

Penggunaan AJAX pada Pengembangan Aplikasi Web

Hendro Steven Tampake

Universitas Kristen Satya Wacana
Jl. Diponegoro 52-60 Salatiga 50711
Telp. (0298) – 321212
Email: hendro.steven@gmail.com

Abstract

AJAX is an acronym of Asynchronous JavaScript And XML. AJAX is a technique of programming that popular at 2005 by Google with their Google Suggest web application. AJAX is not a new programming language, but it is a web programming method that used current web programming standard. With AJAX web application is more fast and user-friendly .

Key Words : Ajax, Web, Internet, Pemrograman, Javascript, HTML, XML

1. Perkembangan Teknologi Web

Perkembangan aplikasi web saat ini sangat pesat. Aplikasi web yang semula hanya merupakan aplikasi statik, sekarang telah menjadi aplikasi web yang sangat dinamis dan interaktif. Aplikasi web juga tidak lagi digunakan untuk media promosi semata. Aplikasi web bahkan saat ini sudah digunakan untuk aplikasi bisnis dengan skala yang besar.

Pengembangan aplikasi web juga bukan merupakan suatu hal yang sulit. Banyak teknologi dan *tools* yang dapat digunakan untuk mengembangkan aplikasi berbasis web. Mulai dari teknologi *scripting language* seperti HTML sampai ke bahasa pemrograman seperti ASP, PHP, JSP dan lainnya. Bentuk aplikasi web juga sudah sangat menarik dengan adanya teknologi yang dapat memberikan *style* seperti CSS pada aplikasi web. Selain itu banyak juga teknologi lain yang dapat digunakan untuk mengembangkan aplikasi web agar lebih interaktif seperti Flash.

2. Hyper Text Markup Language

Hyper Text Markup Language atau biasa disingkat HTML merupakan *markup language* untuk pengembangan aplikasi berbasis web. HTML mendeskripsikan bagaimana sebuah halaman ditampilkan pada *browser*.

HTML menggunakan *tag-tag* khusus untuk mendeskripsikan bagaimana konten sebuah halaman ditampilkan. *Tag-tag* khusus ini akan diterjemahkan

oleh *browser* sehingga *browser* mengerti bagaimana menampilkan suatu konten.

HTML merupakan sebuah standar yang dimaintain oleh *World Wide Web Consorcium* (W3C). Sebagai standar maka *tag-tag* HTML tentunya akan diterjemahkan sama oleh setiap *browser* yang ada. Hal ini yang menyebabkan aplikasi-aplikasi web yang dikunjungi relatif sama tampilannya walaupun dibuka atau diakses menggunakan *browser* yang berbeda-beda seperti Internet Explorer, Firefox, Netscape, Opera dan lain-lain.

Sebuah halaman HTML memiliki struktur seperti berikut ini:

```
<html>
<head>
<title>Title of page</title>
</head>
<body>
This is my first homepage. <b>This text is bold</b>
</body>
</html>
```

HTML sebenarnya file teks biasa yang mengandung *tag-tag* khusus dan disimpan dengan ekstensi .html atau .htm. Untuk menjalankan *source* di atas dibutuhkan *browser* misalnya Firefox atau Internet Explorer.

Beberapa *tag* HTML yang biasa digunakan antara lain, *tag* yang digunakan untuk format teks ditunjukkan pada Tabel 1. Untuk mengatur warna dari huruf, digunakan *tag* *some text*. *Color* dapat berisi kode warna nilai konstan warna seperti *blue*, *red*, *green*, *yellow*, dan lain-lain. Untuk menentukan tebal judul suatu topik atau *heading* dapat digunakan *tag* <hN>...</hN>. N dapat diganti dengan angka 1 sampai dengan 6. N=1 merupakan *heading* paling besar, dan sebaliknya N=6 merupakan *heading* paling kecil. Untuk membuat paragraf digunakan *tag* <p>...</p>. Paragraf juga dapat diatur untuk rata kiri, kanan, tengah atau *justify* dengan menambahkan properti *align* <p align="...">...</p>. Nilai *align* dapat berupa: *left*, *right*, *center*, dan *justify*. Kelebihan utama dari aplikasi web adalah kemampuannya untuk berpindah antar halaman atau bahkan antar aplikasi web yang berbeda dengan menggunakan *hyperlink*. Untuk keperluan ini digunakan *tag* Untuk menyisipkan gambar pada halaman web digunakan *tag* .

Tabel 1 Tag HTML untuk Format Teks

Tag HTML	Keterangan
...	Huruf tebal
<i>...</i>	Huruf miring
<u>...</u>	Garis bawah
^{...}	Super script
_{...}	Sub script
<small>...</small>	Small text

3. JavaScript

JavaScript digunakan oleh milyaran aplikasi web yang ada untuk desain, validasi data, deteksi *browser*, membuat *cookie*, dan lain-lain. JavaScript merupakan *scripting language* yang paling populer pada pemrograman web atau internet. JavaScript dapat bekerja atau dieksekusi pada berbagai jenis *browser* seperti Internet Explorer, Firefox, Netscape, dan Opera.

JavaScript didesain untuk menambah interaktifitas dari sebuah aplikasi web. Kode JavaScript biasanya disisipkan pada sebuah halaman HTML atau dapat juga disimpan pada *file* terpisah dan dipanggil dari sebuah halaman HTML yang membutuhkannya.

Untuk menyisipkan JavaScript pada halaman HTML digunakan *tag* `<script>`.

```
<html>
<body>
<script type="text/javascript">
    document.write("Hello World!");
</script>
</body>
</html>
```

JavaScript juga dapat disisipkan pada *tag* `<head>` dari sebuah halaman HTML. JavaScript yang disisipkan pada *tag* `<body>` seperti contoh di atas, akan dieksekusi saat halaman HTML tersebut dipanggil. Sedangkan JavaScript yang disisipkan pada *tag* `<head>` akan dieksekusi jika dipanggil.

```
<html>
<head>
<script type="text/javascript">
function message()
{
    alert("Hello Welcome to Javascript!")
}
</script>
</head>
<body >
    <input type="button" value="Click Me"
onclick="message()">
</body>
</html>
```

4. XML

XML adalah singkatan dari *eXtensible Markup Language*. XML seperti halnya HTML merupakan sebuah *markup language*. Perbedaan utama antara XML dan HTML adalah bahwa XML didesain untuk membawa dan menyimpan data, sedangkan HTML didesain untuk menampilkan data. XML

bukan untuk menggantikan HTML karena didesain untuk tujuan yang berbeda.

Berbeda dengan HTML yang telah memiliki *tag-tag* yang terdefinisi, XML harus didefinisikan sendiri sesuai dengan kebutuhan atau dengan kata lain diciptakan sendiri.

XML saat ini menjadi standar yang umum untuk *data transmission* antar aplikasi dan menjadi sangat populer sebagai *tool* untuk menyimpan dan menjelaskan informasi.

Contoh dokumen XML

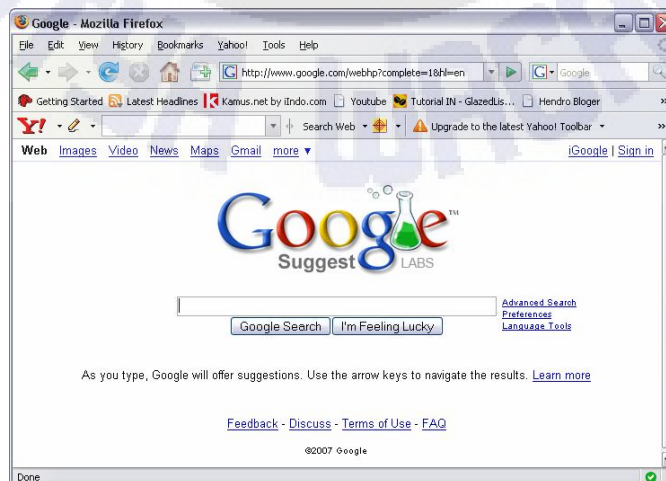
```
<mahasiswa nim="234234234">
  <biodata>
    <nama>Hendro</nama>
    <ttl>Salatiga, 11 Des 1980</ttl>
    <alamat>Salatiga</alamat>
  </biodata>
  <fakultas>Teknologi Informasi</fakultas>
</mahasiswa>
```

Tag-tag pada contoh dokumen di atas tidak didefinisikan oleh standar dokumen XML, tetapi didefinisikan sendiri.

5. Pengantar AJAX

AJAX bukanlah suatu bahasa pemrograman yang baru. AJAX adalah merupakan teknik yang menggunakan standar web yang sudah ada untuk mengembangkan aplikasi web yang lebih baik, lebih cepat, dan lebih *user friendly* serta lebih interaktif.

Dengan teknik menggunakan AJAX, maka JavaScript yang ada pada sebuah halaman web dapat berkomunikasi langsung ke *server*, menggunakan objek JavaScript **XMLHttpRequest**. Dengan objek ini, kode JavaScript dapat mengakses data di *server* tanpa harus *me-reload* seluruh halaman web. [1]



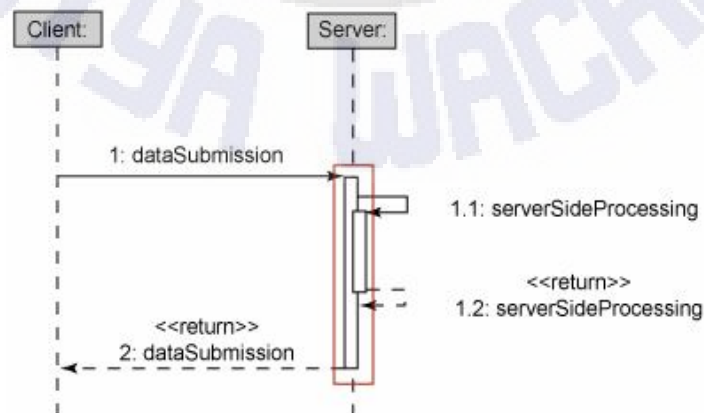
Gambar 1 Google Suggest Menggunakan AJAX

Pada Gambar 1 di atas adalah Aplikasi *Google Suggest*. Aplikasi *Google Suggest* adalah salah satu contoh aplikasi web yang menggunakan teknologi AJAX untuk aplikasinya. Pada Aplikasi *Google Suggest*, AJAX digunakan untuk *auto completion*-nya. Pada saat diketikkan sesuatu pada inputan teks pencariannya, maka aplikasi *Google Suggest* akan mengirimkan inputan tersebut dan *server* mengirimkan kemungkinan kata-kata yang cocok dengan kata yang diinputkan. Aplikasi *Google Suggest* dapat dengan cepat menampilkan daftar kemungkinan kata-kata tanpa terlihat bahwa *browser* melakukan *reload page*. Ini dimungkinkan karena teknologi AJAX melakukan proses secara *Asynchronous* lewat *object* JavaScript XMLHttpRequest [6].

Pada aplikasi web kebanyakan yang belum menggunakan teknologi AJAX hal di atas tidak terjadi. Setiap ada *request* yang dikirimkan dari *user* maka keseluruhan halaman akan *reload* lagi. Jika halaman yang di-*reload* memiliki ukuran yang cukup besar, maka tentunya akan memperlambat akses ke halaman tersebut. Di sinilah kelebihan menggunakan AJAX pada aplikasi web.

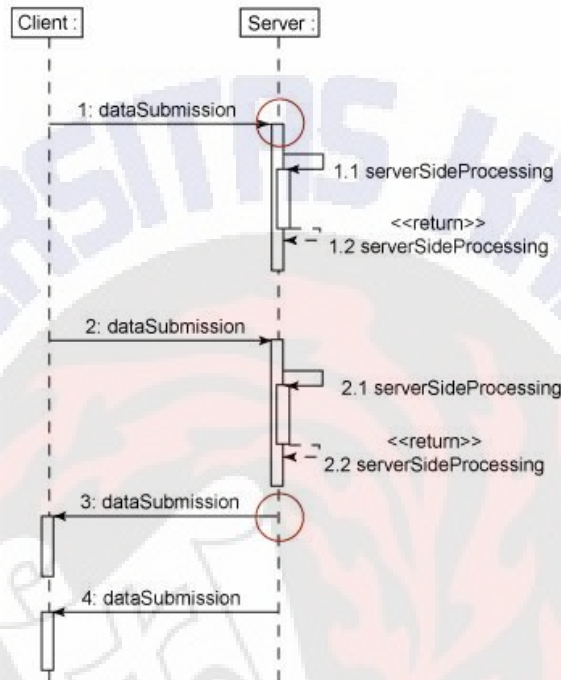
6. Komunikasi *Synchronous* dan *Asynchronous*

Untuk memahami AJAX maka perlu diketahui perbedaan komunikasi *Synchronous* dan *Asynchronous* dalam aplikasi web. Pada komunikasi *Synchronous*, *client* dalam hal ini *browser* akan melakukan *request* ke *server*. Kemudian *server* akan menciptakan *response* terhadap *request* yang diterima tersebut. Selama proses tersebut (*request/response*) maka *browser* tidak dapat mengirimkan *request* yang lain selama *response* untuk *request* sebelumnya belum selesai. Atau dengan kata lain *user/pengguna* harus menunggu untuk dapat menggunakan lagi *browser*-nya [6]. Sebagai contoh saat ditekan tombol atau mengklik pada sebuah *link* di halaman web maka tidak dapat melakukan apa-apa sampai proses atau *reload* halaman selesai. Biasanya ditandai dengan halaman yang *blank* sejenak.



Gambar 2 Komunikasi *Synchronous*

Pada komunikasi *Asynchronous*, *user* atau pengguna masih tetap dapat menggunakan *browser* walaupun *asynchronous request* masih diproses. Setelah *asynchronous request* selesai diproses, maka sebuah *asynchronous response* dikirimkan ke *client*. Atau dengan kata lain proses yang terjadi tidak mempengaruhi *user*, *user* tidak perlu menunggu sampai *response* selesai.



Gambar 3 Komunikasi *Asynchronous*

7. Objek XMLHttpRequest

Kemampuan AJAX sebenarnya karena adanya objek JavaScript XMLHttpRequest. Objek XMLHttpRequest didukung oleh Internet Explorer 5.0+, Safari 1.2, Mozilla 1.0/Firefox, Opera 8+, dan Netscape 7. Dengan didukung oleh banyak *browser*, maka aplikasi web yang menggunakan teknologi AJAX tidak perlu kuatir, karena pasti dapat berjalan pada berbagai *browser* populer tersebut.

Tabel 1 menampilkan macam *attributes* pada objek XMLHttpRequest, sedangkan Tabel 2 menampilkan macam *methods* pada objek XMLHttpRequest.

Tabel 2 *Attributes* Objek XMLHttpRequest [2]

index.isn	
ATTRIBUTE	KETERANGAN
readyState	Kode berhasil berubah dari 0 ke 4
status	200 adalah berhasil dan 404 adalah halaman tidak ditemukan
responseText	Menyimpan hasil response dalam bentuk string
responseXML	Menyimpan hasil response dalam bentuk xml
onreadystatechange	Property yang mengambil fungsi sebagai nilai yang akan dieksekusi

Tabel 3 *Methods* Objek XMLHttpRequest [2]

METHOD	KETERANGAN
open(mode,url,boolean)	mode: jenis request, GET or POST url: lokasi dari file yang akan melakukan proses, bisa berupa path. boolean: true (asynchronous)/false (synchronous).
send("string")	null for a GET command.

8. Menggunakan AJAX pada Aplikasi Web Berbasis Java Server Page

Untuk menunjukkan bagaimana AJAX digunakan pada aplikasi web, maka dibuat contoh aplikasi sederhana menggunakan Java Server Page (JSP) sebagai *server side scripting*-nya. Sebagai catatan AJAX dapat digunakan pada bahasa pemrograman web lainnya seperti PHP, ASP, dan lain-lain. [5]

Aplikasi yang akan dibuat adalah aplikasi registrasi *online* dimana akan menggunakan AJAX untuk mengecek apakah *user id* sudah digunakan atau belum. Aplikasi akan segera mengecek apakah *user id* sudah digunakan atau belum begitu user mengetikkan sesuatu pada teks inputan *user id*. [4]

index.jsp

```

<html>
  <head>
    <title>Contoh AJAX</title>
  </head>
  <script>
    function createRequestObject() {
      var req;
      if(window.XMLHttpRequest){
        // Firefox, Safari, Opera...
        req = new XMLHttpRequest();
      } else if(window.ActiveXObject) {
        // Internet Explorer 5+
        req = new ActiveXObject("Microsoft.XMLHTTP");
      }
    }
  </script>

```

```

    } else {
        // There is an error creating the object,
        // just as an old browser is being used.
        alert('Problem creating the XMLHttpRequest object');
    }
    return req;
}

// Make the XMLHttpRequest object
var http;

function sendRequest(user) {
    http = createRequestObject();
    // Open JSP script for requests
    http.open('get', 'proses.jsp?userid='+user.value);
    http.onreadystatechange = handleResponse;
    http.send(null);
}

function handleResponse() {
    if(http.readyState == 4 && http.status == 200){
        // Text returned FROM the JSP script
        var response = http.responseText;
        if(response) {
            // UPDATE ajaxTest content
            document.getElementById("ajaxTest").innerHTML =
response;
        }
    }
}
</script>
</head>
<body>
    <h2>Contoh AJAX</h2>
    <form>
        <table>
            <tr>
                <td>User ID</td>
                <td>: </td>
                <td><input type="text" name="userid"
onkeyup="javascript: sendRequest(this);"/><span
id="ajaxTest"></span></td>
            </tr>
            <tr>
                <td>Password</td>
                <td>: </td>
                <td><input type="password"
name="password"/></td>
            </tr>
            <tr>
                <td>Real Name</td>
                <td>: </td>
                <td><input type="text"
name="real name"/></td>

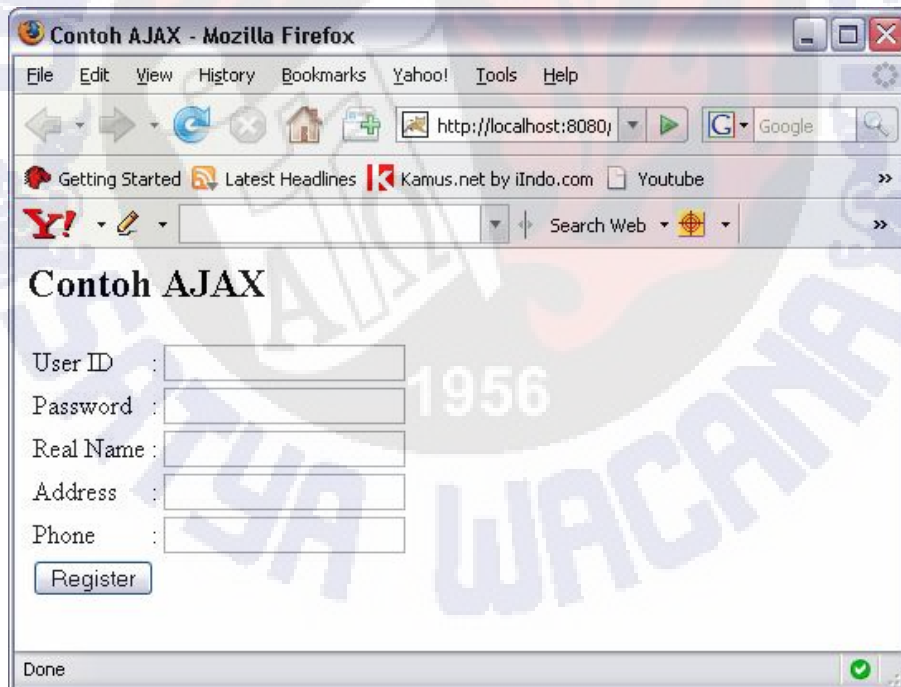
```



```
</tr>
<tr>
    <td>Address</td>
    <td>:</td>
    <td><input type="text" name="address" /
></td>

</tr>
<tr>
    <td>Phone</td>
    <td>:</td>
    <td><input type="text" name="phone" /
></td>

</tr>
<tr>
    <td colspan="3"><input type="submit"
value="Register"/></td>
</tr>
</table>
</form>
</body>
</html>
```



Gambar 4 *index.jsp* Ditampilkan di *Browser*

proses.jsp

```
<%@ page import="java.util.*" %>

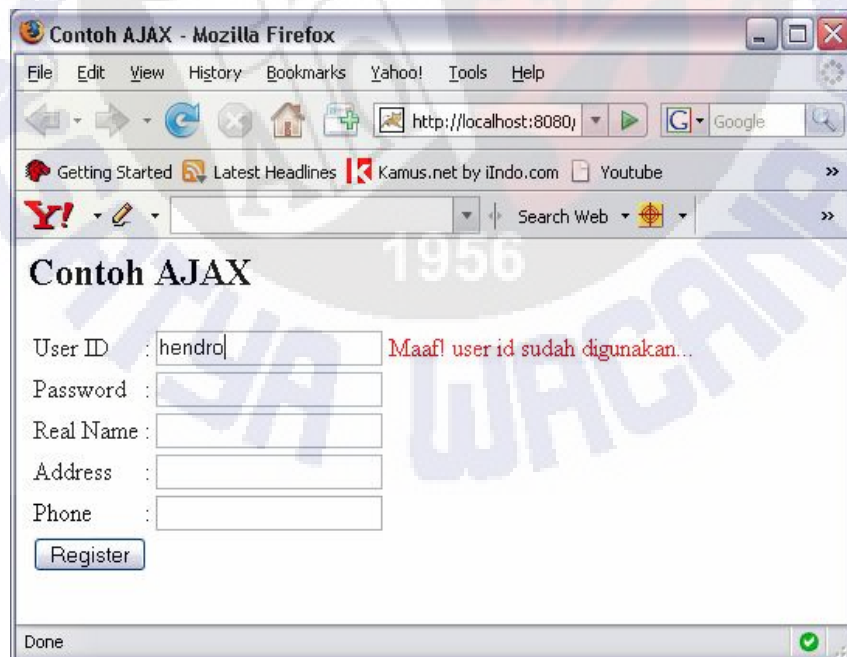
<%
```

```
List regID = new ArrayList();
regID.add("hendro");
regID.add("steven");
regID.add("bambang");
regID.add("bayu");
regID.add("joko");

String userid = request.getParameter("userid");
System.out.println(userid);
if(regID.contains(userid)){
    out.print("<font color=\"red\">Maaf! user id sudah
digunakan...</font>");
}
%>
```

File proses.jsp adalah *file* yang akan melakukan validasi untuk *user id*. Jika *user id* sudah digunakan maka akan dikembalikan pesan “Maaf! User id sudah digunakan” dalam bentuk huruf berwarna merah.

Pada aplikasi sebenarnya atau aplikasi serius, seharusnya proses.jsp melakukan pengecekan ke *database*, tetapi pada tulisan ini hanya didefinisikan lima *user id* yang tidak boleh digunakan lagi yaitu: hendro, steven, bambang, bayu, dan joko. Kelima *user id* ini dideklarasikan dalam objek List. Kemudian setiap *user id* yang diinput oleh *user* akan dibandingkan dengan isi dari objek List tersebut.



Gambar 5 *User Id* yang Diinput Sudah Digunakan

Jika *user id* yang diinputkan sudah digunakan (ada di dalam objek List

yang dijelaskan di atas), maka akan ditampilkan pesan kesalahan seperti pada Gambar 5.

9. Penjelasan Kode Program

```
function createRequestObject() {  
    var req;  
    if(window.XMLHttpRequest){  
        // Firefox, Safari, Opera...  
        req = new XMLHttpRequest();  
    } else if(window.ActiveXObject) {  
        // Internet Explorer 5+  
        req = new ActiveXObject("Microsoft.XMLHTTP");  
    } else {  
        // There is an error creating the object,  
        // just as an old browser is being used.  
        alert('Problem creating the XMLHttpRequest object');  
    }  
    return req;  
}
```

Fungsi di atas melakukan inisialisasi objek XMLHttpRequest. Pertama dibuat objek kosong **req**, kemudian dicek apakah objek window.XMLHttpRequest dikenali oleh *browser* atau tidak, dalam hal ini untuk *browser* Firefox, Safari, dan Opera. Jika tidak dikenali, berarti *browser* adalah Internet Explorer 5+, jadi digunakan objek alternatif yaitu ActiveXObject("Microsoft.XMLHTTP").

Pada kode di atas juga terdapat penanganan kesalahan jika terjadi *error* atau kesalahan lain pada saat inisialisasi objek XMLHttpRequest. Ini terjadi misalnya *user* menggunakan *browser* yang tidak mendukung XMLHttpRequest misalnya Internet Explorer 4.

```
// Make the XMLHttpRequest object  
var http;  
  
function sendRequest(user) {  
    http = createRequestObject();  
    // Open JSP script for requests  
    http.open('get', 'proses.jsp?userid='+user.value);  
    http.onreadystatechange = handleResponse;  
    http.send(null);  
}
```

Objek XMLHttpRequest kemudian diciptakan. Selanjutnya dapat digunakan oleh AJAX untuk mengirimkan *asynchronous request* ke *server*. Fungsi di atas juga menggunakan *method* open() dan send() dari objek XMLHttpRequest. Fungsi open() membuka *file proses.jsp*. *File* ini dapat diganti dengan teknologi bahasa pemrograman web lainnya seperti php, asp atau asp.Net.

Pada kode di atas juga dikirimkan *user id* yang diinput *user* menggunakan *method* GET, dimana semua informasi dikirimkan lewat URL.

```
function handleResponse() {
    if(http.readyState == 4 && http.status == 200){
        // Text returned FROM the JSP script
        var response = http.responseText;
        if(response) {
            // UPDATE ajaxTest content
            document.getElementById("ajaxTest").innerHTML =
response;
        }
    }
}
```

Kemudian akan dieksekusi kode program di atas. Fungsi di atas bertugas untuk menangani *response* dari *server*, kemudian melakukan *update* pada halaman web. Bagian halaman web yang di-*update* adalah elemen web dengan *id* "ajaxTest".

10. AJAX Framework

Untuk memudahkan penggunaan AJAX pada aplikasi web khususnya aplikasi web berbasis Java (JSP dan Servlet), maka saat ini sudah banyak *framework* atau *library* AJAX yang siap pakai dan diintegrasikan pada aplikasi web yang sudah ada.

Framework Echo2 misalnya yang dapat di-*download* dari <http://www.nextapp.com/platform/echo2/echo/>. Echo2 merupakan *platform* untuk pengembangan *rich client web-base application*. Aplikasi yang dikembangkan berorientasi pada *component* dan menggunakan *event-driven API*.

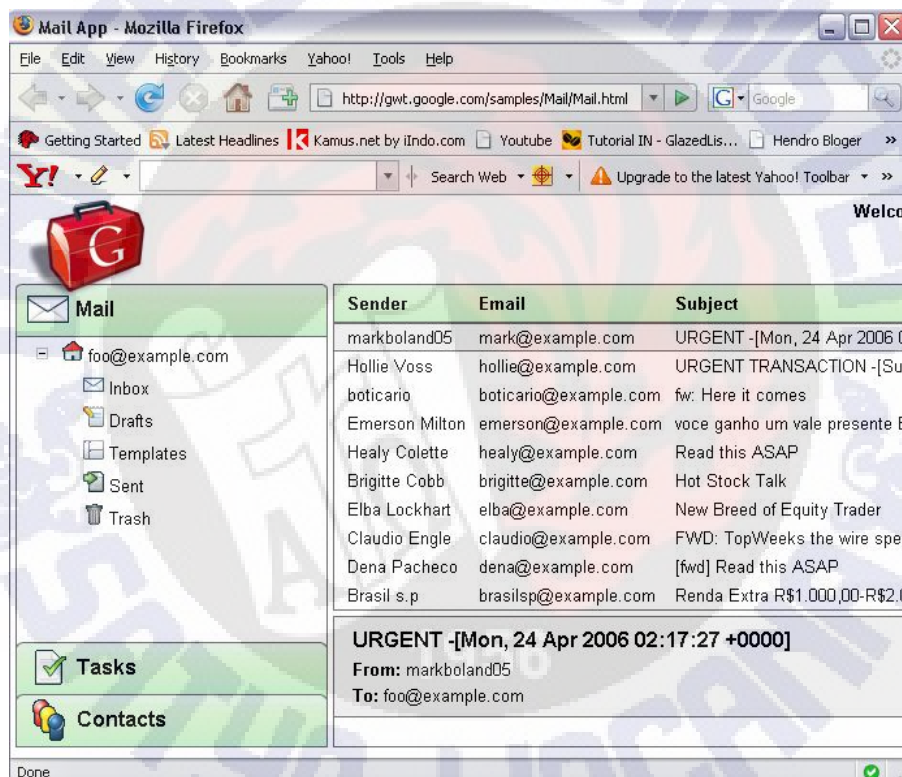


Gambar 6 Contoh Echo2

Selanjutnya ada DWR yang dapat di-download secara gratis dari <http://getahead.org/dwr>. Sama seperti Echo2, DWR merupakan *framework* AJAX berbasis Java. DWR memungkinkan JavaScript yang ada pada aplikasi *client* dalam hal ini *browser* untuk berinteraksi dengan Java yang berada di sisi *server* untuk memanipulasi konten pada halaman web.

DWR adalah sebuah *library* RPC (*Remote Procedure Call*) yang memudahkan untuk memanggil fungsi-fungsi Java langsung dari JavaScript dan sebaliknya memudahkan memanggil fungsi-fungsi JavaScript dari Java.

DWR memiliki banyak *features* seperti *call batching*, konversi otomatis struktur data antara Java dan JavaScript, penanganan kesalahan, dan integrasi dengan *framework-framework* Java lainnya.

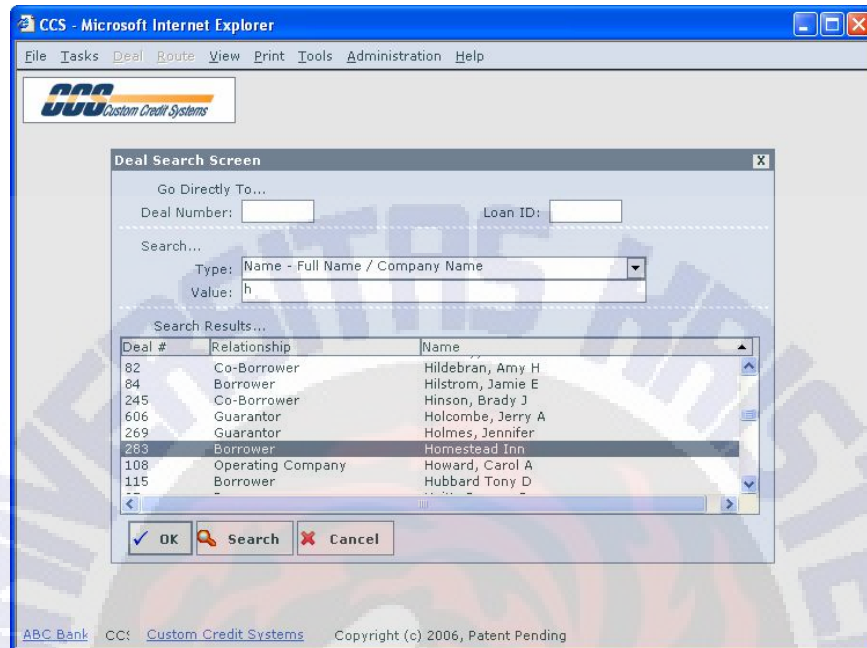


Gambar 7 Contoh aplikasi GWT

Framework Ajax yang lain adalah GWT (<http://code.google.com/webtoolkit/>). Google Web Toolkit (GWT) adalah *framework open source* berbasis Java yang memudahkan pengembangan aplikasi web yang menggunakan AJAX. Dengan GWT, *developer* dapat membuat aplikasi web berbasis AJAX menggunakan bahasa pemrograman Java seperti halnya mengembangkan aplikasi berbasis Java Swing.

Saat aplikasi di-deploy, *compiler* GWT akan mengubah aplikasi Java menjadi JavaScript dan HTML yang dapat berjalan pada berbagai jenis *browser*.

Berikutnya adalah ThinWire yang juga dapat digunakan secara gratis. ThinWire dapat di-download dari alamat <http://www.thinwire.com>.



Gambar 8 Contoh Aplikasi yang Menggunakan Thinwire yang Berjalan di Browser Internet Explorer

Thinwire adalah *framework open source* dengan lisensi LGPL untuk memudahkan pengembangan aplikasi web yang *responsive*, *expressive*, dan *interactive*. Thinwire dapat digunakan untuk aplikasi berskala apapun termasuk aplikasi *enterprise*. Thinwire memiliki komponen-komponen yang digunakan pada *layer view* dari aplikasi *enterprise*.

11. Simpulan

AJAX bukan merupakan teknologi baru dalam pengembangan aplikasi web. AJAX merupakan metode yang menggunakan standar pemrograman web yang sudah ada.

XMLHttpRequest merupakan objek JavaScript yang digunakan AJAX untuk berhubungan dengan *server* secara langsung. Dengan AJAX *request* dan *response* dilakukan secara *asynchronous*, yang menyebabkan aplikasi web lebih baik, cepat, dan lebih interaktif serta lebih *user friendly*.

AJAX dapat diimplementasikan bersama berbagai bahasa pemrograman web seperti Java (JSP), PHP, ASP, dan ASP.Net.

12. Daftar Pustaka

- [1] *Ajax Introduction*, Dokumentasi online: <http://www.w3schools.com/ajax>

- ajax_intro.asp, Tanggal akses: Desember 2006.
- [2] *Ajax Tutorial*, Dokumentasi online: <http://www.xul.fr/en-xml-ajax.html#ajax-in-depth>, Tanggal akses: Desember 2006.
 - [3] *The Complete List of Ajax Tools*, Dokumentasi online: http://www.realsoftwaredevelopment.com/2006/10/the_complete_li_1.html. Tanggal akses: Desember 2006.
 - [4] *Pengenalan Ajax dan Integrasinya dengan Struts*, Dokumentasi online: <http://www.benpinter.net/article.php?story=20060519053547529>. Tanggal akses: Desember 2006
 - [5] Crane, Dave, Eric Pascarello, Darren James, 2005, *Ajax In Action*, Manning Publications, Sound View
 - [6] Ballard, Phil, 2006, *Ajax in 10 Minutes*, Sams Publishing, Indianapolis.

