|  |  |
| --- | --- |
|  | 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 : GANANG ANDIKA KURNIA PUTRA**

**NIM : 244107060008**

**CLASS : SIB-2G**

**MATKUL : PEMPROGRAMAN WEB**

**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

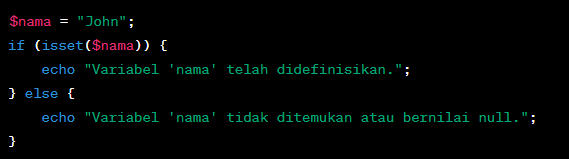
**Attention**

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()**:**



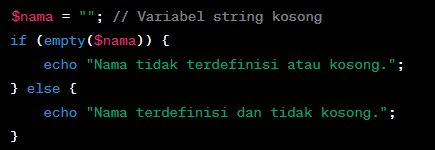
The above code checks if $nama variable has been defined. If yes, then the message "Variable 'name' has been defined." will be displayed, if not, then the message "Variable 'name' not found or null value." will be displayed.

**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)    The isset() function is used to check if a variable has been created and contains a value other than NULL. In this file, since $umur is not assigned a value, isset($umur) returns false, so the program outputs that the variable ‘umur’ is not found or that the user is not an adult. It helps prevent errors when using variables that might not exist. |
| 5 | Add the contents of the isset.php file with the code below. |
| 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)    The isset() function is used to check whether a variable or array element has been declared and is not NULL. In this code, isset($age) is false because the $age variable does not yet have a value, while isset($data[“name”]) is true because the “name” element exists in the $data array. With isset(), we can avoid errors when checking or using variables that have not been created. |

**Function** empty()

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



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)  The empty() function is used to check whether a variable has no value or is empty. In this code, $myArray is an empty array, so empty($myArray) returns true and the program displays the message “Array is undefined or empty.” This function is useful for ensuring that a variable contains data before it is used in a program. |
| 4 | Add the contents of the empty.php file with the code below |
| 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)    The empty() function is used to check whether a variable has not been defined or has no value (empty). In this code, the variable $nonExistentVar has never been created, so empty($nonExistentVar) returns true and displays the message “Variable not defined or empty.” This function helps avoid errors when checking variables that have no value. |

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

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new file named 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)    This file uses a form to send data to the server using the POST method. After the Submit button is pressed, PHP will process the data sent via $\_POST and display it back on the same page. The action="" attribute causes the form to be sent to this file itself. The $\_SERVER[“REQUEST\_METHOD”] function ensures that data is only processed when the form is submitted. With a form like this, users can input data and immediately see the results on the same page. |
| 5 | Create a new file named 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)    In this file, the form is used to send data to the same page (self-processing form) using $\_SERVER[“PHP\_SELF”]. This means that when the Submit button is pressed, the form data is not sent to another file, but is processed directly in the same file. The program will perform a simple validation, which is to check whether the name field has been filled in. If it is empty, an error message “Name must be filled in!” will appear, and if it has been filled in, the message “Data successfully saved!” will appear.  In this way, the form can display the input, validation, and results on a single web page without moving to another page. |

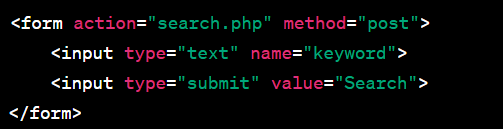
**HTML Injection**

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

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

**Example HTML Injection:**

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



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:



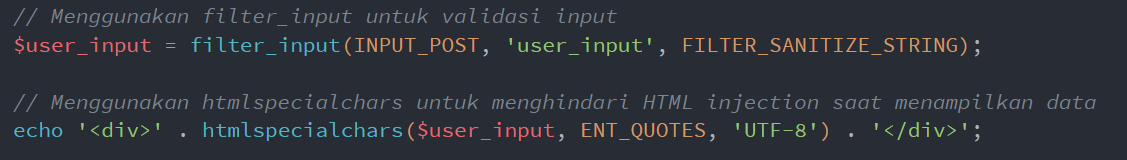
The malicious JavaScript code will be executed by the user's browser and will bring up a "You have been hacked!" warning box. This is a simple example of HTML injection. As a web developer, it is important to always sanitize and validate input from users, as well as avoid displaying user input directly 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)  I noticed that when users fill out the form and press the Submit button, the data entered in the “Name” field is sent to the server using the POST method. The PHP code then processes the input with the htmlspecialchars() function to protect it from dangerous characters such as HTML tags or scripts (preventing XSS). Additionally, there is a simple validation using empty() that ensures the name field cannot be left blank. If the field is empty, the message “Name cannot be empty!” appears, while if it is filled in, the result of the name input entered by the user will be displayed. |
| 5 | Type the additional code in step 6 inside the html\_safe.php |
| 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)  From the code addition above, I observe that the filter\_var() function is used to check whether the email input is in the correct format. If the format is invalid, the program will display an error message, whereas if it is valid, the email data is processed and displayed securely using htmlspecialchars(). In this way, the program becomes more secure and can filter out incorrect input from the outset. |

**Regular Expression (Regex)**

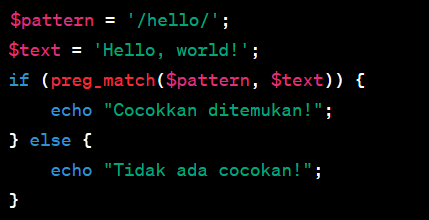
Regex is a powerful tool for searching, matching, or manipulating text based on specific patterns. You can use it for various purposes such as input validation, text search, text replacement, and more.

**Basic Regex Patterns:**

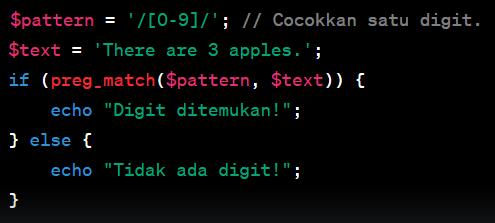
* **Single Character**: Any character will match itself in Regex, except for special characters that must be avoided with escape characters.
  + Example: The **/a/** pattern will match the letter "a" in the text.
* **Specific Characters**: You can match specific characters by mentioning them in a pattern.
  + 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)  From the addition of the code above, I understand that the preg\_match() function is used to match text patterns using Regular Expressions (regex). The pattern /[a-z]/ means searching for lowercase letters (a to z) in the $text variable. If lowercase letters are found, the program displays the message “Lowercase letters found!”, while if none are found, it displays “No lowercase letters!”. Thus, this code functions to check for the presence of lowercase letters in a text using regular expressions. |
| 5 | Type the additional code in step 6 inside the regex.php |
| 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)  Regex /[0-9]+/ searches for one or more digits in the text. If numbers are found, the program displays them (for example, “123”). This shows that regex can be used to recognize number patterns in a string. |
| 9 | Type the additional code in step 10 inside the 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)  Regex /[0-9]+/ searches for one or more digits in the text. If numbers are found, the program displays them (for example, “123”). This shows that regex can be used to recognize number patterns in a string. |
| 13 | Type the additional code in step 14 inside the regex.php |
| 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)  The regex /apple/ is used to search for the word “apple” within a string. If found, the result displays that word. This shows how regex can be used to search for specific words in text.  From the code above, I understand that the preg\_replace() function is used to replace text that matches a specific pattern with new text. In the example, the pattern /apple/ searches for the word “apple” and replaces it with ‘banana’, so the result becomes “I like banana pie.”. Additionally, the pattern /go\*d/ is used to match words that have the letter “o” appearing one or more times, such as “god,” “good,” or “goood.” This regex demonstrates how we can detect and manipulate specific text patterns within a string using regular expressions in PHP. |
| 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)      From the addition of the code above, I understand that the ? symbol in regex is used to indicate that the previous character or pattern is optional (may appear once or not at all). In the example /apples?/, this pattern matches both the words “apple” and “apples.” Thus, this regex allows for more flexible searches for variations of words that have optional endings. |
| 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)      From the addition of the code above, I understand that the {n,m} sign in regex is used to determine the number of occurrences of a particular character or pattern. In the example /a{2,4}/, this pattern matches the letter “a” that appears between 2 and 4 times consecutively, so the results display “aa”, ‘aaa’, and “aaaa”. This regex is useful for limiting or specifying the length of the pattern to be matched. |

**Practical Section 6 : Advanced Form**

Practical Section Steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new file named 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)  The program displays the selected results from various input forms (select, checkbox, radio) using the POST method, as well as checking and displaying the data entered by the user dynamically on the same page. |
| 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)    From the code above, I understand that using jQuery and AJAX allows the form to send data to the server without reloading the page. The data is sent asynchronously to a PHP file, and the response is then displayed directly on the web page via the <div id="result"> element. This method makes user interaction faster and more dynamic than a regular form. |

**Practical Section 7 : Form Validation**

Practical Section Steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new file named 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)  From the addition of the program code above, it can be observed that the form now has validation using \*\*jQuery\*\* before being sent to the server, so that empty inputs can be prevented and users are given immediate warnings without having to reload the page. |
| 5 | Develop a file named 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)  From the code above, I understand that this program adds form validation on the client side using jQuery before the data is sent to the server. Validation is performed immediately when the submit button is pressed. If the “Name” or “Email” fields are still empty, an error message is displayed without reloading the page. This method makes the validation process faster, more interactive, and provides immediate feedback to the user. |
| 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)        With the addition of AJAX code, forms can send data to the server without reloading the page. The validation process continues, and the results from the PHP file are displayed immediately on the same page. This makes user interaction faster, more dynamic, and more efficient than sending regular forms. |
| 10 | Question 7.4: Add validation for the password with a minimum of 8 characters using jQuery and PHP added in step 9. Note here what you observe from the addition of the program code above. (Question No 7.4) |