|  |  |
| --- | --- |
|  | Department of Description Technology, Malang State Polytechnic  **Jobsheet-7: PHP - Form Proccessing**  **Web Design and Programming Courses**  Tutor: Web Design and Programming Teaching Team  *October 2023* |

**Name : Moh. Rafael Abrari**

**Class : SIB\_2G**

**Nim/Number : 244107060039/13**

**Topic**

* The Concept of Form Proccessing with PHP and Jquery

**Purpose**

Students are expected to:

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

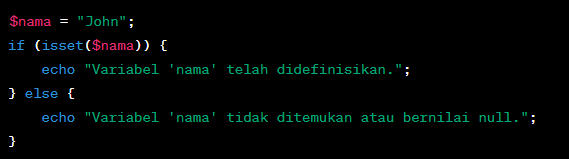
**Attention**

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

**Introduction**

**Function isset**

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



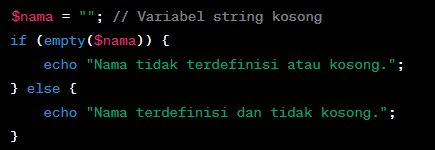
The above code checks if $nama variable has been defined. If yes, then the message "Variable 'name' has been defined." will be displayed, if not, then the message "Variable 'name' not found or null value." will be displayed.

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

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new folder JS07\_PHP-jQuery in dasarWeb and name it isset.php |
| 2 | Type into the isset.php file the code below. |
| 3 |  |
| 4 | Save the file, then open a browser and run localhost/dasarWeb/JS07\_PHP-jQuery/isset.php. What do you understand from using the isset on the file. Give me your answer.  (Question No 1.1)  Answer:  isset() is used to check whether a variable has been created and has a value/is not null. In the code, $umur has not been assigned a value, so isset($umur) is false, and the output is = Anda belum dewasa variabel 'umur' tidak ditemukan. |
| 5 | Add the contents of the isset.php file with the code below. |
| 6 | Save the file, then open the browser and run localhost/dasarWeb/JS07\_PHP-jQuery/isset.php. Ensure that the output does not appear in a single line; the result from the echo should be displayed separately. Explain what you understand from the use of isset() in that file. Write your understanding below. (Question No 1.2)  Answer:  $data["nama"], the key "nama" exists and has a value "Jane", so isset($data["nama"]) returns **true**, and the output is = Nama: Jane |

**Function** empty()

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



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

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

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new file named empty.php inside JS07\_PHP-jQuery folder. Write this code into empty.php file |
| 2 |  |
| 3 | Save the file, then open a browser and run localhost/dasarWeb/JS07\_PHP-jQuery/empty.php  What do you understand from the use of empty on the file. Write your understanding below. (Question No 2.1)  Answer:  empty() checks whether a variable exists and has a non-empty value. If the variable is empty or not defined, it returns true. |
| 4 | Add the contents of the empty.php file with the code below |
| 5 | Save the file, then open the browser and run localhost/dasarWeb/JS07\_PHP-jQuery/empty.php. Ensure that the output does not appear in a single line; the result from the echo should be displayed separately. Explain what you understand from the use of empty() in that file. Write your understanding below. (Question No 2.2)  Answer:  $nonExistentVar variable not defined, so empty($nonExistentVar) also returns true, and the output is = Variabel tidak terdefinisi atau kosong. |

**Practical Section Part 3: PHP Input Form**

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new file named form.php inside JS07\_PHP-jQuery folder. Write this code below |
| 2 |  |
| 3 | Create a new file named form\_process.php inside JS07\_PHP-jQuery folder Type the code in step 2 inside the form\_process.php |
| 4 | Save the file, then open a browser and run localhost/dasarWeb/JS07\_PHP-jQuery/form\_process.php. What do you understand from the use of forms in the file. Write your understanding below. (Question No. 3.1)  Answer:  The form is used to collect user input and send it to a PHP script for processing. The $\_POST array is used in PHP to retrieve the submitted form data, allowing the program to display or process the input further. |
| 5 | Create a new file named form\_self.php inside JS07\_PHP-jQuery folder Type the code below inside form\_self.php |
| 6 | Save the file, then open a browser and run localhost/dasarWeb/JS07\_PHP-jQuery/ form\_self.php. What do you understand from the use of forms in the file. Write your understanding below. (Question No 3.2)  Answer:  The form shows how to submit, validate, and display results on the same page using $\_SERVER["PHP\_SELF"]. It also demonstrates basic input validation (checking if a field is empty) and safe form handling using htmlspecialchars() to prevent security risks |

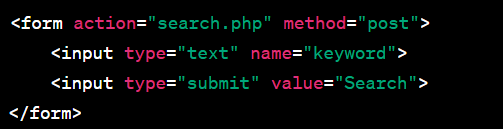
**HTML Injection**

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

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

**Example HTML Injection:**

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



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



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



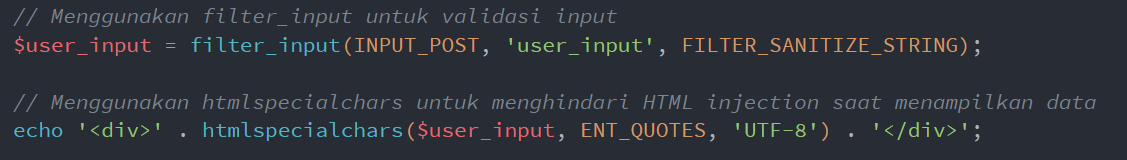
The malicious JavaScript code will be executed by the user's browser and will bring up a "You have been hacked!" warning box. This is a simple example of HTML injection. As a web developer, it is important to always sanitize and validate input from users, as well as avoid displaying user input directly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

**How to Handle HTML Injection**

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

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

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



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

**Practical Section 4 : HTML Injection**

Practical Section Steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new file named html\_safe.php inside JS07\_PHP-jQuery folder. Type the code below inside html\_safe.php |
| 2 |  |
| 3 | Add another script so that step 2 can run normally, save the file. Then open a browser and run localhost/dasarWeb/JS07\_PHP-jQuery/html\_safe.php |
| 4 | Note here what you observe from the addition of the program code above. (Question No. 4.1)  Answer:  is used to make user input safe before displaying or processing it |
| 5 | Type the additional code in step 6 inside the html\_safe.php |
| 6 |  |
| 7 | Complete the program code above so that the result is neat. Save the file, then open a browser and run /refresh localhost/dasarWeb/JS07\_PHP-jQuery/html\_safe.php |
| 8 | Note here what you observe from the addition of the program code above. (Question No. 4.2)  Answer:  The code ensures that only properly formatted email addresses are accepted. It also uses htmlspecialchars() to display the email safely, preventing malicious scripts, improving input validation and security of the web form. |

**Regular Expression (Regex)**

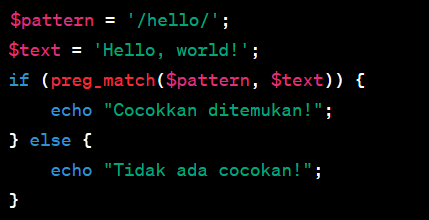
Regex is a powerful tool for searching, matching, or manipulating text based on specific patterns. You can use it for various purposes such as input validation, text search, text replacement, and more.

**Basic Regex Patterns:**

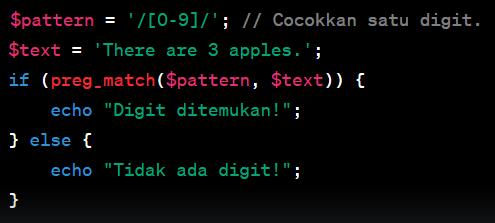
* **Single Character**: Any character will match itself in Regex, except for special characters that must be avoided with escape characters.
  + Example: The **/a/** pattern will match the letter "a" in the text.
* **Specific Characters**: You can match specific characters by mentioning them in a pattern.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

**Examples of Regex Usage:**

1. Matching patterns



1. Match set characters



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

Practical Section Steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new file named regex.php inside JS07\_PHP-jQuery folder Type the code in step 2 inside the regex.php |
| 2 |  |
| 3 | Save the file, then open a browser and run /refresh localhost/dasarWeb/JS07\_PHP-jQuery/ regex.php |
| 4 | Note here what you observe from the addition of the program code above. (Question No 5.1)  Answer:  The program demonstrates how preg\_match() is used with a regular expression to detect specific patterns in text. In this case, it successfully identifies that the string contains lowercase letters. |
| 5 | Type the additional code in step 6 inside the regex.php |
| 6 |  |
| 7 | Complete the program code above so that the result is neat. Save the file, then open a browser and run /refresh localhost/dasarWeb/JS07\_PHP-jQuery/regex.php |
| 8 | Note here what you observe from the addition of the program code above. (Question No. 5.2)  Answer:  The code uses regex with preg\_match() to detect and extract numbers from text. It shows how PHP can identify specific patterns (in this case, digits) and display the matched result neatly. |
| 9 | Type the additional code in step 10 inside the regex.php |
| 10 |  |
| 11 | Complete the program code above so that the result is neat. Save the file, then open a browser and run /refresh localhost/dasarWeb/JS07\_PHP-jQuery/regex.php |
| 12 | Note here what you observe from the addition of the program code above. (Question No 5.3)  Answer:  The code demonstrates how preg\_replace() is used to replace specific text patterns in a string using regex. It changes “apple” to “banana” successfully. |
| 13 | Type the additional code in step 14 inside the regex.php |
| 14 |  |
| 15 | Complete the program code above so that the result is neat. Save the file, then open a browser and run /refresh localhost/dasarWeb/JS07\_PHP-jQuery/regex.php |
| 16 | Note here what you observe from the addition of the program code above. (Question No 5.4)  Answer:  he code shows how the regex \* operator works, it matches zero or more occurrences of a character. Here, it successfully matches words like “god” and “good” that fit the pattern go\*d |
| 17 | Question 5.5: Create a script for step 14 with a variable pattern modified using '?' (0 or 1 time). Note here what you observe from the addition of the program code above. (Question No 5.5)  Answer:  The script demonstrates that ? makes a character optional, so the regex can match words with or without that characte |
| 18 | Question 5.6: Create a script for step 14 with a variable pattern modified using '{n,m}'. Note here what you observe from the addition of the program code above. (Question No 5.6)  Answer:  The pattern /go{1,2}d/ matches the words “god” or “good,” because the letter ‘o’ appears 1–2 times. If it appears more than twice, such as in “goood,” it does not match. |

**Practical Section 6 : Advanced Form**

Practical Section Steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new file named form\_next.php inside JS07\_PHP-jQuery folder site. Type the code in step 2 inside the form\_next.php |
| 2 |  |
| 3 | Save the file, then open a browser and run /refresh localhost/dasarWeb/JS07\_PHP-jQuery/ form\_next.php |
| 4 | Note here what you observe from the addition of the program code above. (Question No 6.1)  Answer:  $\_POST[‘fruit’] displays the selected fruit.  $\_POST[‘color’] (array) displays several colors that are checked using implode().  $\_POST[‘gender’] displays the selected gender.  When the Submit button is clicked, the selected data is sent to the server using the POST method, then displayed on the page. |
| 5 | Create a new file named form\_ajax.php inside JS07\_PHP-jQuery folder Type the code in step 2 inside the form\_ajax.php |
| 6 | Save the file, then open a browser and run /refresh localhost/dasarWeb/JS07\_PHP-jQuery/form\_ajax.php. Note here what you observe from the addition of the program code above. (Question No. 6.2)  Answer:  The code makes jQuery capture the submit event from the form with the id myForm, then e.preventDefault() is used to prevent the form from being submitted normally so that the page is not reloaded. The form input data is converted into a sendable format using $(“#myForm”).serialize(), then sent to proses\_lanjut.php asynchronously with $.ajax(), and the results from PHP are displayed directly in the <div id="hasil"> element. |

**Practical Section 7 : Form Validation**

Practical Section Steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new file named form\_validation.php inside JS07\_PHP-jQuery folder Type the code below inside form\_validation.php |
| 2 |  |
| 3 | Save the file, then open a browser and run /refresh localhost/dasarWeb/JS07\_PHP-jQuery/form\_validation.php |
| 4 | Note here what you observe from the addition of the program code above. (Question No 7.1)  Answer:  The code displays an input form with server-side validation using PHP. When the user clicks the submit button, PHP checks whether the name and email fields have been filled in. If any field is empty, or the email format is invalid, an error message will be displayed. If all inputs are valid, the data will be processed and displayed as the result. |
| 5 | Develop a file named form\_validation.php inside JS07\_PHP-jQuery folder site. Type the code in step 6 inside the form\_validation.php |
| 6 |  |
| 7 | Save the file, then open a browser and run /refresh localhost/dasarWeb/JS07\_PHP-jQuery/form\_validation.php |
| 8 | Note here what you observe from the addition of the program code above. (Question No. 7.2)  Answer:  The code adds form validation using jQuery on the client side. When the submit button is pressed, the script checks whether the Nama and Email fields have been filled in. If either field is empty, an error message appears below the input without sending the form to the server. That way, users can correct their input before the data is sent. |
| 9 | Question 7.3: Create a script for step 6 using ajax. Note here what you observe from the addition of the program code above. (Question No. 7.3)  Answer:  This code adds a form submission process using ajax, so that data is sent to a php file without reloading the page. jQuery validates the name and email input first, then if valid, the data is sent to the server asynchronously, and the results of the PHP process are immediately displayed in the <div id="results"> element. |
| 10 | Question 7.4: Add validation for the password with a minimum of 8 characters using jQuery and PHP added in step 9. Note here what you observe from the addition of the program code above. (Question No 7.4)  Answer:  The code adds a minimum password length of 8 characters on the jQuery side to prevent the form from being submitted if it does not meet the requirements, and on the PHP side to ensure data security. If all inputs are valid, the data is sent via Ajax without reloading the page, and the results are displayed in <div id="results">. |