Department of Information Technology, State Polytechnic of Malang

**Jobsheet-10: PHP – CRUD & Bootstrap Web Design and Programming Courses** Web Design and Programming Teaching Team October 2024

**Topic**

- CRUD concepts with PHP and Bootstrap

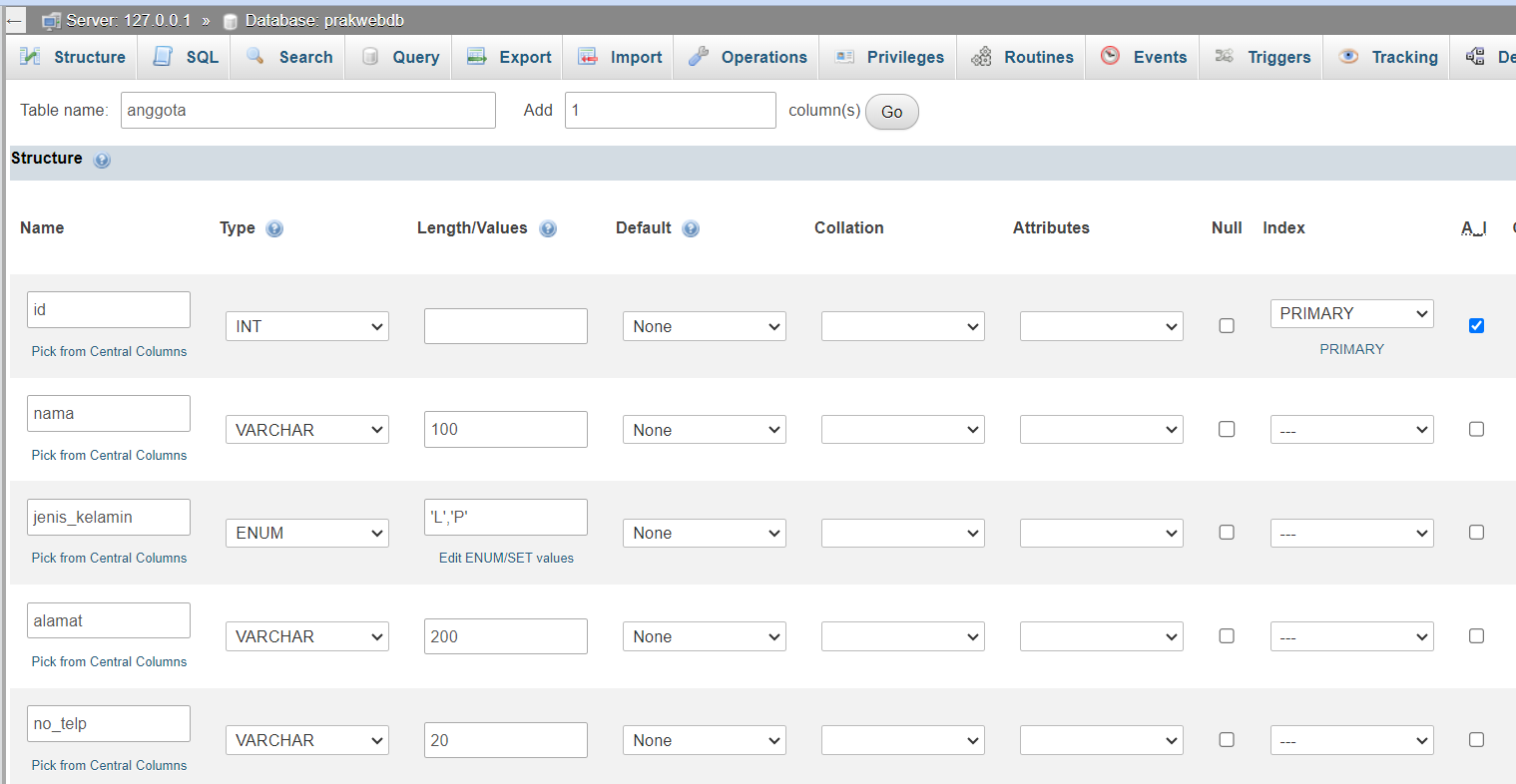
**Objectives**

Students are expected to:

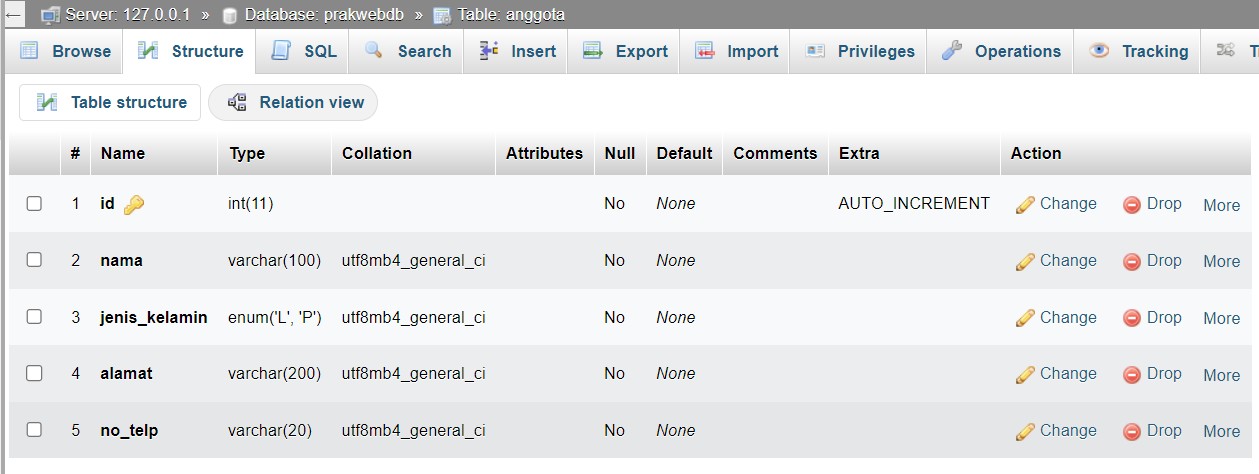
1. Students are able to create CRUD forms using PHP
2. Students are able to create CRUD forms using Bootsrtap
3. Students are able to create CRUD forms using Jquery and Ajax

# Creating a Member (Anggota) Table

The first step in creating a CRUD feature is to create a “anggota” table. In this practicum, we will use *a database*. Create *a database* on MySQL named "prakwebdb". Next, the thing that must be done is to create *a “anggota”* table. The *“anggota” table* consists of *nama, jenis\_kelamin, alamat,* and *no\_telp*. The following are the conditions made in the *“anggota” table*:



If successful, it will appear as below, make sure *jenis\_kelamin* is of the enum type. An enum is a type of data in a Mysql *database* that contains predefined string values. An enum column can only contain one of several specified string values .



**Practical Section 1. CRUD - Read**

|  |  |
| --- | --- |
| **Step** | **Information** |
| 1 | Create a new file in the week10 directory, naming it koneksi.php. |
| 2 | Type into the koneksi.php file the code below. |
| 3 |  |
| 4 | Create a new file in the week10 directory, naming it index.php. |
| 5 | Type into the index.php file the code below. |

|  |  |
| --- | --- |
|  |  |
| 6 | Create a new file in the week10 directory, naming it style.css. |
| 7 | Type into the style.css file the code below. |

|  |  |
| --- | --- |
|  |  |
| 8 | 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 script connects to the database, fetches anggota data, and displays it |

# Practical Section 2. CRUD - Create

|  |  |
| --- | --- |
| **Step** | **Information** |
| 1 | Create a new file in the week10 directory, naming it create.php. |
| 2 | Type into the create.php file the code below. |

|  |  |
| --- | --- |
| 3 |  |
| 4 | Create a new file in the week10 directory, naming it proses.php. |
| 5 | Type into the proses.php file the code below. |
| 6 |  |
| 7 | 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 script collects data through create.php, sends it to proses.php to insert into the anggota table |

**Practical Section 3. CRUD - Update**

|  |  |
| --- | --- |
| **Step** | **Information** |

|  |  |
| --- | --- |
| 1 | Create a new file in the week10 directory, naming it edit.php. |
| 2 | Type into the edit.php file the code below. |
| 3 |  |
| 4 | Add the contents of the file inside the week10 directory, with the name proses.php. |
| 5 | Type into the proses.php file the code below. |
| 6 |  |
| 7 | 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 script fetches existing member data, allows editing in edit.php, and updates the record |

# Practical Section 4. CRUD Section Delete

|  |  |
| --- | --- |
| **Step** | **Information** |
| 1 | Add the contents of the file inside the week10 directory, with the name  proses.php. |
| 2 | Type into the proses.php file the code below. |
| 3 |  |
| 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 script removes the specified member record from the anggota table |

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

|  |  |
| --- | --- |
| **Step** | **Information** |
| 1 | Change the contents of the file to the name index.php in the week10 directory. |
| 2 | Type into the index.php file the code below. |

|  |  |
| --- | --- |
| 3 |  |
| 4 | 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) |

|  |  |
| --- | --- |
| 5 | Change the contents of the file to the name create.php in the week10 directory. |
| 6 | Type into the create.php file the code below. |
| 7 |  |
| 8 | 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) |
| 9 | Change the contents of the file in the week10 directory, with the name edit.php |
| 10 | Type into the edit.php file the code below. |

|  |  |
| --- | --- |
| 11 |  |
| 12 | 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 edit.php script is responsible for allowing users to modify existing records in the database. It retrieves data from the database (based on a specific ID) and displays it in a form. Users can edit the information and submit the form, which updates the corresponding record in the database. |

# Practical Section 6. CRUD Display with Ajax

|  |  |
| --- | --- |
| **Step** | **Information** |
| 1 | Create a new folder inside week10 with the name ajax. In that folder, create a new file with the name koneksi.php. |
| 2 | Type into the koneksi.php file the code below. |

|  |  |
| --- | --- |
| 3 |  |
| 4 | Create a **auth.php** file to authenticate it and for its security. Fill in the code below |
| 5 |  |
| 6 | Create a **index.php** file and fill it in with the script below |

|  |  |
| --- | --- |
| 7 |  |

|  |  |
| --- | --- |
|  | <!-- Bootstrap -->  <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter- bootstrap/4.1.3/css/bootstrap.css" rel="stylesheet">  <!-- Font Awesome -->  <link rel="stylesheet"  href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384- fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9 Sr" crossorigin="anonymous"> |
| 8 | Create a **data.php** file that functions to reload data using ajax so that the application does not need to load when saving data or there is data update. Fill in **data.php** with the script below |
| 9 |  |
| 10 | For security because we use headers in ajax, we need to create a **csrf.php** file and then fill it with the script below |

|  |  |
| --- | --- |
| 11 |  |
| 12 | Save the file, then open a browser and run localhost/week10/ajax. It should appear like the image below    What do you understand from the scripts on those files. Record below your understanding. (Question No. 8)    The AJAX integration allows the CRUD interface to dynamically update without refreshing the entire page.   * **Purpose:** It enhances user experience by making the application more interactive and faster. * **How it works:**   + Data is reloaded asynchronously via a data.php script.   + New records or updates are sent using AJAX calls, reducing the need to reload the whole page.   + Security is handled using CSRF tokens. |
| 13 | Next, create a form with data that can be saved. Open the **index.php** file and add some scripts below so that the ones added are only for the error check section with examples |

|  |  |
| --- | --- |
| 14 |  |
| 15 | Still in the **index.php** file with the addition of the javascript script, insert the script below in the jquery that is already in step 7 |
| 16 |  |

|  |  |
| --- | --- |
|  |  |
| 17 | Next is to create a **form\_action.php** file that functions to save into the database. Then fill it in with the script below |
| 18 |  |
| 19 | Save the file, then open a browser and run localhost/week10/ajax. It should appear like the image below if you click the save button without filling in any data    What do you understand from the scripts on those files. Record below your understanding. (Question No. 9)  The scripts in the files handle data submission dynamically using AJAX. When the "Save" button is clicked, the form data is sent to form\_action.php via an AJAX POST request. The server validates and processes the data, then inserts it into the database. If any errors occur during the process (e.g., missing required fields), appropriate error messages are displayed without reloading the page. |

|  |  |
| --- | --- |
| 20 | Next, create a form with data that can be changed. Open the **data.php** file and add some scripts below so the ones added in the jquery section |
| 21 |  |
| 22 | Create a **get\_data.php** file and fill it in with the script below |
| 23 |  |
| 24 | Finally, change the **form\_action.php** file to the one below so that you can edit the form |

|  |  |
| --- | --- |
| 25 |  |
| 26 | Save the file, then open a browser and run localhost/week10/ajax. When clicking the edit button, it should appear as shown in the image below with the form auto-filled from the previous data    What do you understand from the scripts on those files. Record below your understanding. (Question No. 10)  When editing data using AJAX:   * Clicking the edit button triggers an AJAX call to fetch the specific record using its ID (get\_data.php). * The retrieved data pre-fills a modal form for editing. * Submitting the form sends the updated data to the server (form\_action.php), which processes the update without reloading the page. |
| 27 | On the <script>... </script> in the data.php file add the code below |

|  |  |
| --- | --- |
| 28 |  |
| 29 | Create a **hapus\_data.php** file on your project and type the script below |
| 30 |  |
| 31 | 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 scripts in the hapus\_data.php file and the JavaScript part of data.php handle the deletion of data dynamically without reloading the entire page. When the delete button is clicked, an AJAX request is sent to the hapus\_data.php file with the ID of the record to be deleted. The server processes this request, deletes the corresponding record from the database, and returns a response. The client-side JavaScript then updates the page to reflect the removal of the deleted record. |

Nama : Bayu Putra Laksmana

Class : 2I

Number : 06

NIM : 2341720240

Link GitHub : <https://github.com/putra-afk/Web-Programming/tree/main/week10>