

Learning Objectives

After studying this material, participants are expected to be able to:

- Understanding the basic concept of forms in PHP.
- Managing input data from an HTML form using PHP.
- Perform input data validation for security and accuracy.
- Saving data from the form to the database.
- Using GET and POST methods securely.
- Implementing basic security measures such as input validation and sanitization

Requirements

Having the correct hardware and software components is very important to ensure the successful execution of the tasks outlined in this guide. The hardware and software configuration required to complete the tasks outlined in this guide is as follows:

Hardware Specifications

Minimum hardware specifications to run HTML on the Windows operating system and use software such as Chrome browser and Visual Studio Code are as follows:

Minimum Requirements

| | |
|------------|--|
| Prosesor | Intel Core 2 Duo or Equivalent |
| Memory | 4 GB |
| Hardisk | 120 GB HDD with a minimum available storage of 20 GB |
| Connection | Ethernet and Wi-Fi capabilities. |

Recommendation Requirements

| | |
|------------|--|
| Prosesor | Inter Core i3 or Equivalent |
| Memory | 4 GB |
| Hardisk | 120 GB SSD with a minimum available storage of 20 GB |
| Connection | Ethernet and Wi-Fi capabilities. |

Required Software

By using the correct software on your system to ensure the learning process runs smoothly and meets performance expectations. The required software is as follows:

| | |
|------------------|--|
| Sistem Operation | Windows 8.1 atau terbaru |
| Editor Teks | Visual Studio Code : Versi Stabil Terbaru Diinstal |
| Browser | Chrome, Firefox, Safari, Edge |
| Server | Xampp |
| Koneksi | Ethernet and Wi-Fi capabilities. |

Material Review

1. Understanding the GET Method

GET is an HTTP method used to retrieve or send data from the browser to the server via URL.

Characteristics:

- The data sent will be visible in the address bar (example: form.php?nama=Andi)
- Suitable for use with non-confidential data.
- Can be bookmarked or shared with others

Example of GET Usage:

Search form, product filter, page sorting, etc.

2. Understanding the POST Method

POST is an HTTP method used to send data from the browser to the server in a hidden manner (not displayed in the URL).

Characteristics:

- Data is sent through the body of the request, not the URL.
- Suitable for confidential or important data
- Cannot be bookmarked because the data is not visible.

Example of GET Usage:

Login form, registration, file upload, comments, etc.

Form & Testing Specification Table:

| Category | Specifications & Objectives |
|---------------|---|
| Functionality | <p>1. Receiving Input name</p> <p>✓ The form receives text input from the user through a field named name</p> <p>2. Using the POST Method</p> <p>✓ Data is sent using the POST method so that it does not appear in the URL</p> <p>✓ Suitable for personal or sensitive data</p> <p>3. Validation to Prevent Empty Submission</p> <p>✓ Using HTML required so that users cannot submit the</p> |

| | |
|----------------------|---|
| | <p>form without filling in the data</p> <p>✓ On the server side, check if <code>\$_POST['name']</code> is empty using <code>empty()</code> or <code>trim()</code></p> <p>4. Displaying Results or Error Messages</p> <p>✓ If the input is valid → display "Form successfully processed, Name: [name]" If the input is valid → display "Form successfully processed, Name: [name]"</p> <p>✓ If the input is empty → display "Error: Name is required"</p> |
| UI (Frontend) | <ul style="list-style-type: none"> - Form with <code>method="POST"</code> - Input field: <code>name</code> (with <code>required</code>) - Submit button: <code>Submit</code> - No dynamic error display (server-side validation only) |
| Unit Spec | <ul style="list-style-type: none"> • Action: <code>proses.php</code> • Metode: <code>POST</code> • Input: <code>name</code> (mandatory, text) • Validation: mandatory, avoid empty strings • Security: HTML sanitization (XSS), safe from SQL injection • Response: display name if successful, display error if empty |

Basic Structure Form

Example Form with GET Method:

```

<form action="proses.php" method="GET">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name " required>
  <button type="submit">end</button>
</form>

```

Example Form with GET Method:

```
<form action="proses.php" method="GET">
<label for="name">Name:</label>
<input type="text" id="name" name="name " required>
<button type="submit">end</button>
</form>
```

Explanation:

- action="proses.php" → Determining the PHP file that will process the form data.
- method="POST" → Determining the data transmission method (GET or POST).
- name="name" → Giving a name to the input so it can be accessed in PHP.
- required → Make the input mandatory to fill out before submission.

Metode GET vs POST

- GET → Sending data via URL, suitable for search or shareable links.
- POST → Sending data covertly, suitable for sensitive data.

Testing

- Copy the code from the tutorial material using the POST method and name the file "**proses.php**" then upload it in the "**Case Study Exercise Method POST**" menu and press "**the verify the Answer button**".
- Copy the code from the tutorial material using the GET method and name the file "**proses.php**" then upload it in the "**Case Study Exercise Method GET**" menu and press "**the verify the Answer button**".