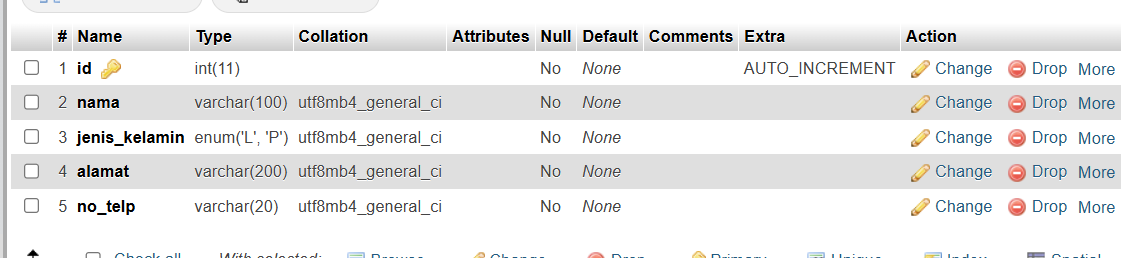
**Jobsheet-10: PHP – CRUD & Bootstrap**

**Web Design and Programming Courses**

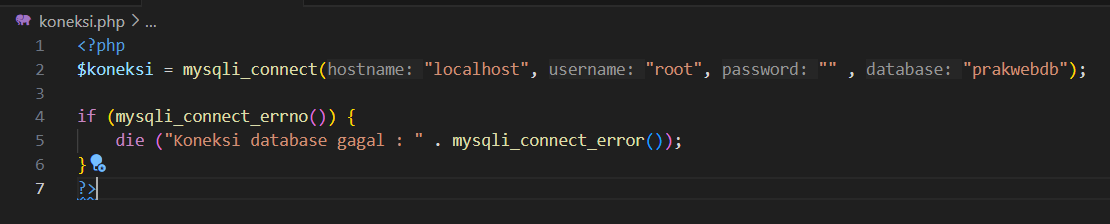
Erwan Majid 08 2i

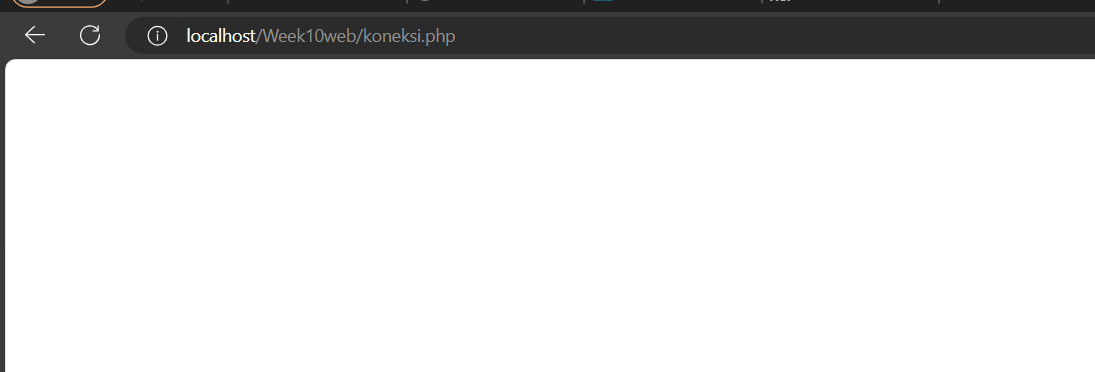
Link Github: <https://github.com/Majid5654/Week10Web>

* Creating a Member (Anggota) Table



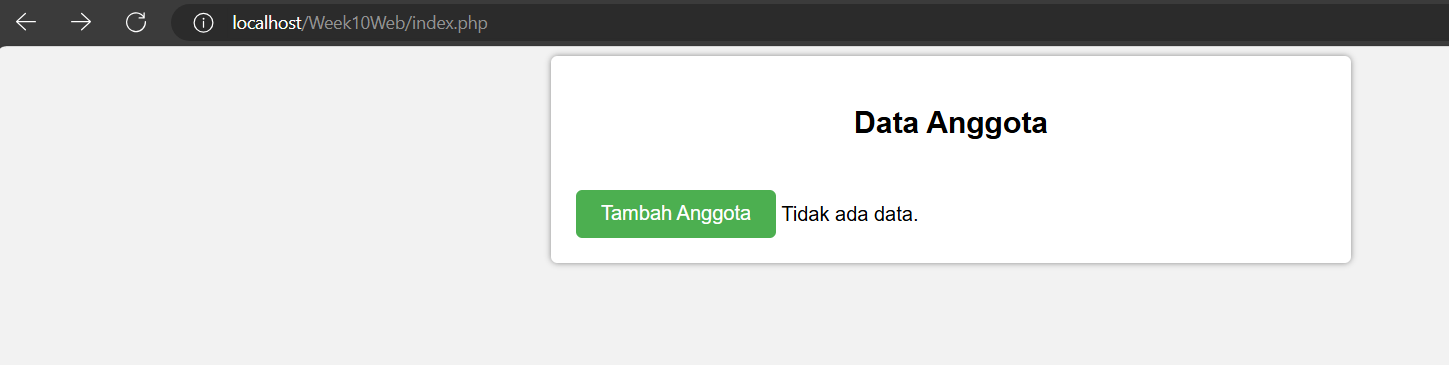
* **Practical Section 1. CRUD – Read**





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.

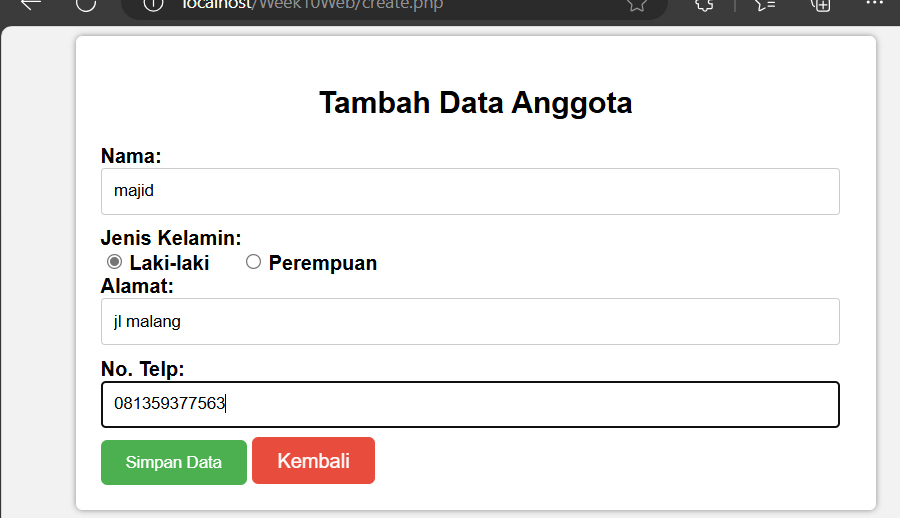
.btn-kembali: Specifically styles the back button with a red background, distinguishing it from the submit button.

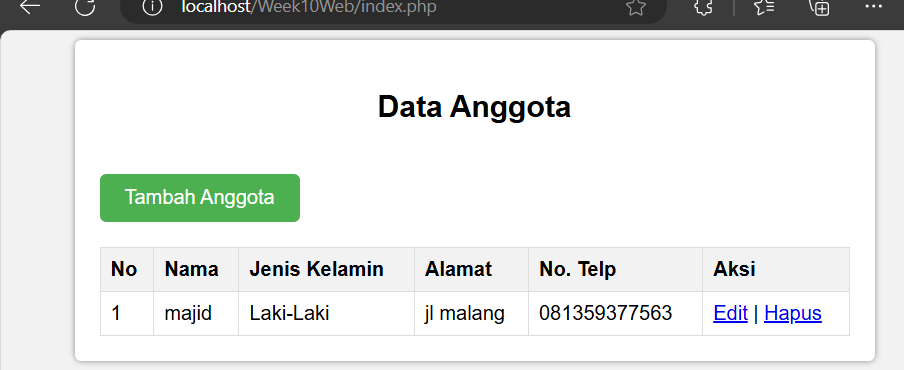
* **Practical Section 2. CRUD – Create**

1. 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)





-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 gende

-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.

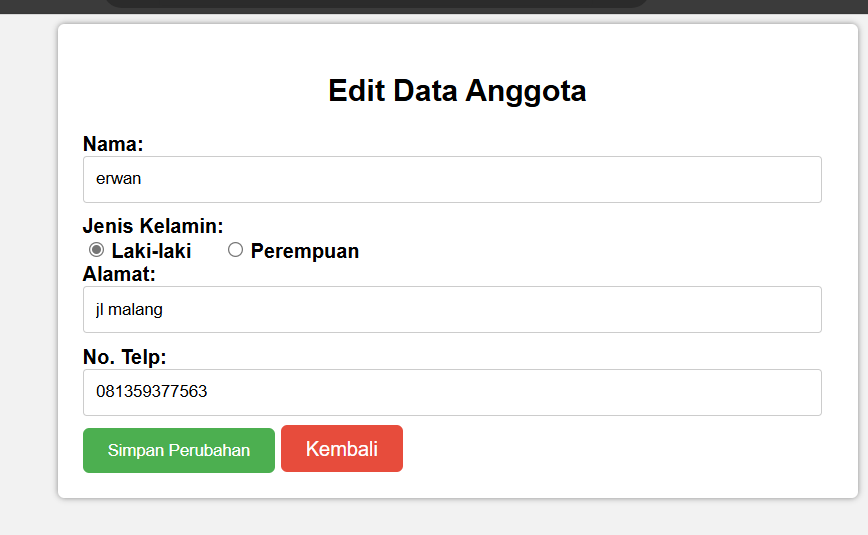
$query = "INSERT INTO anggota (nama, jenis\_kelamin, alamat, no\_telp) VALUES ('$nama', '$jenis\_kelamin', '$alamat', '$no\_telp')";: Creates an SQL INSERT query to add a new row into the anggota table with the values from the form.

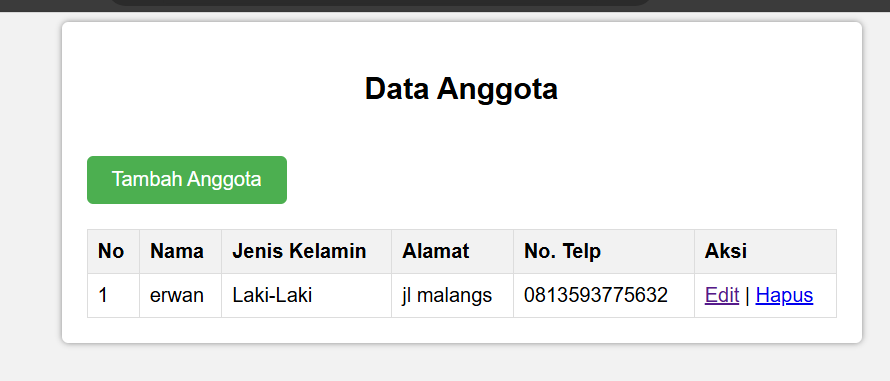
* **Practical Section 3. CRUD – Update**

1. 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)

****

****

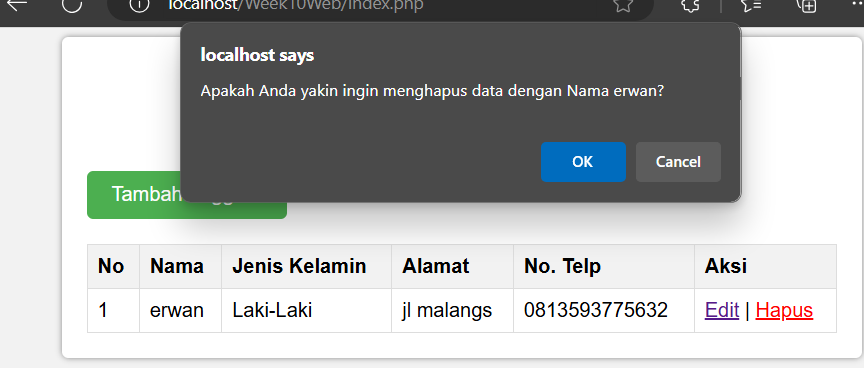
**-**This code provides an interface for managing member data, including functionality to add and edit records in a MySQL database. It includes two HTML forms: one for adding a new member (`tambah` action) and another for editing an existing member (`ubah` action). When adding or editing a record, the 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.

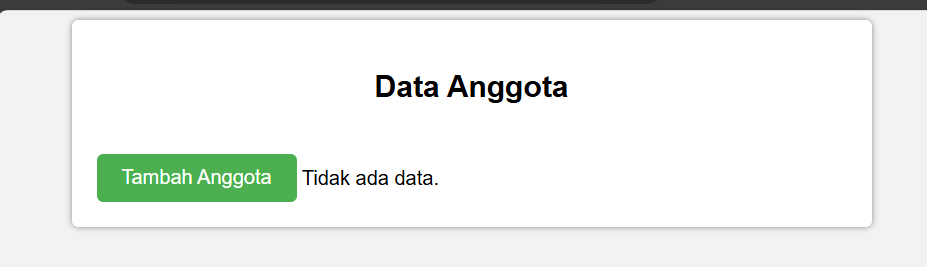
* **Practical Section 4. CRUD Section Delete**

1. 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)

-

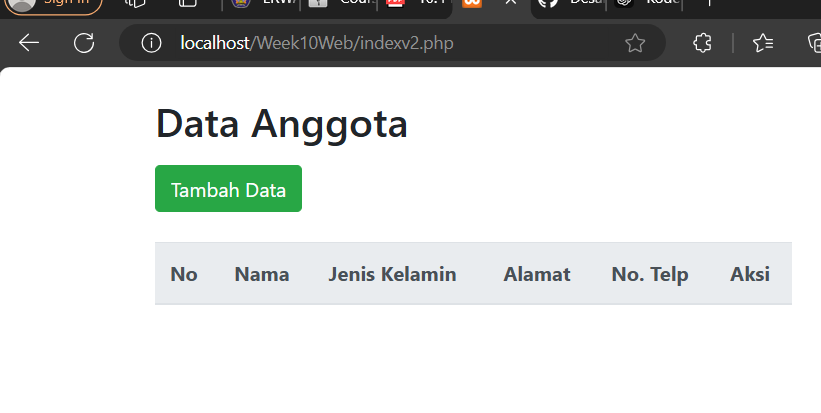




-This code snippet handles the deletion of a member record from the anggota table in the database. If the aksi parameter is set to 'hapus' and a valid id is provided in the URL, it retrieves the id, then executes a DELETE query to remove the record with that id. If the deletion is successful, it redirects 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**

1. 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)

****

-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.

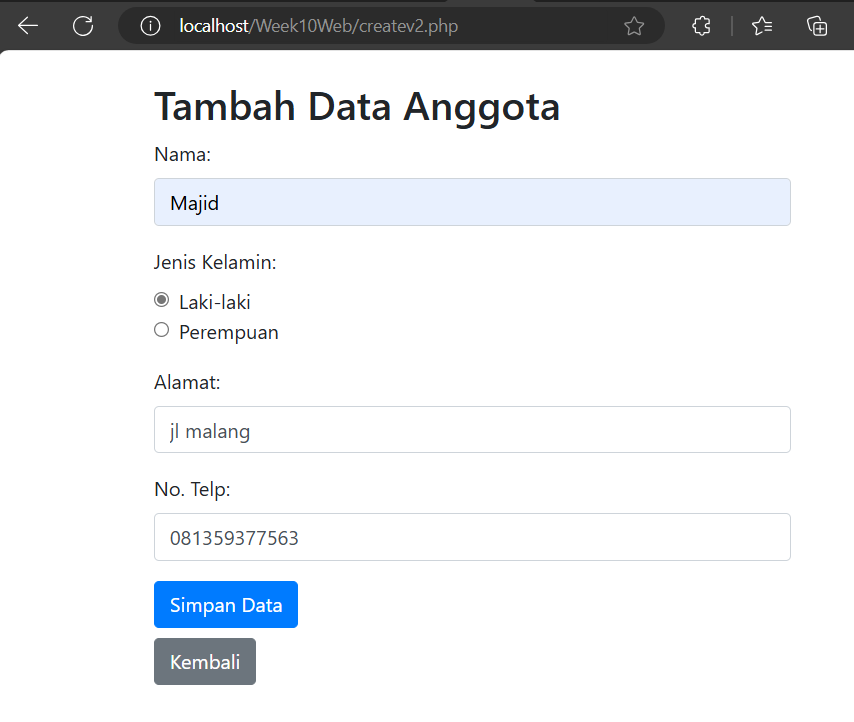
index: Contains a single custom JavaScript function konfirmasiHapus to handle deletion confirmation without using external libraries, relying only on standard JavaScript.

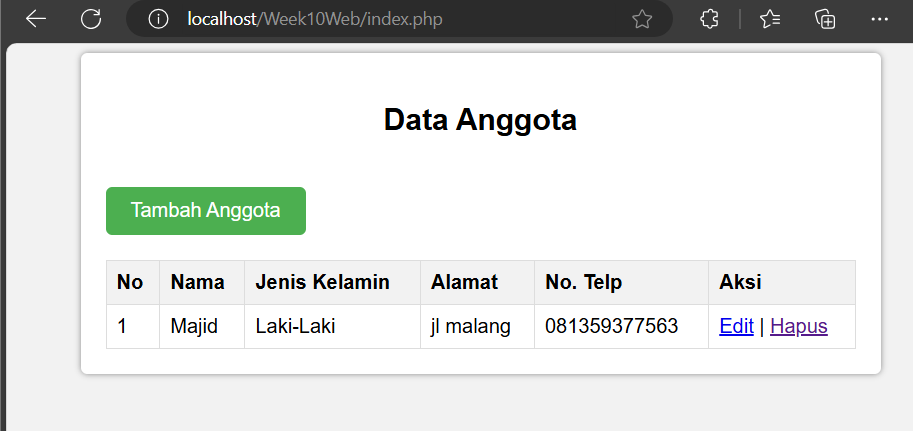
HTML Structure and Elements:

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.

1. 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)

- 

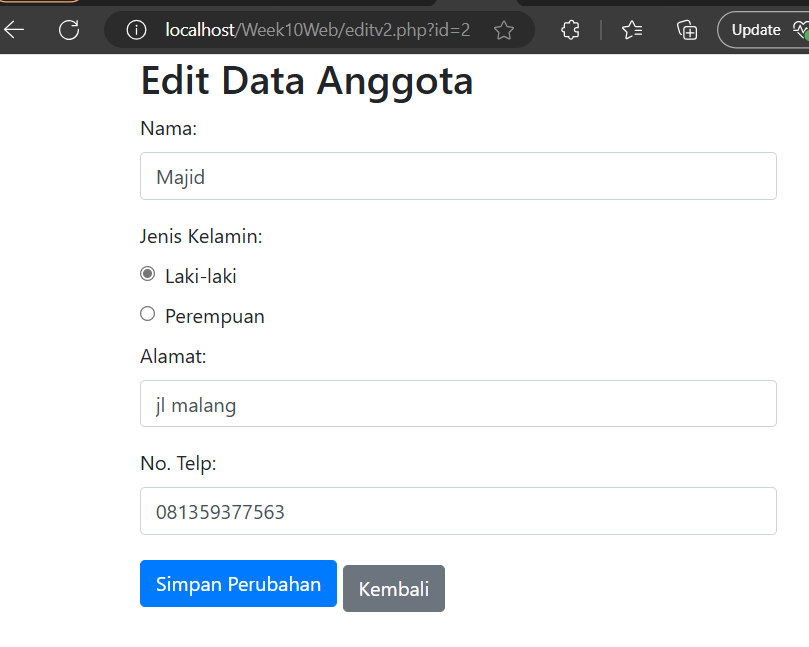


- Styling and Framework:

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.

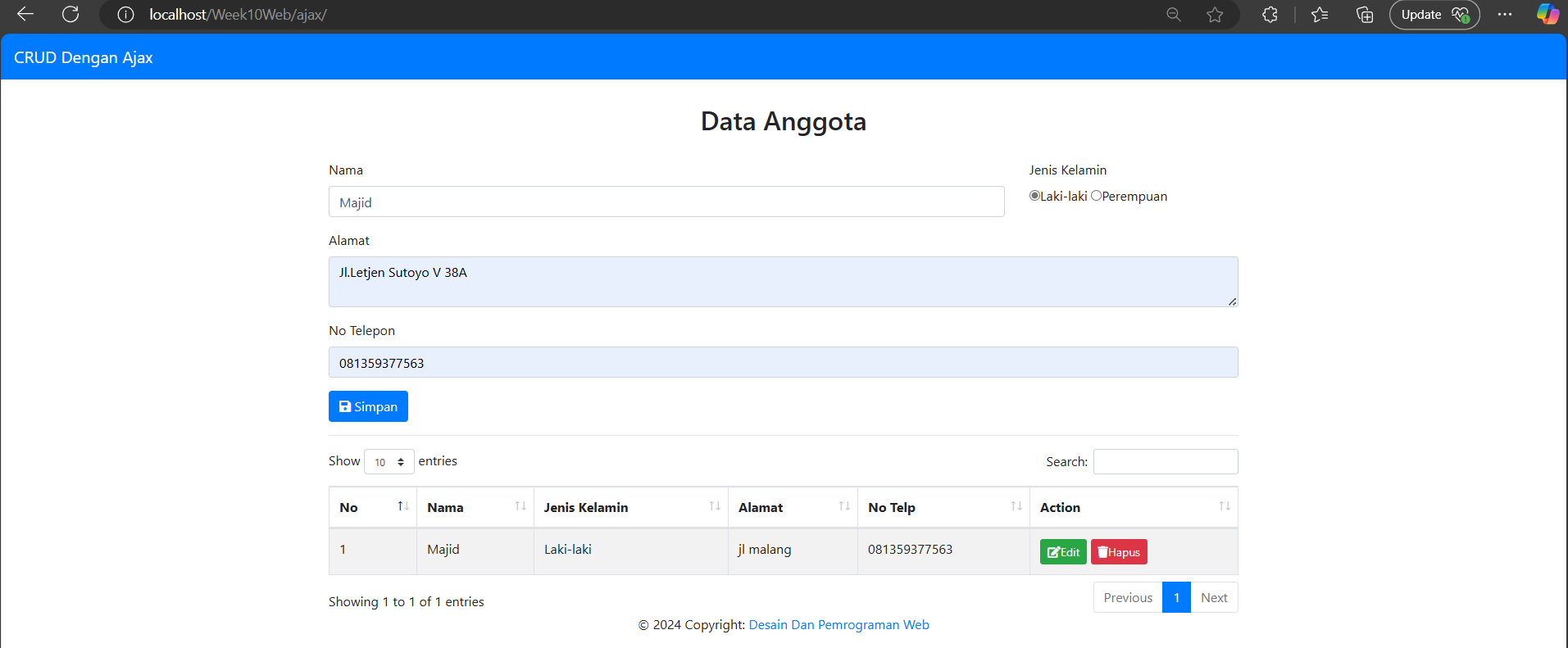
V2 : Uses Bootstrap 4, providing a polished and responsive layout with minimal effort. Bootstrap classes (e.g., form-control, form-group, btn, btn-primary) give the form a consistent and modern appearance with built-in styling.

1. 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)



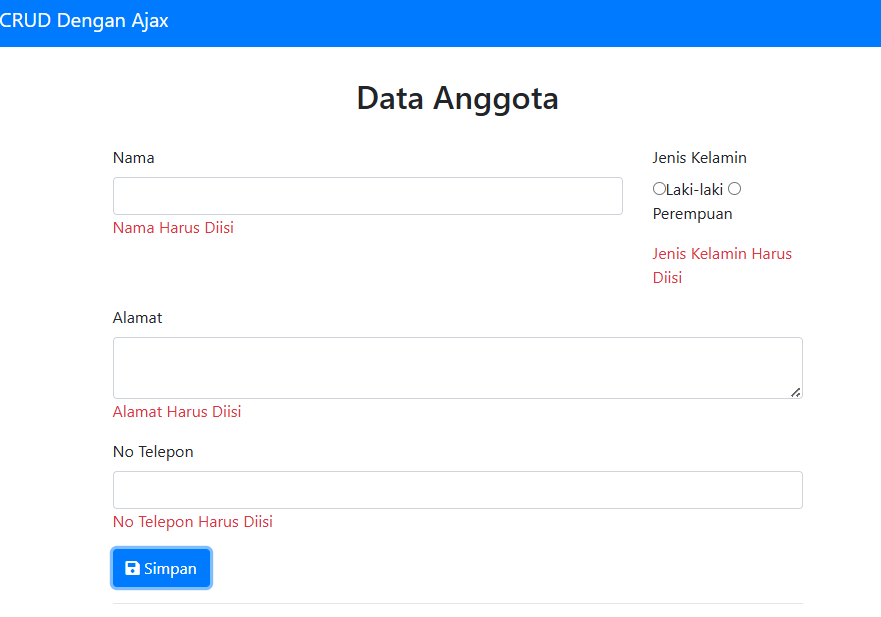
-The different is more styling than v1

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

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

- 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.

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

(Question No. 9)

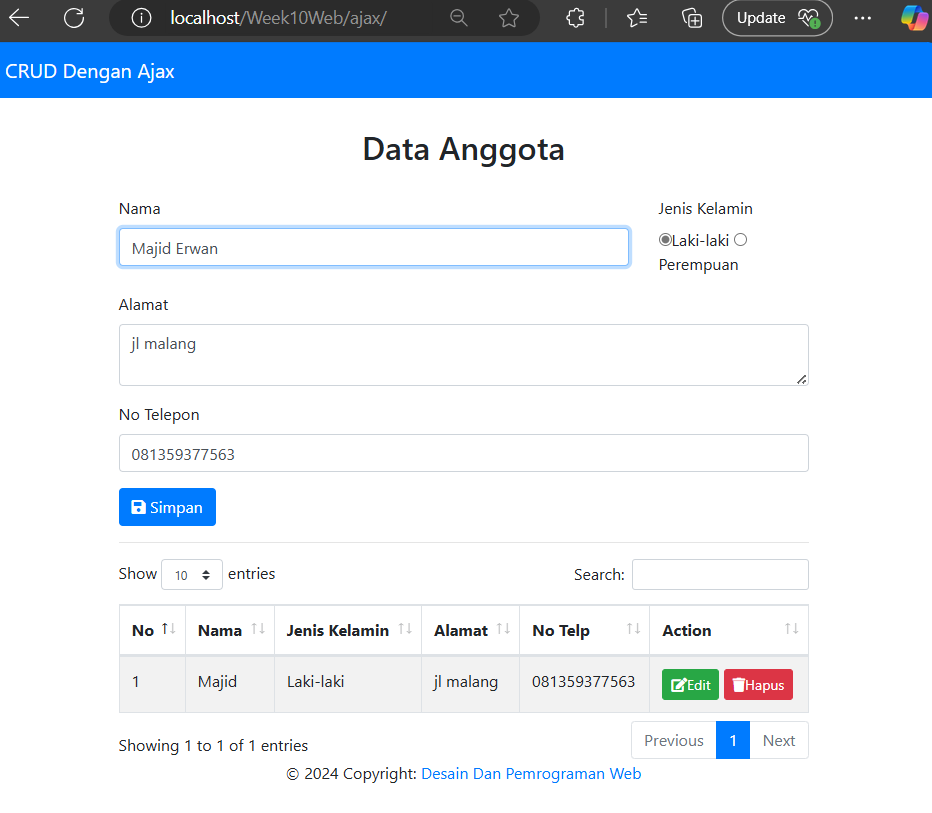
-

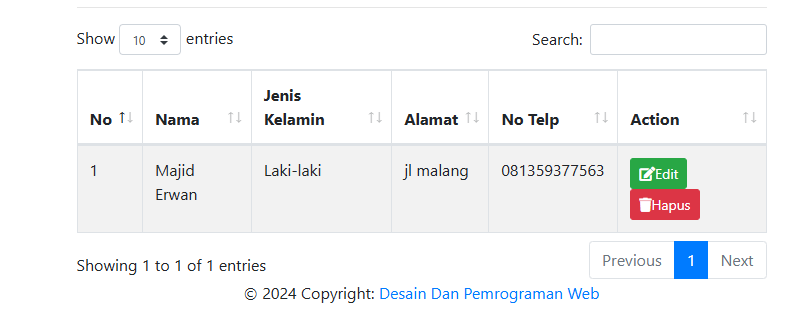
- JavaScript checks if each input field (nama, alamat, jenis\_kelamin, no\_telp) is filled out.

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.

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





- 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").

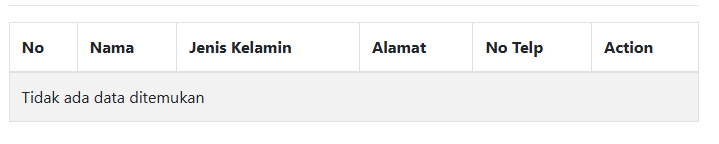
- AJAX Request to form\_action.php:

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.

1. 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)

- 

- 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.