

Jobsheet-10: PHP – CRUD & Bootstrap

Web Design and Programming Courses

Erwan Majid 08 2i

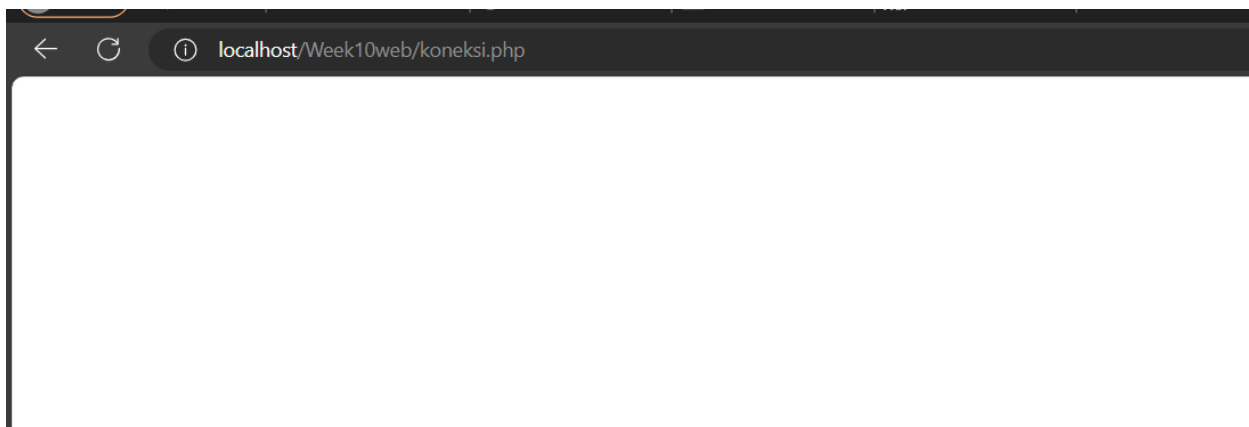
Link Github: <https://github.com/Majid5654/Week10Web>

- Creating a Member (Anggota) Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 nama	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 jenis_kelamin	enum('L', 'P')	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 alamat	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	5 no_telp	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More

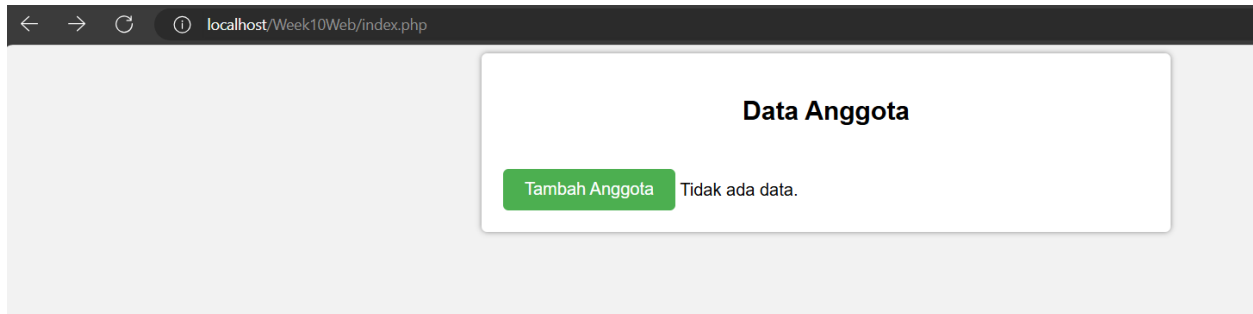
- Practical Section 1. CRUD – Read

```
koneksi.php > ...
1  <?php
2  $koneksi = mysqli_connect(hostname: "localhost", username: "root", password: "", database: "prakwebdb");
3
4  if (mysqli_connect_errno()) {
5      die ("Koneksi database gagal : " . mysqli_connect_error());
6  }
7  <?>
```



It blank, indicate the sql connection is successful connected

1. Save the file, then open a browser and run localhost/week10 . What do you understand from the use of the script. Record below your understanding. (Question No. 1)



-This code sets up a basic webpage for displaying and managing member data. It includes features to display a list of members, edit, and delete entries.

-This part of the code fetches data from the anggota table and displays it in a table format.

-button[type="submit"], .btn-kembali: Styles the submit and back buttons with a green background, white text, padding, and rounded corners.

button[type="submit"]:hover, .btn-kembali:hover: Darkens the background color on hover for the submit and back buttons.

.btn-kembali: Specifically styles the back button with a red background, distinguishing it from the submit button.

- **Practical Section 2. CRUD – Create**

2. Save the file, then open a browser and run localhost/week10. Add anggota data and then save. What do you understand from the script in the file. Record below your understanding. (Question No. 2)

A screenshot of a web browser window showing a form titled 'Tambah Data Anggota'. The form has four input fields: 'Nama' with the value 'majid', 'Jenis Kelamin' with radio buttons for 'Laki-laki' (selected) and 'Perempuan', 'Alamat' with the value 'jl malang', and 'No. Telp' with the value '081359377563'. At the bottom of the form are two buttons: a green 'Simpan Data' button and a red 'Kembali' button.

Data Anggota

[Tambah Anggota](#)

No	Nama	Jenis Kelamin	Alamat	No. Telp	Aksi
1	majid	Laki-Laki	jl malang	081359377563	Edit Hapus

-The form has an action attribute `action="proses.php?aksi=tambah"` which tells the form to send data to `proses.php` and pass an `aksi` parameter with the value `tambah`.

-The `method="post"` attribute specifies that the form data will be sent as a POST request.

-Nama: A required text field for entering the member's name.

-Jenis Kelamin (Gender): A set of radio buttons for selecting gender

-Submit and Back Button:

Simpan Data: A submit button that submits the form data to `proses.php`.

Kembali: A link styled as a button that redirects to `index.php`.

-Database Connection:

`include('koneksi.php');` Includes a separate PHP file (`koneksi.php`) which is responsible for establishing a connection to the database. This file likely contains database connection details such as host, username, password, and database name.

-Fetching Data:

`$aksi = $_GET['aksi'];` Gets the `aksi` parameter from the URL (set to `tambah` by the form).

`$nama = $_POST['nama'];` Retrieves the `nama` (name) entered in the form.

`$jenis_kelamin = $_POST['jenis_kelamin'];` Retrieves the selected gender from the form.

`$alamat = $_POST['alamat'];` Retrieves the address entered in the form.

`$no_telp = $_POST['no_telp'];` Retrieves the phone number entered in the form.

-Database Insertion:

if (\$aksi == 'tambah'): Checks if the action (aksi) is tambah, which indicates that the user wants to add a new record.

\$query = "INSERT INTO anggota (nama, jenis_kelamin, alamat, no_telp) VALUES ('\$nama', '\$jenis_kelamin', '\$alamat', '\$no_telp')";: Creates an SQL INSERT query to add a new row into the anggota table with the values from the form.

- **Practical Section 3. CRUD – Update**

3. Save the file, then open a browser and run localhost/week10. Make changes to the anggota data and then save it. What do you understand from the script in the file. Record below your understanding. (Question No. 3)

Edit Data Anggota

Nama:

Jenis Kelamin:
☒ Laki-laki ☐ Perempuan

Alamat:

No. Telp:

Data Anggota

No	Nama	Jenis Kelamin	Alamat	No. Telp	Aksi
1	erwan	Laki-Laki	jl malangs	0813593775632	Edit Hapus

-This code provides an interface for managing member data, including functionality to add and edit records in a MySQL database. It includes two HTML forms: one for adding a new member ('tambah' action) and another for editing an existing member ('ubah' action). When adding or editing a record, the form data (name, gender, address, and phone number) is submitted to 'proses.php', which performs the appropriate database operations based on the 'aksi' parameter. If the action is 'tambah', a new member

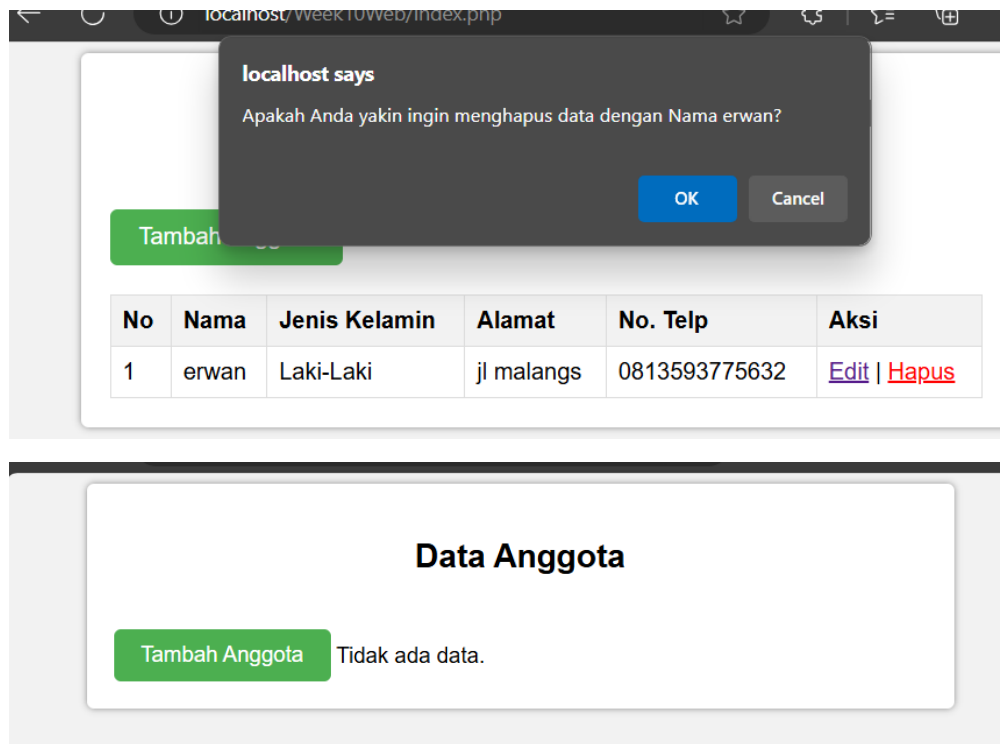
record is inserted; if it's `ubah`, an existing member record (identified by its `id`) is updated. The PHP code also uses `koneksi.php` to establish a connection to the database and `mysqli_query` for executing SQL commands, redirecting to the main `index.php` page upon success.

- **Practical Section 4. CRUD Section Delete**

4. Save the file, then open a browser and run localhost/week10. Delete member data.

What do you understand from the script in the file. Record below your understanding. (Question No. 4)

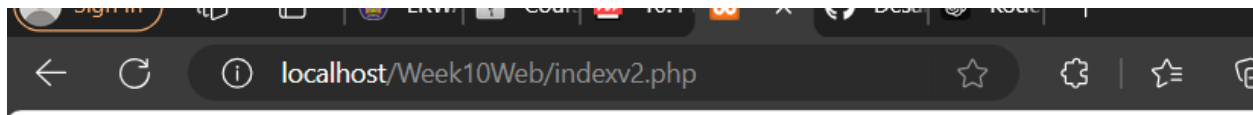
-



-This code snippet handles the deletion of a member record from the anggota table in the database. If the aksi parameter is set to 'hapus' and a valid id is provided in the URL, it retrieves the id, then executes a DELETE query to remove the record with that id. If the deletion is successful, it redirects to index.php; otherwise, it displays an error message. If the id is missing, it outputs "ID tidak valid" (invalid ID). The connection to the database is closed at the end. If aksi is anything other than 'hapus', it redirects to index.php.

- **Practical Section 5. CRUD View with Bootstrap**

5. Save the file, then open a browser and run localhost/week10. What do you understand from the script in the file. Record below your understanding. (Question No. 5)



Data Anggota

Tambah Data

No	Nama	Jenis Kelamin	Alamat	No. Telp	Aksi
----	------	---------------	--------	----------	------

-Styling and Frameworks:

Indexv2: Uses Bootstrap for styling (bootstrap.min.css), providing a more modern and responsive design. Buttons and tables are styled with Bootstrap classes like btn, btn-success, btn-primary, and table.

index: Uses a custom CSS file (style.css), which requires separate styles for buttons and tables and likely a simpler design.

Delete Confirmation Modal:

Indexv2 Implements delete functionality with a Bootstrap modal for confirmation. Each delete button triggers a unique modal for the respective member's ID, providing a user-friendly UI.

index: Uses a JavaScript confirm dialog (confirm()) to confirm deletion, which is a simpler approach but lacks the customized look of the modal.

JavaScript Libraries:

Indexv2: Includes external scripts for jQuery, Popper, and Bootstrap's JavaScript to support modal functionality.

index: Contains a single custom JavaScript function konfirmasiHapus to handle deletion confirmation without using external libraries, relying only on standard JavaScript.

HTML Structure and Elements:

Indexv2: Contains extra <div class="modal fade"> elements inside the loop to generate a unique delete confirmation modal for each member entry.

index : Has a simpler structure without modals. It outputs a single HTML table with <tr> elements inside a PHP loop.

6. Save the file, then open a browser and run localhost/week10. Add anggota data and then save. What do you understand from the script in the file. Record below your understanding. (Question No. 6)

localhost/Week10Web/createv2.php

Tambah Data Anggota

Nama:

Jenis Kelamin:

☒ Laki-laki
☐ Perempuan

Alamat:

No. Telp:

[Simpan Data](#)

[Kembali](#)

localhost/Week10Web/index.php

Data Anggota

[Tambah Anggota](#)

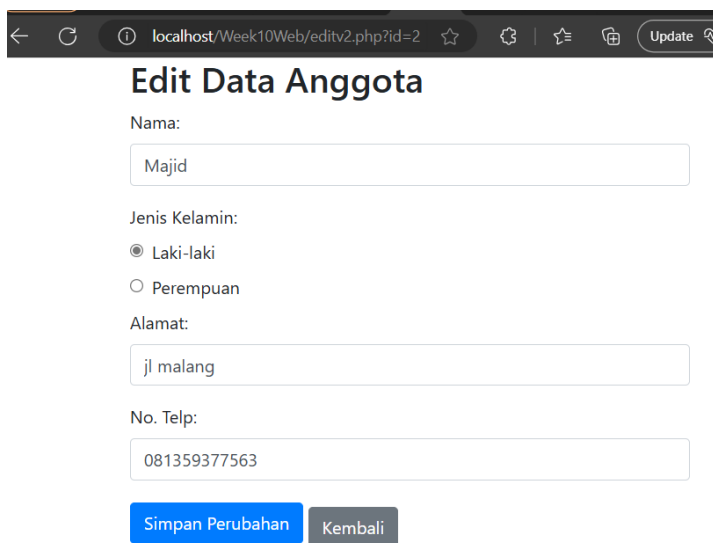
No	Nama	Jenis Kelamin	Alamat	No. Telp	Aksi
1	Majid	Laki-Laki	jl malang	081359377563	Edit Hapus

- Styling and Framework:

V1: Uses custom CSS styling via style.css, so each element's appearance is customized individually. This approach provides more control over the style but requires more effort to achieve a professional look.

V2 : Uses Bootstrap 4, providing a polished and responsive layout with minimal effort. Bootstrap classes (e.g., form-control, form-group, btn, btn-primary) give the form a consistent and modern appearance with built-in styling.

7. Save the file, then open a browser and run localhost/week10. Make changes at anggota data, then save. What do you understand from the script in the file. Record below your understanding. (Question No. 7)



localhost/Week10Web/editv2.php?id=2

Edit Data Anggota

Nama:

Jenis Kelamin:

☒ Laki-laki

☐ Perempuan

Alamat:

No. Telp:

-The different is more styling than v1

- **Practical Section 6. CRUD Display with Ajax**

8. What do you understand from the scripts on those files. Record below your understanding.
(Question No. 8)

The screenshot shows a web application interface for managing member data. At the top, there's a blue header with the text 'CRUD Dengan Ajax'. Below the header, the main content area is titled 'Data Anggota'. It contains a form for adding a new member with fields for 'Nama' (Name), 'Jenis Kelamin' (Gender), 'Alamat' (Address), and 'No Telepon' (Phone Number). The 'Nama' field is filled with 'Majid', and the 'Jenis Kelamin' field has 'Laki-laki' selected. Below the form is a blue 'Simpan' (Save) button. Underneath the form, there's a table showing the list of members. The table has columns for 'No', 'Nama', 'Jenis Kelamin', 'Alamat', 'No Telp', and 'Action'. There is one entry in the table with 'No' 1, 'Nama' Majid, 'Jenis Kelamin' Laki-laki, 'Alamat' jl malang, and 'No Telp' 081359377563. The 'Action' column has 'Edit' and 'Hapus' (Delete) buttons. At the bottom of the table, there's a pagination control showing 'Showing 1 to 1 of 1 entries' and 'Previous', '1', 'Next' buttons. The footer of the page says '© 2024 Copyright: Desain Dan Pemrograman Web'.

- This code is a CRUD (Create, Read, Update, Delete) web application for managing "Data Anggota" (Member Data), using PHP, AJAX, and MySQL, with CSRF protection. It starts with a session to generate a CSRF token, then verifies the token from incoming AJAX requests to prevent CSRF attacks. The form allows users to add, edit, and delete member information (name, gender, address, phone) in the database. The interface is enhanced with Bootstrap for styling and DataTables for a searchable, paginated table. AJAX is used to dynamically load and update the member data without page refreshes. The `koneksi.php` file connects to the MySQL database, and the CSRF token is set as a meta tag for secure AJAX requests.

9. What do you understand from the scripts on those files. Record below your understanding.

(Question No. 9)

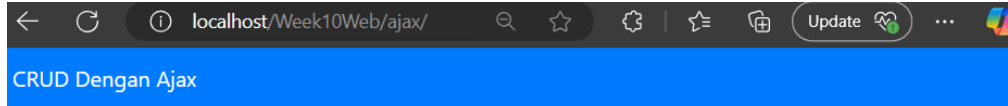
The screenshot shows a web application interface for managing member data. At the top, there's a blue header with the text 'CRUD Dengan Ajax'. Below the header, the main content area is titled 'Data Anggota'. It contains a form for adding a new member with fields for 'Nama' (Name), 'Jenis Kelamin' (Gender), 'Alamat' (Address), and 'No Telepon' (Phone Number). The 'Nama' field is empty and has a red error message 'Nama Harus Diisi' below it. The 'Jenis Kelamin' field has radio buttons for 'Laki-laki' and 'Perempuan', both of which are unselected, and a red error message 'Jenis Kelamin Harus Diisi' below it. The 'Alamat' field is empty and has a red error message 'Alamat Harus Diisi' below it. The 'No Telepon' field is empty and has a red error message 'No Telepon Harus Diisi' below it. At the bottom of the form is a blue 'Simpan' (Save) button.

- JavaScript checks if each input field (nama, alamat, jenis_kelamin, no_telp) is filled out.

If any field is empty, an error message is displayed below it (e.g., "Nama Harus Diisi" for the nama field).

If all fields are filled in, the error messages are cleared.

10. What do you understand from the scripts on those files. Record below your understanding.
(Question No. 10)



Data Anggota

Nama

Jenis Kelamin ☒ Laki-laki ☐ Perempuan

Alamat

No Telepon

Show entries Search:

No	Nama	Jenis Kelamin	Alamat	No Telp	Action
1	Majid	Laki-laki	jl malang	081359377563	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>

Showing 1 to 1 of 1 entries Previous **1** Next

© 2024 Copyright: [Desain Dan Pemrograman Web](#)

Show entries Search:

No	Nama	Jenis Kelamin	Alamat	No Telp	Action
1	Majid Erwan	Laki-laki	jl malang	081359377563	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>

Showing 1 to 1 of 1 entries Previous **1** Next

© 2024 Copyright: [Desain Dan Pemrograman Web](#)

- the user clicks the "Edit" button for a particular record. The button's id attribute holds the unique identifier (id) of that record.

- AJAX Request to get_data.php:

JavaScript sends an AJAX request to get_data.php, passing the id of the selected record.

get_data.php retrieves the record from the anggota table based on the id, then sends the record details as a JSON response to the front end.

- Populate Form with Data:

On success, the form fields are populated with the selected record's data (nama, jenis_kelamin, alamat, and no_telp).

The radio button for jenis_kelamin is set to match the record's value (either "L" for "Laki-laki" or "P" for "Perempuan").

- AJAX Request to form_action.php:

Clicking "Simpan" sends an AJAX request to form_action.php with the form data.

- Feedback to User:

form_action.php sends a JSON response back to the front end indicating a success message, which can be used to confirm to the user that the update was successful.

11. Save the file, then open a browser and run localhost/week10/ajax. When you click the delete button, the data should be deleted. What do you understand from the scripts on those files. Record below your understanding. (Question No. 11)

-

No	Nama	Jenis Kelamin	Alamat	No Telp	Action
Tidak ada data ditemukan					

- The user clicks the "Hapus" button for a specific record. This button's id attribute contains the unique identifier (id) of that record.

- JavaScript sends an AJAX request to hapus_data.php, passing the id of the record to be deleted.

- hapus_data.php retrieves this id and prepares a DELETE SQL statement to remove the record with that id from the anggota table.

- The record is deleted from the database. If the deletion is successful, hapus_data.php sends a JSON response indicating success.
- After receiving the success response, the code reloads the .data section by calling data.php, which retrieves and displays the updated list of records, now excluding the deleted entry.