

## Praktikum 9: Implementasi AJAX Pagination dan Search

### Tujuan

1. Mahasiswa mampu memahami konsep dasar AJAX untuk pagination dan search.
2. Mahasiswa mampu mengimplementasikan pagination dan search menggunakan AJAX dalam CodeIgniter 4.
3. Mahasiswa mampu meningkatkan performa dan User Experience (UX) aplikasi web.

### Instruksi Praktikum

1. Persiapkan text editor (VSCode).
2. Buka kembali folder proyek lab7\_php\_ci.
3. Ikuti langkah-langkah praktikum berikut.

### Langkah-langkah Praktikum

#### 1. Persiapan

- \* Pastikan MySQL Server sudah berjalan.
- \* Buka database `lab\_ci4`.
- \* Pastikan tabel `artikel` dan `kategori` sudah ada dan terisi data.
- \* Pastikan library jQuery sudah terpasang atau dapat diakses melalui CDN.

#### 2. Modifikasi Controller Artikel

Ubah method `admin\_index()` di `Artikel.php` untuk mengembalikan data dalam format JSON jika request adalah AJAX. (Sama seperti modul sebelumnya)

```
...

public function admin_index()
{
    $title = 'Daftar Artikel (Admin)';
    $model = new ArtikelModel();

    $q = $this->request->getVar('q') ?? '';
    $kategori_id = $this->request->getVar('kategori_id') ?? '';
    $page = $this->request->getVar('page') ?? 1;

    $builder = $model->table('artikel')
        ->select('artikel.*, kategori.nama_kategori')
        ->join('kategori', 'kategori.id_kategori =
artikel.id_kategori');

    if ($q != '') {
        $builder->like('artikel.judul', $q);
    }

    if ($kategori_id != '') {
        $builder->where('artikel.id_kategori', $kategori_id);
    }
}
```

```

$artikel = $builder->paginate(10, 'default', $page);
$pager = $model->pager;

$data = [
    'title' => $title,
    'q' => $q,
    'kategori_id' => $kategori_id,
    'artikel' => $artikel,
    'pager' => $pager
];

if ($this->request->isAJAX()) {
    return $this->response->setJSON($data);
} else {
    $kategoriModel = new KategoriModel();
    $data['kategori'] = $kategoriModel->findAll();
    return view('artikel/admin_index', $data);
}
}

...

```

**Penjelasan:**

- ``$page = $this->request->getVar('page') ?? 1;``: Mendapatkan nomor halaman dari request. Jika tidak ada, default ke halaman 1.
- ``$builder->paginate(10, 'default', $page);``: Menerapkan pagination dengan nomor halaman yang diberikan.
- ``$this->request->isAJAX();``: Memeriksa apakah request yang datang adalah AJAX.
- Jika AJAX, kembalikan data artikel dan pager dalam format JSON.
- Jika bukan AJAX, tampilkan view seperti biasa.

**3. Modifikasi View (admin\_index.php)**

- \* Ubah view ``admin_index.php`` untuk menggunakan jQuery.
- \* Hapus kode yang menampilkan tabel artikel dan pagination secara langsung.
- \* Tambahkan elemen untuk menampilkan data artikel dan pagination dari AJAX.
- \* Tambahkan kode jQuery untuk melakukan request AJAX.

```

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

<h2><?= $title; ?></h2>

<div class="row mb-3">
    <div class="col-md-6">
        <form id="search-form" class="form-inline">
            <input type="text" name="q" id="search-box" value="<?= $q; ?>"
placeholder="Cari judul artikel" class="form-control mr-2">

```

```

        <select name="kategori_id" id="category-filter" class="form-
control mr-2">
            <option value="">Semua Kategori</option>
            <?php foreach ($kategori as $k): ?>
                <option value="<?=$k['id_kategori']; ?>" <?=$k['id_kategori']
== $k['id_kategori']) ? 'selected' : ''; ?><?=$k['nama_kategori'];
?></option>
            <?php endforeach; ?>
        </select>
        <input type="submit" value="Cari" class="btn btn-primary">
    </form>
</div>
</div>

<div id="article-container">
</div>

<div id="pagination-container">
</div>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    $(document).ready(function() {
        const articleContainer = $('#article-container');
        const paginationContainer = $('#pagination-container');
        const searchForm = $('#search-form');
        const searchBox = $('#search-box');
        const categoryFilter = $('#category-filter');

        const fetchData = (url) => {
            $.ajax({
                url: url,
                type: 'GET',
                dataType: 'json',
                headers: {
                    'X-Requested-With': 'XMLHttpRequest'
                },
                success: function(data) {
                    renderArticles(data.artikel);
                    renderPagination(data.pager, data.q, data.kategori_id);
                }
            });
        };

        const renderArticles = (articles) => {
            let html = '<table class="table">';

            html +=
            '<thead><tr><th>ID</th><th>Judul</th><th>Kategori</th><th>Status</th><th>Aksi</th></tr></thead><tbody>';

```

```

        if (articles.length > 0) {
            articles.forEach(article => {
                html += `
                    <tr>
                        <td>${article.id}</td>
                        <td>
                            <b>${article.judul}</b>
                            <p><small>${article.isi.substring(0,
50)}</small></p>
                        </td>
                        <td>${article.nama_kategori}</td>
                        <td>${article.status}</td>
                        <td>
                            <a class="btn btn-sm btn-info"
href="/admin/artikel/edit/${article.id}">Ubah</a>
                            <a class="btn btn-sm btn-danger" onclick="return
confirm('Yakin menghapus data?');"
href="/admin/artikel/delete/${article.id}">Hapus</a>
                        </td>
                    </tr>
                `;
            });
        } else {
            html += '<tr><td colspan="5">Tidak ada data.</td></tr>';
        }
        html += '</tbody></table>';
        articleContainer.html(html);
    };

    const renderPagination = (pager, q, kategori_id) => {
        let html = '<nav><ul class="pagination">';
        pager.links.forEach(link => {
            let url = link.url ?
`${link.url}&q=${q}&kategori_id=${kategori_id}` : '#';
            html += `<li class="page-item ${link.active ? 'active' :
''}><a class="page-link" href="${url}">${link.title}</a></li>`;
        });
        html += '</ul></nav>';
        paginationContainer.html(html);
    };

    searchForm.on('submit', function(e) {
        e.preventDefault();
        const q = searchBox.val();
        const kategori_id = categoryFilter.val();
        fetchData(`/admin/artikel?q=${q}&kategori_id=${kategori_id}`);
    });

    categoryFilter.on('change', function() {

```

```
        searchForm.trigger('submit');
    });

    // Initial load
    fetchData('/admin/artikel');
});
</script>

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

### Pertanyaan dan Tugas

1. Selesaikan semua langkah praktikum di atas.
2. Modifikasi tampilan data artikel dan pagination sesuai kebutuhan desain.
3. Tambahkan indikator loading saat data sedang diambil dari server.
4. Implementasikan fitur sorting (mengurutkan artikel berdasarkan judul, dll.) dengan AJAX.

### Laporan Praktikum

1. Lanjutkan praktikum pada repository Lab7Web.
2. Kerjakan semua latihan sesuai urutan.
3. Screenshot setiap perubahan dan hasil uji coba.
4. Update file README.md dengan penjelasan dan screenshot.
5. Commit hasil praktikum.
6. Kirim URL repository.