

Praktikum 8: AJAX

Tujuan

1. Memahami konsep AJAX dan cara kerjanya.
2. Mampu mengimplementasikan AJAX pada aplikasi web dengan CodeIgniter 4.
3. Melatih kemampuan problem solving dan debugging.

Instruksi Praktikum

1. Persiapkan text editor misalnya **VSCode**.
2. Buka kembali folder dengan nama **lab7_php_ci** pada docroot webserver (**htdocs**)
3. Ikuti langkah-langkah praktikum yang akan dijelaskan berikutnya.

Apa itu AJAX?

AJAX merupakan singkatan dari **Asynchronous JavaScript and XML**. Meskipun kepanjangannya menyebutkan XML, pada praktiknya AJAX tidak terbatas pada penggunaan XML saja. AJAX adalah kumpulan teknik pengembangan web yang memungkinkan aplikasi web bekerja secara **asynchronous** (tidak langsung).

Dengan kata lain, AJAX memungkinkan aplikasi web untuk memperbarui dan menampilkan data dari server tanpa harus melakukan reload halaman secara keseluruhan. Hal ini membuat aplikasi web terasa lebih responsif dan dinamis.

Cara Kerja AJAX

AJAX bekerja dengan cara berikut:

1. **Event Trigger:**
Pengguna melakukan suatu aksi di halaman web, misalnya menekan tombol atau mengetikkan sesuatu pada formulir.
2. **Request Dikirim:**
JavaScript di browser akan membuat request HTTP ke server. Request ini biasanya berupa request GET atau POST, dan bisa membawa data tambahan jika diperlukan.
3. **Server Memproses Request:**
Server menerima request dan memprosesnya sesuai dengan kebutuhan. Server bisa mengambil data dari database, melakukan kalkulasi tertentu, atau melakukan aksi lainnya.
4. **Respon Dikembalikan:**
Server mengirimkan respon kembali ke browser. Respon ini biasanya berupa data dalam format JSON (JavaScript Object Notation) atau format lainnya.
5. **Data Ditampilkan:**
JavaScript di browser menerima respon dari server. JavaScript kemudian memproses data tersebut dan memperbarui bagian tertentu dari halaman web tanpa perlu melakukan reload keseluruhan.

Keuntungan menggunakan AJAX:

- **Meningkatkan User Experience (UX):** Hal ini karena halaman web tidak perlu dimuat ulang setiap kali ada interaksi pengguna, sehingga membuat aplikasi web terasa lebih responsif dan dinamis.

- **Menghemat Bandwidth:** Hanya data yang dibutuhkan yang dikirimkan antara browser dan server, sehingga menghemat bandwidth dan mempercepat loading halaman.
- **Mempertahankan State Aplikasi:** Dengan AJAX, state aplikasi (misalnya data yang sedang diedit) bisa dipertahankan walaupun halaman tidak di-reload.

Contoh Penggunaan AJAX:

- Live chat applications
- Autocomplete suggestions
- Real-time updates (misalnya pada papan skor pertandingan olahraga)
- Validasi formulir secara real-time
- Dan masih banyak lagi

Dengan memahami konsep dan cara kerja AJAX, Anda dapat membuat aplikasi web yang lebih interaktif dan menarik bagi pengguna.

Langkah-langkah Praktikum

Persiapan

Buka Kembali project sebelumnya, kemudian kita tambahkan Pustaka yang dibutuhkan pada project tersebut.

Menambahkan Pustaka jQuery.

Kita akan menggunakan pustaka jQuery untuk mempermudah proses AJAX. Download pustaka jQuery versi terbaru dari <https://jquery.com> dan ekstrak filenya.

Salin file **jquery-3.6.0.min.js** ke folder **public/assets/js**.

Membuat Model.

Pada modul sebelumnya sudah dibuat ArtikelModel, pada modul ini kita akan memanfaatkan model tersebut agar dapat diakses melalui AJAX.

Membuat AJAX Controller

```
<?php

namespace App\Controllers;

use CodeIgniter\Controller;
use CodeIgniter\HTTP\Request;
use CodeIgniter\HTTP\Response;
use App\Models\ArtikelModel;

class AjaxController extends Controller
{
    public function index()
    {
        return view('ajax/index');
    }

    public function getData()
```

```

    {
        $model = new ArtikelModel();
        $data = $model->findAll();

        // Kirim data dalam format JSON
        return $this->response->setJSON($data);
    }

    public function delete($id)
    {
        $model = new ArtikelModel();
        $data = $model->delete($id);

        $data = [
            'status' => 'OK'
        ];

        // Kirim data dalam format JSON
        return $this->response->setJSON($data);
    }
}

```

Membuat View

```

<?= $this->include('template/header'); ?>

<h1>Data Artikel</h1>

<table class="table-data" id="artikelTable">
    <thead>
        <tr>
            <th>ID</th>
            <th>Judul</th>
            <th>Status</th>
            <th>Aksi</th>
        </tr>
    </thead>
    <tbody></tbody>
</table>

<script src="<?= base_url('assets/js/jquery-3.6.0.min.js') ?>"></script>
<script>
    $(document).ready(function() {
        // Function to display a loading message while data is fetched
        function showLoadingMessage() {
            $('#artikelTable tbody').html('<tr><td colspan="4">Loading
data...</td></tr>');

```

```

    }

    // Buat fungsi load data
    function loadData() {
        showLoadingMessage(); // Display loading message initially

        // Lakukan request AJAX ke URL getData
        $.ajax({
            url: "<?= base_url('ajax/getData') ?>",
            method: "GET",
            dataType: "json",
            success: function(data) {
                // Tampilkan data yang diterima dari server
                var tableBody = "";
                for (var i = 0; i < data.length; i++) {
                    var row = data[i];
                    tableBody += '<tr>';
                    tableBody += '<td>' + row.id + '</td>';
                    tableBody += '<td>' + row.judul + '</td>';
                    // Add a placeholder for the "Status" column (modify
as needed)
                    tableBody += '<td><span class="status">---
</span></td>';

                    tableBody += '<td>';
                    // Replace with your desired actions (e.g., edit,
delete)
                    tableBody += '<a href="<?= base_url('artikel/edit/'
?>' + row.id + '" class="btn btn-primary">Edit</a>';
                    tableBody += ' <a href="#" class="btn btn-danger
btn-delete" data-id="' + row.id + '">Delete</a>';
                    tableBody += '</td>';
                    tableBody += '</tr>';
                }
                $('#artikelTable tbody').html(tableBody);
            }
        });
    }

    loadData();

    // Implement actions for buttons (e.g., delete confirmation)
    $(document).on('click', '.btn-delete', function(e) {
        e.preventDefault();
        var id = $(this).data('id');
        // Add confirmation dialog or handle deletion logic here

        // hapus data;
        if (confirm('Apakah Anda yakin ingin menghapus artikel ini?'))
    {

```

```
        $.ajax({
            url: "<?= base_url('artikel/delete/') ?>" + id,
            method: "DELETE",
            success: function(data) {
                loadData(); // Reload Datatables to reflect changes
            },
            error: function(jqXHR, textStatus, errorThrown) {
                alert('Error deleting article: ' + textStatus +
errorThrown);
            }
        });
    }
    console.log('Delete button clicked for ID:', id);
});
});
</script>

<?= $this->include('template/footer'); ?>
```

Pertanyaan dan Tugas

Selesaikan programnya sesuai Langkah-langkah yang ada. Tambahkan fungsi untuk tambah dan ubah data. Anda boleh melakukan improvisasi.

Laporan Praktikum

1. Melanjutkan praktikum sebelumnya pada **repository** dengan nama **Lab11Web**.
2. Kerjakan semua latihan yang diberikan sesuai urutannya.
3. Screenshot setiap perubahannya.
4. Update file **README.md** dan tuliskan penjelasan dari setiap langkah praktikum beserta screenshotnya.
5. **Commit** hasilnya pada **repository** masing-masing.
6. Kirim **URL repository** pada **e-learning** ecampus