Analisis dan Implementasi Validasi Form Berbasis Web dengan Menggunakan AJAX (Asynchronous JavaScript And XML)

Aceng Misbah
a.mishbah@yahoo.com
Pembimbing I : Khusnul Novianingsih, S.T, M.T
Pembimbing II : Irawan afrianto, S.T

Fakultas Teknik dan Ilmu Komputer, Jurusan Teknik Informatika
Universitas Komputer Indonesia

Abstrak

Validasi form registrasi merupakan proses pemeriksaan terhadap data input user yang dilakukan pada sisi klien dan pada sisi server. Untuk menjadikan proses validasi pada form registrasi lebih *responsif* dan *user friendly* maka diperlukan proses pengiriman data (*request http*) yang tidak mengganggu aktifitas user dan diperlukan proses penampilan status data secara *realtime*.

Untuk melakukan proses validasi pada sisi server yang bersifat *realtime* maka penulis menggunakan teknik AJAX (*Asynchronous JavaScript And XML*), untuk pengiriman data ke server akan ditangani oleh objek XMLHttp Requests, proses pengiriman data ini berlangsung secara asinkron dengan aktifitas user yang sedang berinteraksi dengan form validasi, sehingga user dapat terus berinteraksi dengan aplikasi sementara itu data input pada field form registrasi dikirimkan keserver.

Kata Kunci : *validasi, XMLHttp Request Objek, Asinkron.*

1. Pendahuluan

Latar Belakang Masalah

Validasi data adalah proses pemeriksaan terhadap data input apakah sudah sesuai aturan yang ditetapkan pada suatu sistem. Umumnya aplikasi berbasis web menempatkan validasi data pada sisi server (*server-side*), namun ada juga yang menempatkan validasi di sisi klien dengan menggunakan *client-side* scripting, misalnya menggunakan kode JavaScript .

Pada aplikasi web yang menempatkan proses validasi bersifat *server-side.* Kenyataan yang ada pada saat ini adalah apabila user memasukkan data input pada sebuah form, jika pada data tersebut terdapat data yang salah, kemudian user mengirimkan data tersebut ke server, maka balasan dari server adalah berupa sebuah halaman web berisi form kosong yang harus diisi ulang user mulai dari awal, hal ini akan dapat mengganggu user (tidak *user friendly*). Selain itu selama proses pengiriman data, proses validasi dan proses balasan dari server user harus menunggu sehingga user merasa aplikasi tersebut kurang responsif. Proses menunggu ini biasanya di tandai dengan halaman kosong (*blank page*) yang ditandai dengan munculnya *icon* jam pasir.

AJAX (Asynchronous JavaScript And XML) merupakan suatu teknik pengembangan web yang dapat diterapkan pada pengembangan aplikasi validasi form registrasi berbasis web, berkaitan dengan permasalahan di atas. Pada perkembangannya, istilah AJAX merujuk pada penggunaan beberapa teknik dalam pengembangan web, antara lain XHTML, CSS, DOM, XML atau XSLT, XMLHttpRequest dan JavaScript.

Pada aplikasi berbasis web dalam hal ini sebuah form registrasi menggunakan AJAX komunikasi dengan server akan ditangani oleh AJAX. Untuk proses *request* HTTP berupa pengiriman data dari klien ke server, dan proses *response* HTTP berupa pengiriman data balasan dari server akan ditangani oleh AJAX dengan menggunakan objek XMLHttp Requests. Proses *request* HTTP dan *response* HTTP terjadi secara asinkron (*asynchronous*) dengan proses tampilan pada halaman browser, sehingga user dapat terus berinteraksi dengan aplikasi sementara data input dikirimkan ke server.

Identifikasi Masalah

Berdasarkan latar belakang masalah diatas, penulis mengidentifikasi masalah sebagai berikut :

1. Bagaimana melakukan *request* HTTP asinkron dan menangani *respon* HTTP pada aplikasi form validasi berbasis web?
2. Bagaimana menampilkan hasil validasi pada sebuah form aplikasi berbasis web yang tidak mengganggu aktifitas user?
3. Bagaimana merancang dan mengimplementasikan aplikasi form registrasi yang divalidasi menggunakan AJAX?

Maksud dan Tujuan

Maksud dari pembuatan Tugas Akhir ini adalah untuk merancang dan mengimplementasikan validasi form berbasis web dengan menggunakan AJAX.

Tujuan dari pembuatan Tugas Akhir ini adalah sebagai berikut:

1. Aplikasi yang akan dibuat dapat dijalankan pada semua browser yang mendukung penggunaan JavaScript dengan pengaturan JavaScript diaktifkan.
2. Aplikasi yang akan dibuat dapat melakukan *request* asinkron untuk mengirimkan data yang akan divalidasi pada server-side, dan mampu untuk menangani *response* dari server.
3. Aplikasi yang akan dibuat diharapkan lebih *responsif* dan *user friendly*.
4. Dapat merancang dan mengimplementasikan sebuah aplikasi form validasi berbasis web dengan menggunakan AJAX.

Batasan Masalah

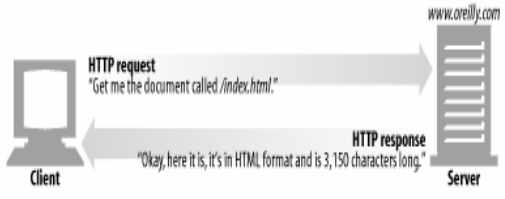
Pembatasan masalah dalam pembuatan tugas akhir ini adalah sebagai berikut:

1. Aplikasi ini menggunakan AJAX pada sisi klien, sementara pada sisi server menggunakan script PHP, dengan web server Apache dan database server MySQL.
2. Pada aplikasi ini akan melakukan proses validasi pada sisi klien dan proses validasi pada sisi server.
3. Pada aplikasi ini tidak melakukan proses penyimpanan data pada saat proses validasi berhasil tetapi hanya melakukan validasi (membandingkan) data input dengan yang telah disimpan pada *database (tabel user)*.
4. Browser yang digunakan adalah browser yang mendukung penggunaan JavaScript dan dengan pengaturan JavaScript diaktifkan.
5. Pada aplikasi ini tidak membahas dan menggunakan capcha.
6. Pada tugas akhir ini tidak menggunakan dan tidak membahas frameworks AJAX.

2. Landasan Teori

HTTP (HyperText Transfer Protocol)

HTTP adalah protokol yang di gunakan untuk mengirimkan halaman web, gambar, dan tipe data lainnya melalui internet dari *server* ke browser dan sebaliknya. HTTP terdiri dari dua bagian yaitu *request, dan response* seperti diperlihatkan pada gambar di bawah ini.



Gambar 1. Proses transaksi antara browser dengan server.

HTTP *request* merupakan pesan HTTP yang dikirimkan oleh client (browser) ke server, sedangkan HTTP *response* adalah pesan balasan dari server ke client.

Format HTTP request adalah sebagai berikut:

```
<request-line>
<header>
[<request-body>]
```

Pada baris pertama HTTP *request* adalah menunjukan tipe dari request, *resources* yang diakses dan versi dari HTTP yang digunakan, pada bagian *header* menunjukan informasi tambahan yang akan digunakan server, setelah header adalah *blank line,* yang akan diikuti dengan data tambahan yaitu body.

Di bawah ini adalah contoh dari HTTP request:

```
GET / HTTP/1.1
Host : www.yahoo.com
User-agent : Mozilla/5.0 (windows; u; windows NT 5.1; en –us;rv;1.7.6) Gecko/20050225 firefox/1.0.1
Connection : keep-alive
```

Format dari HTTP *Response* memiliki persamaan dengan http request, format http response seperti diperlihatkan dibawah ini.

```
<status-line>
<header>
<blank-line>
{<response-body>}
```

Perbedaan antara request HTTP dengan response HTTP terletak pada baris pertama yang berisi informasi status dari informasi *request*. *Status line* memberikan keterangan tentang *resource* yang diminta dengan menggunakan sebuah kode status. Contoh dari respon http adalah sebagai berikut:

```
HTTP / 1.1 200 OK
Date: Sat, 31 Dec 2005 23:45:59 GMT
Content-Type: text/ html;charset=ISO-8859-1
Content-Length: 122

<html>
<head>
<title>yahoo homepage</title>
</head>
<body>
< --   isi dari halaman web--- >
</body>
</html>
```

Pada contoh diatas status line memberikan sebuah http berstatus 200 dan pesan OK, pada *status line* selalu berisi kode status dan sebuah pesan, dibawah ini adalah kode status yang sering ditemukan.

- 200 (OK): *resource* di temukan dan dapat digunakan.
- 304 (NOT MODIFIED): *resource* belum dimodifikasi sejak terakhir *request.*
- 401 (UNAUTHORIZED) : klien tidak diizinkan menggunakan *resource.*
- 403 (FORBIDDEN): klien gagal dalam proses authorisasi, terjadi apabila user gagal pada prose login.
- 404 (NOT FOUND): resource tidak ditemukan.

XmlHTTPRequest object

XmlHtttpRequest object adalah suatu antarmuka yang dibuka oleh suatu *script engine* membolehkan scrip untuk menampilkan secara fungsional HTTP *client*, seperti pengiriman (submit) data atau proses pengambilan (download) data dari suatu web site [W3C,2006].

XmlHtttpRequest merupakan bagian yang sangat penting dari AJAX, XmlHTTPRequest dapat menggunakan JavaScript, Jscript, VBscript atau bahasa scripting yang lain untuk mentransfer dan memanipulasi data XML dari & ke web server dengan menggunakan HTTP. Selain data dalam format XML (eXtensible Markup Language), XmlHTTPRequest juga digunakan untuk *feching* format data lain seperti JSON bahkan sebuah plaint text, termasuk menyediakan data (data provided) dengan cara mengeksekusi *query* back-end database. Spesifikasi dari XmlHTTPRequest diatur oleh W3C.

HTML (HyperText Markup Language)

HTML merupakan bahasa markup yang digunakan untuk membuat halaman web yang akan ditampilkan pada halaman web browser.

PHP (HyperText Preprocessor)

PHP dikenal sebagai bahasa scripting yang menyatu dengan tag-tag HTML yang digunakan untuk membuat satu web site yang interaktif. PHP merupakan bahasa scripting yang bersifat server side, artinya script disimpan dan di eksekusi di server sedangkan hasilnya dikirim ke klien. Untuk menjalankan script yang berada di server, klien menggunakan web browser untuk meminta hasil dari proses yang terjadi di server.

JavaScript

JavaScript merupakan bahasa scripting, berupa kumpulan script yang di tempatkan atau di-*include* pada dokumen HTML. Penggunaan utama dari JavaScript pada pemrograman berbasis web adalah untuk menuliskan fungsi yang ditempatkan atau di-*include* dari file HTML yang akan berinteraksi dengan DOM.

DOM (Dokumen Object Model)

sDOM merupakan suatu API (*Aplication Programming Interface*) untuk dokumen HTML dan XML. DOM menggambarkan bagaimana dokumen HTML atau XML dianggap sebagai suatu objek. DOM menyediakan sebuah aplikasi pemrograman antarmuka untuk mengakses dan memodifikasi isi, susunan maupun style suatu dokumen.

XML (eXtensible Markup Language)

Penggunaan XML adalah untuk memfasilitasi pertukaran data atau informasi pada sistem yang berbeda platform, terutama untuk pertukaran data yang berkoneksi menggunakan internet.

AJAX (Asynchronous JavaScript And XML)

AJAX merupakan suatu teknik pengembangan web untuk menghasilkan aplikasi web lebih interaktif dengan memindahkan sebagian proses komputasi ke browser. Pada perkembangannya AJAX merujuk pada penggunaan beberapa teknik yang digunakan secara bersama, namun pada penggunaanya hanya teknik yang diperlukan saja yang benar-benar di gunakan antara lain :

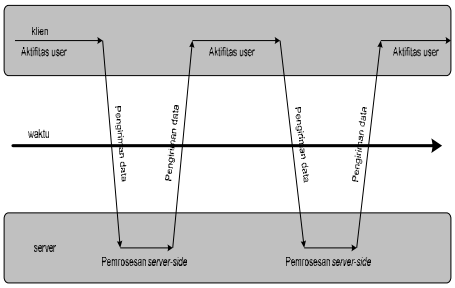
- HTML dan CSS digunakan untuk presentasi standar.
- DOM di gunakan untuk tampilan dinamis dan interaktif.
- XML digunakan sebagai format pertukaran dan manipulasi data.
- XMLHTTPRequest digunakan untuk request data asinkron dengan server.
- JavaScript digunakan untuk yang menyatukan semua teknik diatas.

AJAX memiliki beberapa keunggulan yaitu :

- Kompatibilitas terhadap JavaScript, XML, CSS, dan DOM
- Aktifitas user tidak terganggu
- Mengurangi atau menghilangkan waktu tunggu
- Pemakaian bandwidth yang minimal
- Konsep antarmuka pada aplikasi menyerupai aplikasi desktop

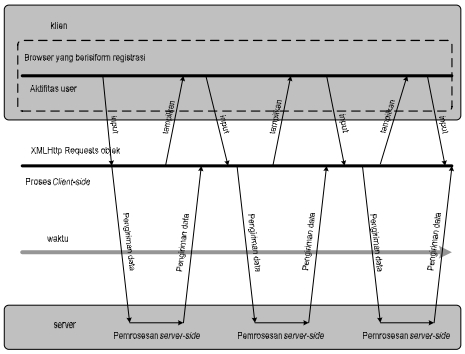
Perbandingan model aplikasi web

1. Model aplikasi web klasik bekerja seperti berikut:
   - Setiap aksi yang dilakukan user pada antarmuka memicu HTTP request ke web server.
   - Server melakukan beberapa proses seperti pengambilan data dan proses yang lainnya.
   - Kemudian server mengirimkan data berupa halaman HTML ke server.



Gambar 2. Gambar interaksi aplikasi web klasik

2. Model aplikasi web menggunakan AJAX bekerja dengan cara sebagai berikut:
   - Aksi user pada antarmuka memicu request fungsi JavaScript yang di kirim ke engine AJAX.
   - Engine AJAX melakukan proses pengiriman request HTTP ke web server.
   - Web server melakukan proses seperti pengambilan data dan proses lainnya.
   - Hasil proses dikirimkan ke klien dalam bentuk XML oleh AJAX.
   - Engine AJAX menerima respon web server kemudian mengirimkan hasil pengolahan data XML ke client.



Gambar 3. Gambar interaksi aplikasi web menggunakan AJAX

3. Perancangan Sistem

Pembuatan XMLHttp Requests object

Supaya aplikasi yang akan dibuat dapat dijalankan pada berbagai macam browser (*cross-browser*) maka pembuatan XMLHttp Requests objek adalah sebagai berikut:

```
function createXmlHttpRequestObject()      {
var xmlHttp;
 try          {
              xmlHttp = new XMLHttpRequest(); }
 catch(e)    {
       var XmlHttpVersions =
              new Array('MSXML2.XMLHTTP.6.0',
                    'MSXML2.XMLHTTP.5.0',
                    'MSXML2.XMLHTTP.4.0',
                    'MSXML2.XMLHTTP.3.0',
                    'MSXML2.XMLHTTP',
                    'Microsoft.XMLHTTP');
       for (var i=0; i<XmlHttpVersions.length && !xmlHttp; i++)
              {
              try          {
                           xmlHttp =
                           new ActiveXObject(XmlHttpVersions[i]);
                           }
              catch (e) {}
              }
       }
if (!xmlHttp)
       alert("Error creating the XMLHttpRequest object.");
 else
       return xmlHttp;
}
```

Penggunaan XMLHttp Requests object

Adapun urutan dalam menggunakan XMLHttp Requests untuk melakukan sebuah request asinkron adalah sebagai berikut:

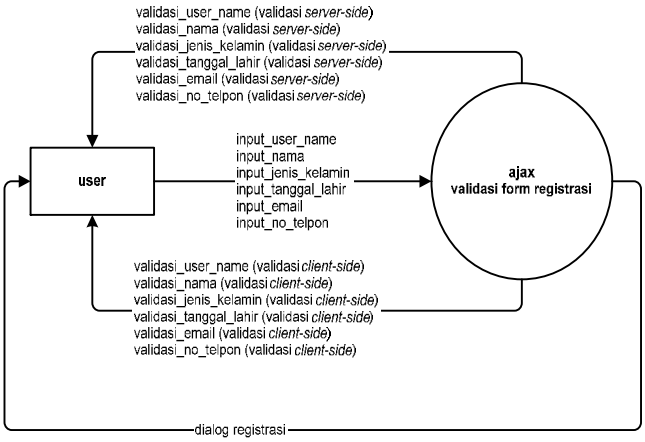
- Periksa XMLHttp Requests
- Coba koneksi ke server
- Atur properti XMLHttp *open* (parameter *async* diisi *true*)
- Atur properti XMLHttp Onreadystatechange
- Kirim request
- Tangani error apabila proses koneksi ke server gagal.

Sedangkan untuk menangani respon dari server adalah sebagai berikut:

- Periksa properti *readystate*
- Lanjutkan proses apabila *readyState* bernilai 4
- Periksa status http
- Lanjutkan proses apabila http status 200
- Terima respon
- Tentukan proses apa yang akan dilakukan pada respon tersebut

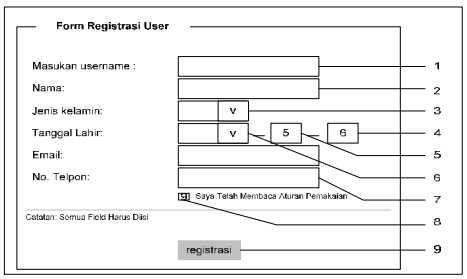
Perancangan Sistem

Sistem yang dibangun merupakan sistem pengembangan dari sistem validasi yang sudah ada yaitu menempatkan proses validasi *Client-side* dan juga menempatkannya pada *server-side,* untuk melakukan validasi *Client-side* digunakan script JavaScript, untuk melakukan validasi *Server-side* menggunakan teknik yang selama ini sudah digunakan yaitu menggunakan script PHP, selain itu digunakan juga teknik AJAX (Asynchronous JavaScript And XML) yang akan menjadikan aplikasi ini lebih responsif dan lebih interaktif.



Gambar 4. Diagram Kontek Ajax validasi form registrasi

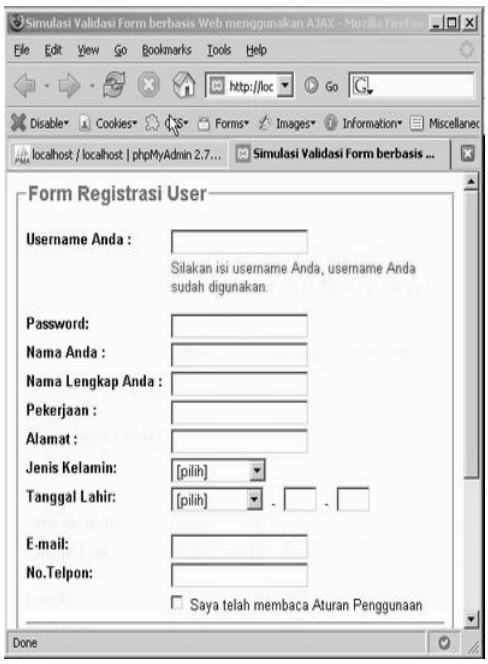
Perancangan Antar Muka



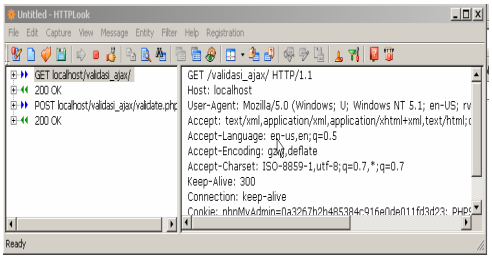
Gambar 5. Perancangan interface form validasi

4. Implementasi

Pengujian kompatibilitas web browser



Gambar 6. Field username mengalami lost on focus



Gambar 7. Data yang di capture pada proses field form lost on focus

5. Kesimpulan dan Saran

Kesimpulan

Berdasarkan hasil pengujian dapat ditarik kesimpulan bahwa aplikasi form berbasis web bebas dari kesalahan sintaks dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

- Aplikasi Form registrasi yang divalidasi menggunakan Ajax dapat berjalan pada browser yang mendukung penggunaan JavaScript.
- Proses validasi menggunakan *Ajax* tidak akan bekerja apabila option pada pegguaan fungsi javascript tidak diaktifkan.
- Proses request data secara asinkron pada form validasi menggunakan AJAX dapat dilakukan dengan menggunakan objek XMLHttp Requests, sedangkan proses menangani dapat dilakukan dengan mengatur fungsi dari handle respon, fungsi handle respon digunakan untuk memicu pesan kesalahan, yang sudah ada sebelumnya pada halaman browser untuk ditampilkan.
- Dengan menggunakan teknik Ajax dalam proses pengiriman data dan menangani tampilan pada halaman browser validasi form berjalan secara *realtime*, sehingga aplikasi ini terasa lebih *responsif* dan *user friendly*.

Saran

Untuk pengembangan selanjutnya pada validasi form registrasi yang menggunakan teknik AJAX penulis menyarankan untuk menyertakan penggunaan *capcha*, penggunaan password yang dienkripsi dan menambahkan lagi field–field yang akan divalidasi di server dan menyediakan fasilitas edit data user.

Untuk melihat keunggulan yang dimiliki teknik AJAX (Asynchronous JavaScript And XML) penulis lebih menyarankan untuk mengimplementasikan pada sebuah aplikasi yang memerlukan proses yang sangat interaktif yaitu pada aplikasi chat berbasis web.

DAFTAR PUSTAKA

[1] Budi Irawam, *Pengantar Jaringan Komputer*, Unikom, 2002.

[2] Cristian Darie, Bogdan Brinzarea, Filip Chereche – Tosa, Mihai Bucica, *AJAX and PHP building responsive web application*, PACKT publishing, Birmingham-mumbai, Maret, 2006.

[3] David Gourley and Brian Totty, Anshu Aggarwal, *HTTP: The Definitive Guide*, O'Reilly, September 2002

[4] Nicholas C. Zakas, Jeremy McPeakandJoe Fawcett, *Professional AJAX*, Wiley Publishing, Inc.

[5] Taryana S, *Internet dan intranet*, Unikom, 2001

[6] http://www.w3.org/TR/2006/WD-XMLHttpRequest/April 05,2005

[7] http://www.adaptivepath.com/publications/essays/February 18, 2005.

[8] http://www.Wikipedia.org/ AJAX (Asynchronous JavaScript And XML) (programming)/ index.php/ February, 18, 2005.