**Jobsheet-7: PHP – Form Processing**

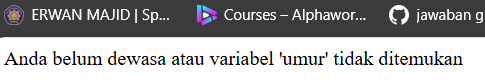
**Web Design and Programming Courses**

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)

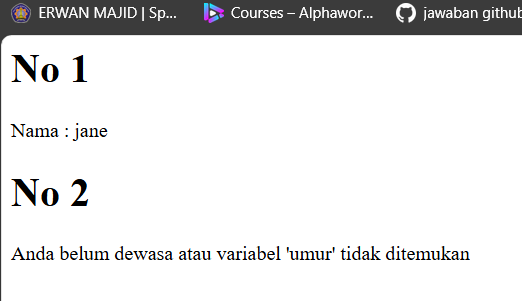


- isset($umur) checks if the variable $umur exists and is not null. If this condition is true, the code proceeds to evaluate whether $umur is greater than or equal to 18.

- 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).

1. 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)



- 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:

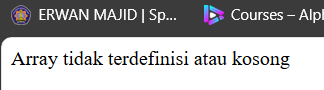
- If "nama" is not set or does not exist, the else block runs, and the message "Variabel 'nama' tidak ditemukan dalam array."

- 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()**

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

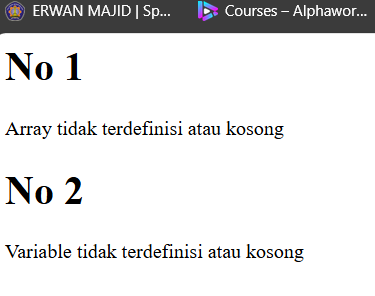
What do you understand from the use of empty on the file? Write your understanding below (Question No. 3)



- 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 ifsett 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).

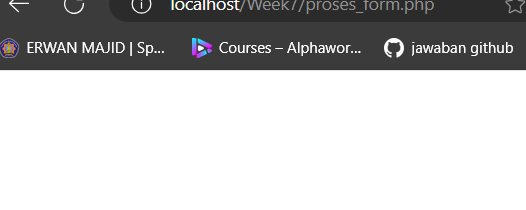
1. Save the file, then open the browser and run localhost/week7/empty.php. Ensure that the output does not appear in a single line; the result from the echo should be displayed separately. Explain what you understand from the use of empty() in that file. Write your understanding below. (Question No. 4)

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

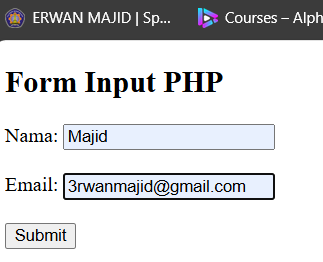
* **Practical Section 3. PHP Input Form**

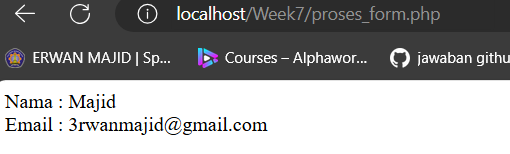
1. Save the file, then open a browser and run localhost/week7/proses\_form.php. Explain what happened and write your understanding below. Then run localhost/week7/form.php. Explain what happened and write your understanding below. (Question No. 5)

Proses\_form.php:



Form.php:





--<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

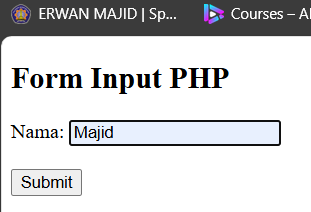
- The PHP script (proses\_form.php) checks if the form was submitted using POST.

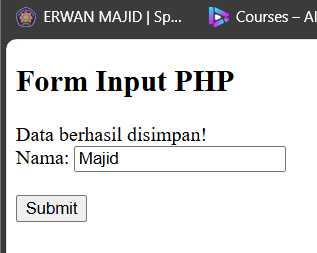
It retrieves the values from the form fields (using $\_POST["nama"] and $\_POST["email"]).

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.

1. 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)





-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).

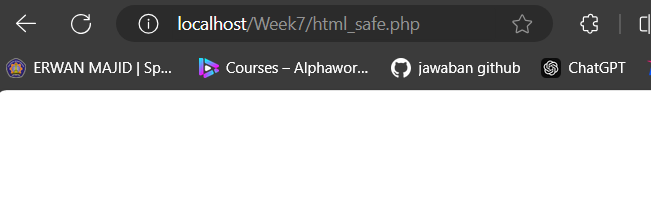
Provide feedback to the user (error messages or success confirmation).

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

* **Practical Section 4. HTML Injection**

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

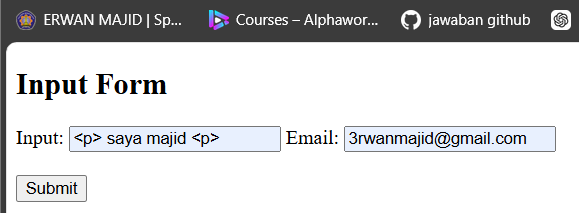




-This line is commented out and therefore not executed. However, if included, it would sanitize the $input by converting special HTML characters into their HTML entities. This is important to prevent Cross-Site Scripting (XSS) attacks

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

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





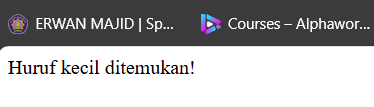
-$email = $\_POST['email']; This line retrieves the value of the email input from an HTTP POST request. It assumes that the form has an input field with the name "email

-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**

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



-$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.

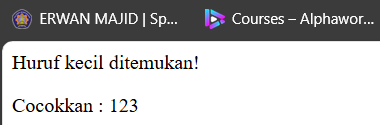
-$text = 'This is a Sample text.'; : The string 'This is a Sample text.' contains a mixture of uppercase and lowercase letters.

-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!").

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

1. 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:

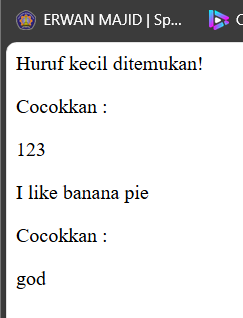
/[a-z]/ searches for lowercase letters.

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

Outputs:

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

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

-

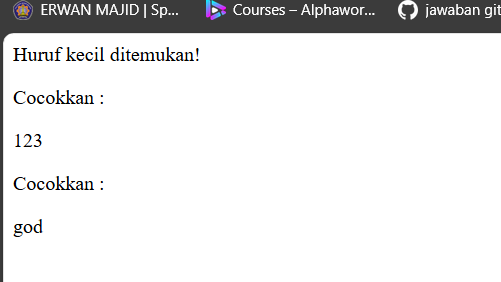
-The script searches for the word "apple" in the string 'I like apple pie'.

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>

1. 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

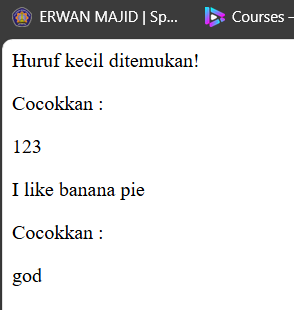
- Conditional Logic: If a match is found: The matched text will be stored in the $matches array, and $matches[0] will contain the first match.

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"

1. 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.

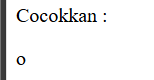
The "o" is optionally matched by o? (it allows for exactly one "o").

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.

1. 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)



-[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.

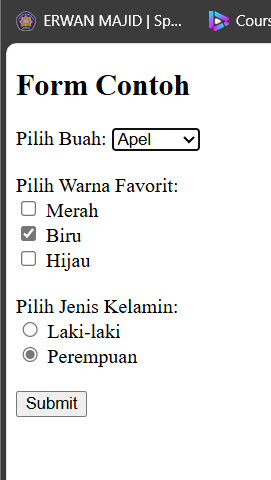
-{1,3} means that the pattern will match at least 1 "o" and up to 3 "o"s.

-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**

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



-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.

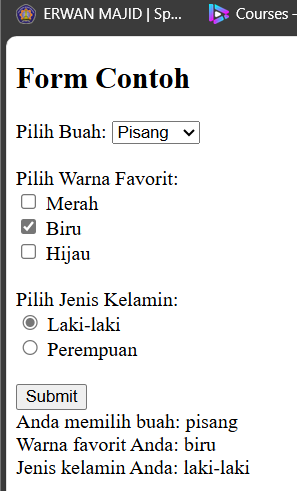
-$selectedBuah: Stores the fruit selected by the user.

$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.

1. 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)

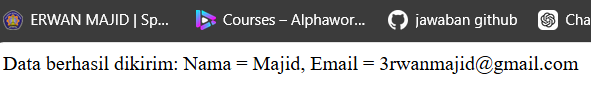


-With AJAX: When you submit the form using AJAX, the data is sent to the server in the background without reloading the web page. This makes the user experience smoother, as only the relevant part of the page updates with new information.

* **Practical Section 7 : Form Validation**

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





-here are two input fields:

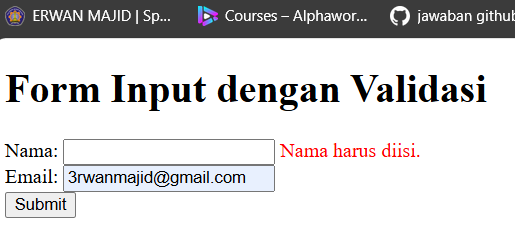
Nama: A text input for the user's name. The label element associates with the input field through the for attribute.

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

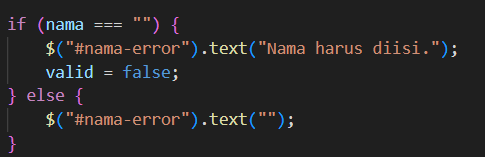
-<input type="submit" value="Submit">:This creates a submit button with the label "Submit." When clicked, the form data will be sent to proses\_validasi.php

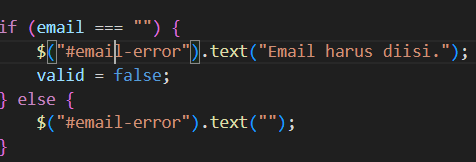
-When the form is submitted, the data is sent to proses\_validasi.php for validation (this is the PHP script you provided earlier).

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

-

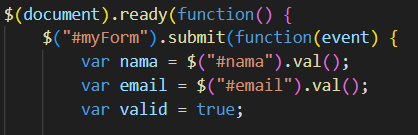
-The different between previous version and last version is Error Handling with jQuery:





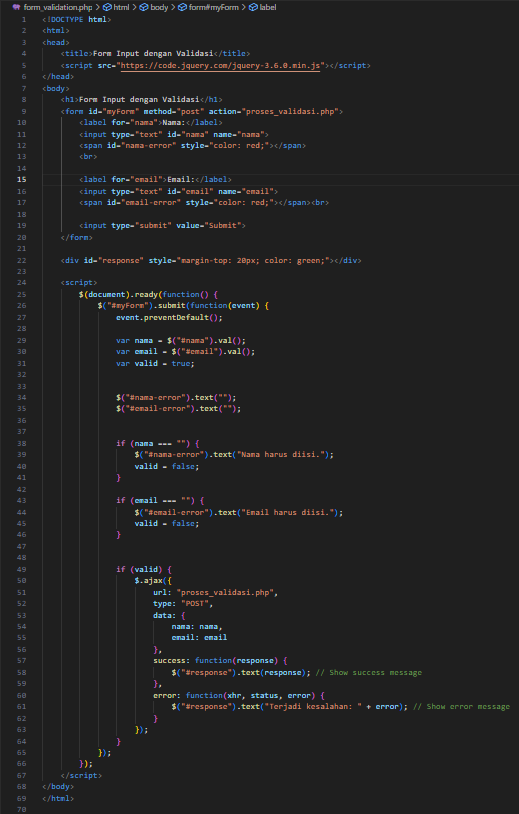
his implementation adds error messages directly next to the respective input fields using the span elements with IDs nama-error and email-error

-Submission Handling:



-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.

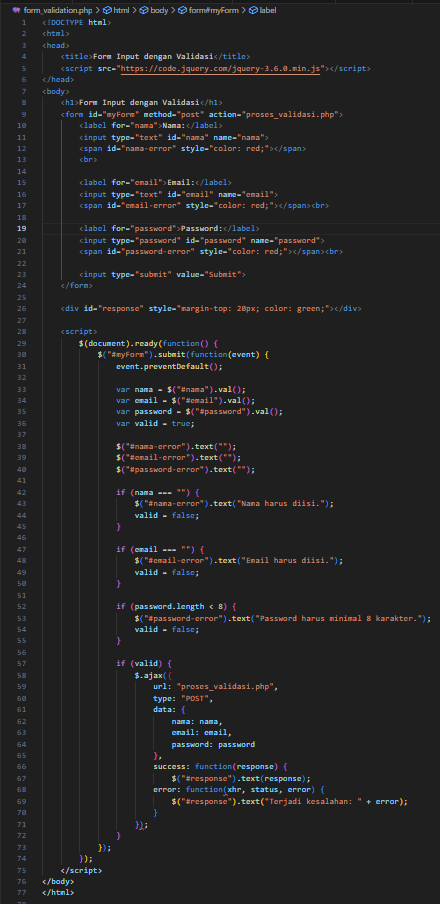
1. Create a script for step 6 using ajax. Screen shoot the code and wrote here what you observe from the addition of the program code. (Question No. 19)



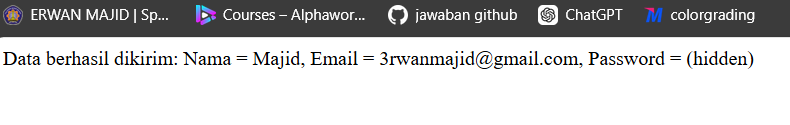
- 

- revent 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

1. 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)

- 



- 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)