

LAPORAN PRAKTIKUM LAB D1
PENGEMBANGAN BERBASIS PLATFORM
“AJAX”



DISUSUN OLEH:
RESMA ADI NUGROHO
24060121120021

DEPARTEMEN INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO
2023

BAB I

PENDAHULUAN

1.1 Tujuan

- a. Membuat objek file XMLHttpRequest sebagai core Ajax
- b. Melakukan request ke server melalui AJAX menggunakan method GET
- c. Melakukan request ke server melalui AJAX menggunakan method POST

1.2 Rumusan Masalah

- a. Buatlah file untuk pencarian data buku berdasarkan judul yang diketik, lalu menampilkan detail buku pada halaman ynag sama menggunakan Ajax.

BAB II

DASAR TEORI

2.1 Ajax

Ajax (Asynchronous JavaScript and XML) adalah sebuah teknologi yang memungkinkan pertukaran data antara server dan browser tanpa harus melakukan refresh pada halaman web. Teknologi ini menggabungkan beberapa teknologi web, seperti JavaScript, XML (meskipun saat ini lebih sering menggunakan JSON), HTML, dan CSS, untuk menciptakan pengalaman web yang lebih dinamis dan responsif. Dengan kata lain, Ajax memungkinkan aplikasi web untuk mengirim dan menerima data dari server dalam latar belakang tanpa mengganggu tampilan atau perilaku halaman yang sedang ditampilkan oleh pengguna.

Salah satu karakteristik utama Ajax adalah "asynchronous," yang berarti operasi-operasi yang melibatkan komunikasi dengan server tidak harus menunggu hingga operasi sebelumnya selesai. Ini berarti bahwa halaman web dapat tetap responsif dan interaktif sementara data diunduh atau diunggah ke server. Misalnya, ketika pengguna mengisi formulir, data yang dikirimkan ke server bisa berlangsung di latar belakang tanpa perlu memuat ulang seluruh halaman.

Teknologi utama yang digunakan dalam Ajax adalah JavaScript, yang digunakan untuk membuat permintaan ke server, mengelola respon, dan memperbarui tampilan halaman web. Selain itu, Ajax biasanya menggunakan XML (meskipun lebih sering menggunakan JSON) untuk membungkus data yang dikirim antara server dan browser. Ini memungkinkan data yang dikirim dalam format terstruktur dan dapat diuraikan dengan mudah.

Keberhasilan Ajax telah mempengaruhi evolusi web modern dan meningkatkan pengalaman pengguna di berbagai aplikasi web, termasuk aplikasi email, jejaring sosial, perbelanjaan online, dan banyak lagi. Ini memungkinkan aplikasi web untuk terasa lebih cepat dan responsif, seiring dengan meningkatnya antarmuka pengguna yang semakin dinamis dan interaktif. Selain itu, Ajax adalah dasar dari banyak kerangka kerja pengembangan web, memudahkan para pengembang untuk mengimplementasikan teknologi ini dengan lebih mudah dan efisien.

BAB III

PEMBAHASAN

1.1 Search Books Ajax

File view_books.php

```
<?php include('./header.php') ?>
<script>
// searchBooks() saat halaman dimuat
window.addEventListener('load', function() {
    searchBooks();
});
</script>
<div class="card mt-5">
    <div class="card-header">Books Data</div>
    <div class="card-body">
        <input class="form-control mb-2" type="text" id="searchInput"
onkeyup="searchBooks()" placeholder="Cari berdasarkan judul...">
        <div id="searchResults"></div>
        <div id="bookDetail"></div>
    </div>
</div>
<?php include('./footer.php') ?>
```

Kode diatas merupakan kode untuk halaman view books yakni halaman yang nantinya akan berisi list buku yang akan dicari beserta detail bukunya apabila menekan tombol detail. Pada bagian awal dan akhir kode terdapat untuk masing-masingnya include file header dan footer. Setelah itu dibawahnya terdapat kode javascript yakni ketika halaman baru diload maka akan menjalankan fungsi searchBooks, yakni fungsi yang akan menampilkan list buku. Namun dalam hal ini karena parameter pencarian masih kosong jadi list buku akan ditampilkan secara semuanya. Setelah itu pada bagian bawahnya terdapat beberapa element div html untuk menampilkan tampilan list buku dan search buku. Dalam element input juga perlu diperhatikan, kita harus menambahkan onkeyup dan akan memanggil fungsi searchBooks agar ketika pengguna memasukan kata kunci pada search bar maka pencarian dapat langsung dilakukan secara realtime. Kemudian pada bagian bawah terdapat div untuk menampilkan list buku dan bagian bawahnya terdapat div untuk menampilkan hasil detail dari buku tersebut.

File ajax.js

```
function searchBooks() {
    var searchTerm = document.getElementById('searchInput').value;
    var xhr = new XMLHttpRequest();
    xhr.open('GET', 'search_books.php?searchTerm=' + searchTerm,
true);
    xhr.onreadystatechange = function () {
        if (xhr.readyState === 4 && xhr.status === 200) {
            document.getElementById('searchResults').innerHTML =
xhr.responseText;
        }
    };

    xhr.send();
}

function showBookDetail(isbn) {
    var xhr = new XMLHttpRequest();
    var url = 'get_book_detail.php?isbn=' + isbn;
    xhr.open('GET', url, true);
    xhr.onreadystatechange = function () {
        if (xhr.readyState === 4 && xhr.status === 200) {

            var bookDetail = document.getElementById('bookDetail');
            bookDetail.innerHTML = xhr.responseText;

        }
    };
    xhr.send();
}
```

Kode diatas merupakan potongan kode dari file ajax.js yang berisi kode untuk melakukan pencarian buku dan melihat detail buku. Pada fungsi searchBooks akan didapatkan value yang akan disearch dengan mendapatkannya melalui element id searchinput yang merupakan elemen input pada html. Kemudian akan dibuat sebuah variable untuk XMLHttpRequest untuk nantinya akan digunakan untuk melakukan request pada file php nantinya. Kemudian setelah itu dapat dilakukan pengaksesan file php untuk searchbooks dengan query merupakan string judul yang sudah didapatkan sebelumnya. Setelah itu akan diperiksa, apabila request berhasil dijalankan maka hasil dari request tersebut akan ditampilkan pada element searchResults. Kemudian pada bagian akhir kode fungsi tersebut terdapat fungsi untuk mengirimkan request tersebut dengan fungsi send. Untuk fungsi selanjutnya yakni showBookDetail sebenarnya kurang lebih memiliki kode yang sama seperti pada searchBooks namun pada fungsi ini url yang akan diakses berbeda dengan searchBooks. Pada showBookDetail akan mengakses file get_book_detail.php dengan query isbn yang didapatkan dari isbn yang sudah terinclude pada button detail

nantinya. Setelah itu apabila request berhasil maka akan ditampilkan hasil detailnya pada element bookDetail.

File search_books.php

```
<?php
require_once('./lib/db_login.php');

if (isset($_GET['searchTerm'])) {
    $searchTerm = $_GET['searchTerm'];
    $query = "SELECT * FROM books WHERE title LIKE
'%" . $searchTerm . "%'";
    $result = $db->query($query);

    if ($result) {
        echo "<table class='table table-striped'>
<tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Action</th>
</tr>";
        while ($row = $result->fetch_object()) {
            echo '<tr>';
            echo '<td>' . $row->isbn . '</td>';
            echo '<td>' . $row->title . '</td>';
            echo '<td><button class="btn btn-primary btn-sm"
onclick="showBookDetail(\'' . $row->isbn .
'\')">Detail</button></td>';
            echo '</tr>';
        }
        echo '</table>';
        echo '<br />';
        echo 'Total Rows = ' . $result->num_rows;
        $result->free();
    } else {
        echo 'Pencarian tidak ditemukan.';
    }

    $db->close();
}
?>
```

1. Pemeriksaan Parameter: Kode pertama memeriksa apakah ada parameter searchTerm yang diterima melalui metode GET (dengan isset(\$_GET['searchTerm'])). searchTerm adalah kata kunci yang akan digunakan untuk mencari buku berdasarkan judul.
2. Pencarian di Database: Jika parameter searchTerm tersedia, maka kata kunci ini digunakan untuk membuat query SQL yang akan mencari buku dengan judul yang mengandung kata kunci tersebut. Query SQL tersebut dibuat dalam variabel \$query.

3. Eksekusi Query: Query SQL kemudian dieksekusi menggunakan `$db->query($query)`. Hasil dari eksekusi query ini adalah sebuah objek hasil (result object).
4. Pengecekan Hasil Query: Kode selanjutnya memeriksa apakah query tersebut berhasil dieksekusi (dengan `if ($result)`). Jika query berhasil, maka hasil pencarian akan ditampilkan dalam bentuk tabel.
5. Menampilkan Hasil Pencarian: Hasil pencarian (buku-buku yang sesuai dengan kata kunci) ditampilkan dalam bentuk tabel. Setiap baris tabel mewakili satu buku, dan dalam setiap baris, ditampilkan ISBN buku, judul buku, dan sebuah tombol "Detail" yang akan memanggil fungsi `showBookDetail()` ketika ditekan.
6. Pembebasan Hasil: Setelah pengguna melihat hasil pencarian, objek hasil query dibebaskan dengan menggunakan `$result->free()`. Ini adalah praktik baik untuk membersihkan sumber daya setelah digunakan.
7. Pesan Pencarian Tidak Ditemukan: Jika query tidak menghasilkan hasil (yaitu tidak ada buku yang sesuai dengan kata kunci), maka pesan "Pencarian tidak ditemukan" akan ditampilkan.
8. Menutup Koneksi Database: Terakhir, koneksi database ditutup menggunakan `$db->close()`.

File `get_book_detail.php`

```
<?php
require_once('../lib/db_login.php');

if (isset($_GET['isbn'])) {
    $isbn = $_GET['isbn'];
    $query = "SELECT * FROM books WHERE isbn = '" . $isbn . "'";
    $result = $db->query($query);

    if ($result->num_rows > 0) {
        $row = $result->fetch_object();
        echo '<h3>Detail Buku</h3>';
        echo '<p><strong>ISBN:</strong> ' . $row->isbn . '</p>';
        echo '<p><strong>Judul:</strong> ' . $row->title . '</p>';
        echo '<p><strong>Penulis:</strong> ' . $row->author .
    '</p>';
        echo '<p><strong>Harga:</strong> $' . $row->price . '</p>';
    } else {
        echo 'Buku tidak ditemukan.';
    }

    $result->free();
    $db->close();
}
?>
```

Kode di atas adalah file PHP yang digunakan untuk mengambil dan menampilkan detail buku berdasarkan ISBN yang diterima melalui parameter GET dari URL. Di bawah ini adalah penjelasan detailnya:

1. `require_once('./lib/db_login.php');`: Ini adalah baris pertama dari kode yang mengimpor file `db_login.php`, yang berisi konfigurasi koneksi ke database MySQL. Dengan demikian, koneksi ke database disiapkan untuk mengambil data buku.
2. `if (isset($_GET['isbn'])) {`: Ini adalah kondisional yang memeriksa apakah parameter `isbn` telah diterima melalui URL. Parameter ini akan digunakan untuk menentukan buku mana yang akan ditampilkan detailnya.
3. `$isbn = $_GET['isbn'];`: Ini adalah baris yang mengambil nilai `isbn` dari URL dan menyimpannya dalam variabel `$isbn`. Nilai ISBN ini akan digunakan dalam query SQL untuk mencari buku yang sesuai.
4. Query SQL: `$query = "SELECT * FROM books WHERE isbn = " . $isbn . ""`;

Query SQL ini mengambil semua kolom (dengan menggunakan tanda `*`) dari tabel `books` di database yang memiliki nilai ISBN yang sesuai dengan nilai yang diterima dari URL.

5. Query dieksekusi dan hasilnya disimpan dalam variabel `$result`. Hasil query ini berisi data buku yang sesuai dengan ISBN yang diterima
6. Pemeriksaan Hasil:

```
if ($result->num_rows > 0) {  
    // ... kode untuk menampilkan detail buku ...  
} else {  
    echo 'Buku tidak ditemukan.';  
}
```

- Kode ini memeriksa apakah query menghasilkan satu atau lebih baris data. Jika hasil query mengandung data buku, maka kode di dalam blok `if` akan mengekstrak dan menampilkan detail buku tersebut.
- Jika query tidak mengembalikan hasil (artinya, buku tidak ditemukan), maka akan ditampilkan pesan "Buku tidak ditemukan."

7. Membaca dan Menampilkan Detail Buku

Jika buku ditemukan (berdasarkan ISBN), kode berikut akan menampilkan detail buku:

```
echo '<h3>Detail Buku</h3>';  
echo '<p><strong>ISBN:</strong> ' . $row->isbn . '</p>';  
echo '<p><strong>Judul:</strong> ' . $row->title . '</p>';  
echo '<p><strong>Penulis:</strong> ' . $row->author . '</p>';  
echo '<p><strong>Harga:</strong> $' . $row->price . '</p>';
```

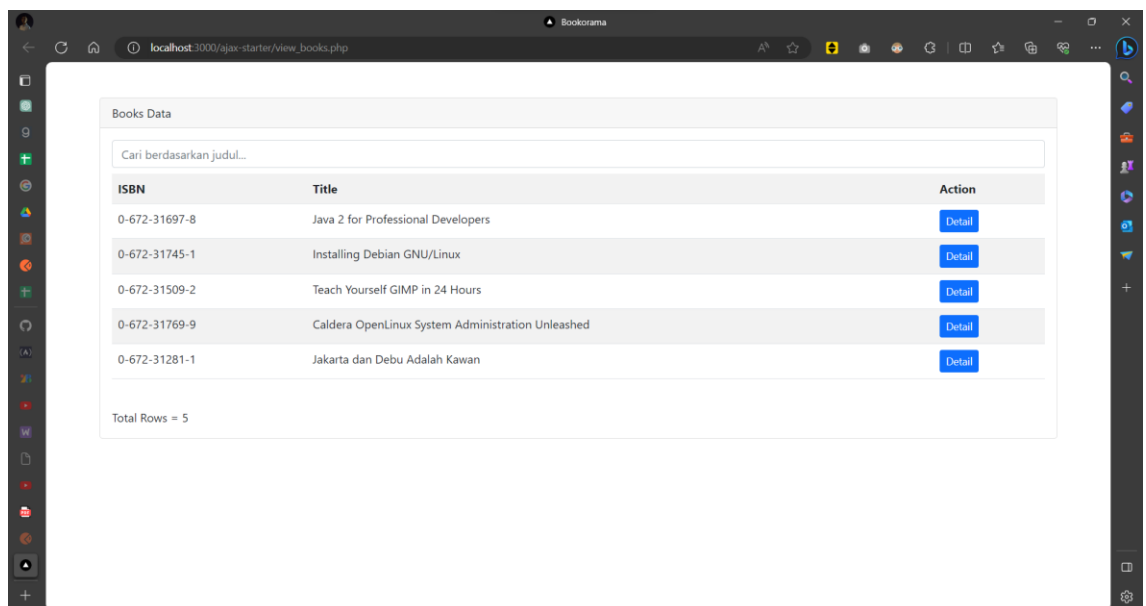
8. Membebaskan Hasil dan Menutup Koneksi:

```
$result->free();  
$db->close();
```

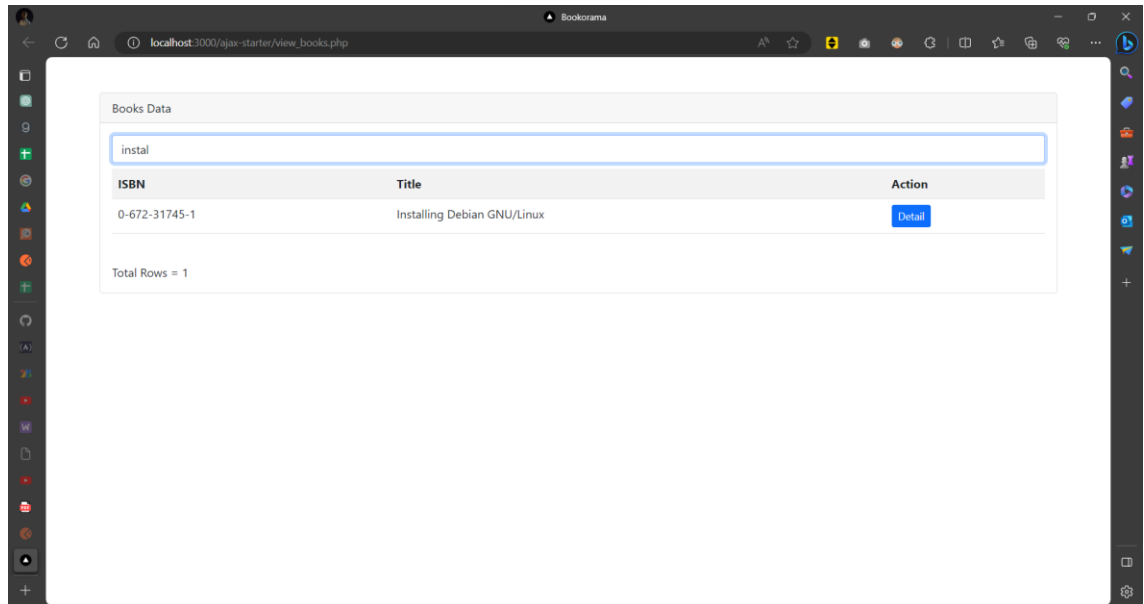
Kode ini membersihkan dan melepaskan hasil query serta menutup koneksi ke database setelah digunakan.

2. Hasil Kode

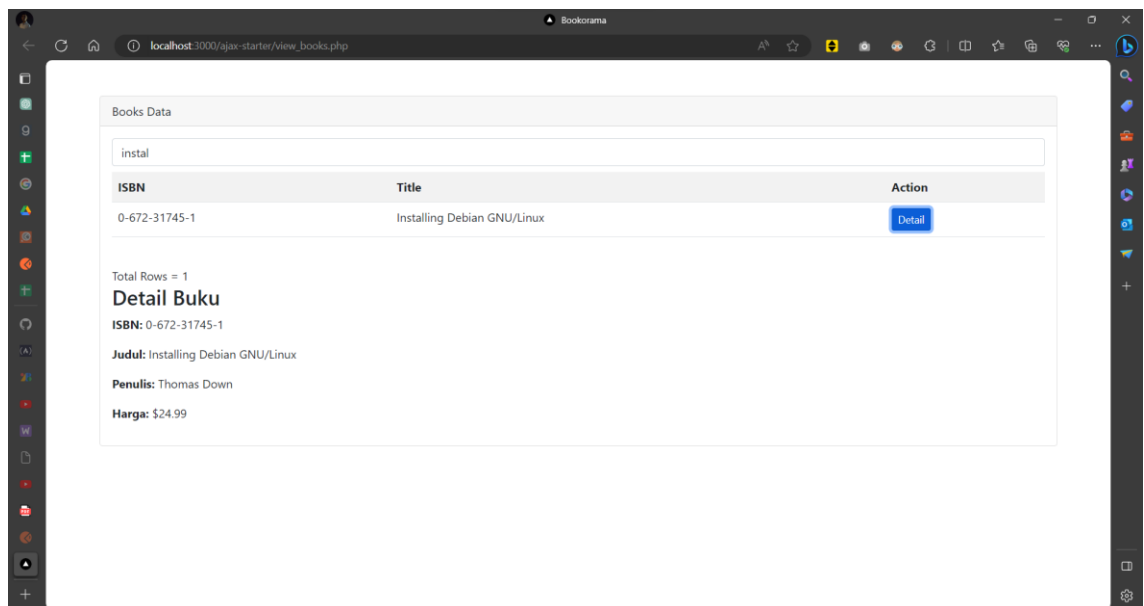
a. List Buku



b. Search Buku



c. Detail Buku



BAB IV

PENUTUP

Implementasi Ajax pada form pencarian buku dan melihat detail buku membawa sejumlah manfaat besar dalam pengalaman pengguna. Dalam hal pencarian buku, pengguna dapat memasukkan kata kunci tanpa harus me-refresh seluruh halaman, yang dapat meningkatkan responsivitas dan kecepatan pencarian. Hasil pencarian kemudian dapat diperbarui secara dinamis di halaman tanpa perlu me-reload seluruh halaman. Ini memberikan pengalaman yang lebih mulus dan efisien bagi pengguna.

Sementara itu, untuk melihat detail buku, pengguna dapat melihat informasi lebih lanjut tentang buku dengan sekali klik tanpa meninggalkan halaman saat ini. Dengan Ajax, informasi detail buku dapat dimuat dari server dan ditampilkan di halaman tanpa harus me-refresh halaman atau membuka halaman baru. Ini menghemat waktu dan usaha pengguna, serta membuat navigasi lebih efisien.

Kesimpulannya, implementasi Ajax pada form pencarian dan detail buku menghadirkan pengalaman pengguna yang lebih baik. Hal ini meningkatkan responsivitas, efisiensi, dan kenyamanan dalam mencari dan mengeksplorasi informasi buku. Teknologi ini telah membantu menciptakan aplikasi web yang lebih dinamis dan responsif, mengoptimalkan antarmuka pengguna, dan memperkaya pengalaman web secara keseluruhan.

DAFTAR PUSTAKA

McFarland, D. (2015). JavaScript & jQuery: The Missing Manual. O'Reilly Media.

Ducket, R. (2014). JavaScript and jQuery for Web Designers. Wiley.