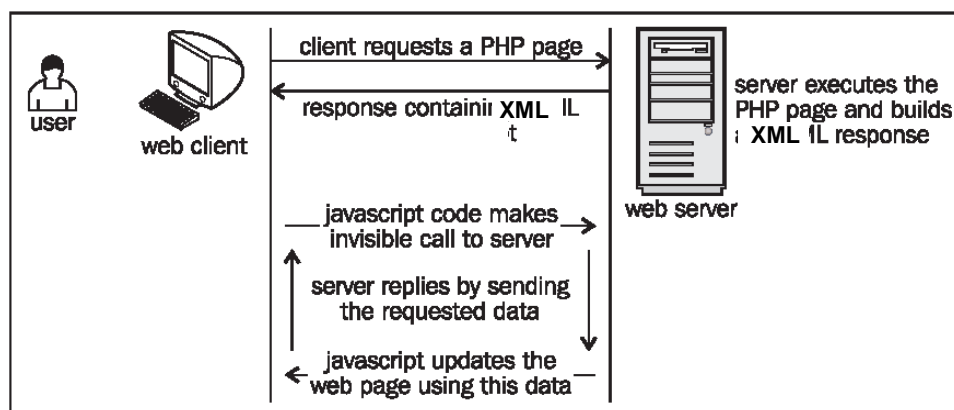


Pendahuluan

AJAX memiliki kepanjangan *Asynchronous Javascript And XML* merupakan suatu teknik baru dalam dunia web. Sejak telah dikembangkan sekitar 2-3 tahun yang lalu, AJAX mulai dilirik oleh para web desainer dan web programmer. Dengan adanya AJAX, akses data ke server yang dikirim melalui client via web dapat lebih cepat daripada mekanisme biasa. Hal ini dikarenakan AJAX tidak perlu melakukan proses loading page (refresh page) atau pindah ke page yang lain.

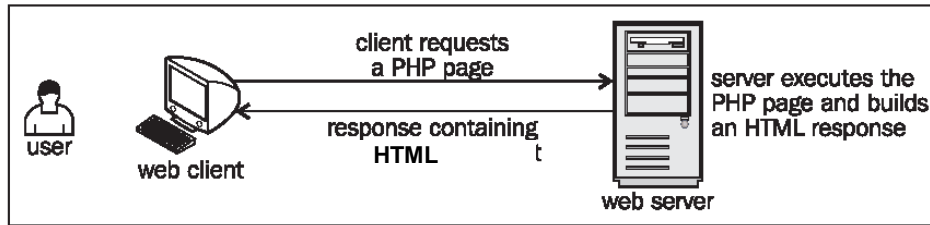
AJAX dapat diintegrasikan dengan server side programming seperti PHP, ASP, JSP dll.

Perhatikan mekanisme proses AJAX menggunakan PHP berikut ini



Pada mekanisme di atas, proses berawal dari web client. Web client merequest sebuah halaman PHP ke server melalui Javascript. Selanjutnya server akan merespon dan menjalankan script PHP. Script PHP akan menghasilkan respon dalam bentuk XML dan data XML tersebut akan dikirim kembali ke web client untuk diolah oleh Javascript. Hasil olahan javascript tersebut akhirnya akan ditampilkan di web client sebagai output tanpa harus merefresh halaman web.

Bandingkan dengan mekanisme biasa dalam menjalankan PHP seperti yang ditunjukkan pada gambar berikut



Pada mekanisme biasa, respon yang dihasilkan oleh server berupa HTML yang mengakibatkan halaman web akan merefresh tampilan. Dilihat dari sisi efisiensi, hal ini akan membutuhkan lebih banyak waktu dan byte data untuk melakukan load daripada XML.

Dengan demikian AJAX ini merupakan teknologi baru yang merupakan hasil perkawinan sempurna antara client side programming (Javascript) dan server side programming.

AJAX tidak hanya dapat diimplementasikan menggunakan PHP, namun juga server side programming yang lain, seperti ASP, JSP dsb.

Sedangkan kekurangan AJAX dibandingkan mekanisme biasa adalah memiliki tingkat kesukaran pemrograman yang lebih tinggi, karena programmer harus menguasai 4 hal yaitu HTML-(X)HTML, Javascript, XML dan juga server side programming.

But it's fine... ☺ Ketrampilan dapat diasah dengan banyak latihan...

Teknologi AJAX telah banyak digunakan oleh situs-situs terkenal. Beberapa diantaranya adalah Google, Yahoo, serta Flickr (www.flickr.com). Oleh Google, AJAX diimplementasikan dalam GMAIL (www.gmail.com), Google Suggest (www.google.com/webhp?complete=1), Google Maps (maps.google.com), serta Google Adsense (www.google.com/adsense).

Bagi Anda yang memiliki email account email di Gmail, tentu sudah pernah merasakan teknologi ini, khususnya ketika membuka email di Inbox. Ketika Anda membuka inbox dan membaca email, serasa halaman web tidak merefresh.

AJAX sudah support di banyak browser, seperti IE (mulai versi 4.0), Firefox (semua versi), Opera (mulai versi 7.6), Konqueror, Netscape (mulai versi 7.1), serta Apple Safari (mulai versi 1.2).

HTTP Request dan Dasar-dasar Respon

Untuk memahami konsep AJAX, pertama kali akan dibahas mengenai bagaimana situs web merequest dan menerima respon dari web server.

Saat ini standar browser untuk memperoleh informasi dari web server adalah berbasis pada metode HTTP (HyperText Transfer Protocol). Hal ini berarti bahwa browser menggunakan metode tersebut untuk mengirim request dan menerima respon dari web server.

HTTP bekerja seperti halnya email, yaitu ketika mengirim request akan terdapat header yang digunakan oleh web server. Oleh web server, header ini digunakan untuk mengetahui tugas apa yang harus dilayaninya dan bagaimana handle request dari HTTP.

Meskipun beberapa header bersifat optional, namun terdapat pula yang mutlak harus ada, misalnya host header. Header ini sangat penting karena akan digunakan web server untuk mengetahui tugas apa yang harus dilayani.

Ketika request sudah diterima oleh server, selanjutnya server akan menentukan jenis respon yang akan dikembalikan kepada HTTP. Berikut ini beberapa jenis kode respon HTTP

KODE	DESKRIPSI
200 OK	Respon yang dikembalikan apabila dokumen atau file ditemukan ketika direquest (sukses)
304 Not Modified	Respon yang dikembalikan apabila browser mengindikasikan bahwa copyan dokumen yang ada local cache (browser) dan server cache sama atau tidak berubah.
401 Unauthorized	Respon yang dikembalikan apabila request membutuhkan otorisasi untuk mengakses dokumen
403 Forbidden	Respon yang dikembalikan apabila pe-request tidak memiliki hak akses untuk dokumen yang diinginkan
404 Not Found	Respon yang dikembalikan apabila dokumen yang akan diakses tidak ditemukan.
500 Internal Server Error	Kode respon ini dikembalikan apabila terdapat kesalahan teknis dalam server.
503 Service Unavailable	Respon yang dikembalikan apabila jenis layanan yang direquest tidak dikenal oleh server

Sedangkan untuk metode request HTTP yang umum digunakan antara lain POST dan GET. Metode GET sering digunakan untuk merequest sumber tertentu ke web server. Biasanya GET digunakan untuk mempassing value untuk diproses ke web server dan selanjutnya value itu diassign ke sebuah variabel untuk diproses oleh server. Sedangkan POST pada prinsipnya sama dengan GET, namun metode ini ideal untuk value yang diperoleh dari isian form dan halaman web.

copyright © 2007 by rosihanari

XMLHttpRequest

Selain metode request HTTP yang telah dijelaskan pada bab sebelumnya, masih ada metode lain yaitu XMLHttpRequest. XMLHttpRequest merupakan metode request HTTP yang beroperasi di belakang layar (tanpa harus merefresh halaman web). Metode ini berwujud suatu obyek. Sehingga dalam hal ini AJAX merupakan konsep yang mendeskripsikan interaksi antara XMLHttpRequest (client side object) dengan server script seperti PHP, ASP dll.

Sayangnya, XMLHttpRequest ini memiliki bentuk obyek yang berbeda dalam beberapa browser sehingga dalam penggunaannya perlu penanganan khusus yang disesuaikan dengan jenis browsernya. Sebagai contoh, dalam IE obyek tersebut dinyatakan sebagai ActiveX Control. Pada Firefox dan Safari, XMLHttpRequest dinyatakan sebagai Javascript Object.

Pada aplikasi AJAX, pendeklarasian penggunaan XMLHttpRequest ini dilakukan dalam sebuah Javascript. Berikut ini adalah script yang digunakan untuk pendeklarasian penggunaan XMLHttpRequest

```
function membuatObyekXmlHttpRequest()  
{  
    var xmlhttp;  
    // pengecekan apabila menggunakan IE  
    if(window.ActiveXObject)  
    {  
        try  
        {  
            // pendeklarasian penggunaan XMLHttpRequest di IE  
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");  
        }  
        catch (e)  
        {  
            xmlhttp = false;  
        }  
    }  
    // jika browser menggunakan Firefox atau yang lain  
    else  
    {  
        try  
        {  
            xmlhttp = new XMLHttpRequest();  
        }  
        catch (e)  
        {  
            xmlhttp = false;  
        }  
    }  
}
```

Script di atas harus ada di dalam halaman web yang menggunakan AJAX.

Metode-metode pada XMLHttpRequest

Dalam obyek XMLHttpRequest terdapat beberapa metode yang digunakan untuk proses request. Berikut ini beberapa diantaranya:

Metode	Deskripsi
<code>abort()</code>	Untuk membatalkan request
<code>open()</code>	Untuk melakukan koneksi ke server secara asynchronous
<code>send()</code>	Untuk mengirim request

Keterangan:

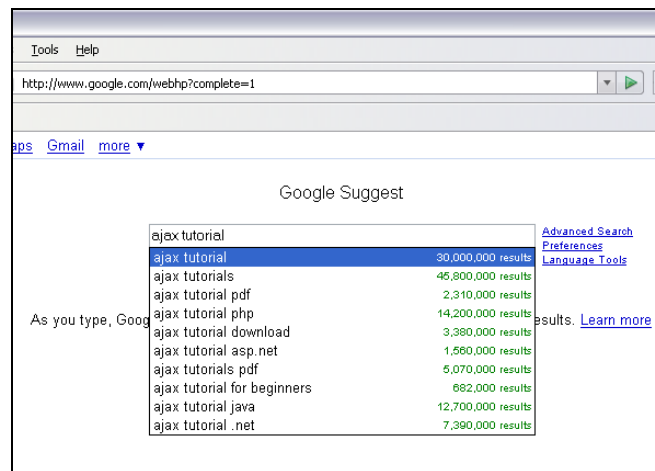
Untuk metode `open()`, secara lengkap memiliki paramater-parameter yaitu

```
open("method", "URL", "async")
```

dengan parameter `method` adalah metode request HTTP yang digunakan yaitu POST atau GET. Parameter `URL` adalah nama URL yang akan dikoneksikan. Sedangkan parameter `async` dapat digantikan dengan `true` apabila koneksi dilakukan secara asynchronous atau `false` apabila tidak secara asynchronous.

Ketika Anda menggunakan request secara asynchronous, proses request tidak akan berhenti meskipun response dari server belum diperoleh. Sedangkan apabila menggunakan request secara synchronous, maka proses request akan terhenti selama response belum diterima. Hal inilah yang menjadikan perbedaan dari kedua jenis request. Kedua jenis request tidak akan menjadi masalah apabila koneksi antara client dengan server sangat cepat. Akan tetapi untuk koneksi yang lambat, penggunaan request synchronous akan sangat mengganggu penggunaan aplikasi.

Sebagai contoh, misalnya pada penggunaan aplikasi Google Suggest yang dapat Anda kunjungi di www.google.com/webhp?complete=1.



Google Suggest merupakan aplikasi web yang mampu memberikan saran untuk keyword pencarian situs web. Pada aplikasi tersebut terdapat suatu text box (untuk menulis keyword) yang apabila setiap kali Anda menulis sebuah karakter akan muncul drop down menu berisi list keyword saran.

Apabila tipe request yang digunakan Google Suggest tersebut adalah secara synchronous, maka setelah Anda menuliskan sebuah karakter, Anda tidak bisa menuliskan karakter berikutnya yang lain (text box tidak bisa ditambahi karakter, alias freezing...), sampai response server diterima (muncul list keyword). Tentu saja hal ini akan sangat mengganggu user apabila koneksinya lambat karena text box akan sering mengalami freezing.

Sedangkan pada kenyataannya aplikasi tersebut tidak demikian. Meskipun belum ada response atau list keyword saran belum muncul, Anda tetap bisa menambahkan karakter pada text box.

Properties XMLHttpRequest

Karena berbentuk object, maka XMLHttpRequest memiliki beberapa properties. Berikut ini beberapa diantaranya yang sering digunakan untuk aplikasi AJAX.

Properties	Deskripsi
Readystate	Menyatakan keadaan proses yang dilakukan obyek saat itu. Properties ini bernilai 0 s.d 4 (0: belum ada inisialisasi, 1: loading, 2: loaded, 3: interaktif, 4: selesai)
status	Menyatakan respon yang diterima dari server. Properties ini akan mengembalikan kode respon (dalam bentuk numerik) seperti yang telah dijelaskan pada bab dasar-dasar respon. Misalnya 404 apabila file tidak

	ditemukan.
<code>responseXML</code>	Mengembalikan respon dalam format XML
<code>statustext</code>	Properti ini mirip dengan properti status, namun yang dikembalikan adalah keterangan statusnya (bukan nilai numeriknya). Misalkan responnya 404, akan mengembalikan <code>statustext: Not Found</code>

copyright © 2007 by rosihanari

Dasar-dasar XML

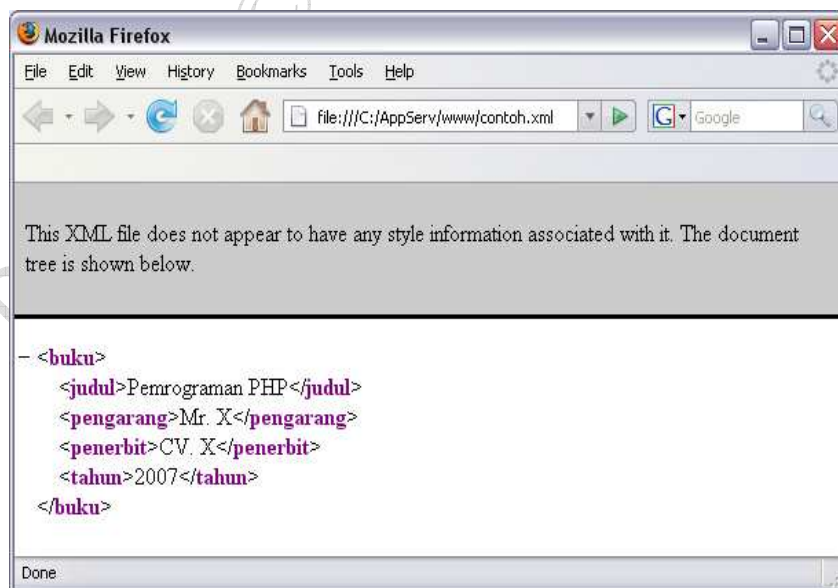
XML kependekan dari *eXtensible Markup Language*, dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998.

Seperti halnya HTML, XML juga menggunakan *elemen* yang ditandai dengan tag pembuka (diawali dengan '<' dan diakhiri dengan '>'), tag penutup (diawali dengan '</' dan diakhiri '>') dan atribut elemen (parameter yang dinyatakan dalam tag pembuka misal <form name="isidata">). Hanya bedanya, HTML mendefinisikan dari awal tag dan atribut yang dipakai didalamnya, serta nama tagnya harus sesuai standard yang ada. Sedangkan pada XML kita bisa menggunakan tag dan menamainya sesuai kehendak kita. XML sering digunakan untuk mendeskripsikan suatu obyek beserta sifatnya.

Untuk lebih jelasnya lihat contoh dibawah yang digunakan untuk mendeskripsikan obyek sebuah buku.

```
<buku>
  <judul>Pemrograman PHP</judul>
  <pengarang>Mr. X</pengarang>
  <penerbit>CV. X</penerbit>
  <tahun>2007</tahun>
</buku>
```

Apabila tag di atas disimpan dalam file berekstensi .xml dan dibuka oleh browser, maka yang tampilannya tidak menarik dan hanya berupa teks saja sebagaimana yang tertulis di atas.



Dokumen XML tersebut akan diolah oleh aplikasi lain untuk menampilkan data obyek buku tersebut (tanpa disertai tag-tagnya). Dengan demikian dapat dikatakan bahwa XML merupakan suatu cara untuk menyajikan data mentah suatu obyek ke dalam web.

XML untuk saat ini bukan merupakan pengganti HTML. Masing-masing dikembangkan untuk tujuan yang berbeda. Kalau HTML digunakan untuk menampilkan informasi dan berfokus pada bagaimana informasi terlihat, XML mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri.

Sintaks XML

Pada prinsipnya sintaks XML sama dengan HTML atau XHTML, yaitu adanya tag pembuka dan penutup. Nama tag sebaiknya disesuaikan dengan field obyek tersebut (perhatikan contoh sebelumnya), meskipun Anda dapat secara bebas menamainya. Sehingga secara umum tag XML adalah `<field>...</field>`, dengan field adalah nama field yang sesuai.

Pada contoh berikut ini

```
<buku>
  <judul>Pemrograman PHP</judul>
  <pengarang>Mr. X</pengarang>
  <penerbit>CV. X</penerbit>
  <tahun>2007</tahun>
</buku>
```

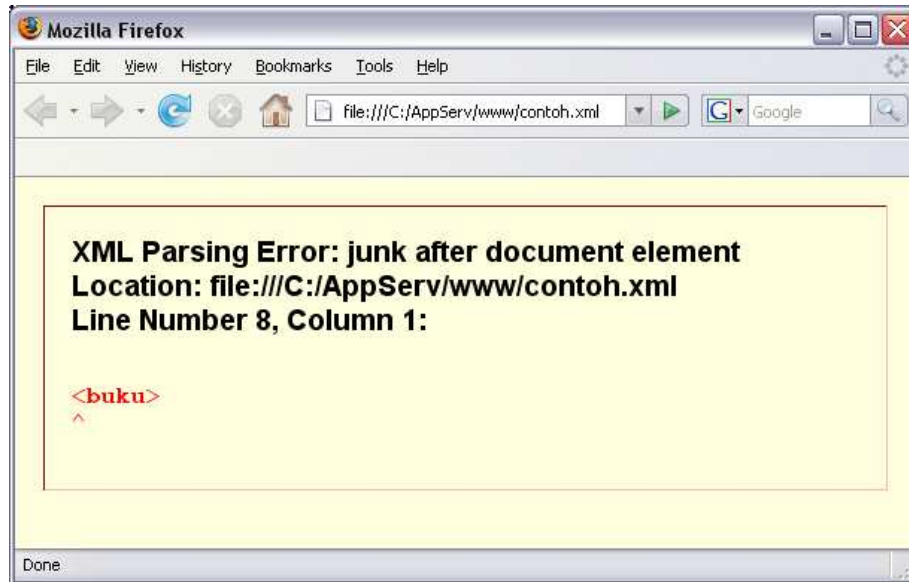
Dalam XML, harus terdapat root tag. Pada contoh di atas, root tag nya adalah buku. Root tag adalah tag pembuka dan penutup yang tunggal.

Dokumen XML hanya mendiskripsikan satu buah obyek buku saja. Lantas... bagaimana kalau jumlah bukunya lebih dari satu (misalnya 2 buku)? Apakah berbentuk seperti ini?

```
<buku>
  <judul>Pemrograman PHP</judul>
  <pengarang>Mr. X</pengarang>
  <penerbit>CV. X</penerbit>
  <tahun>2007</tahun>
</buku>

<buku>
  <judul>Pemrograman ASP</judul>
  <pengarang>Mr. Y</pengarang>
  <penerbit>CV. Y</penerbit>
  <tahun>2007</tahun>
</buku>
```

Ternyata apabila dokumen tersebut dibuka di browser hasilnya adalah sbb:

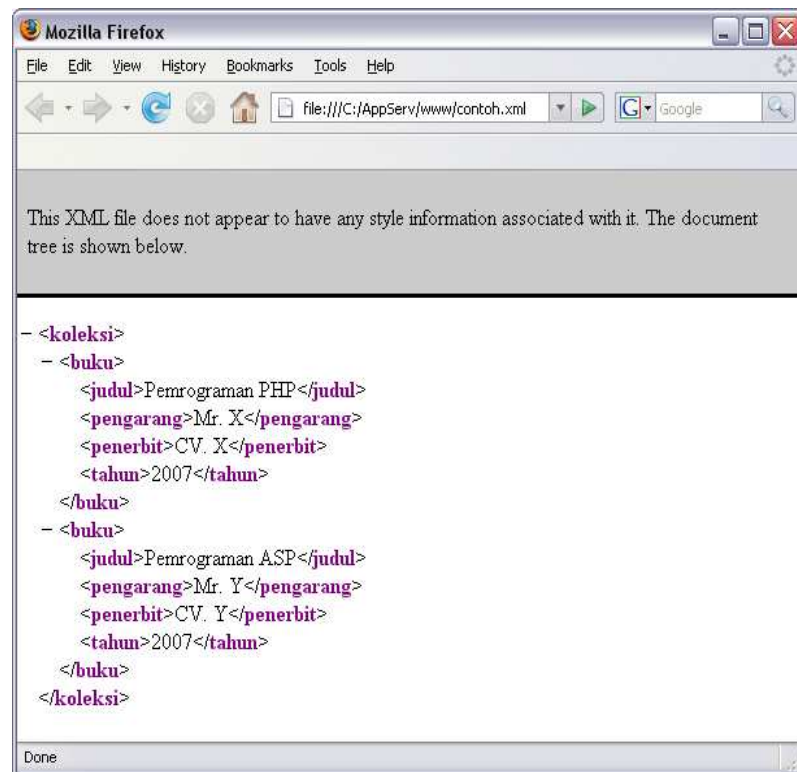


Dari tampilan di atas, jelaslah bahwa terdapat error pada dokumen. Hal ini disebabkan karena tidak adanya root tag. Dalam hal ini tag buku bukan root tag lagi karena berjumlah 2 buah (tidak tunggal). Oleh karena itu dapat dibuat root tag lagi misalnya dengan nama koleksi .

```
<koleksi>
  <buku>
    <judul>Pemrograman PHP</judul>
    <pengarang>Mr. X</pengarang>
    <penerbit>CV. X</penerbit>
    <tahun>2007</tahun>
  </buku>

  <buku>
    <judul>Pemrograman ASP</judul>
    <pengarang>Mr. Y</pengarang>
    <penerbit>CV. Y</penerbit>
    <tahun>2007</tahun>
  </buku>
</koleksi>
```

Adapun hasil tampilan dokumen XML di atas dalam browser adalah sbb:



XML Generator Dengan PHP

Pada bagian ini akan dibahas bagaimana membuat script PHP untuk mengenerate dokumen XML. Data dalam dokumen XML yang akan digenerate diambil dari database MySQL. Untuk lebih memudahkan pemahaman, ada baiknya langsung diterapkan pada contoh.

Misalkan Anda memiliki database buku dengan struktur tabel dan data sebagai berikut

Nama Database : koleksiBuku
Nama Tabel : buku

id	judul	pengarang	penerbit	tahunTerbit
1	Pemrograman PHP	Mr. X	PT. X	2006
2	Pemrograman ASP	Mr. Y	PT. Y	2006
3	Jaringan Komputer	Mr. Z	PT. Z	2007

Berikut ini adalah script PHP untuk mengenerate dokumen XML data di atas.

```
<?
// koneksi ke MySQL
mysql_connect("host","dbUser","dbPass");
mysql_select_db("koleksiBuku");

// membuat header untuk menghasilkan dokumen XML (mime)
header('Content-Type: text/xml');

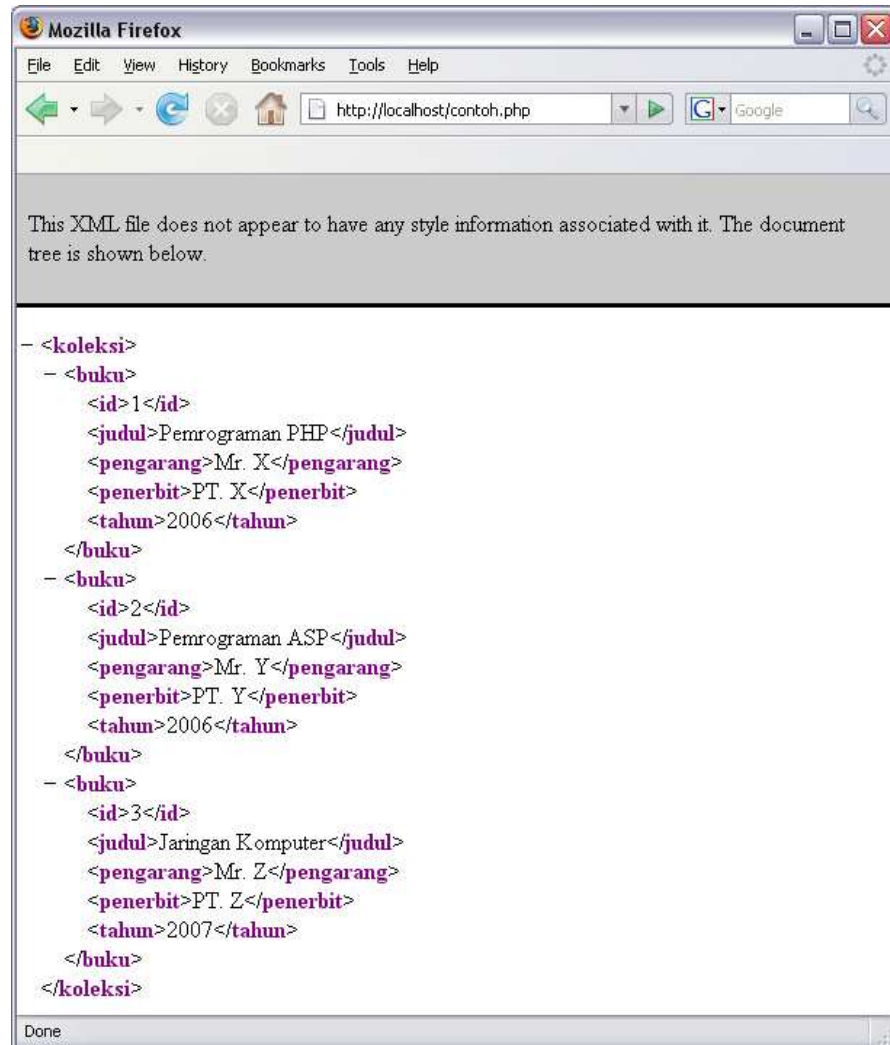
// membuat root tag pembuka
echo "<koleksi>";

// menjalankan query untuk memanggil data dalam tabel
$query = "SELECT * FROM buku";
$hasil = mysql_query($query);

// data hasil query ditampilkan ke dalam dokumen XML
while ($data = mysql_fetch_array($hasil))
{
    echo "<buku>";
    echo "<id>".$data['id']. "</id>";
    echo "<judul>".$data['judul']. "</judul>";
    echo "<pengarang>".$data['pengarang']. "</pengarang>";
    echo "<penerbit>".$data['penerbit']. "</penerbit>";
    echo "<tahun>".$data['tahunTerbit']. "</tahun>";
    echo "</buku>";
}

// membuat root tag penutup
echo "</koleksi>";
?>
```

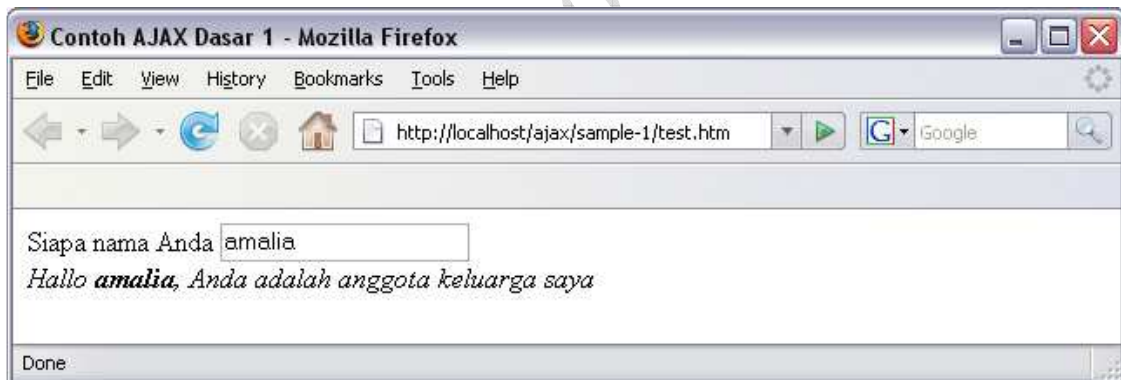
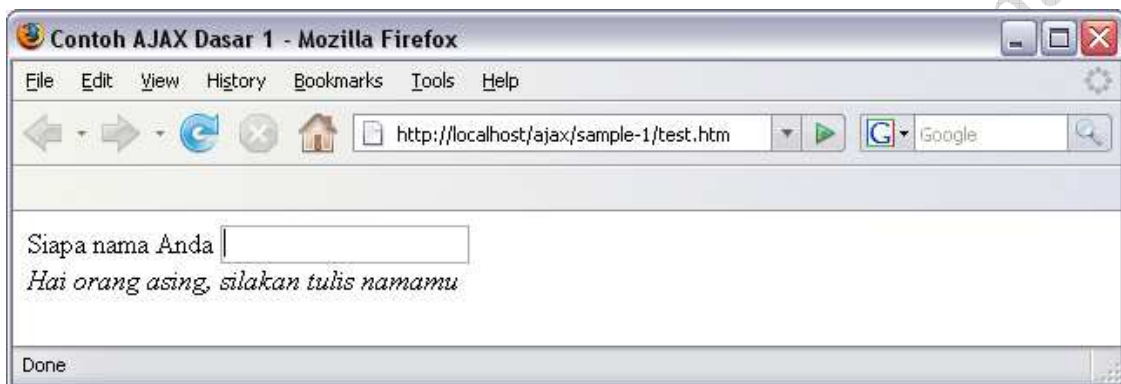
Adapun hasil ketika script di atas dijalankan di browser adalah

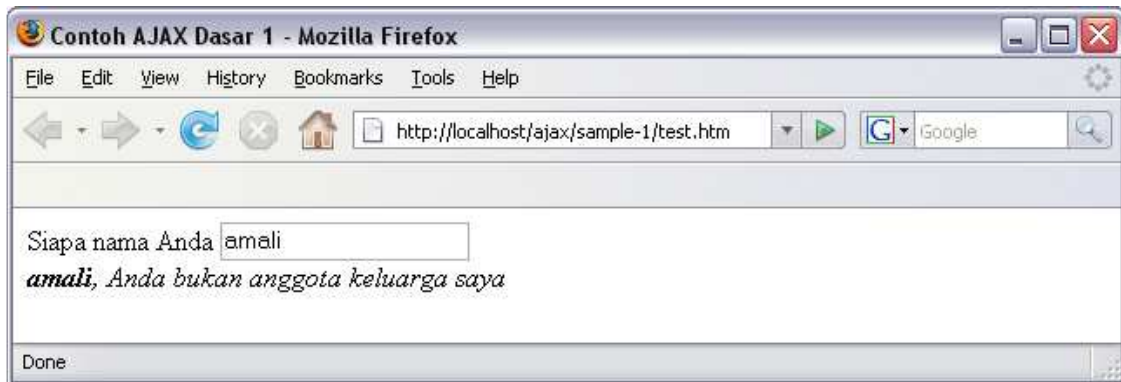


Contoh AJAX Dasar 1

Setelah Anda mengenal dasar-dasar tentang konsep AJAX, XML dan Javascript, berikut ini akan kita bahas mengenai contoh aplikasi AJAX dasar. Pada contoh ini kita akan membuat aplikasi yang akan mendeteksi apakah user termasuk anggota keluarga atau tidak.

Misalkan ada keluarga beranggotakan 4 orang, yaitu bernama ROSIHAN, AMALIA, FAZA dan NADA. Apabila ada seorang user menuliskan namanya dalam suatu text box, maka akan muncul konfirmasi termasuk anggota keluarga atau tidak. Berikut ini screen shotnya





Perhatikan... bahwa aplikasi tersebut tidak memiliki button untuk submit data. Aplikasi ini memiliki efek seperti halnya Google Suggest. Setiap kita menuliskan sebuah karakter maka akan langsung muncul response nya.

Berikut ini adalah script untuk membuat formnya.

Contoh1.htm

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Contoh AJAX Dasar 1</title>
<script type="text/javascript" src="cek.js"></script>
</head>

<!-- bagian untuk menampilkan form -->

<body onload='process() '>
<form name="form1">
Siapa nama Anda <input type="text" id="namaAnda" />
</form>

<!-- bagian untuk menampilkan respon -->

<div id="respon" />

</body>
</html>
```

Selanjutnya kita membuat javascript yang bertugas membuat obyek XMLHttpRequest dan merequest ke server PHP.

Cek.js

```
var xmlHttp = createXmlHttpRequestObject();

// membuat obyek XMLHttpRequest

function createXmlHttpRequestObject()
{
    var xmlHttp;

    // cek untuk browser IE

    if(window.ActiveXObject)
    {
        try
        {
            xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch (e)
        {
            xmlHttp = false;
        }
    }
    // cek untuk browser Firefox atau yang lain
    else
    {
        try
        {
            xmlHttp = new XMLHttpRequest();
        }
        catch (e)
        {
            xmlHttp = false;
        }
    }

    // muncul pesan apabila obyek XMLHttpRequest gagal dibuat

    if (!xmlHttp) alert("Obyek XMLHttpRequest gagal dibuat");
    else
    return xmlHttp;
}

// melakukan request secara asynchronous dengan XMLHttpRequest ke
// server

function process()
{
    // akan diproses hanya bila obyek XMLHttpRequest tidak sibuk

    if (xmlHttp.readyState == 4 || xmlHttp.readyState == 0)
    {

```

```
// mengambil nama dari text box (form)

nama =
    encodeURIComponent(document.getElementById("namaAnda").value);

// merequest file cek.php di server secara asynchronous

xmlHttp.open("GET", "cek.php?nama=" + nama, true);

// mendefinisikan metode yang dilakukan apabila memperoleh
// response server

xmlHttp.onreadystatechange = handleServerResponse;

// membuat request ke server

xmlHttp.send(null);

}
else
{
    // Jika server sibuk, request akan dilakukan lagi setelah
    // satu detik

    setTimeout('process()', 1000);
}
}

// fungsi untuk metode penanganan response dari server

function handleServerResponse()
{
    // jika proses request telah selesai dan menerima respon

    if (xmlHttp.readyState == 4)
    {
        // jika request ke server sukses

        if (xmlHttp.status == 200)
        {
            // mengambil dokumen XML yang diterima dari server

            xmlResponse = xmlHttp.responseXML;

            // memperoleh elemen dokumen (root elemen) dari xml

            xmlDocumentElement = xmlResponse.documentElement;

            // membaca data elemen

            hasil = xmlDocumentElement.firstChild.data;
        }
    }
}
```

```
// akan mengupdate tampilan halaman web pada elemen bernama
// respon

document.getElementById("respon").innerHTML = '<i>' + hasil +
    '</i>';

// request akan dilakukan lagi setelah
// satu detik (automatic request)

setTimeout('process()', 1000);
}
else
{
    // akan muncul pesan apabila terjadi masalah dalam mengakses
    // server (selain respon 200)

    alert("Terjadi masalah dalam mengakses server " +
        xmlhttp.statusText);
}
}
```

Sedangkan untuk script cek.php nya adalah sbb:

Cek.php

```
<?php

header('Content-Type: text/xml');

$nama = $_GET['nama'];

// membuat root tag elemen
echo '<response>';

// daftar anggota keluarga

$myFamily = array('ROSIHAN', 'AMALIA', 'FAZA', 'NADA');

// jika nama berada dalam daftar anggota keluarga

if (in_array(strtoupper($nama), $myFamily))
    echo 'Hallo <strong>'. htmlentities($nama) .
        '</strong>; Anda adalah anggota keluarga saya';

// jika nama masih kosong

else if (trim($nama) == '')
    echo 'Hai orang asing, silakan tulis namamu';
```

```
// jika nama tidak ada dalam daftar anggota keluarga

else
    echo '<strong>'. htmlentities($nama) .
        '</strong>', Anda bukan anggota keluarga saya';

// menutup root tag elemen

echo '</response>';

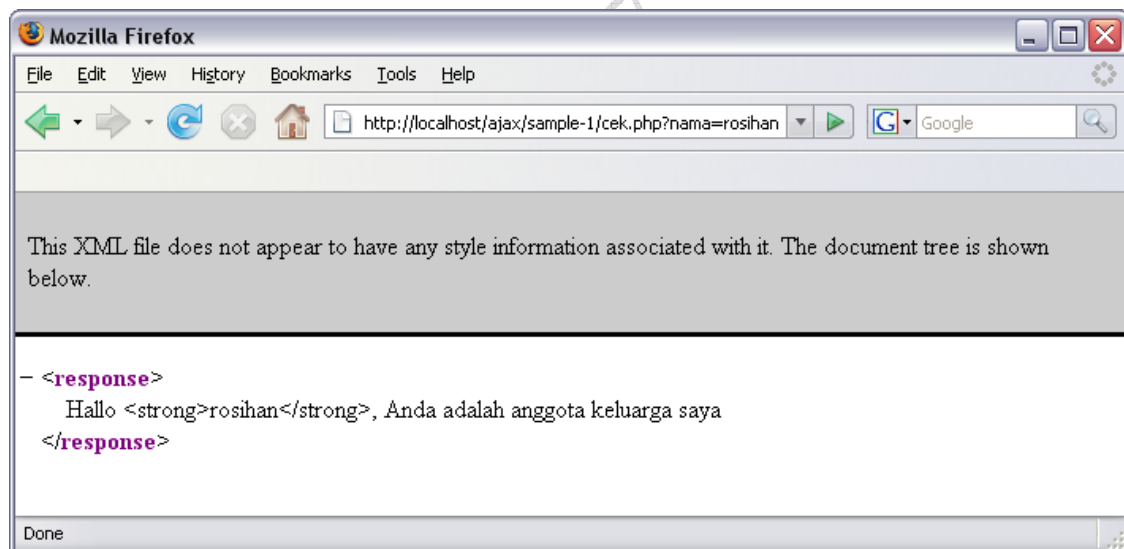
?>
```

Catatan:

Nama root tag tidak harus `response`, namun bisa yang lain (sesuka Anda).

Apabila script di atas dijalankan dalam browser dengan mengetikkan URL berikut ini
<http://.../cek.php?nama=rosihan>

maka tampilannya adalah



Catatan:

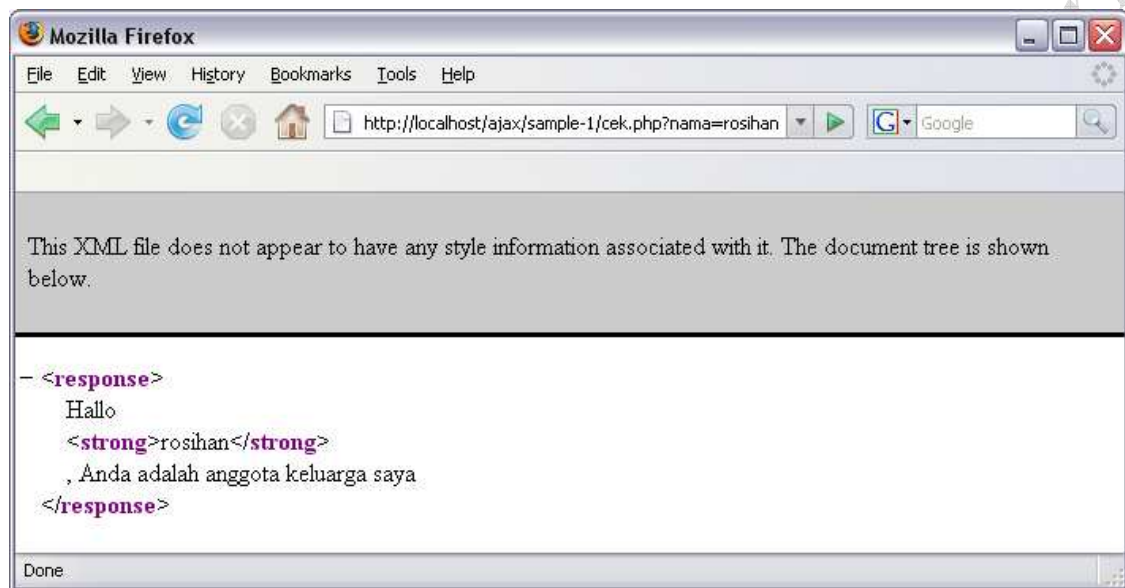
Untuk membuat tag bold (bisa dengan `` atau ``), tanda `<` dan `>` harus dituliskan sebagai `<` dan `>`;

```
echo 'Hallo <strong>'. htmlentities($nama) .
    '</strong>', Anda adalah anggota keluarga saya';
```

Apabila Anda menuliskannya dalam bentuk berikut

```
echo 'Hallo <strong>'. htmlentities($nama) .  
      '</strong>, Anda adalah anggota keluarga saya';
```

Maka akan proses ekstraksi dokumen xml oleh XMLHttpRequest akan ada kesalahan karena `` `` akan dianggap sebagai tag elemen xml.



Pada topik sebelumnya telah dijelaskan bahwa AJAX menggunakan request secara asynchronous. Dalam contoh ini apabila Anda menginginkan request secara synchronous, maka Anda ubah baris perintah berikut dalam fungsi javascript `proses()`.

```
xmlHttp.open("GET", "cek.php?nama=" + nama, true);  
xmlHttp.onreadystatechange = handleServerResponse;  
xmlHttp.send(null);
```

menjadi baris berikut

```
xmlHttp.open("GET", "cek.php?nama=" + nama, false);  
xmlHttp.send(null);  
handleServerResponse();
```

Anda akan melihat efek perbedaan antara kedua metode request apabila koneksi ke servernya lambat, seperti yang telah dijelaskan pada bab sebelumnya. Untuk koneksi yang cepat (seperti localhost), perbedaannya tidak begitu kentara.

Contoh AJAX Dasar 2

Pada contoh kedua ini, kita akan membuat kalkulator sederhana yang mengolah dua bilangan sebagai input dengan operasi penjumlahan, pengurangan, perkalian dan pembagian.

Berikut ini screen shot nya.



Tidak seperti contoh pertama, pada contoh kedua ini output atau respon akan dihasilkan begitu kita klik tombol hitung.

Berikut ini adalah script untuk formnya:

kalkulator.htm

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Contoh AJAX Dasar 2</title>
<script type="text/javascript" src="kalkulator.js"></script>
</head>

<body>
<h1>Kalkulator Sederhana dengan AJAX</h1>

<form name="form1">

<!-- input bilangan pertama -->
```

```
<input type="text" id="bil1" />

<!-- pilihan jenis operasi -->

<select id="operasi">
<option value="penjumlahan">dijumlahkan dengan</option>
<option value="pengurangan">dikurangkan dengan</option>
<option value="perkalian">dikalikan dengan</option>
<option value="pembagian">dibagi dengan</option>
</select>

<!-- input bilangan kedua -->

<input type="text" id="bil2" /><br />

<!-- tombol hitung -->

<input type="button" value="Hitung" onclick="hitung()" />
</form>

<!-- bagian untuk menampilkan output -->

<p><div id="output" /></p>

</body>
</html>
```

Tombol hitung begitu diklik akan menjalankan fungsi `hitung()` yang ada di script `kalkulator.js`

Berikut ini adalah script `kalkulator.js`

kalkulator.js

```
var xmlhttp = createXmlHttpRequestObject();

function createXmlHttpRequestObject()
{
    var xmlhttp;
    if(window.ActiveXObject)
    {
        try
        {
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch (e)
        {
            xmlhttp = false;
        }
    }
}
```

```
else
{
    try
    {
        xmlHttp = new XMLHttpRequest();
    }
    catch (e)
    {
        xmlHttp = false;
    }
}

if (!xmlHttp) alert("Obyek XMLHttpRequest gagal dibuat");
else
return xmlHttp;
}

function hitung()
{
    if (xmlHttp.readyState == 4 || xmlHttp.readyState == 0)
    {
        // mengambil data input dari elemen bernama bill dan dikonversi
        // ke float / riil supaya dapat dioperasikan secara aritmatika

        bill =
parseFloat(encodeURIComponent(document.getElementById("bill").value));

        // mengambil data input dari elemen bernama bil2 dan dikonversi
        // ke float / riil supaya dapat dioperasikan secara aritmatika

        bil2 =
parseFloat(encodeURIComponent(document.getElementById("bil2").value));

        // mengambil data input dari elemen bernama operasi

        operasi =
encodeURIComponent(document.getElementById("operasi").value);

        // proses perhitungan operasi dilakukan di script kalkulator.php

xmlHttp.open("GET", "kalkulator.php?bill=" + bill + "&bil2=" + bil2 +
"&op=" + operasi, true);

xmlHttp.onreadystatechange = handleServerResponse;
xmlHttp.send(null);
    }
    else
    setTimeout('hitung()', 1000);
}

function handleServerResponse()
```



```
{
    if (xmlHttp.readyState == 4)
    {
        if (xmlHttp.status == 200)
        {
            xmlResponse = xmlHttp.responseXML;
            xmlDocumentElement = xmlResponse.documentElement;
            hasil = xmlDocumentElement.firstChild.data;
            document.getElementById("output").innerHTML = hasil;

            // setTimeout('process()', 1000);
        }

        else
        {
            alert("Ada masalah dalam koneksi ke server: " +
                xmlHttp.statusText);
        }
    }
}
```

Perhatikan statement perintah yang dicetak merah di atas. Pada contoh pertama sebelumnya, statement tersebut digunakan. Namun untuk contoh kedua ini tidak digunakan (dibuat komentar saja ya...). Hal ini dikarenakan pada kasus ini proses request dilakukan tidak secara otomatis setiap 1 detik seperti contoh sebelumnya, namun setelah tombol hitung diklik.

Sedangkan berikut ini code untuk kalkulator.php

kalkulator.php

```
<?php
header('Content-Type: text/xml');

$bil1 = $_GET['bil1'];
$bil2 = $_GET['bil2'];
$op    = $_GET['op'];

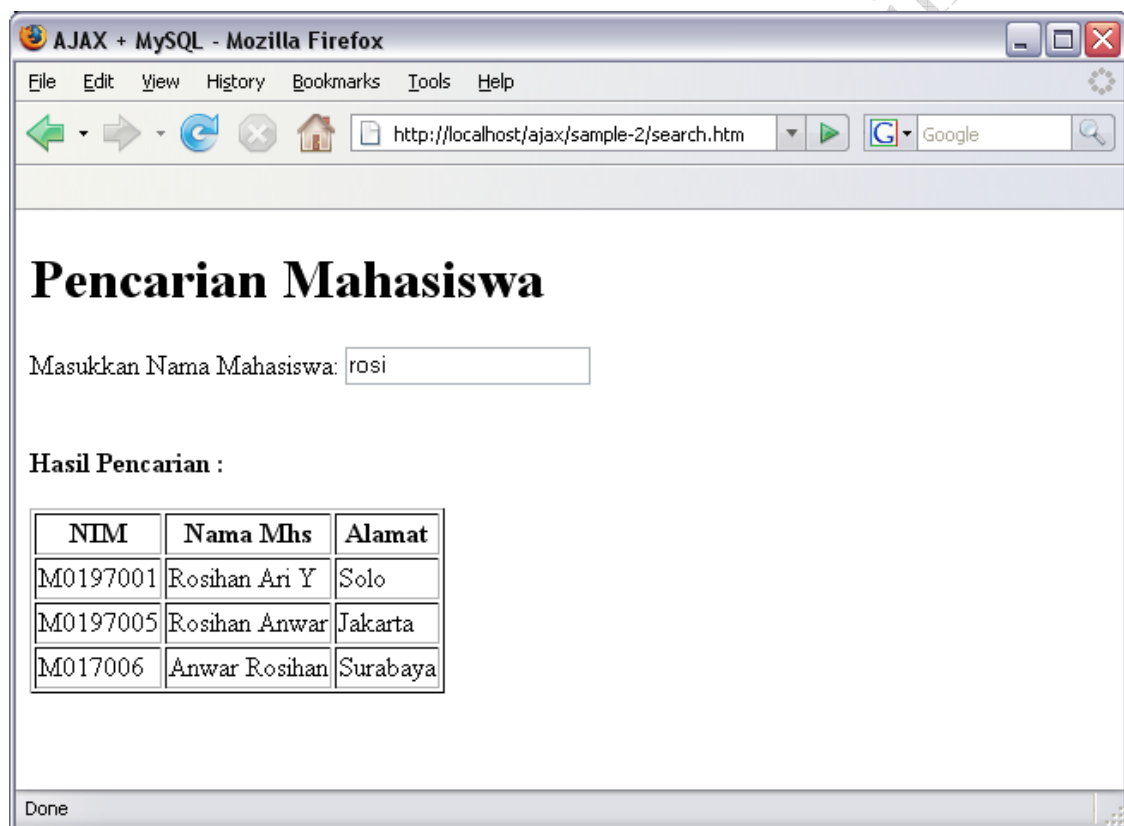
echo '<output>';
if ($op == "penjumlahan") $hasil = $bil1 + $bil2;
else if ($op == "pengurangan") $hasil = $bil1 - $bil2;
else if ($op == "perkalian") $hasil = $bil1 * $bil2;
else if ($op == "pembagian")
{
    if ($bil2 == 0) $hasil = "(ERROR) Divide by zero";
    else $hasil = $bil1 / $bil2;
}
echo "Hasil perhitungannya adalah : ". $hasil;
echo '</output>';
?>
```

Contoh AJAX Database 1

Setelah Anda membuat contoh aplikasi AJAX dasar, berikut ini akan membuat contoh aplikasi AJAX selanjutnya yang akan kita hubungkan dengan database MySQL. Pada contoh ini, aplikasi yang akan kita buat adalah tentang pencarian data mahasiswa.

User diminta memasukkan keyword pencarian data mahasiswa. Tanpa menggunakan tombol, hasil pencarian secara otomatis akan muncul (seperti efek Google Suggest). Adapun pencariannya didasarkan pada nama mahasiswa.

Berikut ini adalah screen shotnya:



Untuk keperluan tersebut terlebih dahulu kita buat database, tabel dan recordnya.

Nama database: dbMhs

```
CREATE TABLE mhs (  
  NIM varchar(9) NOT NULL,  
  NAMAMHS varchar(20) NOT NULL,
```

```
    ALAMAT varchar(20) NOT NULL,  
    PRIMARY KEY (NIM)  
);  
  
#-----  
# Records for table mhs  
#-----  
  
insert into mhs values  
( 'M0197001', 'Rosihan Ari Y', 'Solo'),  
( 'M0197002', 'Dwi Amalia Fitriani', 'Kudus'),  
( 'M0197003', 'Faza Fauzan', 'Solo'),  
( 'M0197004', 'Nada Hasanah', 'Solo'),  
( 'M0197005', 'Rosihan Anwar', 'Jakarta'),  
( 'M0197006', 'Anwar Rosihan', 'Surabaya');
```

Sedangkan untuk formnya adalah sbb:

search.htm

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<title>AJAX + MySQL I</title>  
<script type="text/javascript" src="search.js"></script>  
</head>  
  
<body onload='process() '>  
  
<h1>Pencarian Mahasiswa</h1>  
  
<form name="form1">  
Masukkan Nama Mahasiswa: <input type="text" id="namaMhs" />  
</form>  
  
<p><strong>Hasil Pencarian :</strong></p>  
  
<div id="hasil" />  
  
</body>  
</html>
```

search.js

```
var xmlHttp = createXmlHttpRequestObject();  
  
function createXmlHttpRequestObject()  
{  
    var xmlHttp;  
    if(window.ActiveXObject)
```

```
{
    try
    {
        xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    catch (e)
    {
        xmlHttp = false;
    }
}
else
{
    try
    {
        xmlHttp = new XMLHttpRequest();
    }
    catch (e)
    {
        xmlHttp = false;
    }
}

if (!xmlHttp) alert("Obyek XMLHttpRequest tidak dapat dibuat");
else
return xmlHttp;
}

function process()
{
    if (xmlHttp.readyState == 4 || xmlHttp.readyState == 0)
    {
        nama =
            encodeURIComponent(document.getElementById("namaMhs").value);
        xmlHttp.open("GET", "search.php?namaMhs=" + nama, true);
        xmlHttp.onreadystatechange = handleServerResponse;
        xmlHttp.send(null);
    }
    else
    setTimeout('process()', 1000);
}

function handleServerResponse()
{
    if (xmlHttp.readyState == 4)
    {
        if (xmlHttp.status == 200)
        {
            var xmlResponse = xmlHttp.responseXML;
            xmlRoot = xmlResponse.documentElement;

            nimArray = xmlRoot.getElementsByTagName("nim");
            namaMhsArray = xmlRoot.getElementsByTagName("namamhs");
            alamatArray = xmlRoot.getElementsByTagName("alamat");
        }
    }
}
```

```
        if (nimArray.length == 0)
        {
            html = "Data tidak ditemukan";
        }
        else
        {
            // membentuk tabel untuk menampilkan hasil pencarian

            html = "<table border='1'><tr><th>NIM</th><th>Nama Mhs</th><th>Alamat</th></tr>";

            for (var i=0; i<nimArray.length; i++)
            {
                html += "<tr><td>" + nimArray.item(i).firstChild.data +
                    "</td><td>" +
                        namaMhsArray.item(i).firstChild.data +
                            "</td><td>" +
                                alamatArray.item(i).firstChild.data +
                                    "</td></tr>";
            }
            html = html + "</table>";
        }

        document.getElementById("hasil").innerHTML = html;
        setTimeout('process()', 1000);
    }
    else
    {
        alert("Ada masalah dalam mengakses server: " +
            xmlhttp.statusText);
    }
}
}
```

Yang terakhir, berikut adalah script search.php untuk melakukan proses pencarian.

search.php

```
<?php
header('Content-Type: text/xml');

echo '<hasil>';

$namaMhs = $_GET['namaMhs'];

mysql_connect("localhost","root","root");
mysql_select_db("dbMhs");

$query = "SELECT * FROM mhs WHERE namamhs LIKE '%$namaMhs%'";
$hasil = mysql_query($query);
```

```
while ($data = mysql_fetch_array($hasil))
{
    echo "<mhs>";
    echo "<nim>".$data['NIM']. "</nim>";
    echo "<namamhs>".$data['NAMAMHS']. "</namamhs>";
    echo "<alamat>".$data['ALAMAT']. "</alamat>";
    echo "</mhs>";
}

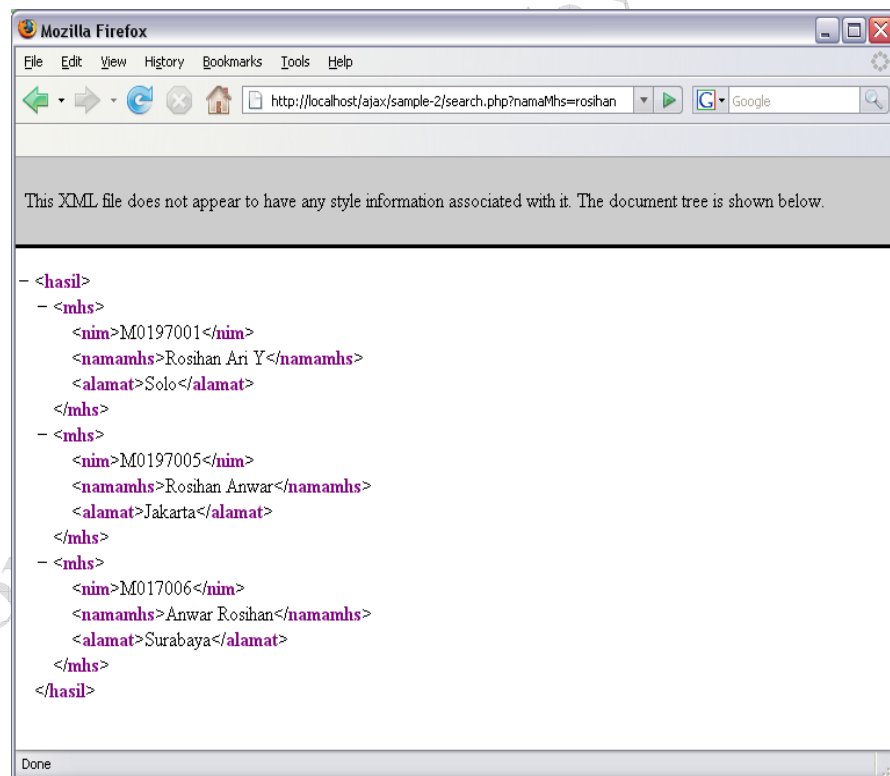
echo '</hasil>';
?>
```

Tips:

Untuk mengecek dokumen XML sudah sukses dibuat atau belum dengan script di atas, silakan buka script tersebut di browser dengan menuliskan URL semisal:

<http://.../search.php?namaMhs=rosihan>

Apabila script benar, maka akan muncul tampilan berikut ini

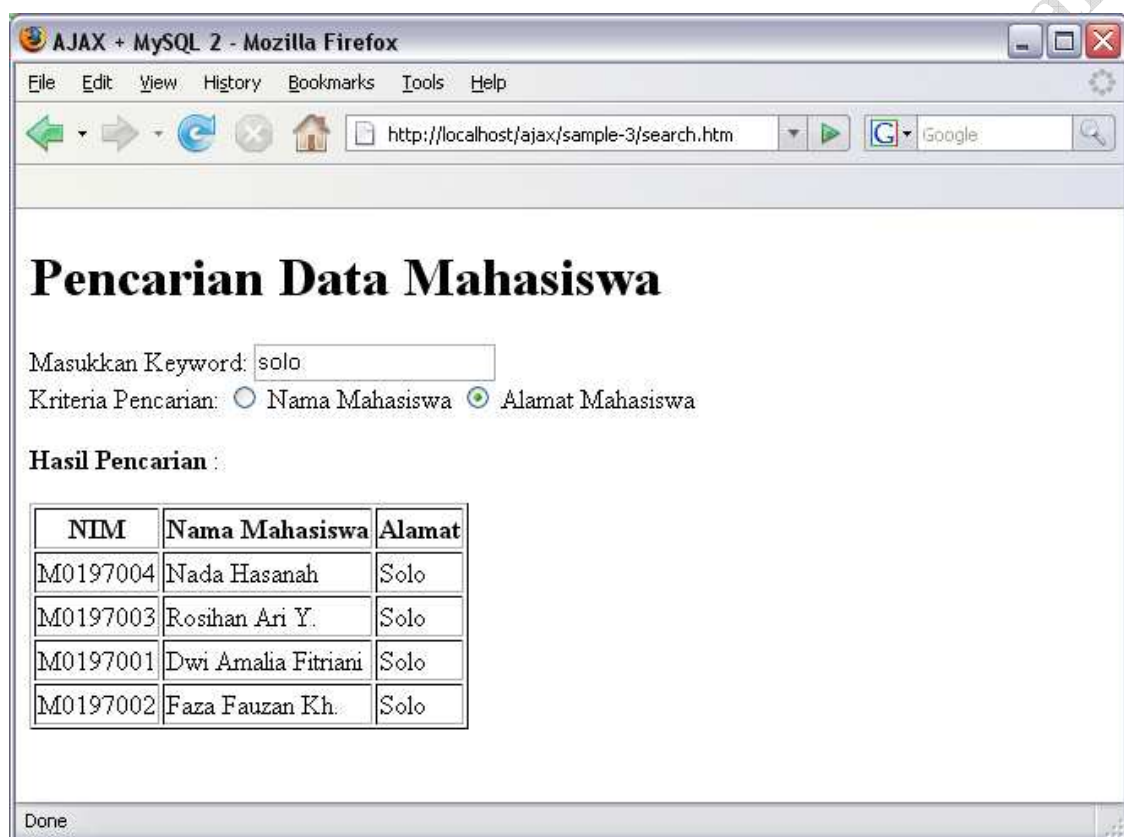


Data yang muncul adalah nama mahasiswa yang mengandung kata 'rosihan' ☺

Contoh AJAX Database 2

Masih melanjutkan contoh sebelumnya tentang pencarian data mahasiswa, pada contoh ini kriteria pencarian bersifat customizable oleh user. User dapat menentukan sendiri kriteria pencariannya, berdasarkan nim, nama atau alamat. Setelah menentukan kriteria pencarian, user diminta menuliskan keywordnya. Aplikasi ini juga tanpa menggunakan tombol.

Berikut ini adalah screen shotnya:



Perhatikan... untuk memilih kriteria pada aplikasi ini digunakan komponen radiobutton.

Untuk aplikasi ini, database, tabel dan datanya sama dengan contoh sebelumnya yaitu

Nama database: dbMhs

```
CREATE TABLE mhs (  
  NIM varchar(9) NOT NULL,  
  NAMAMHS varchar(20) NOT NULL,  
  ALAMAT varchar(20) NOT NULL,  
  PRIMARY KEY (NIM)
```

```
);

#-----
# Records for table mhs
#-----

insert into mhs values
('M0197001', 'Rosihan Ari Y', 'Solo'),
('M0197002', 'Dwi Amalia Fitriani', 'Kudus'),
('M0197003', 'Faza Fauzan', 'Solo'),
('M0197004', 'Nada Hasanah', 'Solo'),
('M0197005', 'Rosihan Anwar', 'Jakarta'),
('M0197006', 'Anwar Rosihan', 'Surabaya');
```

Berikut ini adalah code untuk membuat formnya, javascript dan script PHP nya:

search.htm

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>AJAX + MySQL 2</title>
<script type="text/javascript" src="search.js"></script>
</head>

<body onload="process()">
<h1>Pencarian Data Mahasiswa</h1>
<form name="form1">
Masukkan Keyword: <input type="text" id="keyword" /><br />

Kriteria Pencarian:
<input type="radio" name="kriteria" onclick="process()" value="nama" />
Nama Mahasiswa
<input type="radio" name="kriteria" onclick="process()" value="alamat" />
Alamat Mahasiswa
</form>

<p><strong>Hasil Pencarian</strong> :</p>

<div id="hasil" />

</body>
</html>
```

search.js

```
var xmlhttp = createXmlHttpRequestObject();

function createXmlHttpRequestObject()
```



```
{
    var xmlHttp;
    if(window.ActiveXObject)
    {
        try
        {
            xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch (e)
        {
            xmlHttp = false;
        }
    }
    else
    {
        try
        {
            xmlHttp = new XMLHttpRequest();
        }
        catch (e)
        {
            xmlHttp = false;
        }
    }

    if (!xmlHttp) alert("Obyek XMLHttpRequest gagal dibuat");
    else
    return xmlHttp;
}

function process()
{
    if (xmlHttp.readyState == 4 || xmlHttp.readyState == 0)
    {
        keyword =
            encodeURIComponent(document.getElementById("keyword").value);

        // menentukan kriteria yang dipilih dari radiobutton

        if (document.form1.kriteria[0].checked == true)
            kriteria = "nama";
        else if (document.form1.kriteria[1].checked == true)
            kriteria = "alamat";

        xmlHttp.open("GET", "search.php?keyword=" + keyword +
            "&kriteria=" + kriteria, true);

        xmlHttp.onreadystatechange = handleServerResponse;
        xmlHttp.send(null);
    }
    else
        setTimeout('process()', 1000);
}
```

```
function handleServerResponse()
{
    if (xmlHttp.readyState == 4)
    {
        if (xmlHttp.status == 200)
        {
            var xmlResponse = xmlHttp.responseXML;
            xmlRoot = xmlResponse.documentElement;

            nimArray = xmlRoot.getElementsByTagName("nim");
            namaMhsArray = xmlRoot.getElementsByTagName("namamhs");
            alamatArray = xmlRoot.getElementsByTagName("alamat");

            if (nimArray.length == 0)
            {
                html = "Data tidak ditemukan";
            }
            else
            {
                html = "<table border='1'><tr><th>NIM</th><th>Nama<br>Mahasiswa</th><th>Alamat</th></tr>";

                for (var i=0; i<nimArray.length; i++)
                {
                    html += "<tr><td>" + nimArray.item(i).firstChild.data
                        + "</td><td>" + namaMhsArray.item(i).firstChild.data
                        + "</td><td>" + alamatArray.item(i).firstChild.data
                        + "</td></tr>";
                }
                html = html + "</table>";
            }

            document.getElementById("hasil").innerHTML = html;
            setTimeout('process()', 1000);
        }
        else
        {
            alert("Ada masalah dalam koneksi ke server: " +
                xmlHttp.statusText);
        }
    }
}
```

search.php

```
<?php

header('Content-Type: text/xml');
echo '<response>';

$keyword = $_GET['keyword'];
```

```
$kriteria = $_GET['kriteria'];

mysql_connect("localhost","root","root");
mysql_select_db("dbMhs");

if ($kriteria == "nama")
    $query = "SELECT * FROM mhs WHERE namamhs LIKE '%$keyword%'";
else if ($kriteria == "alamat")
    $query = "SELECT * FROM mhs WHERE alamat LIKE '%$keyword%'";

$hasil = mysql_query($query);

while ($data = mysql_fetch_array($hasil))
{
    echo "<mhs>";
    echo "<nim>".$data['NIM']. "</nim>";
    echo "<namamhs>".$data['NAMAMHS']. "</namamhs>";
    echo "<alamat>".$data['ALAMAT']. "</alamat>";
    echo "</mhs>";
}

echo '</response>';
?>
```

Contoh AJAX Database 3

Pada contoh kali ini, kita akan mencoba membuat aplikasi form yang terkait dengan proses seleksi item.

Perhatikan tampilan berikut ini



Dalam aplikasi tersebut, user diminta memilih propinsi yang ada dalam combo box. Setelah user memilih salah satu item propinsi (misal Jawa Barat), selanjutnya secara otomatis combo box kabupaten akan menampilkan daftar kabupaten yang ada di propinsi Jawa Barat. Daftar kabupaten yang muncul akan diambil dari database.

Untuk keperluan ini, terlebih dahulu kita buat database, tabel dan datanya.

Nama database : dbPropKab

```
CREATE TABLE prop_kab (  
    propinsi varchar(20) NOT NULL,  
    kabupaten varchar(20) NOT NULL  
);
```

```
#-----  
# Records for table prop_kab  
#-----
```

```
insert into prop_kab values  
( 'Jawa Tengah', 'Boyolali' ),  
( 'Jawa Tengah', 'Klaten' ),
```

```
('Jawa Tengah', 'Semarang'),
('Jawa Timur', 'Malang'),
('Jawa Timur', 'Madiun'),
('Jawa Barat', 'Bandung'),
('Jawa Barat', 'Ciamis'),
('Jawa Barat', 'Sumedang');
```

Berikut ini code untuk form, javascript dan PHP nya

select.htm

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>AJAX + MySQL 3</title>
<script type="text/javascript" src="select.js"></script>
</head>

<body>
<h1>Aplikasi Form</h1>

<form name="form1">
<table>
  <tr>
    <td>Pilih Propinsi</td>
    <td>
      <select id="propinsi" name="prop" onchange="select()">
        <option value="Jawa Tengah">Jawa Tengah</option>
        <option value="Jawa Timur">Jawa Timur</option>
        <option value="Jawa Barat">Jawa Barat</option>
      </select>
    </td>
  </tr>
  <tr>
    <td>Pilih Kabupaten</td>
    <td>
      <select name="kab" id="kabupaten" />
    </td>
  </tr>
</table>
</form>

</body>
</html>
```

select.js

```
var xmlHttp = createXmlHttpRequestObject();
```

```
function createXmlHttpRequestObject()
{
    var xmlHttp;
    if(window.ActiveXObject)
    {
        try
        {
            xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch (e)
        {
            xmlHttp = false;
        }
    }
    else
    {
        try
        {
            xmlHttp = new XMLHttpRequest();
        }
        catch (e)
        {
            xmlHttp = false;
        }
    }

    if (!xmlHttp) alert("Obyek XMLHttpRequest gagal dibuat");
    else
    return xmlHttp;
}

function select()
{
    if (xmlHttp.readyState == 4 || xmlHttp.readyState == 0)
    {
        propinsi =
            encodeURIComponent(document.getElementById("propinsi").value);

        xmlHttp.open("GET", "select.php?prop=" + propinsi, true);
        xmlHttp.onreadystatechange = handleServerResponse;
        xmlHttp.send(null);
    }
    else setTimeout('select()', 1000);
}

function handleServerResponse()
{
    if (xmlHttp.readyState == 4)
    {
        if (xmlHttp.status == 200)
        {
            xmlResponse = xmlHttp.responseXML;
        }
    }
}
```

```
xmlRoot = xmlResponse.documentElement;

kabArray = xmlRoot.getElementsByTagName("kab");

html = "";

for (var i=0; i<kabArray.length; i++)
{
    html += "<option value='"
        + kabArray.item(i).firstChild.data
        + "'>"
        + kabArray.item(i).firstChild.data + "</option>";
}

document.getElementById("kabupaten").innerHTML = html;
}

else
{
    alert("Ada kesalahan dalam mengakses server: " +
        xmlHttp.statusText);
}
}
```

select.php

```
<?php

mysql_connect("localhost","root","root");
mysql_select_db("test");

header('Content-Type: text/xml');

$prop = $_GET['prop'];

$query = "SELECT kabupaten FROM prop_kab WHERE propinsi = '$prop'";
$hasil = mysql_query($query);

echo '<output>';

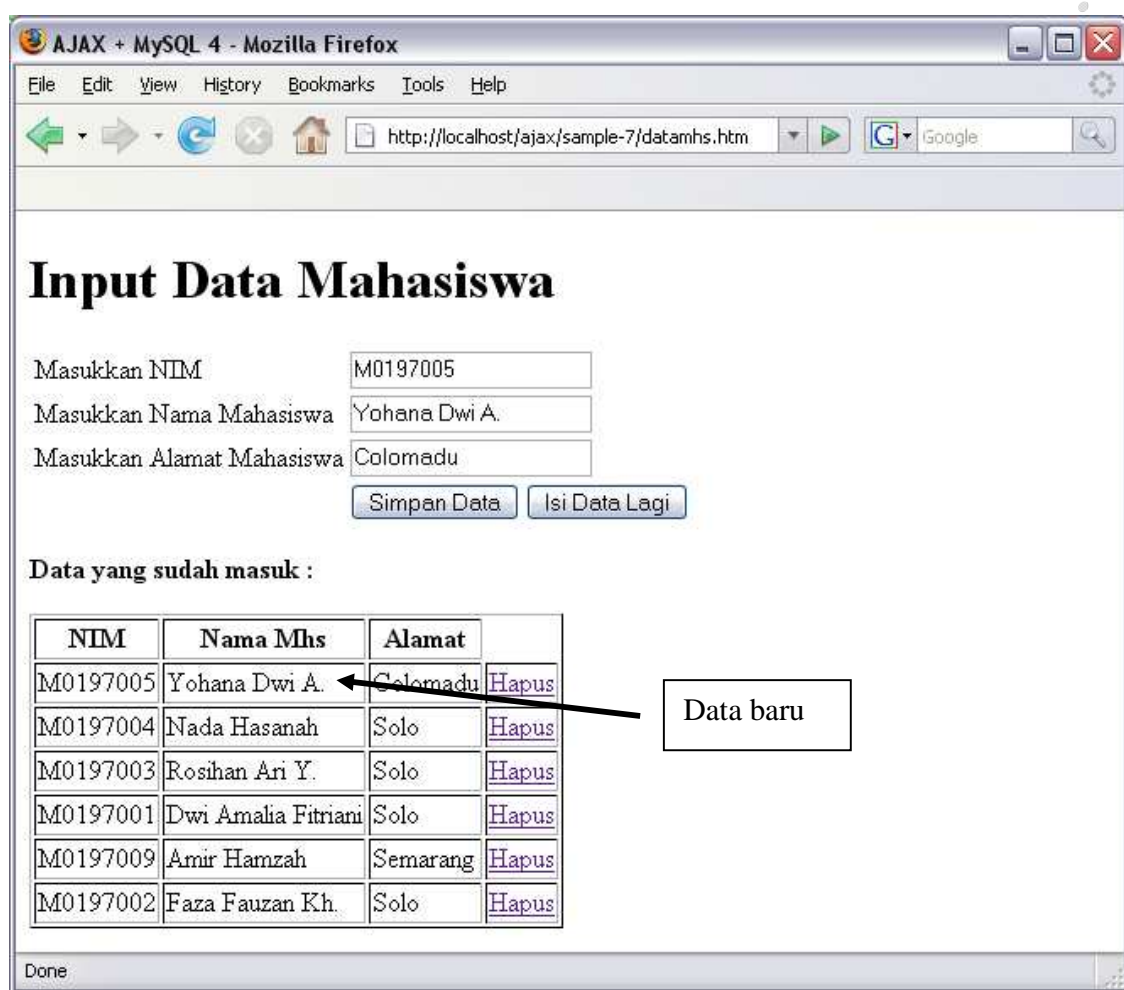
while ($data = mysql_fetch_array($hasil))
{
    echo "<kab>".$data['kabupaten']."</kab>";
}

echo '</output>';
?>
```

Contoh AJAX Database 4

Untuk contoh kali ini, kita akan membuat aplikasi AJAX database yang agak sedikit rumit yaitu terkait dengan tambah data dan hapus data mahasiswa.

Berikut ini adalah screen shot aplikasinya:



Dalam aplikasi ini, user dapat memasukkan data mahasiswa untuk disimpan ke dalam database. Setelah user mengisi data mahasiswa, selanjutnya tombol Simpan Data diklik. Secara langsung tabel yang menampilkan semua data akan berubah sendiri (tanpa refresh halaman).

Untuk menghapus data mahasiswa tertentu, user dapat langsung mengklik link Hapus pada baris data yang diinginkan. Lagi-lagi, dengan tanpa merefresh, tampilan tabel akan berubah setelah proses penghapusan.

Struktur database dan tabel untuk aplikasi ini adalah:

Nama database: dbMhs

```
CREATE TABLE mhs (  
    NIM varchar(9) NOT NULL,  
    NAMAMHS varchar(20) NOT NULL,  
    ALAMAT varchar(20) NOT NULL,  
    PRIMARY KEY (NIM)  
);
```

Berikut ini listing code nya:

datamhs.htm

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<title>AJAX + MySQL 4</title>  
<script type="text/javascript" src="action.js"></script>  
</head>  
  
<body onload="tampil()">  
<h1>Input Data Mahasiswa</h1>  
<form name="form1">  
<table>  
<tr>  
<td>Masukkan NIM</td><td><input type="text" id="nimMhs" /></td>  
</tr>  
<tr><td>Masukkan Nama Mahasiswa</td><td>  
<input type="text" id="namaMhs" /></td></tr>  
<tr><td>Masukkan Alamat Mahasiswa</td>  
<td><input type="text" id="alamatMhs" /></td></tr>  
<tr><td></td><td>  
<input type="button" value="Simpan Data" onclick="simpan()" />  
<input type="reset" value="Isi Data Lagi" /></td></tr>  
</table>  
  
</form>  
  
<p><strong>Data yang sudah masuk :</strong></p>  
<div id="data" />  
  
</body>  
</html>
```

action.js

```
var xmlHttp = createXmlHttpRequestObject();

function createXmlHttpRequestObject()
{
    var xmlHttp;
    if (window.ActiveXObject)
    {
        try
        {
            xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch (e)
        {
            xmlHttp= false;
        }
    }
    else
    {
        try
        {
            xmlHttp=new XMLHttpRequest();
        }
        catch(e)
        {
            xmlHttp=false;
        }
    }

    if (!xmlHttp) alert ("Object XMLHttpRequest gagal dibuat !");
    else
    return xmlHttp;
}

function tampil()
{
    if (xmlHttp.readyState ==4 || xmlHttp.readyState ==0 )
    {
        xmlHttp.open ("GET","datamhs.php?op=tampildata",true);
        xmlHttp.onreadystatechange = handleServerResponse;
        xmlHttp.send(null);
    }
    else
    setTimeout('tampil()',1000);
}

function simpan()
{
    if (xmlHttp.readyState==4 || xmlHttp.readyState==0)
    {
```

```
        nim
    =encodeURIComponent(document.getElementById("nimmhs").value);
        nama
    =encodeURIComponent(document.getElementById("namamhs").value);

    alamat=encodeURIComponent(document.getElementById("alamatmhs").value
);
        /* kesalahan semula: kurang tanda sama dengan setelah
    op=simpandata&&nim */

    xmlHttp.open("GET","datamhs.php?op=simpandata&nim="+nim+"&nama="+nam
a+"&alamat="+alamat,true);
    xmlHttp.onreadystatechange = handleServerResponse;
    xmlHttp.send(null);
    }
    else
    setTimeout('simpan()',1000);
}

function hapus(nim)
{
    if (xmlHttp.readyState==4 || xmlHttp.readyState==0)
    {
        xmlHttp.open("GET","datamhs.php?op=hapusdata&nim="+nim,true);
        xmlHttp.onreadystatechange = handleServerResponse;
        xmlHttp.send(null);
    }
    else
    setTimeout('hapus()',1000);
}

function handleServerResponse()
{
    if (xmlHttp.readyState==4)
    {
        if (xmlHttp.status == 200)
        {
            var xmlResponse = xmlHttp.responseXML;
            xmlRoot =xmlResponse.documentElement;

            nimArray = xmlRoot.getElementsByTagName("nim");
            namaMhsArray = xmlRoot.getElementsByTagName("namamhs");
            alamatArray = xmlRoot.getElementsByTagName("alamat");

            html = "<table
                border='1'><tr><th>NIM</th><th>Nama</th><th>
                Alamat</th></tr>";

            for (var i=0; i<nimArray.length; i++)
            {
                html += "<tr><td>" +
                    nimArray.item(i).firstChild.data + "</td><td>" +
                    namaMhsArray.item(i).firstChild.data + "</td><td>" +
```

```
alamatArray.item(i).firstChild.data +
"</td><td><a href=\"datamhs.php\"
onclick=\"hapus('\"+nimArray.item(i).firstChild.data+\"')\";
return false;\">Hapus</a></td></tr>\";
}
html = html + "</table>\";
document.getElementById(\"data\").innerHTML = html;
}
else
{
    alert(\"Ada kesalahan dalam mengakses server : \"
    + xmlhttp.statusText);
}
}
}
```

datamhs.php

```
<?php

header('Content-Type: text/xml');

echo '<hasil>';

$nimMhs = $_GET['nim'];
$namaMhs = $_GET['nama'];
$alamatMhs = $_GET['alamat'];

$op = $_GET['op'];

mysql_connect(\"localhost\", \"root\", \"root\");
mysql_select_db(\"dbMhs\");

if ($op == \"tampildata\")
{
    $query = \"SELECT * FROM mhs\";
    $hasil = mysql_query($query);
}
else if ($op == \"simpandata\")
{
    $query = \"INSERT INTO mhs VALUES('$nimMhs', '$namaMhs', '$alamatMhs')\";
    mysql_query($query);

    $query = \"SELECT * FROM mhs\";
    $hasil = mysql_query($query);
}
else if ($op == \"hapusdata\")
{
    $query = \"DELETE FROM mhs WHERE nim = '$nimMhs'\";
    mysql_query($query);

    $query = \"SELECT * FROM mhs\";
    $hasil = mysql_query($query);
}
```

```
}

while ($data = mysql_fetch_array($hasil))
{
    echo "<mhs>";
    echo "<nim>".$data['NIM']. "</nim>";
    echo "<namamhs>".$data['NAMAMHS']. "</namamhs>";
    echo "<alamat>".$data['ALAMAT']. "</alamat>";
    echo "</mhs>";
}

echo '</hasil>';

?>
```