and submits data back to the same script, indicated by action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>".

2. **Data Validation**:

- The script checks if the form has been submitted using $_SERVER["REQUEST_METHOD"] == "POST".
- It validates the input by checking if the nama field is empty. If it is empty, an error message is generated.

3. **User Feedback**:

- Upon successful submission (when the name is filled), a success message ("Data berhasil disimpan!") is displayed.
- If the input is empty, an error message is shown next to the input field, guiding the user to provide the required information.

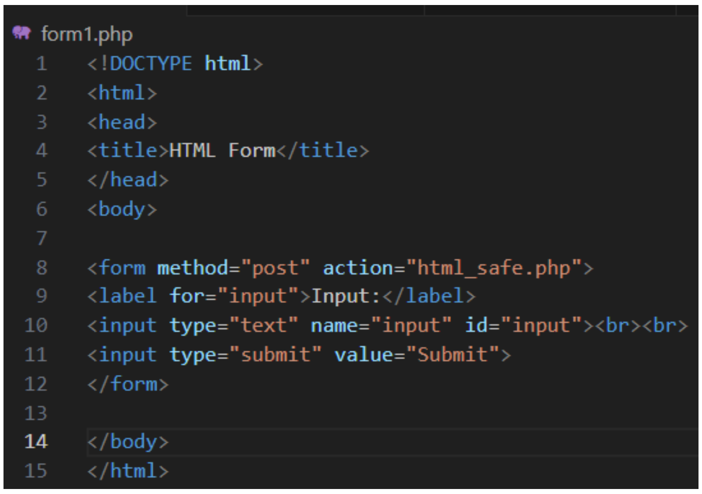
4. **Preserving Input**:

- The form retains the entered name in the input field after submission, improving user experience by allowing them to see what they typed.

Practical Section 4. HTML Injection

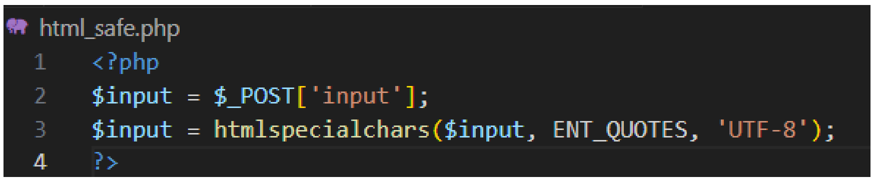
Record here what you observed, give your explanation.

(Question No. 7)



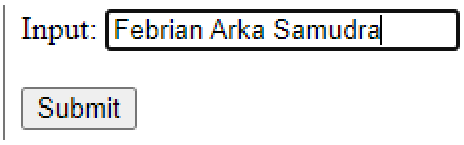
form1.php

```html
<!DOCTYPE html>
<html>
<head>
<title>HTML Form</title>
</head>
<body>

<form method="post" action="html_safe.php">
<label for="input">Input:</label>
<input type="text" name="input" id="input"><br><br>
<input type="submit" value="Submit">
</form>

</body>
</html>
```



html_safe.php

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

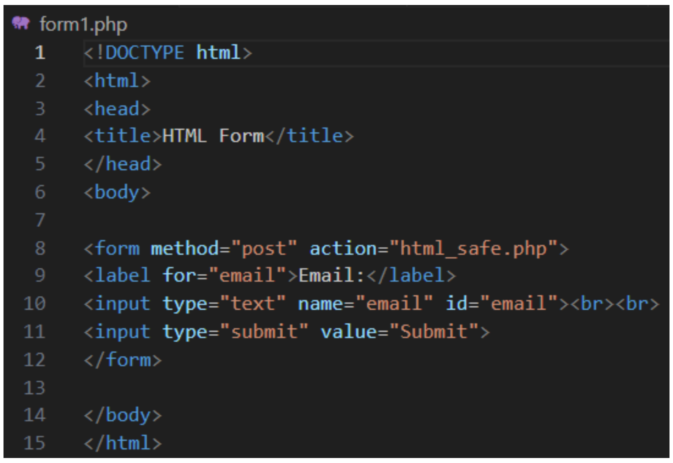

Input: Febrian Arka Samudra

Explanation :

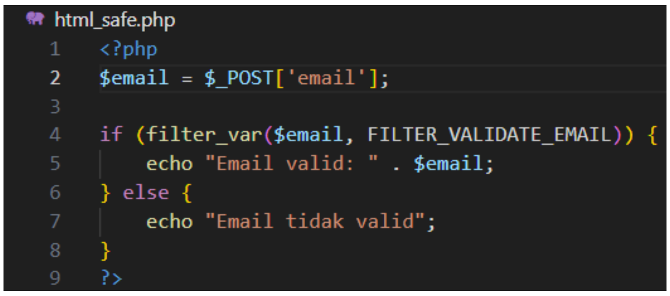
- **With htmlspecialchars()**: When users input HTML tags (e.g., <script>alert('XSS');</script>), this function converts them into HTML entities (like &lt;script&gt;), preventing execution as HTML.
- **Without htmlspecialchars()**: Omitting this function allows the browser to execute input as HTML, leading to potential Cross-Site Scripting (XSS) attacks where malicious scripts can be injected.
- htmlspecialchars() is essential for web application security. It changes special characters (<, >, &, ', and ") into their HTML entities, ensuring they are displayed safely.
- Always use htmlspecialchars() before outputting user-generated content in HTML to safeguard against XSS vulnerabilities.
- This practice ensures that user input is rendered harmless and displayed as intended, maintaining the integrity of the website.

Note here what you observe from the addition of the program code above.

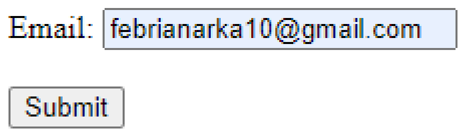
(Question No. 8)

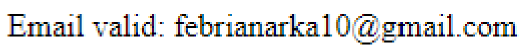

form1.php

```html
<!DOCTYPE html>
<html>
<head>
<title>HTML Form</title>
</head>
<body>

<form method="post" action="html_safe.php">
<label for="email">Email:</label>
<input type="text" name="email" id="email"><br><br>
<input type="submit" value="Submit">
</form>

</body>
</html>
```


html_safe.php

```php
<?php
$email = $_POST['email'];

if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
    echo "Email valid: " . $email;
} else {
    echo "Email tidak valid";
}
?>
```


Email: febrianarka10@gmail.com


Email valid: febrianarka10@gmail.com

Explanation :

1. Email Retrieval:

- The code retrieves the submitted email address from the form and assigns it to the $email variable.

2. Validation Process:

- The filter_var() function checks if the $email is formatted correctly. It returns the sanitized email if valid or false if invalid.

3. Handling Valid Emails:

- If the email is valid, the code executes the if block, displaying a message confirming that the email is valid.

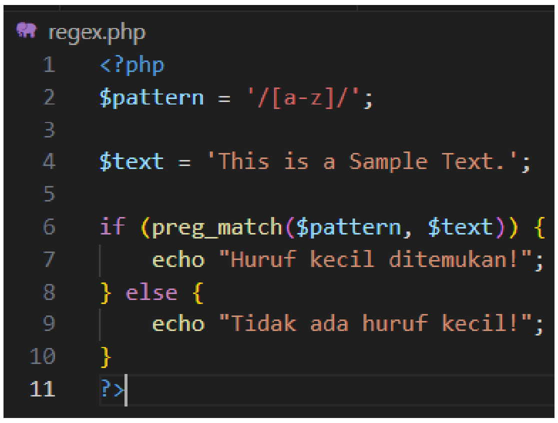
4. Handling Invalid Emails:

- If the email fails validation, the else block runs, showing a message indicating the email is not valid.

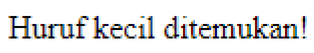
Practical Section 5 : The Use of Regex in PHP

Note here what you observe from the addition of the program code above.

(Question No. 9)


regex.php

```php
<?php
$pattern = '/[a-z]/';

$text = 'This is a Sample Text.';

if (preg_match($pattern, $text)) {
    echo "Huruf kecil ditemukan!";
} else {
    echo "Tidak ada huruf kecil!";
}
?>
```


```
Huruf kecil ditemukan!
```

Explanation :

1. **Regular Expression Pattern:**

- $pattern = '/[a-z]/'; defines a regular expression pattern that matches any single lowercase letter from 'a' to 'z'.
- The pattern is enclosed within forward slashes / to indicate that it's a regular expression.

2. **Text String:**

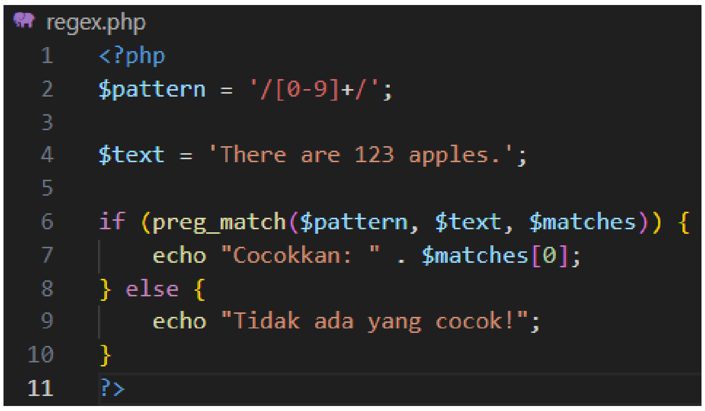
- $text = 'This is a Sample Text.'; sets a text string that will be searched for lowercase letters.

3. **Regular Expression Matching:**

- if (preg_match($pattern, $text)) { ... } else { ... } uses the preg_match function to check if the regular expression pattern matches any part of the text string.
- If a match is found, the if block is executed, and the message "Huruf kecil ditemukan!" (Lowercase letter found) is printed.
- If no match is found, the else block is executed, and the message "Tidak ada huruf kecil!" (No lowercase letter found) is printed.

Note here what you observe from the addition of the program code above.

(Question No. 10)



regex.php

```php
<?php
$pattern = '/[0-9]+/';

$text = 'There are 123 apples.';

if (preg_match($pattern, $text, $matches)) {
    echo "Cocokkan: " . $matches[0];
} else {
    echo "Tidak ada yang cocok!";
}
?>
```

```
Cocokkan: 123
```

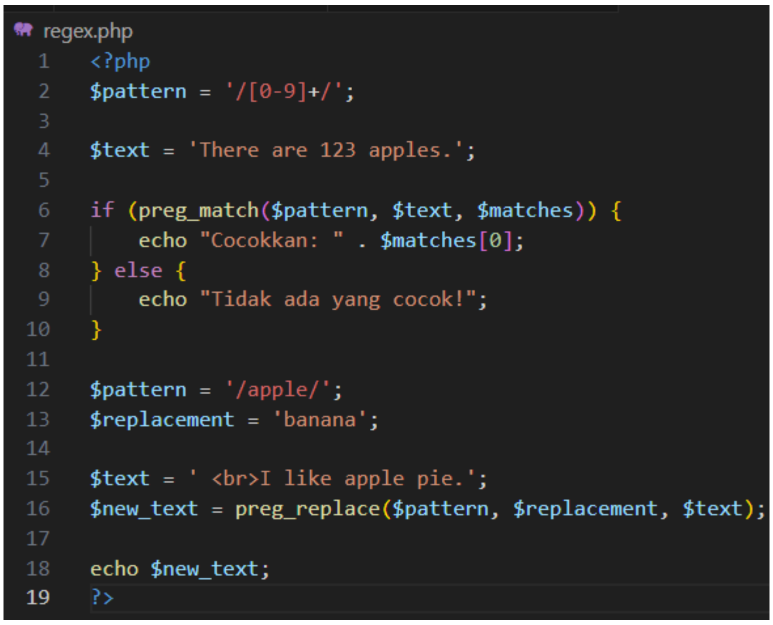
Explanation :

- The pattern '/[0-9]+/' is used to match one or more digits (0-9) in the input string.
- The code searches the string 'There are 123 apples.', which contains the numeric sequence 123.
- The preg_match() function checks the input string against the pattern. If a match is found, it returns 1, and the matched digits are stored in the $matches array.
- If a match is found, the output is "Cocokkan: " followed by the matched digits (123).

- If no match is found, the output would be "Tidak ada yang cocok!", indicating no matches were detected.

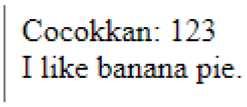
Note here what you observe from the addition of the program code above.

(Question No. 11)



regex.php

```php
<?php
$pattern = '/[0-9]+/';

$text = 'There are 123 apples.';

if (preg_match($pattern, $text, $matches)) {
    echo "Cocokkan: " . $matches[0];
} else {
    echo "Tidak ada yang cocok!";
}

$pattern = '/apple/';
$replacement = 'banana';

$text = ' <br>I like apple pie.';
$new_text = preg_replace($pattern, $replacement, $text);

echo $new_text;
?>
```



```
Cocokkan: 123
I like banana pie.
```

Explanation :

1. **Pattern Matching**:

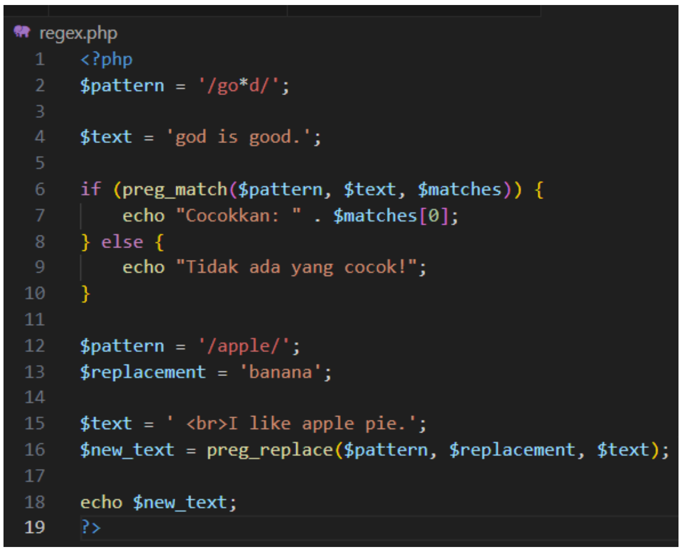
- The code uses the pattern '/[0-9]+/' to search for numeric sequences in the string 'There are 123 apples.'.
- The preg_match() function identifies and captures the digits, outputting "Cocokkan: 123" if a match is found. If no match is found, it would display "Tidak ada yang cocok!".

2. **Pattern Replacement**:

- The code then defines a new pattern '/apple/' and specifies banana as the replacement string.
- The preg_replace() function searches for the word "apple" in the string 'I like apple pie.' and replaces it with "banana".
- The resulting output after replacement is "I like banana pie.".

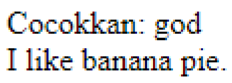
Note here what you observe from the addition of the program code above.

(Question No. 12)



regex.php

```php
<?php
$pattern = '/go*d/';

$text = 'god is good.';

if (preg_match($pattern, $text, $matches)) {
    echo "Cocokkan: " . $matches[0];
} else {
    echo "Tidak ada yang cocok!";
}

$pattern = '/apple/';
$replacement = 'banana';

$text = ' <br>I like apple pie.';
$new_text = preg_replace($pattern, $replacement, $text);

echo $new_text;
?>
```



```
Cocokkan: god
I like banana pie.
```

Explanation :

1. **Pattern Matching**:

- The code uses the pattern '/go*d/', which matches the letter 'g' followed by zero or more occurrences of 'o' and ending with 'd'.
- It searches the string 'god is good.', successfully finding the match god.
- The output for this part is "Cocokkan: god", confirming the successful match.

2. **Pattern Replacement**:

- The code then applies the pattern '/apple/', intending to replace occurrences of "apple" with "banana".
- It processes the string 'I like apple pie.', where "apple" is present.
- The preg_replace() function replaces "apple" with "banana", resulting in "I like banana pie."

3. **Output**:

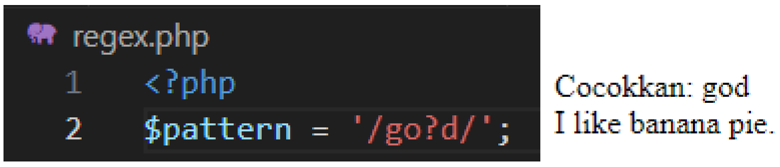
- The final output displays both results: the successful match ("Cocokkan: god") and the modified string ("I like banana pie.").

In the script in step 14, change the variable pattern from '*' to '?'.

Save the file, then open a browser and run /refresh localhost/week7/regex.php

Note here what you observe from the addition of the program code above.

(Question No. 13)



```
regex.php
<?php
$pattern = '/go?d/';
```

```
Cocokkan: god
I like banana pie.
```

Explanation :

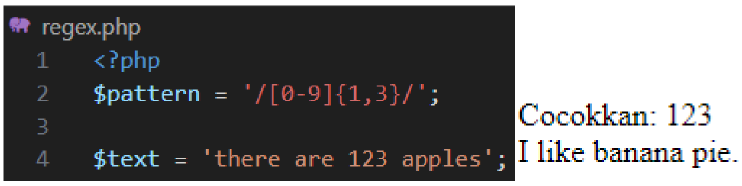
- The regex pattern '/go?d/' is designed to match strings that begin with "g", followed by zero or one occurrence of "o", and end with "d". The ? quantifier makes the "o" optional.
- In the input string 'god is good.', the pattern successfully matches the substring god.
- The output indicates a successful match with "Cocokkan: god".

In the script in step 14, change the variable pattern to '/[o]{1,3}/'.

Save the file, then open a browser and run /refresh localhost/week7/regex.php

Note here what you observe from the addition of the program code above.

(Question No 14)



```
regex.php
<?php
$pattern = '/[0-9]{1,3}/';

$text = 'there are 123 apples';
```

```
Cocokkan: 123
I like banana pie.
```

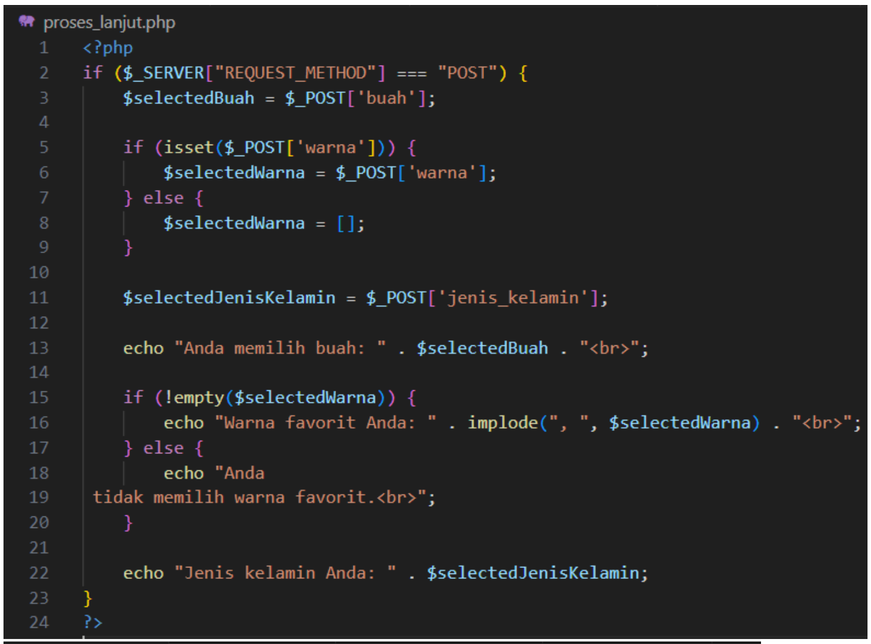
Explanation :

- The regex pattern '/[0-9]{1,3}/' is employed to find sequences of one to three consecutive digits within a text. Here, [0-9] matches any digit, while {1,3} indicates that the match can consist of one to three occurrences.
- In the input string 'There are 123 apples.', the pattern successfully matches the sequence 123.
- The output for this match is "Cocokkan: 123", confirming the successful identification of the digit pattern.

Practical Section 6 : Advanced Form

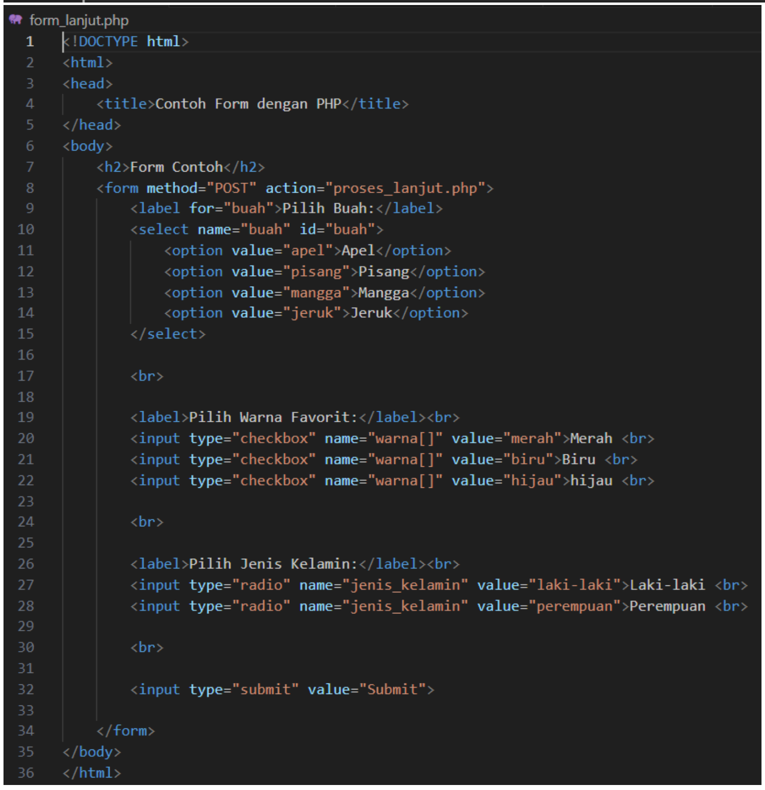
Note here what you observe from the program code above.

(Question No. 15)



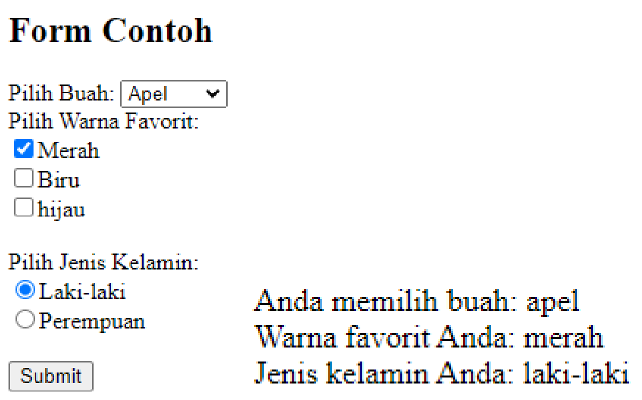
proses_lanjut.php

```php
<?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;
}
?>
```



form_lanjut.php

```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>

        <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>
</body>
</html>
```



Form Contoh

Pilih Buah: Apel

Pilih Warna Favorit:

[✓ Merah]
Biru
hijau

Pilih Jenis Kelamin:

[✓ Laki-laki]
Perempuan

Submit

```
Anda memilih buah: apel
Warna favorit Anda: merah
Jenis kelamin Anda: laki-laki
```

Explanation :

1. **Form Structure**:

- The HTML form utilizes the POST method to send data to proses_lanjut.php.
- It includes a dropdown menu for selecting one fruit, checkboxes for selecting multiple favorite colors, and radio buttons for gender selection.

2. **Input Elements**:

- **Fruit Selection**: A <select> element lets users choose from four fruit options: Apple, Banana, Mango, and Orange.
- **Color Preferences**: Checkboxes allow users to select multiple favorite colors (Red, Blue, Green).
- **Gender Selection**: Radio buttons enable users to specify their gender (Male or Female).

3. **Data Handling**:

- The PHP script checks if the form is submitted via POST.
- It retrieves the selected fruit and handles the selected colors as an array.
- If no colors are selected, it initializes an empty array for the colors.
- The selected gender is also retrieved.

4. **Output Display**:

- The script outputs the selected fruit.
- It lists the selected favorite colors using implode() if any are chosen; if none are selected, it informs the user.
- The selected gender is displayed as well.

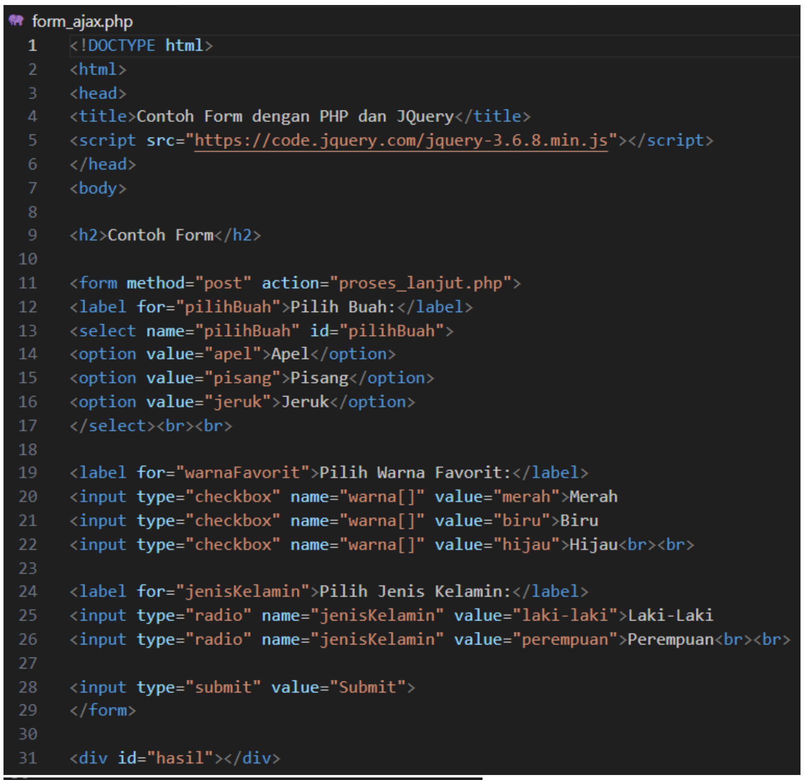
5. **User Feedback**:

- The code effectively provides user-friendly messages based on the selections, ensuring clarity in communication.

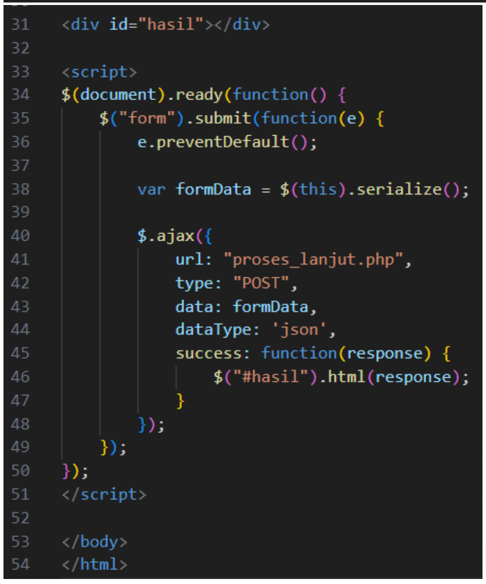
Save the file, then open a browser and run /refresh localhost/week7/form_ajax.php.

Note here what you observe from the addition of the program code above.

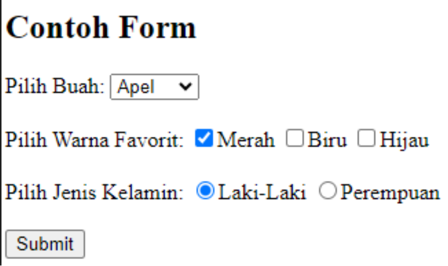
(Question No. 16)


form_ajax.php

```html
<!DOCTYPE html>
<html>
<head>
<title>Contoh Form dengan PHP dan JQuery</title>
<script src="https://code.jquery.com/jquery-3.6.8.min.js"></script>
</head>
<body>

<h2>Contoh Form</h2>

<form method="post" action="proses_lanjut.php">
<label for="pilihBuah">Pilih Buah:</label>
<select name="pilihBuah" id="pilihBuah">
<option value="apel">Apel</option>
<option value="pisang">Pisang</option>
<option value="jeruk">Jeruk</option>
</select><br><br>

<label for="warnaFavorit">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<br><br>

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

<input type="submit" value="Submit">
</form>

<div id="hasil"></div>
```



```html
<div id="hasil"></div>

<script>
$(document).ready(function() {
    $("form").submit(function(e) {
        e.preventDefault();

        var formData = $(this).serialize();

        $.ajax({
            url: "proses_lanjut.php",
            type: "POST",
            data: formData,
            dataType: 'json',
            success: function(response) {
                $("#hasil").html(response);
            }
        });
    });
});
</script>

</body>
</html>
```



Contoh Form

Pilih Buah: Apel

Pilih Warna Favorit: [✓ Merah] Biru Hijau

Pilih Jenis Kelamin: [✓ Laki-Laki] Perempuan

Explanation :

1. **Form Fields**:

- You are presented with a form containing a dropdown to select a fruit, checkboxes to select favorite colors, and radio buttons to select gender.

2. **After Submission**:

- Once the form is submitted, the PHP script processes the POST request.

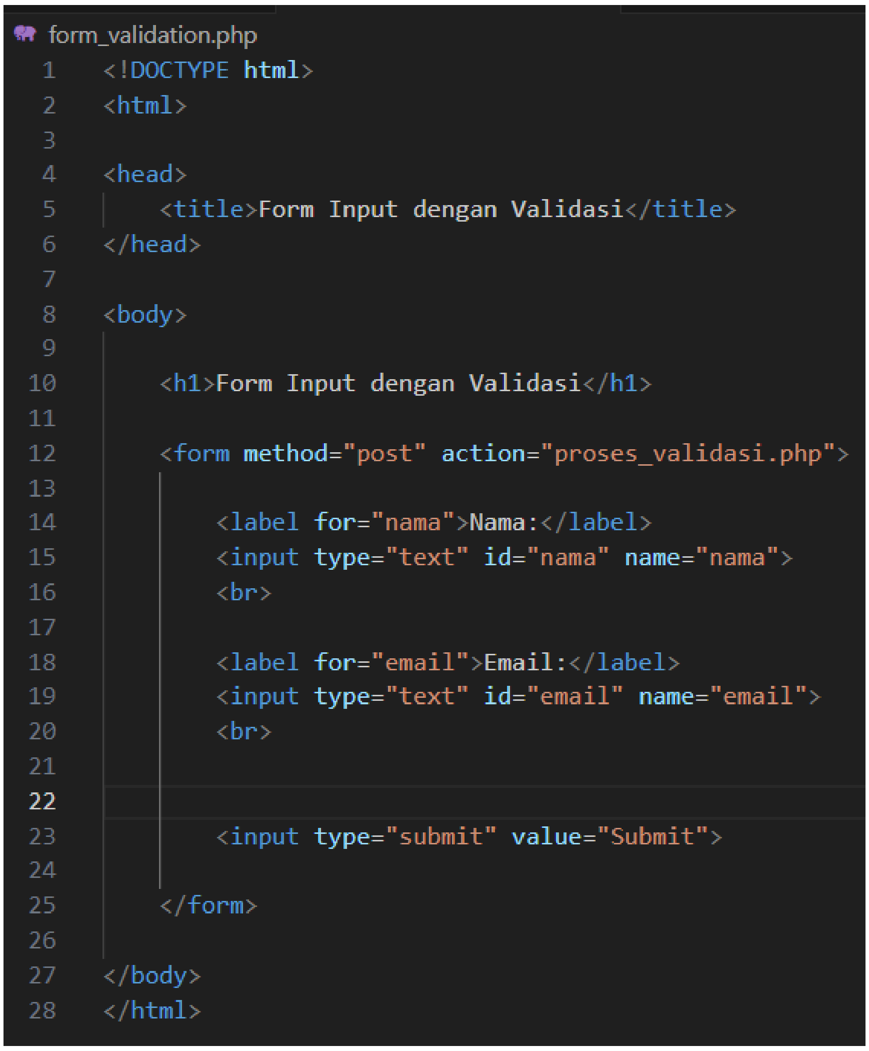
  The page will display the selected options:

  - The fruit you selected will be displayed (e.g., "Anda memilih buah: Apel").
  - If you selected any colors, they will be listed (e.g., "Warna favorit Anda: Merah, Biru").
  - If no colors were selected, it will display a message saying, "Anda tidak memilih warna favorit."
  - The gender you selected will be displayed (e.g., "Jenis kelamin Anda: Laki-laki").

Practical Section 7 : Form Validation

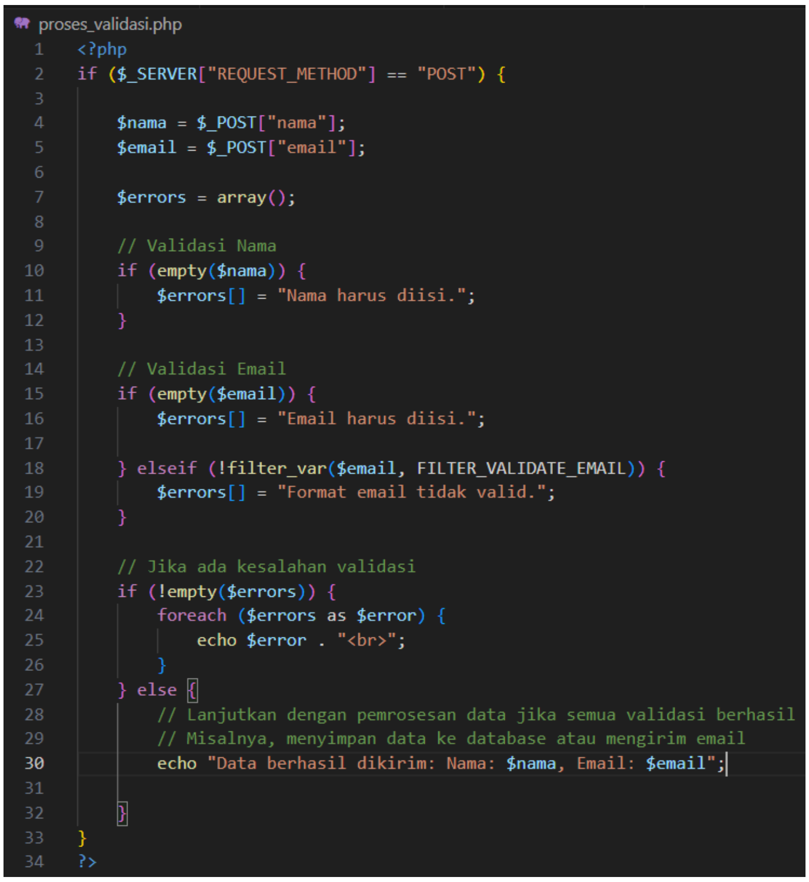
Note here what you observe from the addition of the program code above.

(Question No. 17)



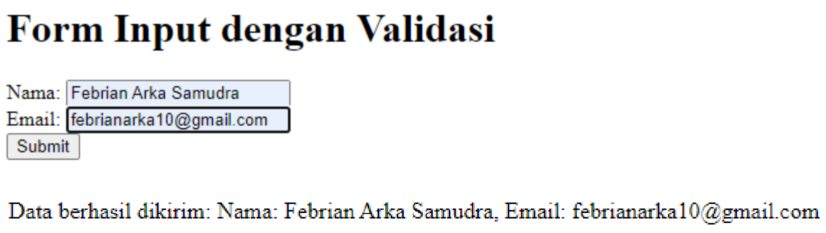
form_validation.php

```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>
```



proses_validasi.php

```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";

    }
}
?>
```



Form Input dengan Validasi

Nama: Febrian Arka Samudra
Email: febrianarka10@gmail.com

Data berhasil dikirim: Nama: Febrian Arka Samudra, Email: febrianarka10@gmail.com

Explanation :

1. **Validation**:

- If the **Name** field is empty, it shows: "Nama harus diisi."
- If the **Email** field is empty or has an invalid format, it shows appropriate error messages.

2. **Error Handling**:

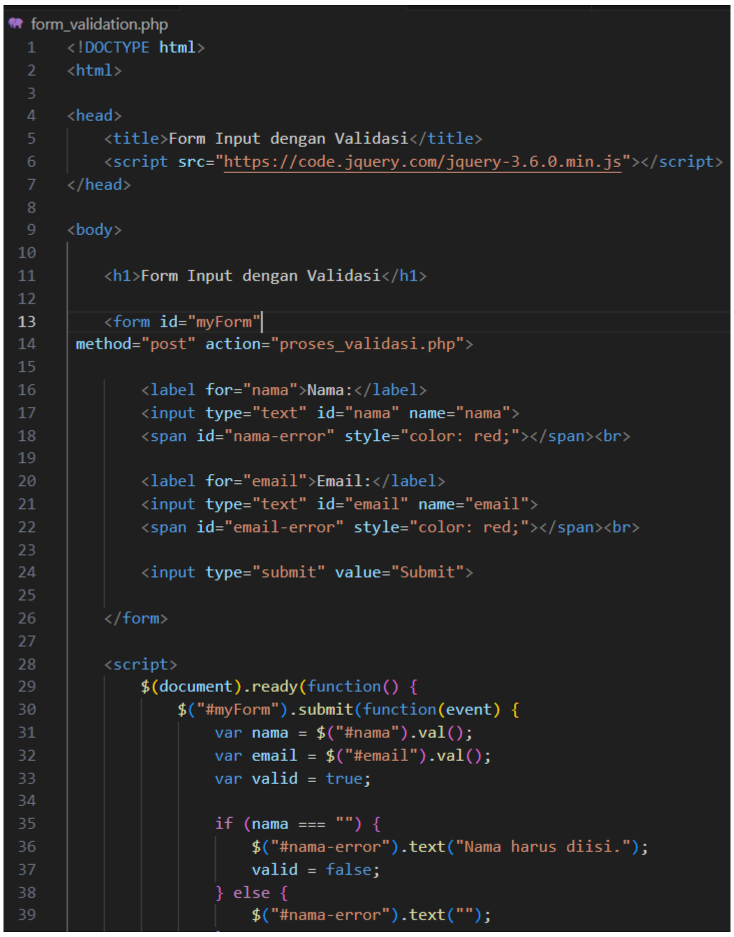
- Errors are displayed if validation fails for either field.

3. **Success**:

- If both fields are valid, the form successfully submits, and a success message is shown with the entered name and email.

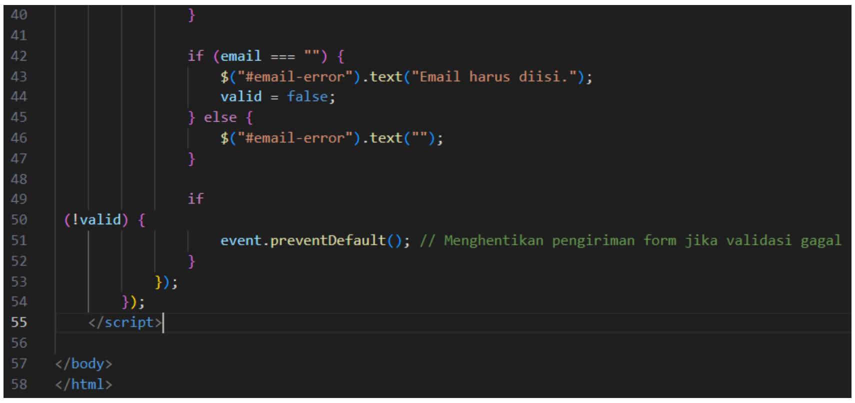
Note here what you observe from the addition of the program code above.

(Question No. 18)

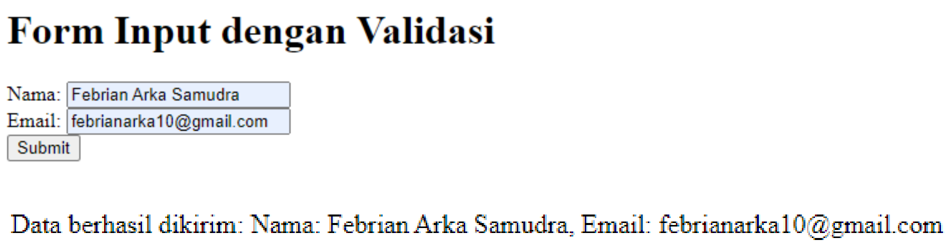


form_validation.php

```
<!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>
```



Form Input dengan Validasi

Nama: Febrian Arka Samudra
Email: febrianarka10@gmail.com
Submit

Data berhasil dikirim: Nama: Febrian Arka Samudra, Email: febrianarka10@gmail.com

Explanation :

1. **Real-Time Validation (Client-Side)**:

- When you try to submit the form, the input fields (Name and Email) are validated on the client side before the form is sent to the server.
- If the **Nama** field is empty, an error message "Nama harus diisi." appears in red next to the field.
- Similarly, if the **Email** field is empty, "Email harus diisi." is shown in red next to the email input.

2. **Prevention of Form Submission**:

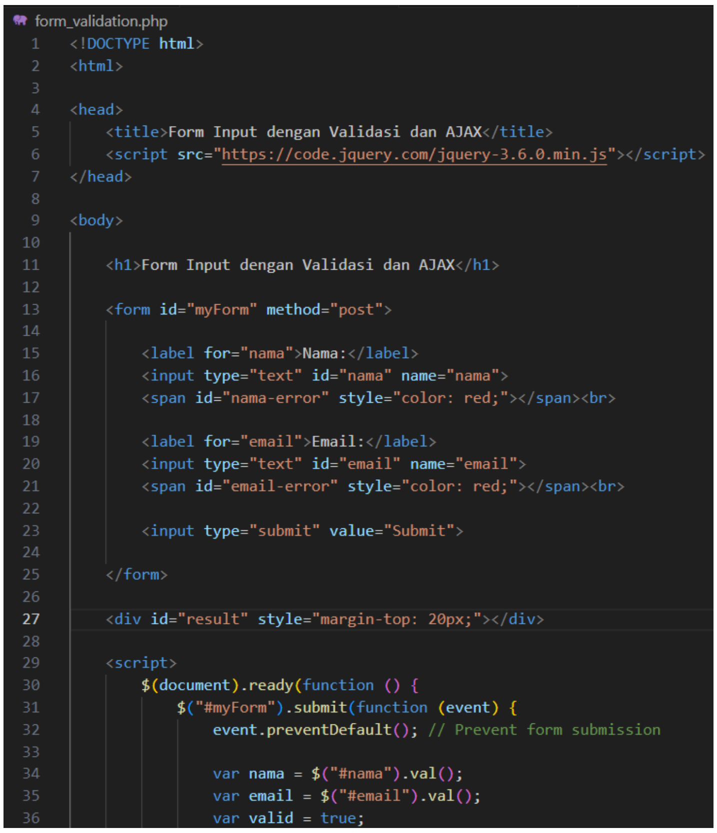
- If either the **Nama** or **Email** fields are empty, the form submission is prevented, and you are required to correct the input before submitting again.
- The error messages clear once valid input is provided.

3. **Improved User Experience**:

- The validation occurs instantly when you attempt to submit, providing immediate feedback, without having to reload the page or wait for server-side validation.

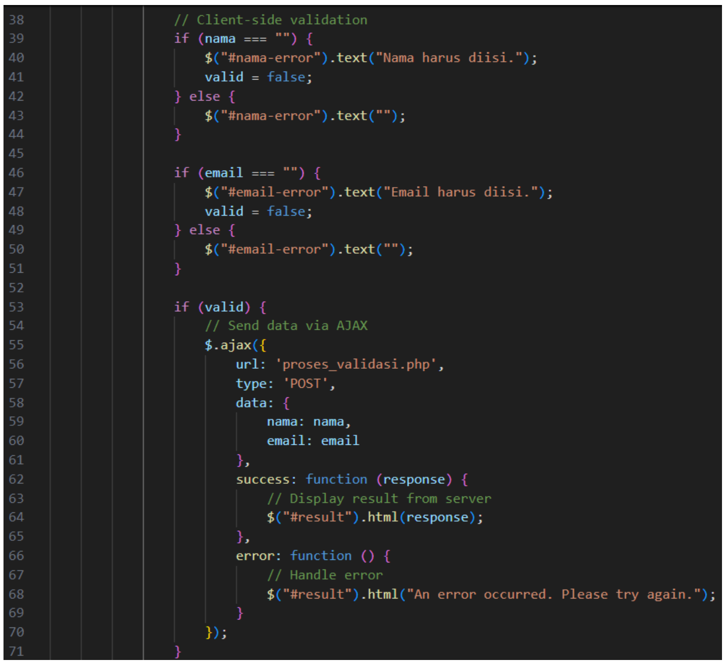
Create a script for step 6 using ajax. Screen shoot the code and wrote here what you observe from the addition of the program code.

(Question No. 19)

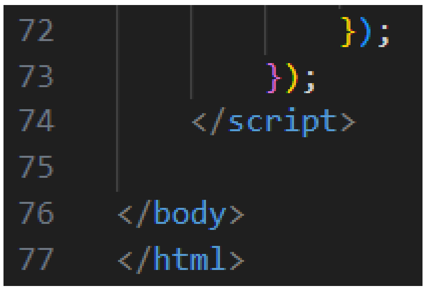


form_validation.php

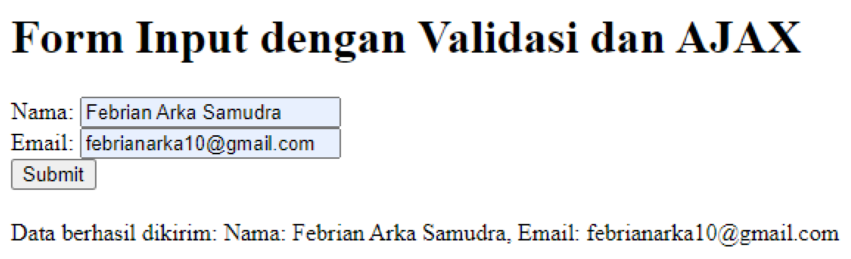
```
<!DOCTYPE html>
<html>

<head>
    <title>Form Input dengan Validasi dan AJAX</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>

<body>

    <h1>Form Input dengan Validasi dan AJAX</h1>

    <form id="myForm" method="post">

        <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>

    <div id="result" style="margin-top: 20px;"></div>

    <script>
        $(document).ready(function () {
            $("#myForm").submit(function (event) {
                event.preventDefault(); // Prevent form submission

                var nama = $("#nama").val();
                var email = $("#email").val();
                var valid = true;
```



```
            // Client-side validation
            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) {
                // Send data via AJAX
                $.ajax({
                    url: 'proses_validasi.php',
                    type: 'POST',
                    data: {
                        nama: nama,
                        email: email
                    },
                    success: function (response) {
                        // Display result from server
                        $("#result").html(response);
                    },
                    error: function () {
                        // Handle error
                        $("#result").html("An error occurred. Please try again.");
                    }
                });
            }
```



```
            });
        });
    </script>

</body>
</html>
```



Form Input dengan Validasi dan AJAX

Nama: Febrian Arka Samudra
Email: febrianarka10@gmail.com
Submit

Data berhasil dikirim: Nama: Febrian Arka Samudra, Email: febrianarka10@gmail.com

Explanation :

1. **Asynchronous Form Submission**:

- When the form is submitted, the data is sent to the server using Ajax. The page doesn't reload, and the data is processed in the background.

2. **Real-Time Feedback**:

- The PHP script processes the form data, and the result (either validation errors or success messages) is displayed in the <div id="result"> element without refreshing the page.

3. **Instant Validation**:

- The form is first validated on the client side using jQuery to ensure all required fields are filled in.
- If the fields are valid, the Ajax request sends the data to the server for further processing.

4. **Error Handling**:

- If there is an issue with the server response (e.g., server downtime), an error message is displayed to the user, improving reliability.

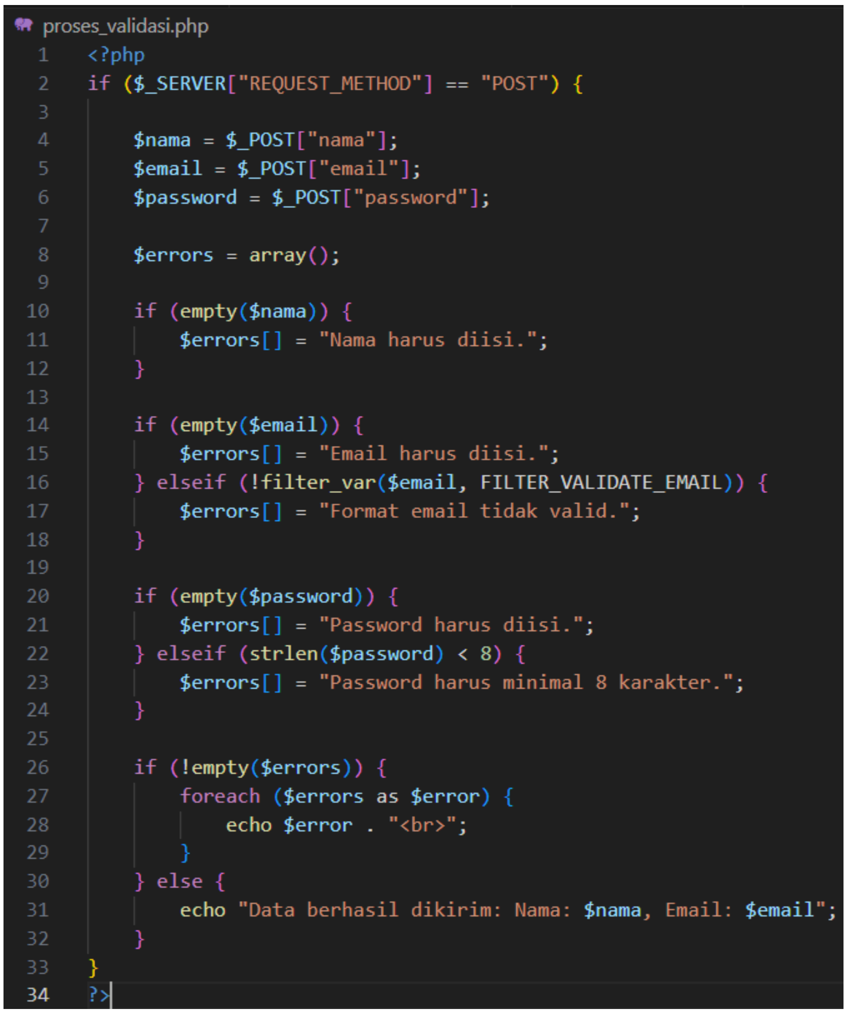
5. **Improved User Experience**:

- The form submission process is more seamless and faster since the page doesn't need to reload. Users receive immediate feedback after submitting their information.

Add code for password validation with a minimum of 8 characters using jQuery and PHP.

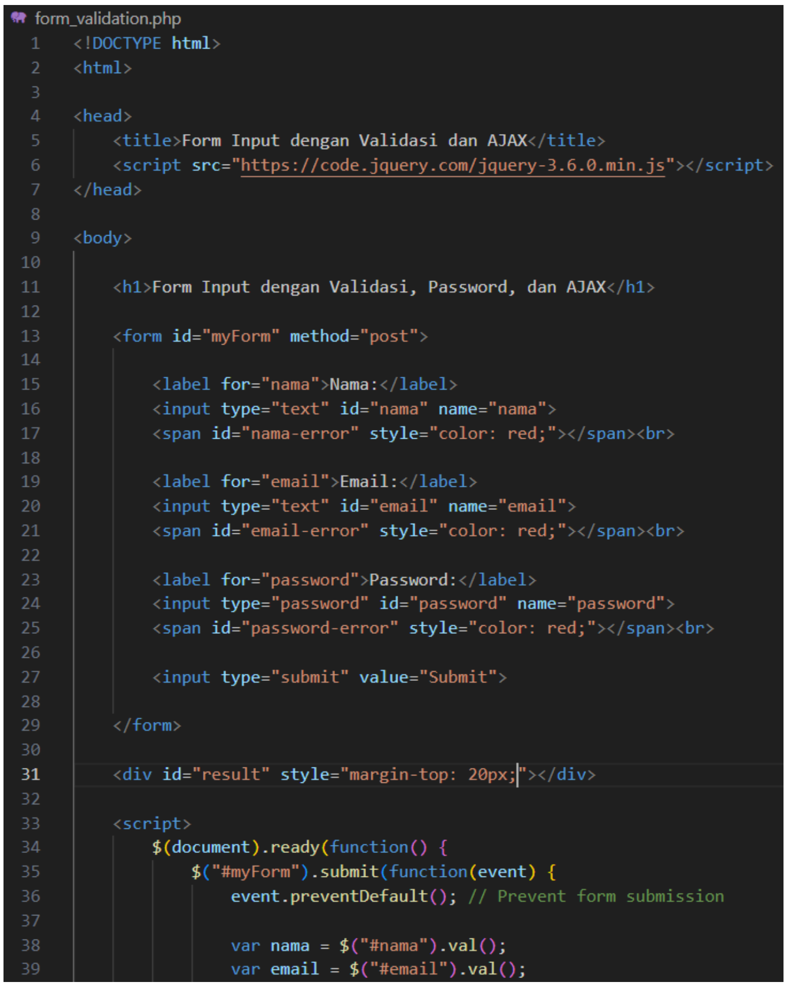
Screen shoot the code and note here what you observe from the addition of the program code.

(Question No. 20)



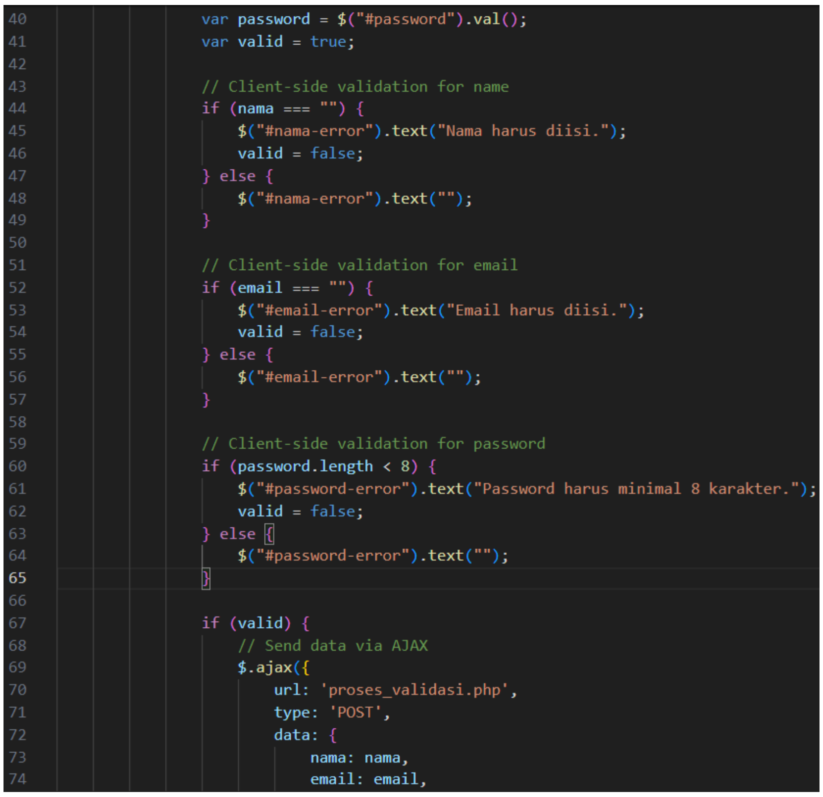
proses_validasi.php

```php
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {

    $nama = $_POST["nama"];
    $email = $_POST["email"];
    $password = $_POST["password"];

    $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($password)) {
        $errors[] = "Password harus diisi.";
    } elseif (strlen($password) < 8) {
        $errors[] = "Password harus minimal 8 karakter.";
    }

    if (!empty($errors)) {
        foreach ($errors as $error) {
            echo $error . "<br>";
        }
    } else {
        echo "Data berhasil dikirim: Nama: $nama, Email: $email";
    }
}
?>
```

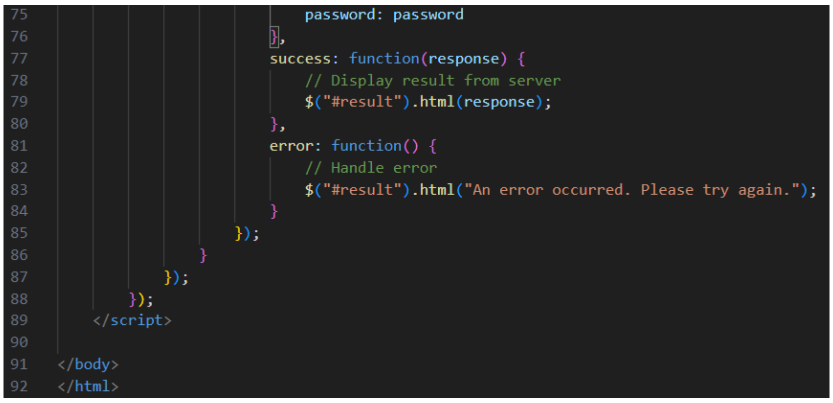


form_validation.php

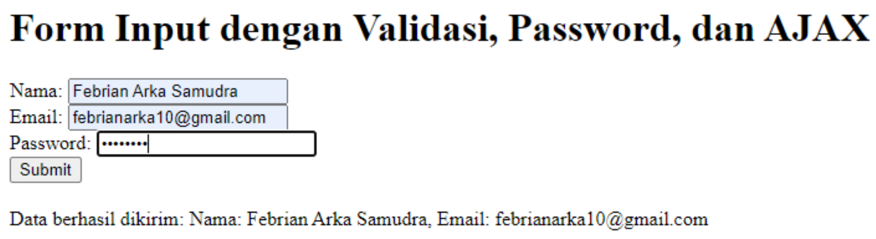
```
<!DOCTYPE html>
<html>

<head>
    <title>Form Input dengan Validasi dan AJAX</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>

<body>

    <h1>Form Input dengan Validasi, Password, dan AJAX</h1>

    <form id="myForm" method="post">

        <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>

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

        <input type="submit" value="Submit">

    </form>

    <div id="result" style="margin-top: 20px;"></div>

    <script>
        $(document).ready(function() {
            $("#myForm").submit(function(event) {
                event.preventDefault(); // Prevent form submission

                var nama = $("#nama").val();
                var email = $("#email").val();
```



```
                var password = $("#password").val();
                var valid = true;

                // Client-side validation for name
                if (nama === "") {
                    $("#nama-error").text("Nama harus diisi.");
                    valid = false;
                } else {
                    $("#nama-error").text("");
                }

                // Client-side validation for email
                if (email === "") {
                    $("#email-error").text("Email harus diisi.");
                    valid = false;
                } else {
                    $("#email-error").text("");
                }

                // Client-side validation for password
                if (password.length < 8) {
                    $("#password-error").text("Password harus minimal 8 karakter.");
                    valid = false;
                } else {
                    $("#password-error").text("");
                }

                if (valid) {
                    // Send data via AJAX
                    $.ajax({
                        url: 'proses_validasi.php',
                        type: 'POST',
                        data: {
                            nama: nama,
                            email: email,
```



```
                            password: password
                        },
                        success: function(response) {
                            // Display result from server
                            $("#result").html(response);
                        },
                        error: function() {
                            // Handle error
                            $("#result").html("An error occurred. Please try again.");
                        }
                    });
                }
            });
        });
    </script>

</body>
</html>
```



Form Input dengan Validasi, Password, dan AJAX

Nama: Febrian Arka Samudra
Email: febrianarka10@gmail.com
Password: ••••••••
Submit

Data berhasil dikirim: Nama: Febrian Arka Samudra, Email: febrianarka10@gmail.com

Explanation :

1. **Password Validation (Client-Side)**:

- The jQuery script checks if the password has at least 8 characters before form submission. If not, an error message ("Password harus minimal 8 karakter.") appears below the password field.
- If the password is valid, the message clears, and the form data is submitted via Ajax.

2. **Password Validation (Server-Side)**:

- The PHP script ensures that the password meets the minimum length requirement. If the password is less than 8 characters, the server returns an error message ("Password harus minimal 8 karakter.").

3. **Improved User Experience**:

- The client-side validation provides immediate feedback, preventing invalid passwords from being submitted without a full page reload.
- The server-side validation ensures security and correctness even if JavaScript is disabled on the client side, as the data is validated again on the server.

4. **Error Handling**:

- Both the client-side and server-side validation work together to ensure the data is correct. If the password or any other field is invalid, the form is not submitted, and appropriate error messages are displayed.

Github : https://github.com/FebrianArkaSamudra/PemrogramanWeb/tree/main/Week7