

# TELAAH AJAX UNTUK MENGEJAR KETERTINGGALAN APLIKASI WEB TERHADAP APLIKASI DESKTOP

Abdul Kadir  
Program Studi Teknik Informatika  
STMIK Jenderal A. Yani Yogyakarta

## ABSTRACT

*This paper focused on strengths and weaknesses of Ajax as a technique for implementing web applications. The goal is to motivate web application developers to use this technique in developing web applications by realizing its weaknesses. By using Ajax, gaps between desktop applications with interactive capability and conventional web applications which is not so responsive can be decreased.*

*The material of this paper was based on empirical experiences and literatures. It discusses several area of Ajax applications. Hopefully, web application developers can create or provide delightful features for customers. By supporting such applications, customers are not only satisfied but also very happy.*

**Keyword:** Ajax, web client, server

## INTISARI

Makalah ini menyoroti kelebihan dan kekurangan Ajax sebagai teknik untuk mengimplementasikan aplikasi web. Diharapkan dengan kajian ini, para perancang/pengembang aplikasi web mulai berpikir untuk menerapkan teknik ini dalam merancang/membangun aplikasi web, dengan mempertimbangkan segala kekurangannya. Dengan demikian, kesenjangan kemampuan aplikasi desktop yang begitu interaktif dan aplikasi web konvensional yang kurang begitu responsif bisa berkurang.

Kajian yang dilakukan, yang didasarkan pengalaman empiris dan literatur, mengupas berbagai area penerapan yang cocok kalau menggunakan Ajax. Harapan yang dikehendaki, perancang/pengembang bisa menciptakan atau menyediakan fasilitas yang memanjakan pemakai. Dengan demikian, pemakai tidak hanya sekedar puas tetapi sekaligus juga akan memberikan apresiasi yang tinggi.

**Kata-kunci:** Ajax, web, klien, server

## PENDAHULUAN

Ajax bukanlah teknologi baru dan juga bukan bahasa pemrograman. Ajax populer semenjak Garret (2005) menuliskan artikel berjudul "Ajax: A New Approach to Web Applications". Menurut Garret, julukan Ajax hanya untuk mempermudah komunikasi dengan kliennya disebabkan terlalu panjang kalau menyebutkan semua teknologi yang mendasarinya. Kenyataan yang terjadi, pada tahun 1989 Microsoft memang telah mengimplementasikan XMLHttpRequest, yang *notabene* adalah salah satu komponen dari Ajax, yang dikemas dalam ActiveX dan diterapkan pada Internet Explorer (Darie, dkk., 2006). Selain itu, JavaScript yang menjadi tulang punggung untuk menyusun mesin Ajax telah diperkenalkan pada tahun 1995 (Zakas, dkk., 2006).

Lantas, dimana keunggulan Ajax sebagai pendukung aplikasi Web? Hal inilah yang akan dibahas terlebih dulu, sebelum menuju ke pembahasan area yang cocok untuk menggunakan Ajax. Mengingat Ajax merupakan teknik yang belum sempurna, segala kekurangannya juga akan dibicarakan.

Hal ini dimaksudkan agar bisa menjadi wacana dalam memikirkan implementasi aplikasi web yang mampu mengantisipasi kelemahan-kelemahan tersebut. Selain itu, barangkali bisa memacu para pembuat *tool* untuk berpartisipasi dalam mengembangkan masa depan Ajax.

## TEKNOLOGI UNTUK APLIKASI WEB

Fondasi untuk membangun aplikasi web dimulai dengan penciptaan World Wide Web yang dikembangkan oleh Tim Berners-Lee pada tahun 1989 (Ragget, dkk., 1998). Saat itu dengan hanya menggunakan dokumen HTML, aplikasi web sudah merupakan terobosan besar, karena memungkinkan informasi dari beberapa tempat terhubung. Sayangnya, HTML hanya mendukung aplikasi yang bersifat statis atau tetap. Oleh karena itu, timbul keinginan untuk mengembangkan teknologi pendukung sehingga aplikasi dapat bersifat dinamis. Sebagai perwujudan maka muncul teknologi di sisi server (misalnya PHP dan ASP) dan teknologi di sisi klien (misalnya JavaScript dan VBScript). Dalam praktek, kelebihan kedua

teknologi tersebut digabung untuk mendapatkan hasil yang maksimal.

Namun, kenyataan yang terjadi, aplikasi seperti itu masih saja tertinggal jauh dibandingkan dengan aplikasi *desktop* (misalnya dengan menggunakan Visual Basic atau Delphi). Dimana kesenjangan? Tabel

**Tabel 1 Perbandingan aplikasi desktop dan aplikasi web konvensional**

Aspek	Aplikasi Desktop	Aplikasi Web Konvensional
Kebersahabatan ( <i>user-friendly</i> )	Sangat tinggi	Terbatas, karena masih ada kendala
Keinteraktifan	Sangat interaktif	Terbatas (misalnya bisa menangani validitas data, tetapi masih mengandalkan penyegaran halaman)
Waktu-tanggapan	Terasa cepat	Terasa lama, karena pemakai harus sering menunggu sampai halaman ditampilkan) dan pada saat seperti itu pemakai tidak bisa melakukan apa-apa

Akibat yang terjadi sebagai konsekuensi dari kekurangan-kekurangan yang terjadi pada aplikasi web, pemakai seringkali tidak puas; apalagi kalau mereka pernah merasakan dalam menggunakan aplikasi berbasis *desktop*. Kenyataan yang terjadi pada aplikasi web konvensional memang demikian. Ketika klien membutuhkan data yang berada di server, segala kegiatan pemakai menjadi terhenti sampai server memberikan tanggapan dan browser menyegarkan halaman.

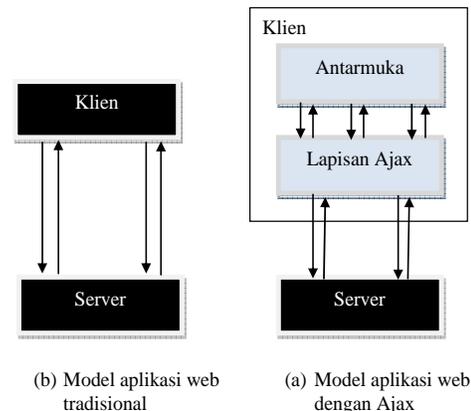
Fakta bahwa aplikasi web lebih disukai daripada aplikasi *desktop* untuk penyebaran informasi pada lokasi yang jauh tidak lepas dari manfaat yang diperoleh. Keuntungan aplikasi web (Darie, dkk., 2006):

1. Aplikasi web murah dan mudah untuk disebar. Tidak perlu biaya untuk orang TI yang bertanggung jawab memasang di setiap mesin. Aplikasi cukup ditempatkan di server dan klien cukup menginstal *browser*.
2. Aplikasi web murah dan mudah untuk ditingkatkan. Begitu aplikasi dipasang di server setelah mengalami perubahan, setiap orang bisa mengaksesnya.
3. Aplikasi web memiliki persyaratan yang fleksibel bagi pemakai. Pemakai bebas menggunakan *hardware* atau *software* apa saja.
4. Aplikasi web membuat lebih mudah untuk menggunakan penyimpan data yang terpusat. Setiap lokasi dapat mengakses data yang sama dari pusat data yang sama. Tidak perlu mengakses pada tempat penyimpanan yang berbeda-beda, sehingga mencegah operasi sinkronisasi data.

1 memperlihatkan perbedaan antara aplikasi web konvensional yang menggabungkan teknologi di sisi klien dan di sisi server dengan aplikasi *desktop*. Acuan perbandingan dilakukan dengan asumsi bahwa data berada dalam server dan pemakai menggunakan komputer yang bertindak sebagai klien.

### SOLUSI DENGAN AJAX

Ajax, dengan memanfaatkan minimal JavaScript, DOM, dan XMLHttpRequest, dapat digunakan untuk mengurangi kesenjangan antara aplikasi web konvensional dan aplikasi *desktop*. Gambar 1 menunjukkan skema perbedaan antara aplikasi web konvensional dan aplikasi web berbasis Ajax.



**Gambar 1 Perbedaan model aplikasi web tanpa dan dengan Ajax**

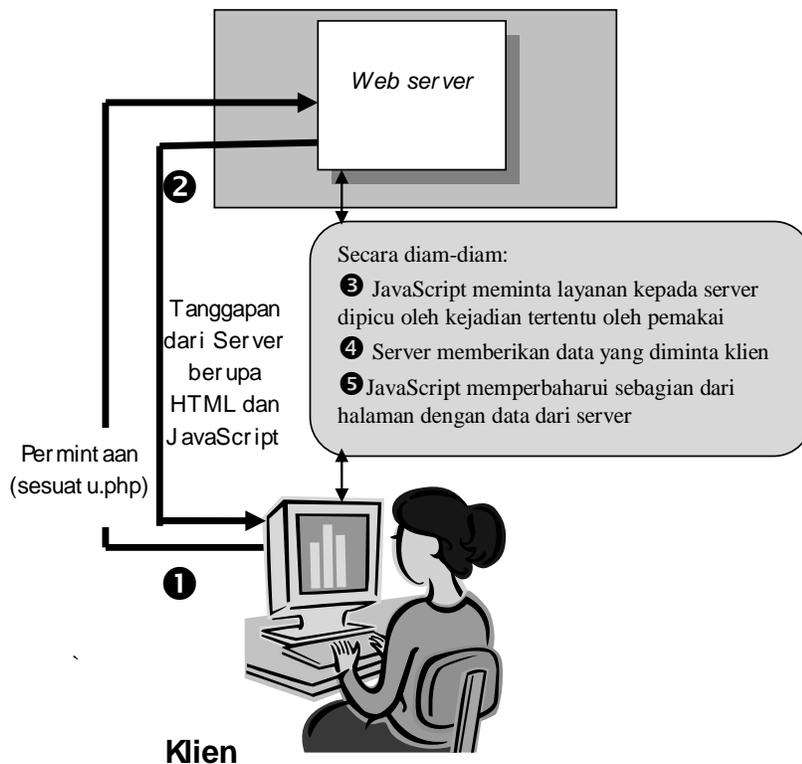
Nampak bahwa pada aplikasi web yang berbasis Ajax, terdapat bagian yang disebut lapisan Ajax (atau terkadang disebut mesin Ajax). Lapisan inilah yang mengatur komunikasi antara klien dan server.

Butir penting yang perlu diperhatikan dalam lapisan Ajax:

1. Lapisan Ajax meminta layanan ke server atau tidak, tergantung pada jenis layanan dari antarmuka. Sebagai contoh, validasi terhadap objek formulir tidak perlu melakukan permintaan ke server, tetapi

- ditangani di sisi klien. Namun, pemeriksaan keabsahan *password* mau tidak mau harus dilayangkan ke server karena harus berhubungan dengan *database*.
2. Sekalipun penambahan lapisan Ajax berkesan memperlambat proses (karena komunikasi ke server tidak lagi dilakukan secara langsung), kenyataannya sebaliknya. Hal ini disebabkan permintaan yang diakukan oleh lapisan Ajax menghasilkan informasi yang jauh lebih sedikit dibandingkan kalau yang diminta adalah untuk keseluruhan halaman.
  3. Lapisan Ajax bekerja dengan pendekatan asinkron. Asinkron berarti bahwa klien bisa meminta layanan dari server dan tidak perlu menunggu sampai server melayaninya. Dengan demikian, saat informasi dari server belum didapat, layanan kepada pemakai tetap bisa dilakukan.
  4. Komunikasi antara klien dan server tidak perlu mengubah keseluruhan antarmuka dalam halaman web. Hanya bagian tertentu yang terpengaruh. Hal seperti ini tidak bisa dilakukan pada aplikasi web yang tidak menggunakan Ajax.

Gambar 2 memberikan ilustrasi pemrosesan dengan Ajax.



Gambar 2 Proses dengan Ajax

### KEUNTUNGAN DAN KELEMAHAN AJAX

Berdasarkan sifat-sifat yang terkandung dalam Ajax, sejumlah keuntungan bisa diperoleh. Beberapa manfaat yang bisa dipetik (Darie, dkk., 2006):

1. Memungkinkan untuk membuat aplikasi web yang lebih baik dan lebih responsif.
2. Karena kepopulerannya, akan memicu pengembangan pola (pustaka atau kerangka) yang membuat para pengembang tidak harus membangun dari nol melainkan cukup menggunakan yang telah dibuat oleh orang lain.
3. Teknologi-teknologi yang digunakan untuk membangun aplikasi web selama ini tetap terpakai.
4. Keahlian para pengembang tetap bisa termanfaatkan.
5. Fitur-fitur Ajax terintegrasi dengan baik terhadap fungsionalitas yang disediakan browser (misalnya mengubah ukuran halaman dan navigasi halaman).

Namun, yang juga perlu disadari, Ajax juga menawarkan beberapa kelemahan. Antara lain:

1. Halaman menjadi tidak bisa di-*bookmark*.
2. Bagian dari aplikasi Ajax bisa jadi tidak bisa diindeks oleh mesin pencari.
3. Tombol Back pada browser menjadi tidak berfungsi seperti biasanya.
4. Aplikasi Ajax menjadi tidak berfungsi kalau klien mematikan JavaScript.
5. Browser yang digunakan oleh klien harus mendukung Ajax.

Butir 4 dan butir 5 perlu untuk diperhatikan. Suatu tantangan muncul di sini, "Bagaimana membuat aplikasi web yang bisa memungkinkan fungsi tetap berjalan walau terbatas kalau persoalan tersebut muncul?"

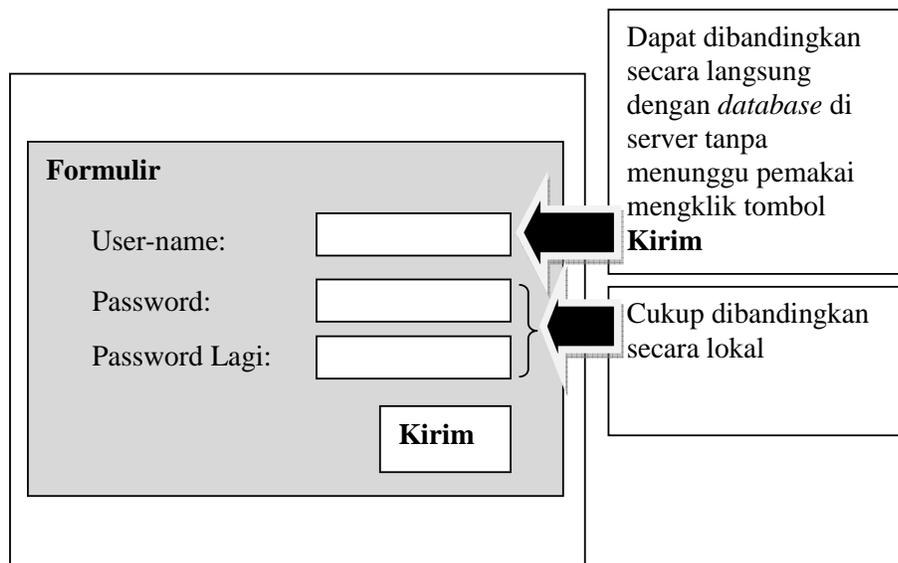
Hal lain yang juga penting untuk diperhatikan adalah masalah keamanan. Sebagai sesuatu yang baru diterapkan, keamanan menggunakan Ajax juga perlu mendapatkan fokus. Kemungkinan-kemungkinan serangan dan penanganan pada aplikasi web seperti ini dibahas oleh Wells (2007).

## AREA YANG COCOK UNTUK PENERAPAN AJAX

Apakah Ajax bisa diterapkan untuk semua aplikasi web? Tentu saja jawabannya bergantung pada jenis aplikasi yang dibuat. Tiga pedoman yang perlu diperhatikan dalam menggunakan Ajax:

1. Lalu lintas yang minimal. Ajax hanya mengirim dan menerima informasi dengan ukuran yang sekecil mungkin ke dan dari server. Informasi yang tidak relevan tidak diikutsertakan.
2. Mencegah pengunduhan halaman keseluruhan. Apabila ada bagian yang tidak perlu diubah dan cukup banyak, hindarkan untuk menampilkan kembali. Hanya bagian yang memang perlu diubah yang dimodifikasi.
3. Menghindari aplikasi web yang membuat pemakai kehilangan semua yang telah ia ketikkan di halaman web ketika ia mengklik suatu tombol dan sesuatu hal yang tidak valid terjadi.

Gambar 3 memberikan suatu ilustrasi penerapan Ajax untuk menangani registrasi anggota baru pada suatu *website*. Pada gambar tersebut bisa dilihat, kapan aplikasi perlu berkomunikasi dengan server dan kapan tidak.



**Gambar 3** Proses komunikasi asinkron Ajax pada formulir

Berikut adalah beberapa contoh yang membuat kehadiran Ajax diperlukan:

1. Pembeding-an suatu nilai kunci dengan isi suatu tabel dalam *database* (misalnya, NIP pegawai sudah ada atau belum, *user-*

- name* sudah digunakan oleh orang lain atau belum, *email* sudah terdaftar atau belum).
2. Penambahan data ke dalam *database* (cukup ditangani dengan menggunakan satu dokumen).
  3. Menyajikan informasi produk.
  4. Validasi data dengan data yang tercantum dalam *server*.
  5. Menampilkan daftar pilihan yang didasarkan data dalam *database*, yang ditentukan oleh suatu pilihan dalam halaman web (misalnya hanya menampilkan daftar kota yang ada di Jawa Tengah ketika pemakai memilih Jawa Tengah dari daftar provinsi).
  6. Melakukan komunikasi tertulis secara langsung (*chatting*).
  7. Memberikan saran ketika pemakai melakukan pengetikan.

Dengan menerapkan Ajax untuk menangani persoalan-persoalan seperti di atas, kesenjangan antara aplikasi web konvensional dan aplikasi *desktop* akan berkurang.

Hal yang juga menarik, aplikasi berbasis Ajax memungkinkan untuk memanfaatkan sumber dalam Internet. Sebagai contoh, dimungkinkan untuk memanfaatkan Google Maps atau Google Suggestion. Hasil eksperimen yang dilakukan penulis, Ajax juga bisa dimanfaatkan untuk mendapatkan data buku menurut ISBN dari berbagai sumber di Internet.

Sejauh ini, pengembangan *tool* atau *library* untuk aplikasi web dengan tujuan untuk mengurangi kesenjangan terhadap aplikasi *desktop* terus dilakukan. Beberapa contoh ditunjukkan berikut ini.

- *Tool* bernama *Script.aculo.us* (<http://script.aculo.us>), yang dibangun dengan memanfaatkan *tool* Prototype, menyediakan berbagai fitur interaktif seperti *click and drag* (memungkinkan sebuah objek ditarik dan diletakkan ke tempat lain), *sortable* (memungkinkan penukaran daftar item melalui *mouse*), dan *in place editor* (fasilitas yang memungkinkan suatu data bisa diedit

secara langsung melalui *field* masukan)

- SVG (*Scalable Vector Graphics*) yang merupakan rekomendasi dari World Wide Web Consortium (W3C) sejak Januari 2003 memungkinkan pembangkitan grafik dua dimensi secara *real-time*.
- XSLT (*Extensible Stylesheet Language Transformation*) memungkinkan pembuatan komponen grid dalam aplikasi web.

## PENUTUP

Kehadiran Ajax dalam upaya mengurangi kesenjangan antara aplikasi web konvensional dan aplikasi *desktop* perlu menjadi bahan pertimbangan bagi para perancang/pengembang aplikasi web. Segala kekurangan yang terjadi adalah fakta pada masa kini. Namun, tidak tertutup kemungkinan bahwa kekurangan-kekurangan yang masih ada akan tertutupi dengan teknologi-teknologi baru atau yang sedang dikembangkan. Bagaimanapun, rancangan besar dengan memikirkan hal-hal yang akan memanjakan pemakai akan membawa ke arah yang positif. Berpikir ke arah kepuasan pemakai akan membawa arah ke kesuksesan. Apalagi kalau bisa memanjakan pemakai.

## REFERENSI

- Darie, C.; Brinzarea, B.; Chereches-Tessa, F.; Bucica, M.; 2006; *Ajax and PHP Building Responsive Web Applications*; Packt Publishing Ltd.; Mumbai.
- Garret, J.J. ; 2005; *Ajax: A New Approach to Web Applications*; <http://www.adaptivepath.com/ideas/essays/archives/000385.php>; Diakses dari Internet pada tanggal 20 Januari 2008.
- Zakas, N.; McPeak, J.; Fawcett, J.; 2006; *Professional Ajax*; Wiley Publishing, Inc.; Indianapolis.
- Ragget, D.; Lam, J.; Alexander, I. F.; Kmiec, M.; 1998; *Raggets on HTML 4*; Addison Wesley.
- Wells, C.; 2007; *Securing Ajax Applications*; O'Reilly; Sebastopol.