

|  |
| --- |
| **Web Programming II** |
|  |
| **Modul 8** |
| **Javascript** |

**Disusun oleh:**

**Dwi Intan Af’idah, S.T., M.Kom**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**POLITEKNIK HARAPAN BERSAMA**

**TAHUN AJARAN 2020/2021**

Daftar Isi

[Daftar Isi ii](#_Toc57180645)

[1 Pengantar 1](#_Toc57180646)

[1.1 Ajax 1](#_Toc57180647)

[1.2 Kelebihan dan Kekurangan Javascript 2](#_Toc57180648)

[2 XML dan AJAX 4](#_Toc57180649)

[2.1 Mengenal XML 4](#_Toc57180650)

[2.2 XML dan AJAX 4](#_Toc57180651)

[3 Ajax 5](#_Toc57180652)

[3.1 HTTP Request dan Dasar-dasar Respon 6](#_Toc57180653)

[3.2 XMLHttp Request 7](#_Toc57180654)

[4 Contoh Pemrograman AJAX Dasar 9](#_Toc57180655)

[4.1 Contoh1 9](#_Toc57180656)

[4.2 Search 15](#_Toc57180657)

[4.3 Aplikasi Form 19](#_Toc57180658)

1. Pengantar
   1. Ajax

* AJAX memiliki kepanjangan Asynchronous Javascript And XML merupakan suatu teknik baru dalam dunia web.
* 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.
* Tujuannya adalah untuk memindahkan sebagian besar interaksi pada komputer web surfer, melakukan pertukaran data dengan server di belakang layar, sehingga halaman web tidak harus dibaca ulang secara keseluruhan setiap kali seorang pengguna melakukan perubahan.
* Hal ini akan meningkatkan interaktivitas, kecepatan, dan usability.
* Adapun beberapa contoh website telah menggunakan fasilias AJaX ini adalah Facebook, Google Map, Google Mail, Yahoo Mail dlL.

|  |  |
| --- | --- |
| Website Konvensional | Website AJAX |

* Sedangkan kekurangan AJAX dibandingkan mekanisme biasa adalah memiliki tingkat kesukaran pemrograman yang lebih tinggi, karena programmer harus menguasai 4 hal yaitu:

1. HTML-(XHTML),
2. Javascript,
3. XML,
4. dan juga server side programming.
   1. Kelebihan dan Kekurangan Javascript
5. **Kelebihan Javascript**

**High Interactivity,** Aplikasi AJAX lebih interaktif dibanding dengan aplikasi web konvensional.

**High Usability,** Update data tidak me-reload keseluruhan halaman, melainkan hanya yang relevan.

**High Speed,** Aplikasi AJAX lebih cepat dibanding dengan aplikasi web konvensional.

Dengan Ajax, javascript akan berkomunikasi secara langsung dengan server melalui objek javascript yaitu XMLHttpRequest tersebut.

Dengan objek XMLHttpRequest, suatu halaman web dapat membuat request ke, dan mendapatkan respon dari server web tanpa me-reload halaman secara keseluruhan. User akan selalu tetap dengan halaman yang sama. Bahkan user tidak akan tahu kalau ada data yang dikirim dan diterima dari server, karena javascript melakukan transaksi data di balik layar.

Bagusnya lagi permintaan dikirim asynchronous, yang berarti bahwa kode JavaScript (dan pengguna) tidak menunggu pada server untuk merespon. Sehingga pengguna dapat terus

memasukkan data, bergulir sekitar, dan menggunakan aplikasi.

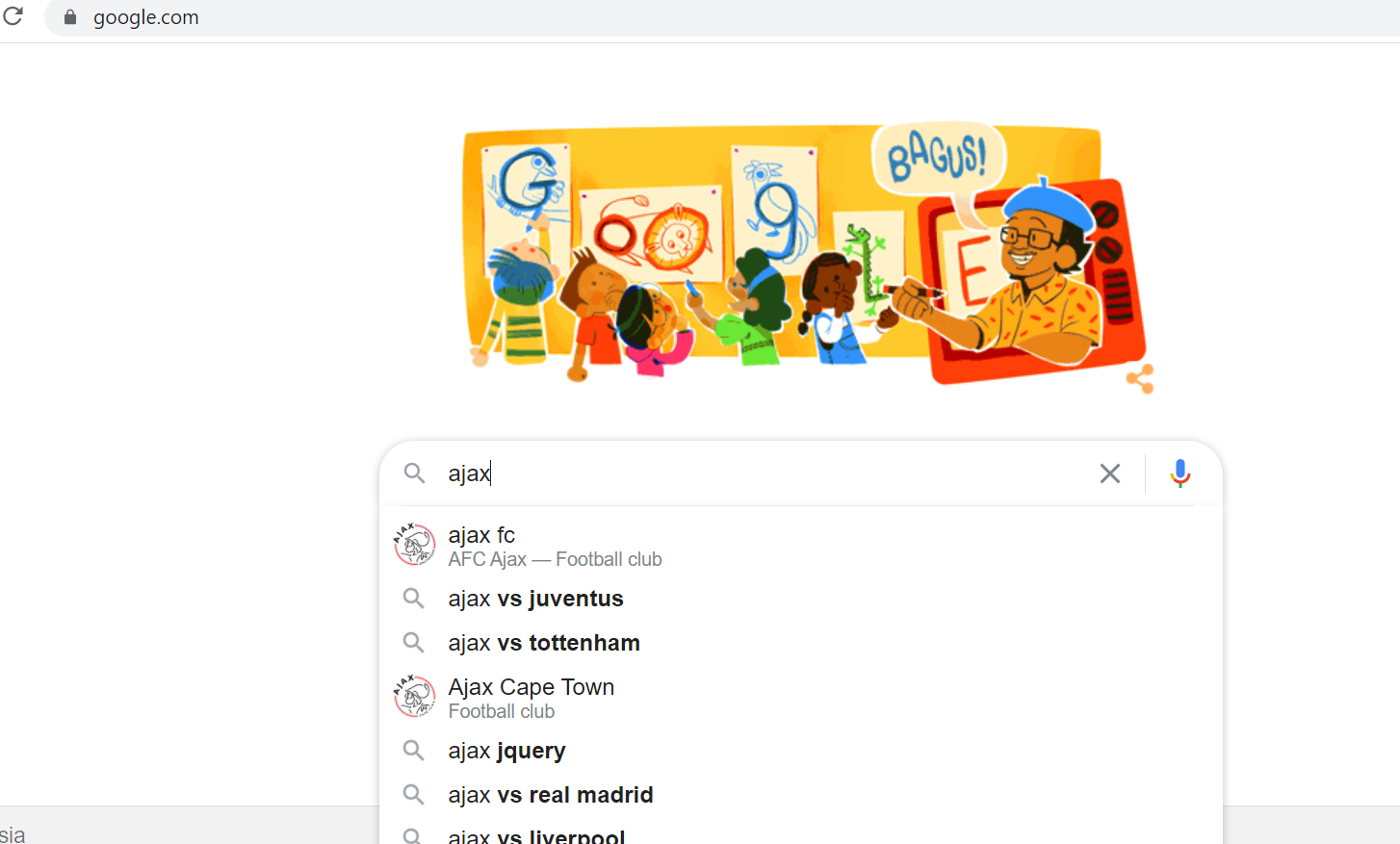
1. **Kekurangan Javascript**
2. Integrasi browser: Karena konten halaman menganut prinsip asinkron, di mana data bisa di-update tanpa halaman di-refresh, maka perubahan tampilan tidak tercatat di bagian history dari browser. Jadi ketika Anda mengklik tombol Back, yang muncul bukan tampilan sebelum mengeksekusi kode AJAX, namun halaman sebelumnya.
3. Kekurangan lainnya adalah kesulitan bagi seorang user untuk melakukan bookmark/ favorite pada bagian tertentu dari halaman web.
4. Search engine optimization: Karena konten di-generate menggunakan JavaScript, search engine tidak bisa mengindeksnya sehingga mengurangi efektivitas halaman ditinjau dari SEO.
5. Terlalu mengandalkan JavaScript: AJAX menggunakan JavaScript, yang kadang diimplementasikan secara berbeda di berbagai browser atau verse tertentu dari sebuah browser. Karena. itu sering kale sebuah website yang mengandung JavaScript harus dites menggunakan berbagai jenis browser untuk memastikan tampilannya tidak ada yang salah, begitu juga dengan AJAX. Namun mengingat browser sekarang seperti Mozilla dan IE 7 menggunakan lapisan abstraksi semisal JavaScript Framework, maka kekurangan ini sepertinya bisa ditanggulangi nanti.
6. Alat bantu pemrograman berupa IDE (integrated development environment) untuk JavaScript sangatlah jarang dan langka. Anda bisa menggunakan beberapa tool, seperti Firebug, IE Developer toolbar, dan Venkman.
7. Termasuk kelemahan JavaScript adalah apabila user men-disable JavaScript di browser-nya, maka AJAX tidak akan bisa digunakan.
8. Web analytics: Berbagai solusi web analytic biasanya memiliki anggapan bahwa sebuah halaman barn di-loading tiap kali konten di-update ke user. Mengingat AJAX mengubah paradigma seperti ini, maka programmer harus mengatur peletakan kode web analytic sehingga proses tracking akan lebih baik ditampilkan di browser. User bisa melihat dan menirunya dari sourcenya.
9. XML dan AJAX
   1. Mengenal XML

* XML (Extensible Markup Language) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam.
* XML merupakan kelanjutan dari HTML (HyperText Markup Language) yang merupakan bahasa standar untuk melacak Internet.
* XML (eXtensible Markup Language) merupakan bahasa web turunan dari SGML (Standart Generalized Markup Language) yang ada sebelumnya.
  1. XML dan AJAX
* 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-(XHTML), Javascript, XML dan juga server side programming.**

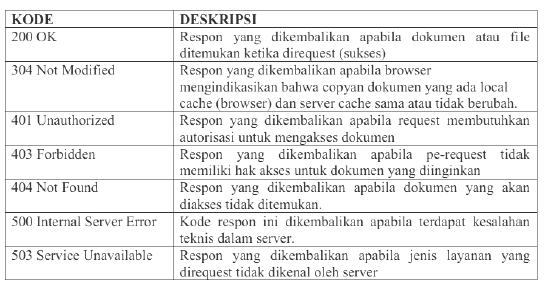
1. Ajax

Contoh teknologi AJAX:

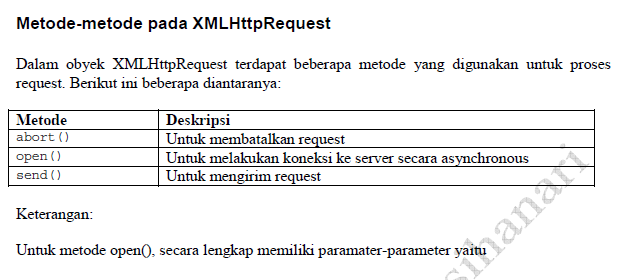
1. Bagi yang memiliki email account email di Gmail, tentu sudah pernah merasakan teknologi ini, khususnya ketika membuka email di Inbox. Ketika Anda membuka inbox dan membaca email, serasa halaman web tidak merefresh.
2. Sebagai contoh, misalnya pada penggunaan aplikasi Google Suggest.



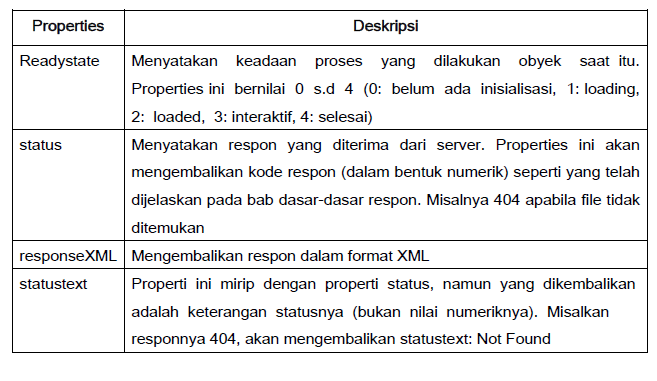
* 1. HTTP Request dan Dasar-dasar Respon
* Untuk memahami konsep AJAX, pertama kali akan dibahas mengenai bagaimana situs web merequest dan menerima respon dari web server.
* Saat ini standar browser untuk memperoleh informasi dari web server adalah berbasis pada metode HTTP (HyperText Transfer Protocol). 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 menghandle 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



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

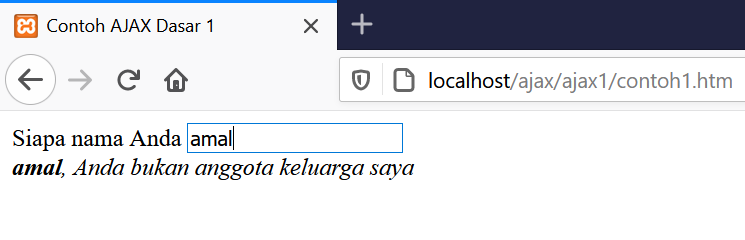


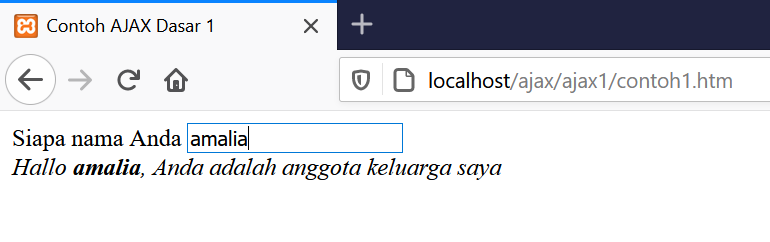
* Karena berbentuk object, maka XMLHttpRequest memiliki beberapa properties. Berikut ini beberapa diantaranya yang sering digunakan untuk aplikasi AJAX.



1. Contoh Pemrograman AJAX Dasar
   1. Contoh1

* 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 &lt;strong&gt;'. htmlentities($nama) .

'&lt;/strong&gt;, 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 '&lt;strong&gt;' . htmlentities($nama) .

'&lt;/strong&gt;, Anda bukan anggota keluarga saya';

// menutup root tag elemen

echo '</response>';

?>

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.

* 1. Search



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>

Sedangkan untuk formnya adalah sbb:

**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("namahost","dbuser","dbpass");

mysql\_select\_db("dbname");

$query = "SELECT \* FROM mhs WHERE namamhs LIKE '%$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>';

?>

* 1. Aplikasi Form



* 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" />

</select>

</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("namahost","dbuser","dbpass");

mysql\_select\_db("dbname");

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>';

?>