

AJAX (Asynchronous JavaScript and XML)

Andi Sunyoto

Dosen STMIK AMIKOM Yogyakarta

Abstraksi

AJAX is not a new programming language, but a technique for creating better, faster, and more interactive web applications. With AJAX, your JavaScript can communicate directly with the server, using the JavaScript XMLHttpRequest object. With this object, your JavaScript can trade data with a web server, without reloading the page. AJAX uses asynchronous data transfer (HTTP requests) between the browser and the web server, allowing web pages to request small bits of information from the server instead of whole pages. The AJAX technique makes Internet applications smaller, faster and more user-friendly. Web applications have many benefits over desktop applications; they can reach a larger audience, they are easier to install and support, and easier to develop. However, Internet applications are not always as "rich" and user-friendly as traditional desktop applications. With AJAX, Internet applications can be made richer and more user-friendly.

Keywords : picture, watermark, picture, javascript, protection

Pendahuluan

AJAX diperkenalkan oleh Jesse James Garret dari Adaptive Path pada tahun 2005. Ia mendeskripsikan bagaimana mengembangkan web yang berbeda dengan metode tradisional. Ia mempublikasikan sebuah artikel yang berjudul "AJAX: A New Approach to Web Applications". Pada artikelnya, Garret yakin bahwa aplikasi web dapat menutup jurang pemisah antara web dan aplikasi desktop.

Pengembangan web secara tradisional bekerja secara *synchronously*, antara aplikasi dan server, setiap kali melakukan link atau melakukan operasi "submit" pada form. Caranya, browser

mengirim data ke server, server merespons dan seluruh halaman akan di refresh.

Aplikasi web yang bekerja dengan AJAX bekerja secara *asynchronously*, yang berarti mengirim dan menerima data dari user ke server tanpa perlu me-load kembali seluruh halaman, melainkan hanya melakukan penggantian pada bagian web yang hendak diubah. Penggunaan AJAX mulai populer ketika digunakan oleh Google pada tahun 2005.

AJAX bukanlah bahasa pemrograman baru, tetapi merupakan teknik baru penggunaan standar yang telah ada. Dengan AJAX kita dapat menjadi lebih baik, cepat dan menambah unsur user-friendly dan interaktif pada aplikasi web kita. AJAX berbasiskan pada JavaScript dan request HTTP.

AJAX berbasiskan standar terbuka seperti:

- JavaScript
- XML
- HTML/XHTML
- CSS

Dengan menggunakan JavaScript AJAX dapat mengirim dan menerima data antara web server dan web browser. Teknik yang dimiliki AJAX akan bergantian bertukar data dan mere-load ulang seluruh halaman.

Dengan demikian, untuk mempelajari AJAX, harus dipahami pula konsep standar di atas. AJAX di-support sebagian besar browser populer sehingga aplikasi AJAX adalah aplikasi *cross-platform* dan *cross-browser*.

Melalui AJAX, JavaScript dapat dikomunikasikan secara langsung dengan server menggunakan obyek JavaScript XMLHttpRequest. Obyek JavaScript ini dapat *men-trade* data sebuah web server tanpa harus *me-reload (refresh)* halaman web.

AJAX menggunakan *asynchroneuse* data transfer (pada HTTP request) antara browser dan web server, yang memperbolehkan halaman web *me-request* bit yang kecil atau seluruh informasi dari

server. Teknik AJAX membuat aplikasi internet menjadi kecil, cepat dan lebih user-friendly.

AJAX adalah aplikasi web yang lebih baik. Aplikasi web menambah keuntungan dibanding aplikasi desktop:

- Dapat menjangkau pengguna yang luas
- Mudah diinstal
- Mudah dikembangkan
- Mudah dipelihara

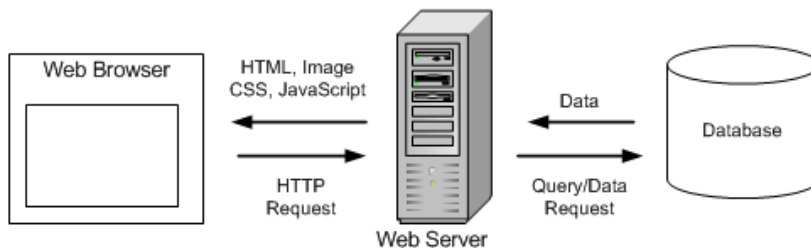
Seperti yang kita ketahui, aplikasi internet tidak selalu susah dan user friendly seperti aplikasi desktop. Dengan AJAX aplikasi internet semakin kecil, cepat dan mudah digunakan.

Pembahasan

Model-Model Web

Model Tradisional

Yang dimaksud dengan model tradisional di sini adalah model yang sering digunakan tanpa AJAX. Pertama, browser membuat sebuah HTTP request dikirim ke server, misalnya /index.html.



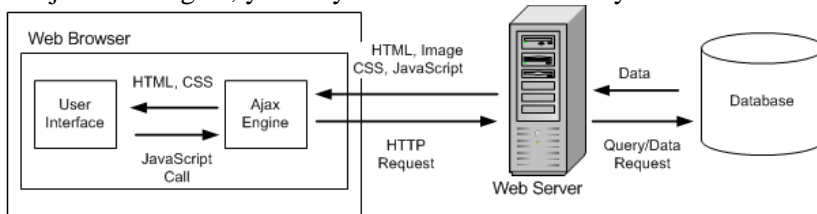
Gambar 1 Arsitektur model tradisional

Pada model ini, server mengirimkan response berisi seluruh halaman termasuk header, logo, navigasi, footer, dll. Ketika mengklik next maka akan menampilkan halaman baru lagi (artinya, header, logo, footer, navigasi dikirim ulang) dan seterusnya akan mengirimkan data halaman baru lagi setiap diminta request dari user. Halaman seperti ini tidak masalah ketika data yang ditampilkan tidak memerlukan response yang cepat. Namun, akan menjadi masalah jika

user menginginkan response yang cepat, misalnya ketika dipilih *drop-down* tertentu maka data yang ditampilkan berubah menurut nilai dari *drop-down*.

Model AJAX

Perlu diingat, AJAX akan berkomunikasi dengan server tanpa harus me-refresh semua halaman. Konsep ini berbeda dengan aplikasi web sistem tradisional. Dalam model AJAX, aksi dari sisi klien dibagi menjadi dua bagian, yaitu layer user interface dan layer AJAX.



Gambar 2 Arsitektur model AJAX

Ketika user mengklik sebuah link atau mengirimkan sebuah form maka input tersebut akan ditangani oleh layer AJAX dan diinteraksikan dengan server, kemudian meng-update user interface (UI). Jadi, dalam AJAX, interaksi UI secara logika terpisah dengan interaksi jaringan.

Pendekatan yang digunakan AJAX adalah bagaimana mengirimkan jumlah data yang kecil dari dan ke server atas request dari user. Model aplikasi web tradisional di mana browser akan merespons sendiri atas inialisasi request, memproses request dari web server. Dalam AJAX terdapat layer yang diberi nama "AJAX engine" untuk menangani komunikasi. "AJAX Engine" adalah sebuah obyek JavaScript atau function yang dipanggil ketika informasi diperlukan dari server.

Proses yang dilakukan mesin AJAX (*AJAX Engine*) adalah menerima respons dari server, melakukan parsing data dan melakukan sedikit perubahan untuk keperluan menampilkan informasi. Oleh karena proses mengirimkan sedikit informasi, jika dibanding dengan model aplikasi web tradisional maka user interface di-update lebih cepat dan user mampu melakukan pekerjaannya dengan cepat.

Beberapa poin penting untuk menggambarkan AJAX:

- Layer AJAX tidak memerlukan komunikasi dengan server (contohnya untuk validasi form karena dapat ditangani sepenuhnya oleh *client-side*).
- Oleh karena request antara layer AJAX dan server berupa bagian kecil dari informasi (tidak komplit satu halaman) maka sering digunakan untuk interaksi dengan database sehingga waktu render dan waktu pengiriman menjadi pendek.
- Layer UI secara langsung tergantung pada respons server sehingga user dapat melanjutkan interaksi dengan sebuah halaman selama aktivitas dikerjakan di background (*background process*). Berarti, untuk beberapa interaksi, waktu tunggu user hampir tidak ada.
- Komunikasi antarhalaman dan server tidak selama memerlukan AJAX untuk mengubah perubahan UI. Contoh, beberapa aplikasi menggunakan AJAX untuk notifikasi dengan halaman, tetapi tidak melakukan apa pun terhadap response dari server.

Teknologi di Balik AJAX

Teknologi AJAX di dalamnya meliputi beberapa komponen, yaitu:



Gambar 3 Teknologi di balik AJAX

- XHTML (*Extensible HyperText Markup Language*) adalah bahasa markup seperti HTML, tetapi dengan gaya bahasa lebih baik. XHTML digunakan membuat halaman web dan dokumen-dokumen lain yang dapat ditampilkan dalam browser. XHTML merupakan standard internasional dengan spesifikasi yang ditetapkan oleh W3C (*World Wide Web Consortium*). Versi terakhir saat tulisan ini dibuat adalah HTML 2.0.
- CSS (*Cascading Style Sheets*) adalah sebuah mekanisme sederhana untuk memberikan style (seperti font, warna, jarak spasi, dll) kepada dokumen web yang ditulis dalam HTML atau XML (termasuk beberapa variasi bahasa XML seperti XHTML dan SVG).
- JavaScript adalah bahasa scripting kecil, ringan, berorientasi-objek dan lintas platform. JavaScript tidak dapat berjalan dengan baik sebagai bahasa mandiri, melainkan dirancang untuk ditanamkan pada produk.
- DOM (*Document Object Model*) adalah sebuah API (*Application Program Interface*) untuk dokumen HTML dan XML. DOM menyediakan representasi dokumen secara terstruktur, dimungkinkan untuk mengubah isi dan presentasi visual. Pada dasarnya, DOM menghubungkan halaman web dengan script atau bahasa pemrograman.
- XML (*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan khusus. Keperluan utama XML adalah untuk pertukaran data antarsistem yang beraneka ragam.
- XSLT (*Extensible Stylesheet Language Transformations*) adalah sebuah bahasa berbasis-XML untuk transformasi dokumen XML. Walaupun proses merujuk pada transformasi, dokumen asli tidak berubah melainkan dokumen XML baru dibuat dengan basis isi dokumen yang sudah ada. XSLT

biasanya digunakan untuk mengubah skema XML ke halaman web atau dokumen PDF.

- Objek XMLHttpRequest berkemampuan melakukan pertukaran data secara asinkron dengan web server. AJAX menggunakan obyek XMLHttpRequest untuk melakukan pertukaran data dengan web server.
- JSON (*JavaScript Object Notation*) yaitu format pertukaran data komputer yang ringan dan mudah. Keuntungan JSON dibandingkan dengan XML adalah pada proses penerjemahan data menggunakan Javascript. Javascript dapat menerjemahkan JSON menggunakan built-in procedure eval().

Dalam kenyataannya, semua teknologi dapat digunakan untuk AJAX, tetapi hanya tiga yang dibutuhkan: HTML/XHTML, DOM dan JavaScript. XHTML ternyata diperlukan untuk menampilkan informasi, sedangkan DOM diperlukan untuk halaman XHTML tanpa di-reload ulang.

Pengguna AJAX

Beberapa web komersial sudah menggunakan teknologi AJAX untuk menaikkan pelayanan terhadap user. Berikut beberapa contoh web yang menggunakan AJAX.

- Google Suggest: <http://www.google.com/webhp?complete=1>
- Gmail: <http://www.gmail.com>
- Google Maps: <http://maps.google.com>
- Yahoo! News: <http://news.yahoo.com>
- Bitflux Blog: <http://blog.bitflux.ch>
- Dll

Keistimewaan AJAX

AJAX sebagai sebagai teknik pemrograman baru memiliki beberapa keistimewaan. Keistimewaan AJAX antara lain:

- Membuat permintaan kepada server tanpa memuat kembali (*reload*) halaman.

- Mengurai (*parse*) dan bekerja dengan dokumen XML dan atau JSON.
 - Data yang dikirim sedikit sehingga menghemat bandwidth dan mempercepat koneksi.
 - Proses dilakukan di belakang layar.
 - Banyak didukung oleh browser-browser modern yang populer.
 - Aplikasi yang dibangun semakin interaktif dan dinamis.
- Penggunaan AJAX akan mendatangkan beberapa keuntungan.

Keuntungan tersebut antara lain.

- *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

Contoh penerapan AJAX

Berikut contoh penerapan AJAX. Pada penerapan ini ada dua file yang dibutuhkan, yaitu file HTML dan file PHP. Isi file JavaScript adalah sebagai berikut (time.html):

```
<html>
<body>
<script type="text/javascript">
function AJAXFunction()
{
var xmlHttp;
try
{
// Firefox, Opera 8.0+, Safari
xmlHttp=new XMLHttpRequest();
}
catch (e)
{
// Internet Explorer
```



```
try
{
xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
try
{
xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
}
catch (e)
{
alert("Your browser does not support AJAX!");
return false;
}
}
}
xmlHttp.onreadystatechange=function()
{
if(xmlHttp.readyState==4)
{
document.myForm.time.value=xmlHttp.responseText;
}
}
xmlHttp.open("GET","time.php",true);
xmlHttp.send(null);
}
</script>
<form name="myForm">
Name: <input type="text"
onkeyup="AJAXFunction();" name="username" />
Time: <input type="text" name="time" />
</form>
</body>
```

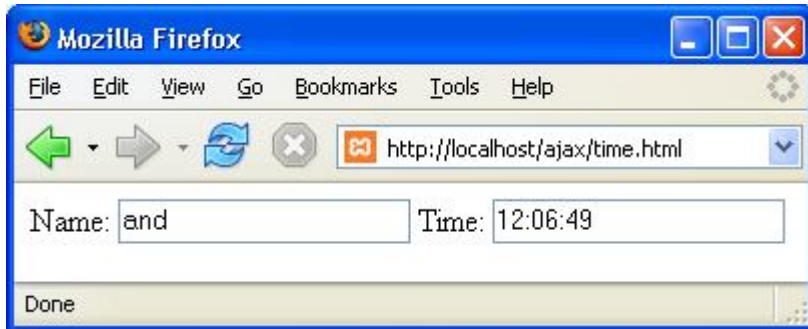
```
</html>
```

Sebelum menjalankan skrip di atas, perlu disiapkan dahulu skrip *server side*-nya. Skripnya adalah sebagai berikut (time.php):

```
<?  
echo date('H:m:s');  

```

Jalankan file “time.htm”, kemudian ketikkan username. Dengan demikian, setiap mengetikkan huruf di user name, jam langsung ter-update. Output program ketika dijalankan adalah sebagai berikut:



Gambar 4 Tampilan time dengan AJAX

Penutup

JavaScript dibutuhkan untuk komunikasi *client-server*, sementara manipulasi DOM untuk meng-update halaman web. Teknologi lain digunakan agar AJAX lebih bagus, tetapi bukan merupakan hal yang pokok. Untuk pembuatan aplikasi agar lebih *powerful* maka AJAX dapat digabung dengan aplikasi *server-side programming* seperti PHP, ASP, Java servlet atau .NET.

Daftar Pustaka

Sunyoto, Andi., “AJAX: Membangun Web dengan Teknologi Asynchronous JavaScript & XML”., Penerbit Andi., 2007

- Castagnetto, Jesus., et.al., "Profesional PHP Programming", Wrox Press, 1999.
- Dubois, Paul., "MySQL: The Definitive Guide to Using, Programming, and Administering MySQL 4.1 and 5.0 Third Edition, 2005.
- Eichorn, Joshua., "Understanding AJAX: Using JavaScript to Create Rich Internet Applications", Prentice Hall, 2006.
- Heilmann, Christian., "Beginning JavaScript with DOM Scripting and Ajax: From Novice to Professional", Apress, 2006.
- Langerstrom, Randles, Larry., "Programming the Web using XHTML and JavaScript", McGraw-Hill, 2003.
- W3Schools, "Learn PHP, Learn AJAX, Learn XML, Learn JavaScript, Learn XML and XHTML DOM, Learn MySQL", www.w3schools.com, 2007.
- Woychowsky, Edmond., "Ajax: Creating Web Pages with Asynchronous JavaScript and XML", Prentice Hall, 2006.