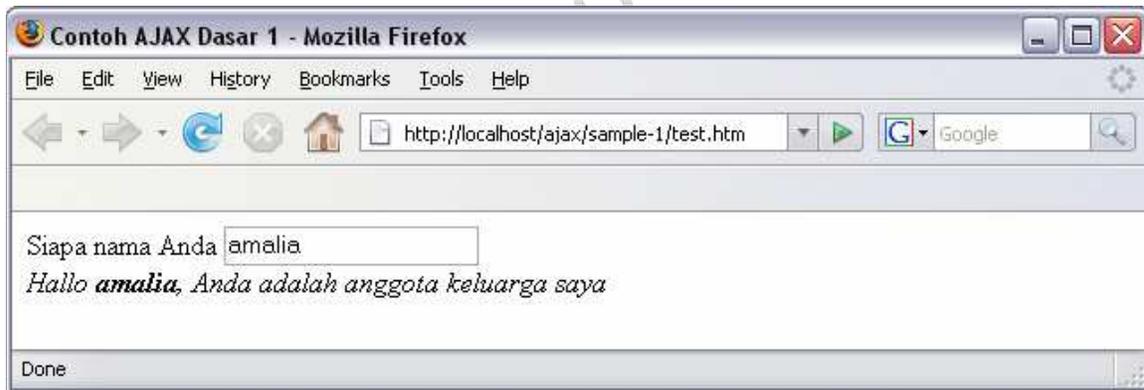
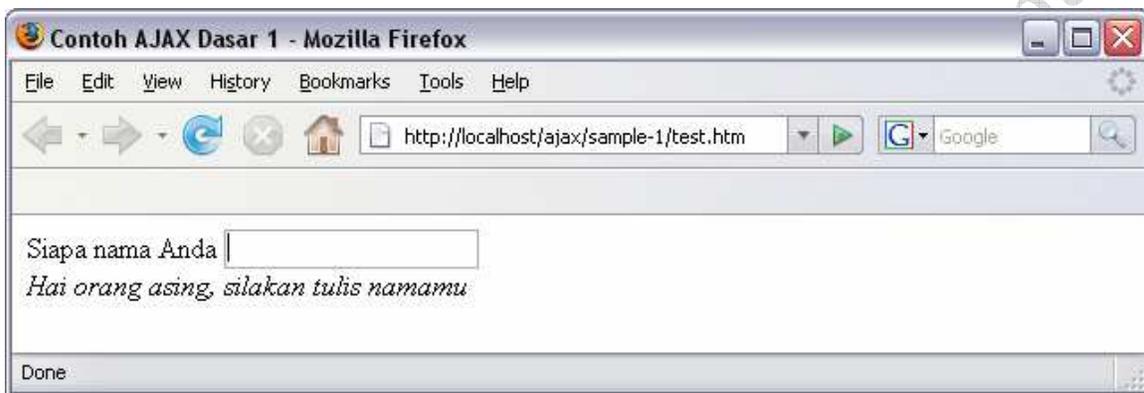
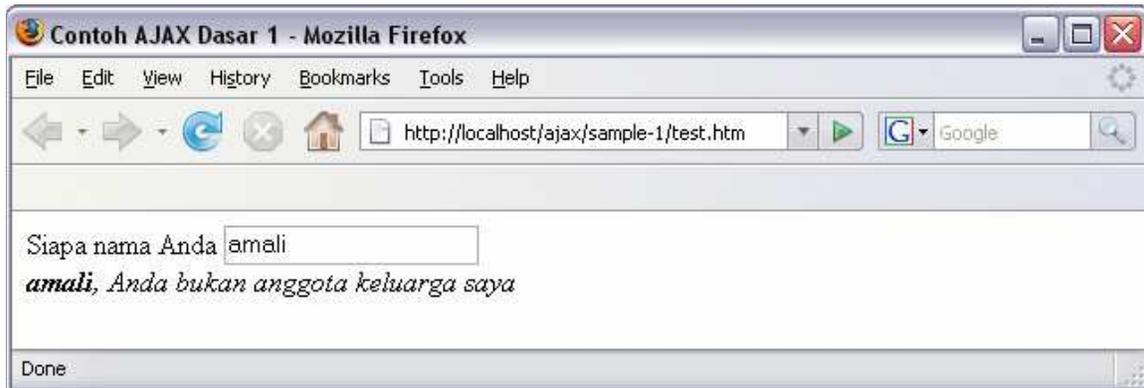


Contoh AJAX Dasar 1

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');

$name = $_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($name), $myFamily))
    echo 'Hallo &lt;strong>'. htmlentities($name) .
        '&lt;/strong>; Anda adalah anggota keluarga saya';

// jika nama masih kosong

else if (trim($name) == '')
    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>';

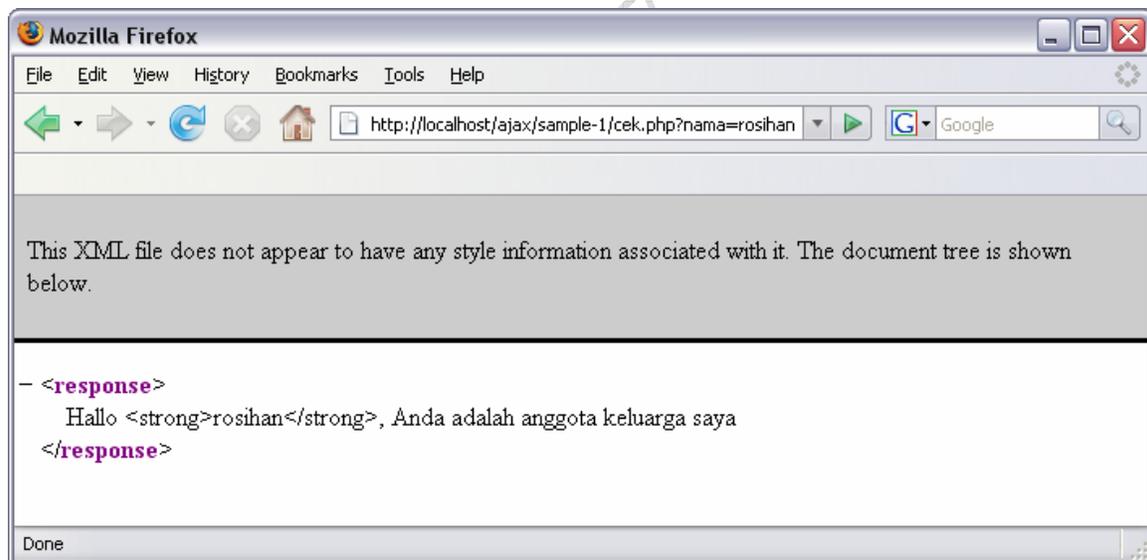
?>
```

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 `>`;

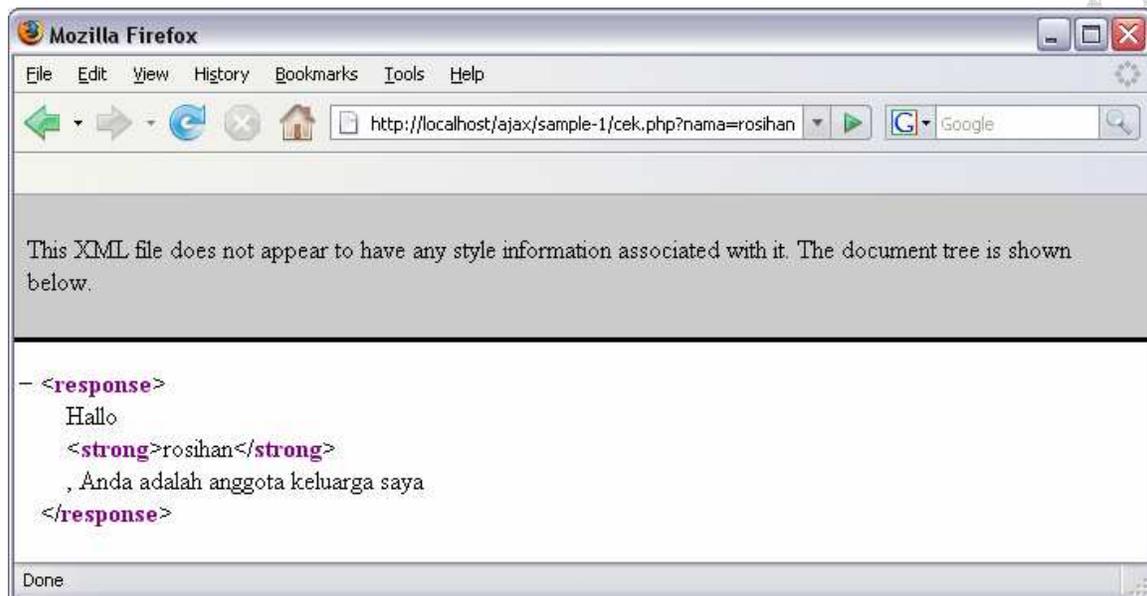
```
echo 'Hallo &lt;strong&gt;'. htmlentities($nama) .
    '&lt;/strong&gt;, Anda adalah anggota keluarga saya';
```



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.