

MODUL VII INTEGRASI AJAX

A. TUJUAN

- Memahami teknologi Ajax dan integrasinya dengan PHP.
- Mampu menghasilkan aplikasi dengan pemrosesan silent.
- Mampu memanfaatkan Ajax untuk menghasilkan aplikasi-aplikasi PHP yang responsif.

B. ALOKASI WAKTU

3 JS (3 x 50 menit)

C. PETUNJUK

- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas praktikum dengan baik, sabar, dan jujur.
- Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

D. DASAR TEORI

1. Sekilas Ajax

Meskipun bukan merupakan topik baru, namun Ajax masih sering diperbincangkan. Pada kenyataannya, Ajax memang mampu menyelesaikan sejumlah persoalan, terutama berkaitan dengan pemrosesan round-trip pada client-server.

Ajax (*Asynchronous JavaScript and XML*) merupakan suatu teknik pengembangan web yang digunakan untuk menghasilkan aplikasi web interaktif. Teknologi Ajax ini dimaksudkan untuk meningkatkan interaktifitas, kecepatan, fungsionalitas, dan kegunaan aplikasi web.

Secara garis besar, ada empat komponen pembentuk Ajax, yaitu:

- Presentasi berbasis browser
- Data (umumnya dalam format XML) yang diambil dari server.
- Pemrosesan data di balik layar dengan menggunakan objek XMLHttpRequest.
- JavaScript.

E. LATIHAN

1. Dukungan Ajax

Pada dasarnya, komponen yang diperlukan untuk mengimplementasikan Ajax hanyalah browser dengan dukungan JavaScript. Terkait hal ini, meski kebanyakan browser sudah mendukung, namun perlu sekali dilakukan identifikasi.

Untuk melakukan identifikasi, buat file dukungan_ajax.html dan isikan kode berikut:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Periksa Dukungan Ajax</title>
</head>

<script language="JavaScript" type="text/javascript">

/**
 * Menciptakan objek XMLHttpRequest
 * @return true jika berhasil
 */
function createXmlHttpRequest() {
  // Untuk menyimpan reference ke objek XMLHttpRequest
  var xmlhttp = false;

  if (window.ActiveXObject) {
    // Jika menggunakan Internet Explorer
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
  } else {
    // Untuk Mozilla dan browser lain
    xmlhttp = new XMLHttpRequest();
  }

  if (!xmlhttp) {
    alert("Gagal Menciptakan Objek XMLHttpRequest");
  } else {
    return xmlhttp;
  }
}

// Menciptakan objek XMLHttpRequest
var xmlhttp = createXmlHttpRequest();

if (xmlhttp) {
  alert("OK, Support Ajax :-");
} else {
  alert("Dukungan Ajax tidak ditemukan :-(");
}

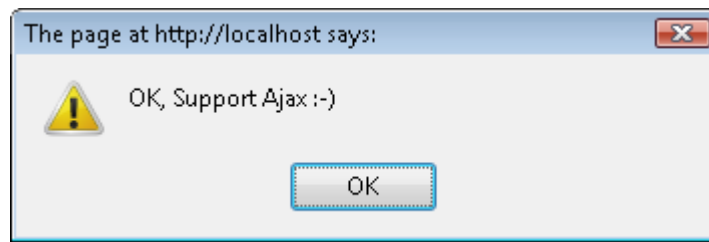
</script>

<body>

</body>

</html>
```

Apabila muncul tampilan seperti Gambar 1, berarti dukungan Ajax tersedia dan sudah bekerja.



Gambar 1. Tampilan hasil pencarian data

2. Pemrosesan Data

Agar dapat lebih memahami implementasi Ajax, bagian ini akan menjelaskan pembuatan halaman pemrosesan dengan melibatkan dan tanpa Ajax.

■ Pemrosesan Reguler

Buat kode program pemrosesan_reguler.php seperti di bawah, kemudian jalankan dan amati hasilnya setelah melakukan submisi.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Pemrosesan Reguler (Tanpa Ajax)</title>
</head>

<body>

<form action="<?php $_SERVER['PHP_SELF'];?>" method="get">
Nama
<input type="text" name="nama" /> <br />

<input type="submit" value="OK" />
</form>

<?php
if (isset($_GET['nama'])) {
  echo 'Hallo, ' . $_GET['nama'];
}
?>

</body>
</html>
```

Pada saat pemrosesan dilakukan, apakah ada indikator yang menyatakan bahwa browser sedang bekerja?

▪ Pemrosesan AJAX

Untuk kasus yang sama, sekarang kita menyelesaikan dengan pendekatan Ajax. Buat kode HTML dan simpan dengan nama **pemrosesan_ajax.html**.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Pemrosesan Menggunakan Ajax</title>
</head>

<script language="JavaScript" type="text/javascript">
var xmlHttp = false;

function createXmlHttpRequest() {
  var xmlHttp = false;

  if (window.ActiveXObject) {
    // Jika menggunakan Internet Explorer
    xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
  } else {
    // Untuk Mozilla dan browser lain
    xmlHttp = new XMLHttpRequest();
  }

  if (!xmlHttp) {
    alert("Gagal Menciptakan Objek XMLHttpRequest");
  } else {
    return xmlHttp;
  }
}

function prosesData(url, param) {
  xmlHttp = createXmlHttpRequest();
  if (xmlHttp.readyState == 4 ||
  xmlHttp.readyState == 0) {
    // Eksekusi source dari server
    xmlHttp.open("GET", url + param, true);
    xmlHttp.onreadystatechange = handleRespon;
    // Membuat request
    xmlHttp.send(null);
  } else {
    setTimeout("prosesData(url, param)", 1000);
  }
}

function handleRespon() {
  if (xmlHttp.readyState == 4) {
    if (xmlHttp.status == 200) {
      document.getElementById('hasil').
      innerHTML = xmlHttp.responseText;
    } else {
      alert("Error: " + xmlHttp.statusText);
    }
  }
}

</script>
</head>

<body>

<form name="my_form">
```

```
Nama: <input type="text" name="nama" /> <br />
<input type="button" value="OK"
  onclick="prosesData('pemrosesan_ajax.php?nama=',
    my_form.nama.value);" />
</form>

Respon Server:<br />

<!-- Penting, ini untuk menampilkan hasil -->
<div id="hasil"></div>

</body>
</html>
```

Langkah selanjutnya, buat kode PHP untuk melakukan pemrosesan silent (di balik layar) dan simpan dengan nama **pemrosesan_ajax.php** (sesuaikan dengan penetapan di HTML).

```
<?php

if (isset($_GET['nama'])) {
    echo 'Hallo, ' . $_GET['nama'];
}

?>
```

Pada saat pemrosesan dilakukan, apakah ada indikator yang menyatakan bahwa browser sedang bekerja? Jelaskan!

Sekilas, pendekatan Ajax terlihat cukup rumit dan memerlukan kode program lebih banyak. Namun dalam implementasinya, kita bisa menerapkan aspek guna ulang (reusability) sehingga lebih praktis.

3. Aplikasi Database Berbasis Ajax

Kemampuan Ajax yang paling menarik adalah akses data di database secara silent. Pendekatan ini secara nyata mampu mereduksi traffic dan meningkatkan performa aplikasi.

Sebagai contoh, di sini kita akan membuat halaman pertama berupa nim mahasiswa yang diletakkan di combobox (seleksi). Begitu dipilih, nantinya akan ditampilkan halaman detail data.

File: seleksi_ajax.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Aplikasi Database Ajax</title>
</head>

<script language="JavaScript" type="text/javascript">
var xmlHttp = false;
```

```
function createXmlHttpRequest() {
    var xmlHttp = false;
    if (window.ActiveXObject) {
        xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
    } else {
        xmlHttp = new XMLHttpRequest();
    }
    if (!xmlHttp) {
        alert("Gagal menciptakan Objek");
    }
    return xmlHttp;
}

function showData(url, id) {
    xmlHttp = createXmlHttpRequest();

    if (xmlHttp.readyState == 4 ||
        xmlHttp.readyState == 0) {
        // Membentuk url dan argumen
        var url = url + "?id=" + id;
        xmlHttp.onreadystatechange = handleRespon;
        xmlHttp.open("GET", url, true)
        xmlHttp.send(null)
    } else {
        // Jika gagal, coba setelah 1 detik
        setTimeout('showData(url, id)', 1000);
    }
}

function handleRespon() {
    if (xmlHttp.readyState == 4) {
        if (xmlHttp.status == 200) {
            document.getElementById('hasil').
                innerHTML = xmlHttp.responseText;
        } else {
            alert("Error: " + xmlHttp.statusText);
        }
    }
}
</script>
</head>

<body>

<?php
require_once './koneksi.php';

$sql = 'SELECT nim FROM mahasiswa';
$res = mysql_query($sql);
if ($res && mysql_num_rows($res)) { ?>
    <form>
        NIM:
        <select name="nim" onchange="showData('show_data.php',
this.value);">
            <?php
                while ($row = mysql_fetch_row($res)) { ?>
                    <option value="<?php echo $row[0];?>"><?php echo
                    $row[0];?></option>
                <?php
                }
            ?>
        </select>
    }
}
```

```
</form>
<?php
}
?>

<!-- Penting, ini untuk menampilkan hasil -->
<p><div id="hasil"></div></p>

</body>
</html>
```

Langkah selanjutnya, siapkan kode PHP untuk menampilkan detail data di file **show_data.php**.

```
<?php
require_once './koneksi.php';

$id = $_GET['id'];

$sql = "SELECT * FROM mahasiswa
        WHERE nim = $id";
$res = mysql_query($sql);
if ($res) {
    if (mysql_num_rows($res)) { ?>

        <table border=1 cellpadding=5>
        <tr>
            <th>#</th>
            <th width=100>NIM</th>
            <th width=150>Nama</th>
            <th width=200>Alamat</th>
        </tr>
        <?php
        $i = 1;
        while ($row = mysql_fetch_row($res)) { ?>
            <tr>
                <td>
                    <?php echo $i;?>
                </td>
                <td>
                    <?php echo $row[0];?>
                </td>
                <td>
                    <?php echo $row[1];?>
                </td>
                <td>
                    <?php echo $row[2];?>
                </td>
            </tr>
            <?php
            $i++;
        }
        ?>
        </table>
        <?php
        } else {
            echo 'Data Tidak Ditemukan';
        }
    }
    ?>
```

4. Indikator Proses

Keberadaan indikator loading terkadang sangat diperlukan untuk mengindikasikan apakah permintaan mendapat tanggapan ataukah tidak. Bagaimanapun juga, sifat dasar aplikasi-aplikasi Ajax yang berjalan tanpa melibatkan indikator browser mengakibatkan kebutuhan akan indikator ini menjadi semakin penting.

Sebagai contoh, di sini kita akan menggunakan kode di pembahasan sebelumnya dan menambahkan indikator proses. Adapun langkahnya adalah menambahkan baris tebal berikut pada fungsi `handleRespon()`.

File: seleksi_loading.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Aplikasi Database Ajax</title>
</head>

<script language="JavaScript" type="text/javascript">
var xmlHttp = false;

function createXmlHttpRequest() {
  var xmlHttp = false;
  if (window.ActiveXObject) {
    xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
  } else {
    xmlHttp = new XMLHttpRequest();
  }
  if (!xmlHttp) {
    alert("Gagal menciptakan Objek");
  }
  return xmlHttp;
}

function showData(url, id) {
  xmlHttp = createXmlHttpRequest();

  if (xmlHttp.readyState == 4 ||
  xmlHttp.readyState == 0) {
    // Membentuk url dan argumen
    var url = url + "?id=" + id;
    xmlHttp.onreadystatechange = handleRespon;
    xmlHttp.open("GET", url, true)
    xmlHttp.send(null)
  } else {
    // Jika gagal, coba setelah 1 detik
    setTimeout('showData(url, id)', 1000);
  }
}

function handleRespon() {
  // Proses jika objek komplit (kode = 4)
  if (xmlHttp.readyState == 4) {
    if (xmlHttp.status == 200) {
      // Set teks
      document.getElementById('hasil').
      innerHTML = xmlHttp.responseText;
    } else {
      alert("Error: " + xmlHttp.statusText);
    }
  }
}
```



```
    }

    // Jika belum komplit
    // tampilkan animasi indikator
    } else {
        document.getElementById('hasil').innerHTML =
            '';
    }

}
</script>
</head>

<body>

<?php
require_once './koneksi.php';

$sql = 'SELECT nim FROM mahasiswa';
$res = mysql_query($sql);
if ($res && mysql_num_rows($res)) { ?>
    <form>
        NIM:
        <select name="nim" onchange="showData('show_data.php',
this.value);">
            <?php
                while ($row = mysql_fetch_row($res)) { ?>
                    <option value="<?php echo $row[0];?>"><?php echo
                    $row[0];?></option>
                <?php
                }
            ?>
        </select>
    </form>
<?php
}

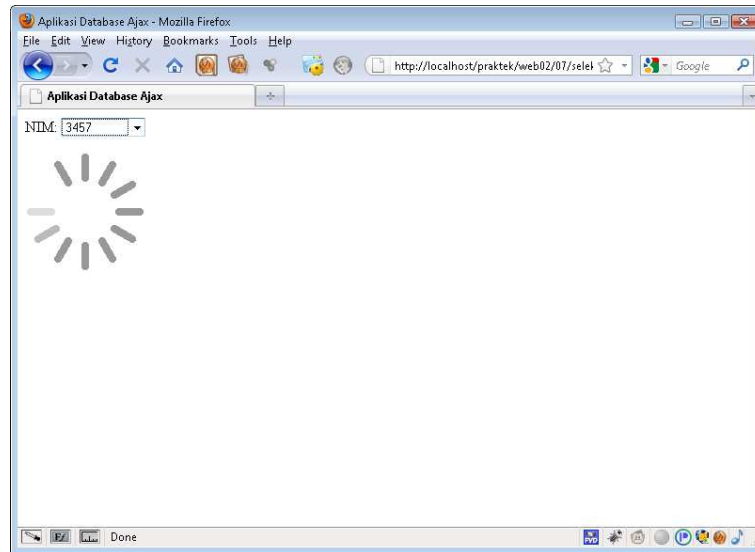
?>

<!-- Penting, ini untuk menampilkan hasil -->
<p><div id="hasil"></div></p>

</body>
</html>
```

Sebelum Anda menguji hasilnya, pastikan bahwa Anda sudah menyiapkan file animasi, misalnya dalam contoh ini adalah **loading.gif**.

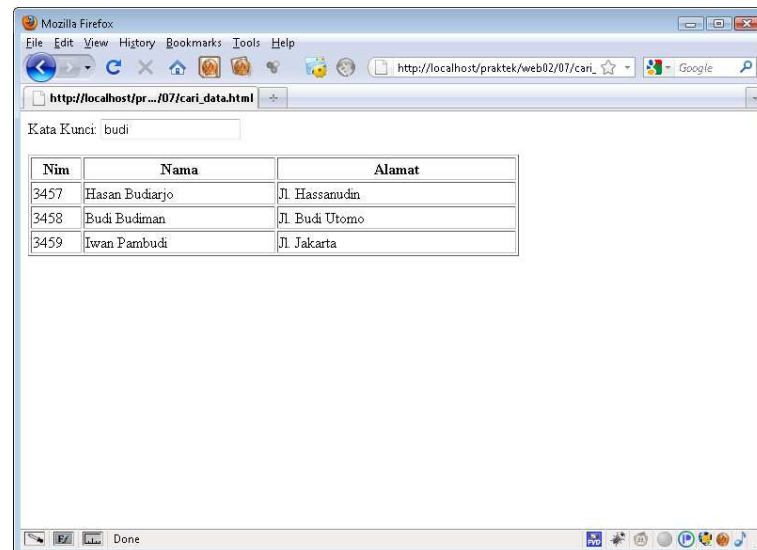
Contoh tampilan indikator progres diperlihatkan seperti pada Gambar 2.



Gambar 2. Tampilan indikator proses Ajax

F. TUGAS PRAKTIKUM

1. Buat aplikasi pencarian data dengan mengimplementasikan Ajax. Contoh tampilan hasil diperlihatkan seperti Gambar 3.



Gambar 3. Pencarian dan penyaringan menggunakan Ajax