**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.

## 2. Requirements

Before starting this lesson, participants are advised to have a basic understanding of:

- HTML dan CSS
- Basics of PHP
- Basic concepts of MySQL database

## 3. Hardware Specifications

### 3.1 Minimum Requirements

- **Prosesor:** Intel Core i3 or higher
- **RAM:** 4 GB
- **Penyimpanan:** 10 GB of free space
- **Sistem Operasi:** Windows 7 / macOS Sierra / Linux Ubuntu 16.04

### 3.2 Recommendation Requirements

- **Prosesor:** Intel Core i5 or higher
- **RAM:** 8 GB or more
- **Penyimpanan:** SSD with 20 GB of free space
- **Sistem Operasi:** Windows 10 / macOS Big Sur / Linux Ubuntu 20.04

## 4. Required Software

- **Web Server:** XAMPP, WAMP, or LAMP
- **Database Management System:** MySQL or MariaDB
- **Editor Kode:** VS Code, Sublime Text, or PHPStorm
- **Browser:** Google Chrome or Mozilla Firefox
- **PHP:** Version 7.4 or higher

**Material Review**

Previously, we have learned how to connect our application to the database. In this session, we will learn how to add data to the database.

1. **The first step is to create a form to store data through a form**

```
<html>
    <body>
      <form action="insert.php" method="post">
         NIP :<input type="text" name="NIP" />
         Initial name: <input type="text" name="initial_name" />
         Last name: <input type="text" name="last_name" />
         Age: <input type="text" name="age" />
         <input type="submit" />
    </form>
  </body>
</html>
```

2. **The second step is to save the data**

The INSERT INTO command is used to add a new record to a database table. The INSERT INTO statement can be written in 2 forms: INSERT INTO table_name VALUES (value1, value2, value3,...) or INSERT INTO table_name (column1, column2, column3,...). VALUES (value1, value2, value3,...). Still using the test_db database and the biodata table, we are trying to design a program to input data using the INSERT INTO instruction.

```
<?php
$servername = "localhost";
$username   = "root"; // Replace with your database username
$password   = ""; // Replace with your database password
$dbname     = "test_db"; // Replace with the name of the database used

// Create a connection to the database
$conn = new mysqli($servername, $username, $password, $dbname);

// Check the connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

// Data to be entered
$name = "John Doe";
$age = 25;
$address = "Jl. Contoh No. 123";

// Query to enter data
$sql = "INSERT INTO biodata (name, age, address) VALUES ('$name', $age, '$address')";
```

```
// Execute the query and check the result
if ($conn->query($sql) === TRUE) {
echo "New record created successfully";
} else {
echo "Error: " . $sql . "<br>" . $conn->error;
}

// Close the connection
$conn->close();
?>
```

## 3. Explanation

- Opening a connection to the database using new mysqli().
- Checking if the connection is successful.
- Determining the data to be input.
- Writing the INSERT INTO command to add data to the biodata table.
- Execute the command with $conn->query($sql).
- Display a success message if successful, or an error message if it fails.
- Closing the connection to the database.