

Jobsheet-7: PHP – Form Processing

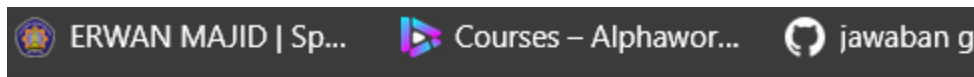
Web Design and Programming Courses

Erwan Majid / 08/2i

Link Github: <https://github.com/Majid5654/Semester-3/tree/master/Week7>

- **Practical Section 1. Function isset()**

1. What do you understand from using the isset on the file? (Question No. 1)



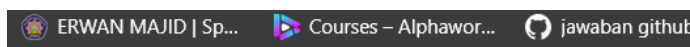
Anda belum dewasa atau variabel 'umur' tidak ditemukan

- `isset($umur)` checks if the variable `$umur` exists and is not null. If this condition is true, the code proceeds to evaluate whether `$umur` is greater than or equal to 18.

- If `$umur` is set and the condition `$umur >= 18` is true, the message "Anda sudah dewasa" (You are an adult) is displayed.

- If `$umur` is either not set (i.e., hasn't been initialized) or the condition fails, the else block runs, displaying the message "Anda belum dewasa atau variabel 'umur' tidak ditemukan" (You are not an adult or the 'umur' variable was not found).

2. 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)



No 1

Nama : jane

No 2

Anda belum dewasa atau variabel 'umur' tidak ditemukan

- `isset($data["nama"])` checks if the "nama" key exists in the associative array `$data` and if its value is set (i.e., not null). If the "nama" key exists and has a value, the script outputs the nama value inside a `<p>` tag:

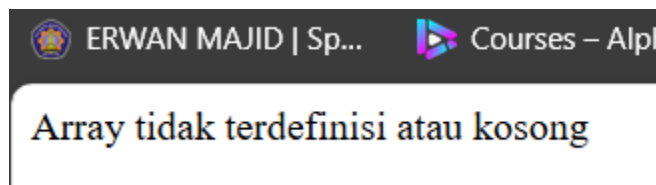
- If "nama" is not set or does not exist, the else block runs, and the message "Variabel 'nama' tidak ditemukan dalam array."

- The `isset()` is used to safely verify whether the "nama" key exists in the `$data` array before trying to access it. This avoids errors that could occur if you try to access a key that doesn't exist in the array.

- **Practical Section 2. Function empty()**

3. 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()` function is used to check whether the `$myArray` variable is empty. `empty($myArray)` checks if `$myArray` is either: Not set, or Set but contains an empty value

- in this case, since `$myArray` is defined as an empty array (`array()`), `empty()` returns true, so the message "Array tidak terdefinisi atau kosong"

-the different between `if isset` and `empty` is,if `isset` It doesn't care if the value is empty or not ,it must not null. Then the `empty` checks if a variable is either undefined or contains an empty value (e.g., null, false, empty array, "", or 0).

4. 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)



No 1

Array tidak terdefinisi atau kosong

No 2

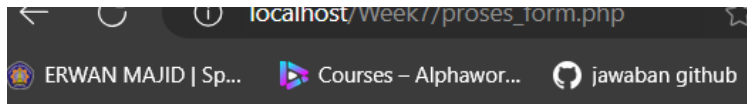
Variable tidak terdefinisi atau kosong

- is used to check whether the variable `$nonExistentVar` is either not set or is set but has an empty value.

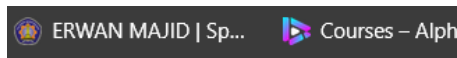
- **Practical Section 3. PHP Input Form**

5. 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)

Proses_form.php:



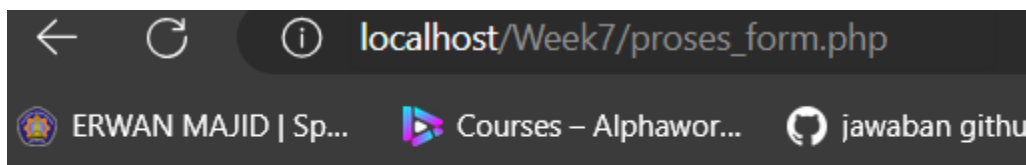
Form.php:



Form Input PHP

Nama:

Email:



Nama : Majid

Email : 3rwanmajid@gmail.com

--<input type="text" name="nama": This creates a text input field where the user can enter their name. The name attribute ("nama") is important because it defines the key used by the `$_POST` array to send the data to PHP.

-<input type="email" name="email": This creates an email input field where the user can enter their email address.

- Submit Button: When the user clicks the Submit button, the form is sent to proses_form.php for processing.

- The user enters their name and email in the form

- Form Submission: When the user clicks "Submit", the form data is sent to proses_form.php via the POST method

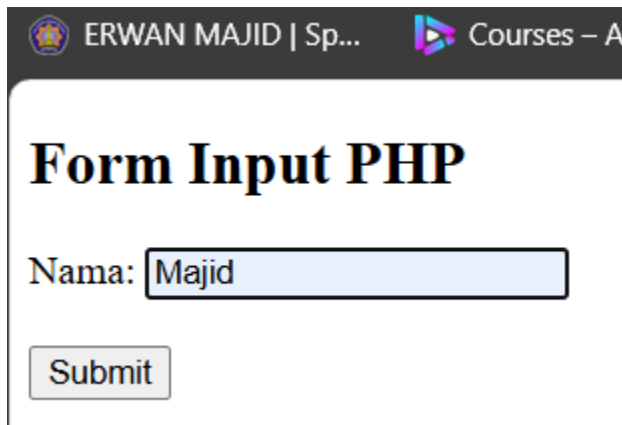
- The PHP script (proses_form.php) checks if the form was submitted using POST.

It retrieves the values from the form fields (using \$_POST["nama"] and \$_POST["email"]).

The submitted name and email are then displayed on the webpage

- The action="proses_form.php" tells the browser to send the form data to the proses_form.php script for processing when the user clicks the "Submit" button.

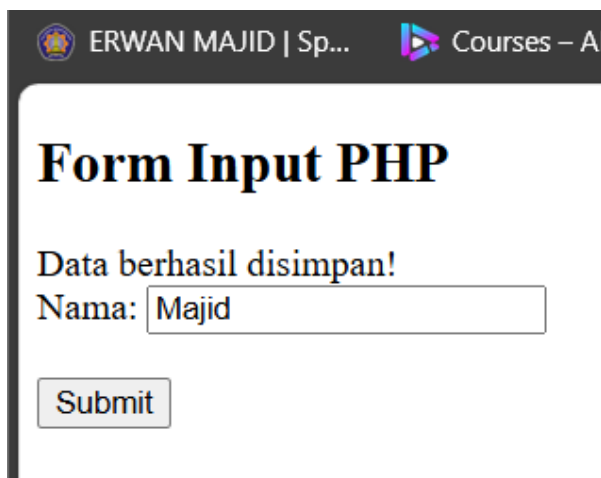
6. Save the file, then open a browser and run localhost/week7/ form_self.php. What do you understand from the use of forms in the file? Write your understanding below. (Question No. 6)



ERWAN MAJID | Sp... Courses - A

Form Input PHP

Nama:



ERWAN MAJID | Sp... Courses - A

Form Input PHP

Data berhasil disimpan!

Nama:

-The form is used to:

Collect user input (in this case, the user's name).

Validate the input on the server (checking if the name is provided).

Provide feedback to the user (error messages or success confirmation).

Dynamically handle user input and retain it in case of errors, making the process user-friendly.

- **Practical Section 4. HTML Injection**

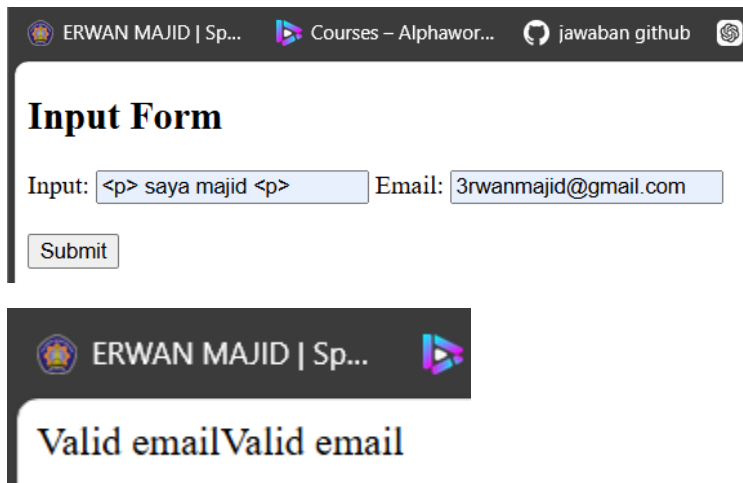
7. Record here what you observed, give your explanation. (Question No. 7)

The screenshot displays a web browser window with a dark theme. At the top, a navigation bar shows the user's name 'ERWAN MAJID | Sp...' and a logo for 'Courses – Alphawor...'. Below this, the page title 'Input Form' is prominently displayed. The main content area features a form with a label 'Input:' followed by a text input field. The input field contains the text '<p> saya majid <p>', which is a demonstration of HTML injection. To the right of the input field is a 'Submit' button. The browser's address bar at the bottom shows the URL 'localhost/Week7/html_safe.php'. Below the address bar, several tabs are visible, including 'ERWAN MAJID | Sp...', 'Courses – Alphawor...', 'jawaban github', and 'ChatGPT'.

-This line is commented out and therefore not executed. However, if included, it would sanitize the \$input by converting special HTML characters into their HTML entities. This is important to prevent Cross-Site Scripting (XSS) attacks

-Validating Email: This line retrieves the value of another form field named email, storing it in the variable \$email

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



The screenshot shows a web browser window with a dark header bar containing the text 'ERWAN MAJID | Sp...' and 'Courses – Alphawor...'. Below the header, the page title 'Input Form' is displayed. The form contains two input fields: 'Input:' with the value '<p> saya majid <p>' and 'Email:' with the value '3rwanmajid@gmail.com'. A 'Submit' button is located below the input fields. Below the form, a message 'Valid emailValid email' is displayed.

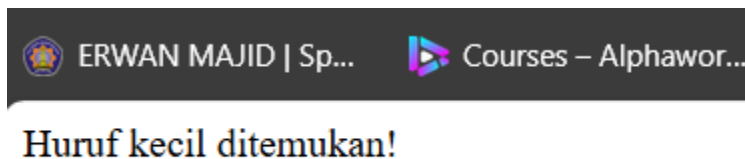
-`$email = $_POST['email'];` This line retrieves the value of the email input from an HTTP POST request. It assumes that the form has an input field with the name "email"

-Email Validation: This line uses PHP's `filter_var` function to validate the email address. The `FILTER_VALIDATE_EMAIL` filter checks if the provided email is a valid format.

-The code is a common method for validating email addresses in PHP.

- **Practical Section 5 : The Use of Regex in PHP**

9. Note here what you observe from the addition of the program code above. (Question No. 9)



The screenshot shows a web browser window with a dark header bar containing the text 'ERWAN MAJID | Sp...' and 'Courses – Alphawor...'. Below the header, the message 'Huruf kecil ditemukan!' is displayed.

-`$pattern = '/[a-z]/';` : `/[a-z]/` looks for any lowercase letter from a to z. The brackets `[]` define a character class, and `a-z` is a range for lowercase letters, The slashes `/` are delimiters that indicate the start and end of the pattern.

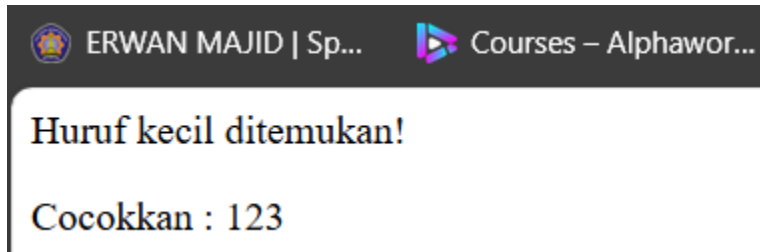
-`$text = 'This is a Sample text.';` : The string 'This is a Sample text.' contains a mixture of uppercase and lowercase letters.

-`if (preg_match($pattern, $text))` : `preg_match($pattern, $text)` is a PHP function that checks if the regular expression pattern matches any part of the text.

-`echo "Huruf kecil ditemukan!";` If `preg_match` finds a lowercase letter, this message is printed: "Huruf kecil ditemukan!" (meaning: "Lowercase letters found!").

-if no lowercase letters are found, the script will print: "Tidak ada huruf kecil!" (meaning: "No lowercase letters!").

10. Note here what you observe from the addition of the program code above. (Question No. 10)



-`$pattern = '[0-9]+/';` : The pattern `[0-9]+/` is intended to search for one or more digits (0-9).

-`$text = 'There are 123 apples.';` :

-`preg_match` Function:

This function is used to search for a match of a regular expression within a string.

The first argument is the pattern, the second is the string to search, and the third (optional) is an array to store matches.

Regular Expressions:

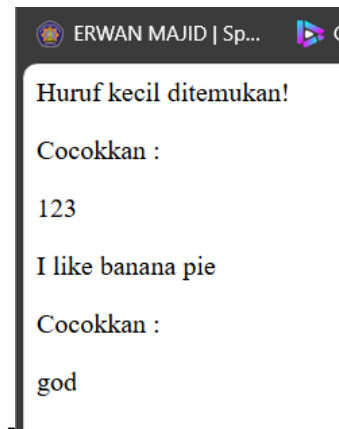
`/[a-z]/` searches for lowercase letters.

`/[0-9]+/` searches for one or more digits.

Outputs:

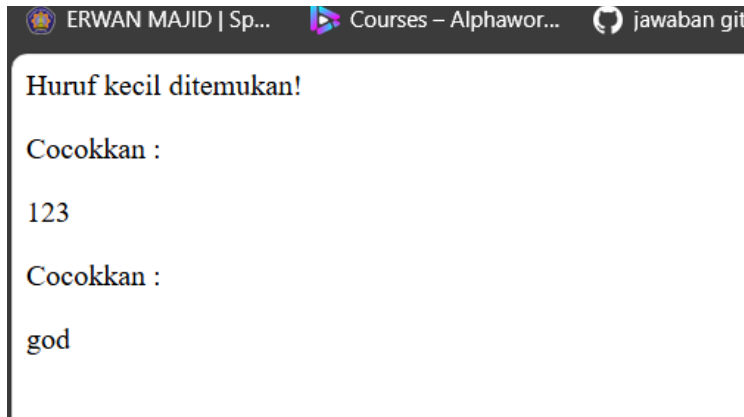
Based on whether matches are found or not, the appropriate messages are printed.

11. Note here what you observe from the addition of the program code above. (Question No. 11)



-The script searches for the word "apple" in the string 'I like apple pie'.
It then replaces the first (and only) occurrence of "apple" with "banana".
The modified string becomes 'I like banana pie'.
The script outputs the result as an HTML paragraph: <p>I like banana pie</p>

12. Note here what you observe from the addition of the program code above. (Question No. 12)



--\$pattern = '/go*d/'; : o*d is the actual pattern being searched:

g matches the character "g".

o* matches zero or more occurrences of the character "o". This means it can match:

No "o" at all (i.e., "gd").

One "o" (i.e., "god").

Multiple "o"s (i.e., "good").

d matches the character "d".

-Text to Search: This is the string where the regular expression will search for matches. The text is 'god is good'.

-Using preg_match: The preg_match function is used to search for the pattern in the text

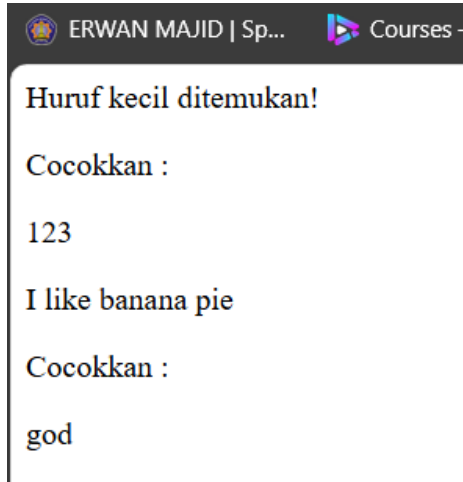
- Conditional Logic: If a match is found: The matched text will be stored in the \$matches array, and \$matches[0] will contain the first match.

If no match is found:

It will output "Tidak ada yang cocok" (meaning "No match found").

-The pattern /go*d/ is designed to match any word that starts with "g", is optionally followed by zero or more "o"s, and ends with "d"

13. 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)



-the pattern now matches either "gd" or "god", but it no longer matches "good" (because "good" contains two "o"s, and ? only allows for zero or one "o").

ch for a match in the string 'god is good'.

Matches

god will match because:

The "g" is matched by g.

The "o" is optionally matched by o? (it allows for exactly one "o").

The "d" is matched by d.

Does not match:

good will not match because it has two "o"s. The pattern o? only allows zero or one "o", not more than one.

14. 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)

Cocokkan :

o

-`[o]{1,3}`: The square brackets `[o]` represent a character set, which means that the pattern is looking for the character "o".

-`{1,3}` is a quantifier that specifies the number of repetitions the character "o" can have.

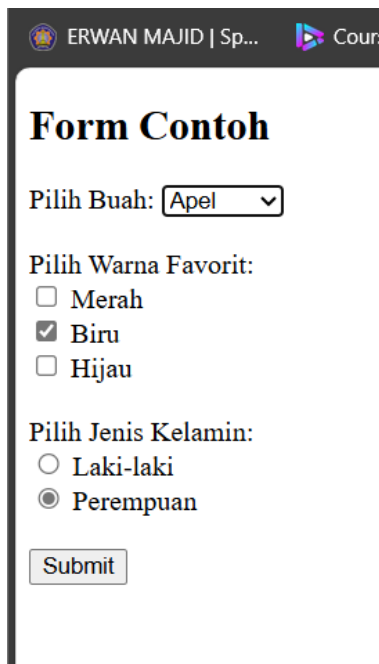
-`{1,3}` means that the pattern will match at least 1 "o" and up to 3 "o"s.

-So this pattern will match occurrences of 1, 2, or 3 "o"s in the text.

-The pattern does not capture both "o"s in the word good because `preg_match` only returns the first match. If you want to match all occurrences, you would need to use `preg_match_all`.

- **Practical Section 6 : Advanced Form**

15. Note here what you observe from the program code above. (Question No. 15)



ERWAN MAJID | Sp... Course

Form Contoh

Pilih Buah:

Pilih Warna Favorit:

☐ Merah

☒ Biru

☐ Hijau

Pilih Jenis Kelamin:

☐ Laki-laki

☒ Perempuan

-Form Structure: The form uses the POST method, which means the data will be sent securely to the server without appearing in the URL.

The action attribute is set to `"proses_lanjut.php"`, which means when the form is submitted, the data will be sent to this PHP file for processing.

-Select Input for Buah (Fruit): A dropdown list with different fruit options (apel, pisang, mangga, jeruk) allows the user to select one fruit.

-Checkbox Inputs for Warna Favorit (Favorite Colors): There are three checkbox options for selecting favorite colors (merah, biru, hijau), and the user can select multiple colors.

-Submit Button: The submit button sends the form data to the server for processing when clicked.

-The PHP script checks if the form was submitted using the POST method.

-\$selectedBuah: Stores the fruit selected by the user.

\$selectedWarna: Stores the array of selected colors. If no colors were selected, it initializes as an empty array.

\$selectedJenisKelamin: Stores the gender selected by the user.

-When the user submits the form, the data is sent to proses_lanjut.php.

16. 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)

ERWAN MAJID | Sp... Courses

Form Contoh

Pilih Buah:

Pilih Warna Favorit:

☐ Merah

☒ Biru

☐ Hijau

Pilih Jenis Kelamin:

☒ Laki-laki

☐ Perempuan

Anda memilih buah: pisang

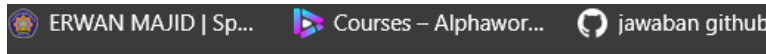
Warna favorit Anda: biru

Jenis kelamin Anda: laki-laki

-With AJAX: When you submit the form using AJAX, the data is sent to the server in the background without reloading the web page. This makes the user experience smoother, as only the relevant part of the page updates with new information.

- **Practical Section 7 : Form Validation**

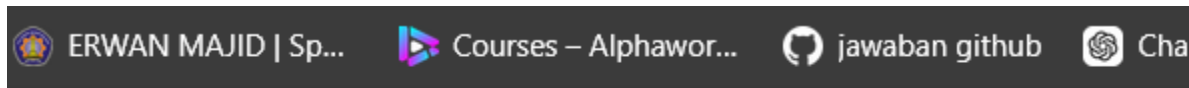
17. Note here what you observe from the addition of the program code above (Question No. 17)



Form Input dengan Validasi

Nama:

Email:



Data berhasil dikirim: Nama = Majid, Email = 3rwanmajid@gmail.com

-here are two input fields:

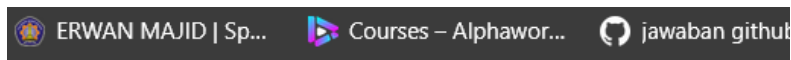
Nama: A text input for the user's name. The label element associates with the input field through the for attribute.

Email: A text input for the user's email address. Both fields use
 for line breaks to make the form visually separated

-<input type="submit" value="Submit">:This creates a submit button with the label "Submit." When clicked, the form data will be sent to proses_validasi.php

-When the form is submitted, the data is sent to proses_validasi.php for validation (this is the PHP script you provided earlier).

18. Note here what you observe from the addition of the program code above.(Question No. 18)



Form Input dengan Validasi

Nama: Nama harus diisi.

Email:

-

-The different between previous version and last version is Error Handling with jQuery:

```

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("");
}

```

his implementation adds error messages directly next to the respective input fields using the span elements with IDs nama-error and email-error

-Submission Handling:

```

$(document).ready(function() {
    $("#myForm").submit(function(event) {
        var nama = $("#nama").val();
        var email = $("#email").val();
        var valid = true;

```

-The form submission is captured by jQuery, allowing for the validation checks to occur when the user attempts to submit the form. If any fields are invalid, `event.preventDefault();` stops the form from submitting.

19. 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 > html > body > form#myForm > label
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>
13 <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 <input type="submit" value="Submit">
20 </form>
21
22 <div id="response" style="margin-top: 20px; color: green;"></div>
23
24 <script>
25     $(document).ready(function() {
26         $("#myForm").submit(function(event) {
27             event.preventDefault();
28
29             var nama = $("#nama").val();
30             var email = $("#email").val();
31             var valid = true;
32
33
34             $("#nama-error").text("");
35             $("#email-error").text("");
36
37
38             if (nama === "") {
39                 $("#nama-error").text("Nama harus diisi.");
40                 valid = false;
41             }
42
43             if (email === "") {
44                 $("#email-error").text("Email harus diisi.");
45                 valid = false;
46             }
47
48
49             if (valid) {
50                 $.ajax({
51                     url: "proses_validasi.php",
52                     type: "POST",
53                     data: {
54                         nama: nama,
55                         email: email
56                     },
57                     success: function(response) {
58                         $("#response").text(response); // Show success message
59                     },
60                     error: function(xhr, status, error) {
61                         $("#response").text("Terjadi kesalahan: " + error); // Show error message
62                     }
63                 });
64             }
65         });
66     });
67 </script>
68 </body>
69 </html>
70
```

Form Input dengan Validasi

Nama:	Majid
Email:	3rwanmajid@gmail.com
<input type="submit" value="Submit"/>	

Data berhasil dikirim: Nama = Majid, Email = 3rwanmajid@gmail.com

- revert Default Submission: The event.preventDefault(); method is called to stop the form from submitting in the traditional way.
- This AJAX-based implementation allows for form submission without refreshing the page

20. 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)

```
form_validation.php > html > body > form#myForm > label
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>
13     <br>
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     <label for="password">Password:</label>
18     <input type="password" id="password" name="password">
19     <span id="password-error" style="color: red;"></span><br>
20     <input type="submit" value="Submit">
21   </form>
22
23   <div id="response" style="margin-top: 20px; color: green;"></div>
24
25   <script>
26     $(document).ready(function() {
27       $("#myForm").submit(function(event) {
28         event.preventDefault();
29
30         var nama = $("#nama").val();
31         var email = $("#email").val();
32         var password = $("#password").val();
33         var valid = true;
34
35         $("#nama-error").text("");
36         $("#email-error").text("");
37         $("#password-error").text("");
38
39         if (nama === "") {
40           $("#nama-error").text("Nama harus diisi.");
41           valid = false;
42         }
43
44         if (email === "") {
45           $("#email-error").text("Email harus diisi.");
46           valid = false;
47         }
48
49         if (password.length < 8) {
50           $("#password-error").text("Password harus minimal 8 karakter.");
51           valid = false;
52         }
53
54         if (valid) {
55           $.ajax({
56             url: "proses_validasi.php",
57             type: "POST",
58             data: {
59               nama: nama,
60               email: email,
61               password: password
62             },
63             success: function(response) {
64               $("#response").text(response);
65             },
66             error: function(xhr, status, error) {
67               $("#response").text("Terjadi kesalahan: " + error);
68             }
69           });
70         }
71       });
72     });
73   </script>
74 </body>
75 </html>
```

```

proses_validasi.php > ...
1  <?php
2  if ($_SERVER["REQUEST_METHOD"] == "POST") {
3      $nama = $_POST['nama'];
4      $email = $_POST['email'];
5      $password = $_POST['password'];
6      $errors = [];
7
8      if (empty($nama)) {
9          $errors[] = "Nama harus diisi.";
10     }
11     if (empty($email)) {
12         $errors[] = "Email harus diisi.";
13     } elseif (!filter_var(value: $email, filter: FILTER_VALIDATE_EMAIL)) {
14         $errors[] = "Format email tidak valid.";
15     }
16     if (strlen(string: $password) < 8) {
17         $errors[] = "Password harus minimal 8 karakter.";
18     }
19
20     if (count(value: $errors) > 0) {
21         echo implode(separator: "<br>", array: $errors);
22     } else {
23         echo "Data berhasil dikirim: Nama = $nama, Email = $email, Password = (hidden)";
24     }
25 }
26 ?>

```

ERWAN MAJID | Sp...
Courses – Alphawor...
jawaban github

Form Input dengan Validasi

Nama:
Email:
Password:

ERWAN MAJID | Sp...
Courses – Alphawor...
jawaban github
ChatGPT
colorgrading

Data berhasil dikirim: Nama = Majid, Email = 3rwanmajid@gmail.com, Password = (hidden)

- HTML Form: Added a password field with an associated error message span, The type attribute of the password input is set to password for secure input.

- Client-Side Validation: The jQuery script now checks if the password length is less than 8 characters. If it is, an error message is display

- Server-Side Validation: The PHP script now checks the length of the password. If it is less than 8 characters, an error message is added to the \$errors array. On successful validation, the password is not displayed in the response just (hidden)