

Panduan Lengkap PHP Ajax jQuery

Oleh : Desrizal

Pengenalan Ajax
XMLHttpRequest
Format Data Respon Ajax
Pengenalan jQuery
Ajax jQuery

Sponsored By:



www.codingwear.com
baju kaosnya programmer

| Daftar isi | Halaman |
|--|---------|
| 1. Pengenalan Ajax | 4 |
| Apa Yang Harus Anda Ketahui | 5 |
| Apakah Ajax | 5 |
| XMLHttpRequest | 5 |
| Komunikasi dengan Server | 7 |
| 2. XMLHttpRequest Object | 13 |
| Properti atau Atribut..... | 14 |
| 3. Format Data Respon Ajax | 26 |
| JSON..... | 27 |
| AJAX | 40 |
| XML | 44 |
| 4. Pengenalan jQuery | 46 |
| Apa itu jQuery | 47 |
| Sintaks jQuery | 48 |
| jQuery Selector | 48 |
| jQuery Events | 50 |
| Efek-efek dengan jQuery | 51 |
| Manipulasi HTML dengan jQuery | 59 |
| Manipulasi CSS dengan jQuery | 63 |
| Web Tab Scroll Content (Contoh) | 67 |
| 5. jQuery Ajax | 70 |
| Fungsi-fungsi Ajax jQuery | 71 |
| Insert, Delete, Update dan Animasi Loading | 73 |

1

Pengenalan AJAX

Apa yang harus anda ketahui

Sebelum anda memulai belajar PHP AJAX anda harus sudah mempunyai pengetahuan dasar mengenai :

1. HTML
2. Javascript
3. PHP

Silahkan membaca buku atau dari internet untuk mempelajari HTML, Javascript dan PHP

Apakah AJAX

AJAX adalah singkatan dari Asynchronous JavaScript and XML.

AJAX, terdiri dari HTML, Javascript, DHTML dan DOM yang kemudian digabungkan dengan bahasa pemrograman web di sisi server seperti PHP dan ASP, sehingga membentuk suatu aplikasi berbasis web yang interaktif.

AJAX bukanlah bahasa pemrograman baru, tetapi adalah teknik baru untuk membuat aplikasi web lebih baik, lebih cepat dan lebih interaktif.

Dengan AJAX, Javascript dapat langsung berkomunikasi dengan server dengan menggunakan objek **XMLHttpRequest**. Dengan objek ini, javascript dapat melakukan transaksi data dengan server web, tanpa harus me-*reloading* halaman web tersebut secara keseluruhan.

Berikut adalah teknologi yang termasuk dalam aplikasi AJAX :

- HTML yang digunakan untuk membuat Web forms dan mengidentifikasi filed-field yang akan anda gunakan dalam aplikasi.
- JavaScript adalah kode inti untuk menjalankan aplikasi Ajax dan untuk membantu memfasilitasi komunikasi dengan aplikasi .
- DHTML, atau Dynamic HTML, membantu anda untuk membuat form atau web anda dinamis. Anda akan menggunakan <div>, dan elemen HTML dinamis lainnya.
- DOM, Document Object Model, akan digunakan (melalui kode JavaScript) untuk bekerja dengan kedua struktur dari HTML dan XML anda yang dalam beberapa kasus berasal dari server.

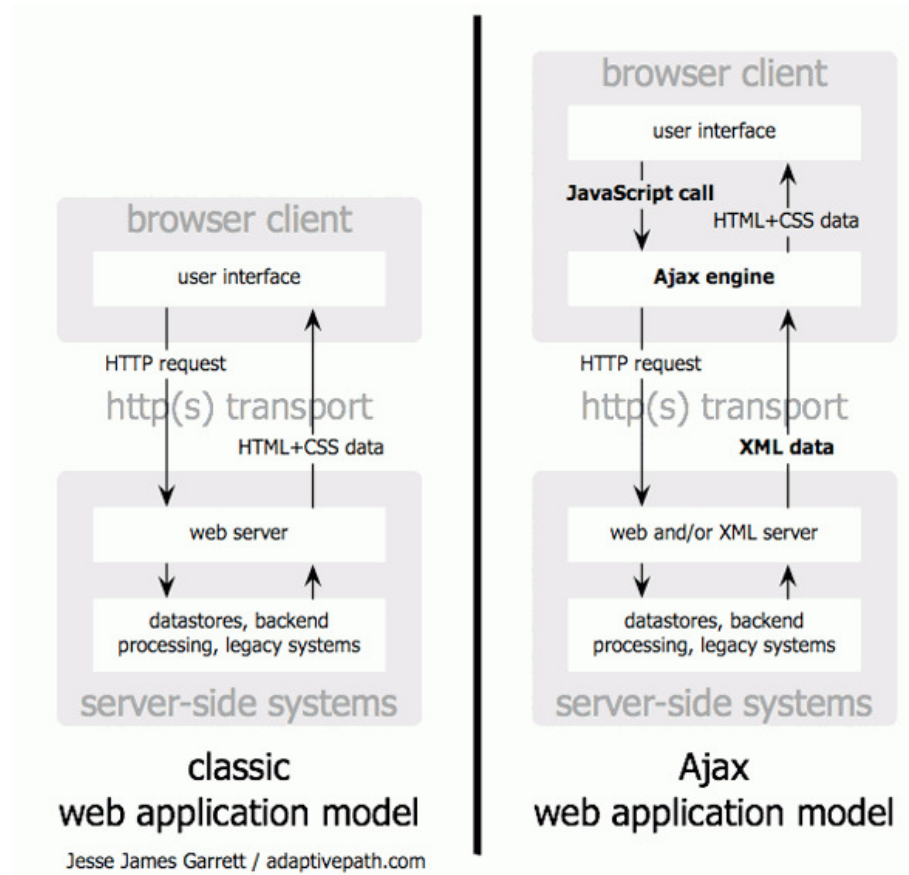
XMLHttpRequest

Objek pertama yang harus anda mengerti adalah XMLHttpRequest, mungkin bagi anda terdengar sesuatu yang baru. Ini adalah objek javascript dan membuatnya sederhana seperti ditunjukkan di kode 1 di bawah ini :

Kode 1. Membuat objek XMLHttpRequest.

```
<script language="javascript" type="text/javascript">
var xmlhttp = new XMLHttpRequest();
</script>
```

Untuk mendapatkan dan mengirim data dari/ke suatu database atau file di server menggunakan javascript tradisional, maka anda harus membuat HTML Form. Dan user harus mengklik tombol "submit" untuk mengirim/mendapatkan informasi, menunggu respon dari server, kemudian halaman yang baru berupa hasilnya akan di-load. Karena server selalu memberikan halaman baru setiap user tekan tombol submit, aplikasi web sederhana akan berjalan lambat dan akan kurang user-friendly.



Gambar 2. perbandingan web aplikasi tradisional dengan Ajax

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.

Kode JavaScript bahkan bisa mendapatkan data, melakukan perhitungan, dan mengirim permintaan lain, semua tanpa campur tangan pengguna! Ini adalah kekuatan dari XMLHttpRequest. Hal ini dapat bicara bolak-balik dengan server semua yang diinginkan, tanpa pernah tahu pengguna tentang apa yang sebenarnya terjadi. Hasilnya adalah, dinamis responsif, pengalaman yang sangat interaktif seperti aplikasi desktop, tapi dengan semua kekuatan Internet di belakangnya.

Objek XMLHttpRequest disupport hampir semua browser (Internet Explorer, Firefox, Chrome, Opera, dan Safari).

Untuk membuat objek XMLHttpRequest supaya bisa didukung oleh beberapa browser adalah sebagai berikut :

Kode 2. Membuat objek XMLHttpRequest untuk beberapa browser.

```
if (window.XMLHttpRequest)
{
    // kode untuk IE7+, Firefox, Chrome, Opera, Safari
    return new XMLHttpRequest();
}
if (window.ActiveXObject)
{
    // kode untuk IE6, IE5
    return new ActiveXObject("Microsoft.XMLHTTP");
}
```

Komunikasi Dengan Server

Oke, dari pada bingung-bingung dengan penjelasan di atas, mari kita lihat contoh berikut, bagaimana cara membuat objek ajax dan bagaimana caranya berkomunikasi dengan server. Ketika Ajax melakukan komunikasi dan transaksi data dengan server, berarti anda harus mempunyai suatu program web yang berjalan di sisi server. Ada banyak bahasa pemrograman yang berjalan di sisi server, seperti PHP, ASP dan sebagainya. Pada contoh berikut kita akan menggunakan bahasa PHP.

Pada contoh berikut kita akan membuat suatu program yang mengambil data yang kita letakkan di dalam database MySQL. Jadi misal kita punya tabelDataKaryawan di database MySQL.

```
-- Table structure for table `tabelDataKaryawan`
--
CREATE TABLE `tabeldatakaryawan` (
  `NIP` int(10) NOT NULL,
  `Nama` varchar(50) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `Alamat` tinytext NOT NULL,
  PRIMARY KEY (`NIP`)
)
```

Berikut adalah isi dari tabelDataKaryawan tersebut

| NIP | Nama | Email | Alamat |
|--------|--------------|---------------|--|
| 889456 | Desrizal | drz@fmi.com | Street 11C no. 332, Tembagapura, Papua |
| 889457 | Nurmi Yulita | nurmi@fmi.com | Teras Estate No. 234, Tembagapura, Papua |
| 896543 | Budi Hartono | budi@fmi.com | Barak Q Ridge camp |

Lalu kita membuat halaman HTML untuk menampilkan data tersebut, menggunakan semacam drop-down box, yang kalau diklik maka akan mengambil data di MySQL dan menampilkannya di halaman karyawan.html tanpa harus me-reloading keseluruhan halaman

Kode 3. datakaryawan.html

```
<html>
<head>
<script>
var ajaxku;
function ambildata(nip){
    ajaxku = buatajax();
    var url="ambildata.php";
    url=url+"?q="+nip;
    url=url+"&sid="+Math.random();
    ajaxku.onreadystatechange=stateChanged;
    ajaxku.open("GET",url,true);
    ajaxku.send(null);
}

function buatajax(){
    if (window.XMLHttpRequest){
        return new XMLHttpRequest();
    }
    if (window.ActiveXObject){
        return new ActiveXObject("Microsoft.XMLHTTP");
    }
    return null;
}

function stateChanged(){
var data;
    if (ajaxku.readyState==4){
        data=ajaxku.responseText;
        if(data.length>0){
            document.getElementById("alamat").value = data
        }else{
            document.getElementById("alamat").value = "";
        }
    }
}

</script>
</head>
<body>
Nama :
```



```
<select size="1" name="karyawan" id="karyawan"
onchange=ambildata(this.value)>
<option selected>--Pilih Karyawan--</option>
<option value="889456">Desrizal</option>
<option value="889457">Nurmi Yulita</option>
<option value="896543">Budi Hartono</option>
</select>
</p>
<p>
Alamat : <textarea rows="4" id="alamat" name="alamat"
cols="42"></textarea>

</p>
Pada contoh berikut anda akan memilih nama dan mengambil data alamat di
database<br>
berdasarkan nama tersebut dan menampilkannya di halaman ini tanpa harus
me-reload<br>
keseluruhan halaman
</body>
</html>
```

Pada kode di atas, pada baris kode berikut :

```
var url="ambildata.php";
```

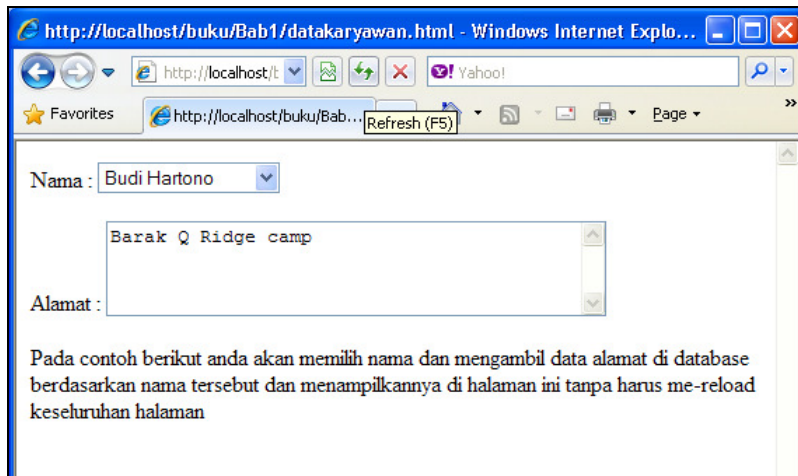
Anda akan melakukan transaksi data dengan server dengan bantuan program PHP yang bernama **ambildata.php**, untuk itu mari kita buat kode untuk file **ambildata.php**.

Kode 4. ambildata.php

```
<?php
mysql_connect("localhost","root","");
mysql_select_db("test");

$nip = $_GET['q'];
if($nip){
    $query = mysql_query("select alamat from tabelDataKaryawan where
nip=$nip");
    while($d = mysql_fetch_array($query)){
        echo $d['alamat'];
    }
}
?>
```

Berikut tampilan dari **datakaryawan.html**



Gambar 2. Tampilan datakaryawan.html di browser

Coba anda pilih pada drop-down box nama, maka otomatis akan tampil alamat sesuai nama yang teradapat di database.

Berikut penjelasan untuk kode pada **datakaryawan.html**

```
<body>
Nama :
<select size="1" name="karyawan" id="karyawan"
onchange=ambildata(this.value)>
<option selected>--Pilih Karyawan--</option>
<option value="889456">Desrizal</option>
<option value="889457">Nurmi Yulita</option>
<option value="896543">Budi Hartono</option>
</select>
</p>
<p>
Alamat : <textarea rows="4" id="alamat" name="alamat"
cols="42"></textarea>
```

Kode di atas kita membuat form HTML biasa, di mana kita akan membuat drop-down box nama-nama karyawan. Pada elemen `<select>` terdapat kode `onchange=ambildata(this.value)`, yang artinya apabila ada event perubahan terhadap combo box, dengan cara memilih salah satu option, maka segera eksekusi fungsi `ambildata(nip)` yang terdapat di javascript, dan kode `this.value` berfungsi untuk mengambil nilai dari option yang dipilih.

Kemudian, mari kita lihat kode javascript.

```
function ambildata(nip){
    ajaxku = buatajax();
    var url="ambildata.php";
    url=url+"?q="+nip;
    url=url+"&sid="+Math.random();
    ajaxku.onreadystatechange=stateChanged;
    ajaxku.open("GET",url,true);
    ajaxku.send(null);
}
```

Fungsi `ambildata(nip)` akan dieksekusi setiap user memilih nama karyawan pada combobox.

- `ajaxku = buatajax()`, kita membuat objek `XMLHttpRequest` dengan cara menjalankan fungsi `buatajax()`, di mana pada fungsi tersebut terdapat kode untuk membuat objek `XMLHttpRequest`.
- `var url="ambildata.php"`, Kita meng-set file PHP yang akan bekerja di sisi server.
- `url=url+"?q="+nip`, kita memasukkan nilai `nip`, yang diambil dari drop-down box ke dalam variabel `q`
- `url=url+"&sid="+Math.random()`, kita memasukan angka random ke dalam variabel `sid`, hal ini berguna untuk mencegah server menggunakan chached file atau mencegah data yang diberikan adalah data yang lama.
- `ajaxku.onreadystatechange=stateChanged`, setiap terjadi perubahan terhadap properti `readyState`, akan segera menjalankan fungsi `stateChanged()`.
- `ajaxku.open("GET",url,true)`, membuka objek `XMLHttpRequest` dengan metode GET dan URL yang akan melakukan action di sisi server.
- `ajaxku.send(null)`, mengirim HTTP request ke server.

Fungsi `buatajax()` berguna untuk membuat `XMLHttpRequest`, penjelasannya bisa anda lihat di kode 2. Membuat objek `XMLHttpRequest` untuk beberapa browser.

```
function stateChanged(){
var data;
  if (ajaxku.readyState==4){
    data=ajaxku.responseText;
    if(data.length>0){
      document.getElementById("alamat").value = data
    }else{
      document.getElementById("alamat").value = "";
    }
  }
}
```

- Fungsi `stateChanged()` berguna untuk menangani jika terjadi perubahan properti `readyState`, coba lihat kode `ajaxku.onreadystatechange=stateChanged`.
- `ajaxku.readyState==4`, Jika nilai properti `readyState` adalah sama dengan 4 (4 artinya adalah respon dari server sudah komplet), maka kita mengambil teks/output yang dihasilkan oleh program PHP (`ambildata.php`)
Jika kita lihat output yang dihasilkan program PHP `ambildata.php` adalah pada baris kode `echo $d['alamat'];`
- `document.getElementById("alamat").value = data`, Lalu nilai atau output yang dihasilkan dari `ambildata.php` tersebut, kita masukkan ke dalam elemen `textarea` yang mempunyai id "alamat".

Penjelasan untuk kode `ambildata.php` :

- Melakukan koneksi ke database MySQL
- Mengambil nilai variabel `q` dengan metode GET, karena ajax mengirim variabel `q` dengan metode GET, lalu memasukkannya ke dalam variabel `$nip`.

- Melakukan query untuk mengambil alamat yang sesuai nip.
- Lalu memberikan output alamat yang didapat dengan menggunakan echo .

2

XMLHttpRequest Object

Pada Bab ini kita akan mengenal lebih lanjut dengan objek XMLHttpRequest. Sebagai sebuah objek, maka XMLHttpRequest memiliki properti-properti dan juga metode metode di dalamnya.

Properti atau Atribut

Berikut adalah Properti dari objek XMLHttpRequest

| Property | Description |
|--------------|--|
| constructor | Menghasilkan referensi ke <i>constructor</i> suatu objek. |
| readyState | Mengambil <i>state</i> saat ini dari suatu <i>request</i> . |
| responseBody | Mengambil <i>response body</i> sebagai suatu array. |
| responseText | Mengambil <i>response body</i> sebagai suatu string. |
| responseXML | Mengambil <i>response body</i> sebagai suatu <i>XML Document Object Model (DOM) object</i> . |
| status | Mengambil kode status HTTP dari suatu <i>request</i> . |
| statusText | Mengambil status friendly HTTP dari suatu request. |
| timeout | Mendapatkan atau men-set nilai time-out. |

Berikut adalah Events dari objek XMLHttpRequest :

| Event | Description |
|--------------------|--|
| onreadystatechange | Men-set atau mengambil event handler untuk suatu request asinkron. |
| ontimeout | Akan terpanggil ketika ada eror timeout. |

Berikut adalah Metode dari objek XMLHttpRequest :

| Method | Description |
|-----------------------|--|
| abort | Meng-cancel <i>HTTP request</i> . |
| getAllResponseHeaders | Menghasilkan daftar komplet dari respon <i>headers</i> . |
| getResponseHeader | Menghasilkan respon header yang dispesifikkan. |
| open | Meng-assign metode, URL tujuan , atribut lainnya dari request yang akan dikirim. |
| send | Mengirim <i>HTTP request</i> ke server dan menerima suatu respons. |
| setRequestHeader | Menambah <i>HTTP headers</i> yang disesuaikan ke request. |

readyState

Syntax

```
object.readyState == nState
```

Nilai

| | |
|--------|--|
| nState | Nilai dari readyState berupa integer, yaitu : 0 (Uninitialized) Objek sudah dibuat, tapi belum diinisialisasi (metode open belum dipanggil). 1 (Open) Objek sudah dibuat, tetapi metode send belum dipanggil. 2 (Sent) Metode send sudah dipanggil. responseText dan responseBody belum bisa dipanggil. 3 (Receiving) Beberapa data sudah diterima. responseText dan responseBody belum bisa dipanggil. 4 (Loaded) Semua data sudah diterima. responseText dan responseBody sudah bisa dipanggil. |
|--------|--|

readyState bersifat read-only, tidak memiliki nilai *default*.

Catatan

Anda tidak bisa memanggil properti **responseBody** dan **responseText** untuk mendapat hasil dari suatu request yang kita kirim. Jika memanggil **responseBody** atau **responseText** akan menghasilkan error, karena respons belum diterima secara keseluruhan. Kita harus menunggu sampai semua data diterima (readyState == 4). **readyState** diperkenalkan di Internet Explorer 7.

Contoh :

```
function stateChanged(){  
var data;  
    if (ajaxku.readyState==4){  
        data=ajaxku.responseText;  
    }  
}
```

status

Syntax

```
object.status == nStatus
```

Nilai

| nStatus | Kode status HTTP yang diterima berupa Integer. |
|---------|--|
| Number | Description |
| 100 | Continue |
| 101 | Switching protocols |
| 200 | OK |
| 201 | Created |
| 202 | Accepted |
| 203 | Non-Authoritative Information |
| 204 | No Content |
| 205 | Reset Content |
| 206 | Partial Content |
| 300 | Multiple Choices |
| 301 | Moved Permanently |
| 302 | Found |
| 303 | See Other |
| 304 | Not Modified |
| 305 | Use Proxy |
| 307 | Temporary Redirect |
| 400 | Bad Request |
| 401 | Unauthorized |
| 402 | Payment Required |
| 403 | Forbidden |
| 404 | Not Found |
| 405 | Method Not Allowed |
| 406 | Not Acceptable |
| 407 | Proxy Authentication Required |
| 408 | Request Timeout |

| | |
|-----|------------------------------|
| 409 | Conflict |
| 410 | Gone |
| 411 | Length Required |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 414 | Request-URI Too Long |
| 415 | Unsupported Media Type |
| 416 | Requested Range Not Suitable |
| 417 | Expectation Failed |
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 502 | Bad Gateway |
| 503 | Service Unavailable |
| 504 | Gateway Timeout |
| 505 | HTTP Version Not Supported |

status bersifat read-only, tidak memiliki nilai default. **status** diperkenalkan di Windows Internet Explorer 7.

Contoh :

```
if (ajaxku.status == 401)
    alert('Access denied. ');
else
    alert(ajaxku.responseText);
}
```

timeout

Syntax

```
object.timeout = v
```

Nilai

| | |
|---|---|
| v | Berupa Integer milliseconds berapa lama batas browser menunggu respon dari server. |
|---|---|

timeout bersifat read/write, nilai defaultnya adalah 0.

Catatan

Jika periode period time-out sudah habis, **responseText** akan bernilai *null*. Kita harus men-set nilai time-out lebih lama dari dugaan waktu respon dari request kita. Properti **timeout** hanya di-set di antara ketika memanggil metode **open** dan metode **send**.

Jika anda men-set nilai **XMLHttpRequest** time-out lebih besar dari pada nilai time-out network ketika stack, network stack akan muncul pertama kali dan **ontimeout** event tidak akan pernah terpanggil.

Contoh

```
var ajaxku;  
ajaxku = new XMLHttpRequest();  
ajaxku.open("GET", url, true);  
ajaxku.timeout = 10000;  
ajaxku.ontimeout = timeoutPeringatan;  
ajaxku.send(null);
```

onreadystatechange Event

Syntax

Object.onreadystatechange = fungsi_yang_menangani

Catatan

onreadystatechange event telah diperkenalkan di Windows Internet Explorer 7.

Contoh

```
var ajaxku = new XMLHttpRequest();  
ajaxku.onreadystatechange = reportStatus;  
ajaxku.open("GET", "http://localhost/test.xml", true);  
ajaxku.send();  
function reportStatus()  
{  
    if (ajaxku.readyState == 4 /* complete */) {  
        if (ajaxku.status == 200 || ajaxku.status == 304) {  
            alert('Transfer complete.');        }  
        else { // terjadi error  
        }  
    }  
}
```

ontimeout Event

Syntax

Objek.ontimeout = fungsi_yang_menangani

Catatan

ontimeout event terjadi jika periode **timeout** lewat sebelum **onload** event terjadi.

Ketika **ontimeout** event terjadi, properti [responseText](#) tidak tersedia dan kalau kita mencoba meng-aksesnya akan terjadi error.

Contoh

```
<script type="text/javascript">
function timeo()
{
    alert("XDR ontimeout");
}
...
ajaxku.ontimeout = timeo;
```

abort Method

Cancel HTTP request.

Syntax

```
XMLHttpRequest.abort()
```

Catatan

abort telah diperkenalkan di Windows Internet Explorer 7.

Metode **abort** men-cancel atau meng-interupsi operasi HTTP Request asinkron yang sedang terjadi. Memanggil metode **abort** akan mereset objek. **onreadystatechange** event juga di-cancel, dan **readyState** di-ubah ke 0 (*uninitialized*).

open Method

Meng-assign metode, URL tujuan , atribut lainnya dari request yang akan dikirim.

Syntax

```
object.open(sMethod, sUrl [, bAsync] [, sUser] [, sPassword])
```

Parameters

| | |
|------------------|--|
| <i>sMethod</i> | Wajib. String yang menentukan metode HTTP digunakan untuk membuka koneksi: seperti GET, POST, atau HEAD. Tidak case-sensitive. |
| <i>sUrl</i> | Wajib. String yang menentukan absolut atau relative URL dari server-side web programming/services atau data XML data. |
| <i>bAsync</i> | Optional. Variant yang menentukan true untuk operasi asynchronous (hasil segera diberikan, tidak perlu menunggu respon dari server, sehingga pengguna dapat terus memasukkan data, bergulir sekitar, dan menggunakan aplikasi), atau false untuk operasi synchronous. Jika tidak ditentukan maka defaultnya adalah true . Jika anda memberikan nilai false , Operasi send adalah <i>synchronous</i> , dan Windows Internet Explorer tidak menerima input atau menghasilkan output ketika operasi send sedang berjalan. |
| <i>sUser</i> | Optional. Variant untuk menentukan nama user untuk proses autentikasi. |
| <i>sPassword</i> | Optional. Variant untuk menentukan password untuk proses autentikasi. |

Berikut adalah HTTP verbs dan metode World Wide Web Distributed Authoring and Versioning (WebDAV) yang di-support:

| Verb / Method | Defined In HTTP (RFC-2616) | Defined In WebDAV (RFC-2518) | Function |
|---------------|----------------------------|------------------------------|---------------------------------|
| GET | HTTP | WebDAV | Request URI |
| POST | HTTP | WebDAV | Send data to server |
| HEAD | HTTP | WebDAV | Request URI without body |
| PUT | HTTP | WebDAV | Store data for URI |
| DELETE | HTTP | WebDAV | Delete data for URI |
| MOVE | | WebDAV | Move URI to to new location |
| PROPFIND | | WebDAV | Request URI Properties |
| PROPPATCH | | WebDAV | Update or Delete URI Properties |
| MKCOL | | WebDAV | Create collection at URI |
| COPY | | WebDAV | Create copy of URI |
| LOCK | | WebDAV | Create Lock |
| UNLOCK | | WebDAV | Remove Lock |

| | | | |
|---------|------|--------|---------------------|
| OPTIONS | HTTP | WebDAV | Request URI Options |
|---------|------|--------|---------------------|

send Method

Mengirim HTTP request ke server dan menerima respon.

Syntax

```
object.send( [varBody])
```

Parameters

| | |
|----------------|--|
| <i>varBody</i> | Optional. Variant yang menentukan body dari message request yang dikirim. |
|----------------|--|

Metode send bisa *synchronous* atau *asynchronous*, tergantung dari nilai parameter *varAsync* di metode open yang dipanggil.

Parameter *varBody* bisa berupa **String**, **array** of unsigned bytes, atau objek XML Document Object Model (DOM).

setRequestHeader Method

Menambah HTTP headers yang disesuaikan ke request.

Syntax

```
object.setRequestHeader(sName, sValue)
```

Parameters

| | |
|---------------|--|
| <i>sName</i> | Wajib. String yang menentukan nama header. |
| <i>sValue</i> | Wajib. String yang menentukan nilai header. |

Contoh

```
var ajaxku = new XMLHttpRequest();  
ajaxku.open("POST", sURL, false);  
ajaxku.setRequestHeader("Content-Type", "text/xml");  
ajaxku.send(sRequestBody);
```

Oke sekarang kita akan membuat aplikasi auto complete menggunakan ajax, di mana pada contoh berikut kita akan membuat auto complete dari suatu database, pada contoh ini datanya berupa nama negara-negara di dunia. Jika kita mengetik suatu huruf atau kata, maka otomatis akan ditampilkan Negara yang memungkinkan diawali oleh huruf atau kata tersebut.

Pertama, mari kita buat table Negara, dengan struktur sebagai berikut :

```
-- Table structure for table `negara`  
--  
CREATE TABLE `negara` (  
  `nama` varchar(50) NOT NULL,  
  PRIMARY KEY (`nama`)  
)
```

Kemudian isi dari table tersebut adalah nama-nama Negara seperti di bawah :

| nama |
|------------------|
| Afghanistan |
| Amerika Serikat |
| Arab Saudi |
| Bahrain |
| Belanda |
| Brunei Darusalam |
| Chili |
| China |
| Filipina |
| Indonesia |
| Inggris |
| Jepang |
| Jerman |
| Korea Selatan |
| Korea Utara |
| Malaysia |
| Singapura |
| Thailand |

Sekarang, mari kita buat kode HTML dan Ajax

Kode4. autonegara.html

```
<html>  
<head>  
<script>  
var drz;  
function lihat(kata){  
  if(kata.length==0){  
    document.getElementById("kotaksugest").style.visibility =  
"hidden";  
  }else{  
    drz = buatajax();  
    var url="cari.php";  
    drz.onreadystatechange=stateChanged;  
    var params = "q="+kata;  
    drz.open("POST",url,true);  
    //beberapa http header harus kita set kalau menggunakan POST  
    drz.setRequestHeader("Content-type", "application/x-www-form-  
urlencoded");  
    drz.setRequestHeader("Content-length", params.length);  
    drz.setRequestHeader("Connection", "close");
```

```
        drz.send(params);
    }
}

function buatajax(){
    if (window.XMLHttpRequest){
        return new XMLHttpRequest();
    }
    if (window.ActiveXObject){
        return new ActiveXObject("Microsoft.XMLHTTP");
    }
    return null;
}

function stateChanged(){
var data;
    if (drz.readyState==4 && drz.status==200){
        data=drz.responseText;
        if(data.length>0){
            document.getElementById("kotaksugest").innerHTML = data;
            document.getElementById("kotaksugest").style.visibility = "";
        }else{
            document.getElementById("kotaksugest").innerHTML = "";
            document.getElementById("kotaksugest").style.visibility =
"hidden";
        }
    }
}

function isi(kata){
    document.getElementById("kata").value = kata;
    document.getElementById("kotaksugest").style.visibility = "hidden";
    document.getElementById("kotaksugest").innerHTML = "";
}
</script>
</head>
<body>
<form>
<div>
<input type=text id=kata name=kata onkeyup=lihat(this.value)>
</div>
<div id=kotaksugest style="position:absolute;top:36;left:15;background-
color:lightblue;width:200;visibility:hidden;z-index:100">
</div>
<!--//sesuaikan posisi kotaksugest terhadap teksbox--->
</center>
</form>
</body>
</html>
```

Pada contoh di atas kita menggunakan metode POST untuk mengirim parameter-parameter ke server.

```
drz.open("POST",url,true);
```

Beberapa Header harus kita set bila menggunakan metode POST, supaya program bisa berjalan dengan baik

```
drz.setRequestHeader("Content-type", "application/x-www-form-  
urlencoded");  
drz.setRequestHeader("Content-length", params.length);  
drz.setRequestHeader("Connection", "close");
```

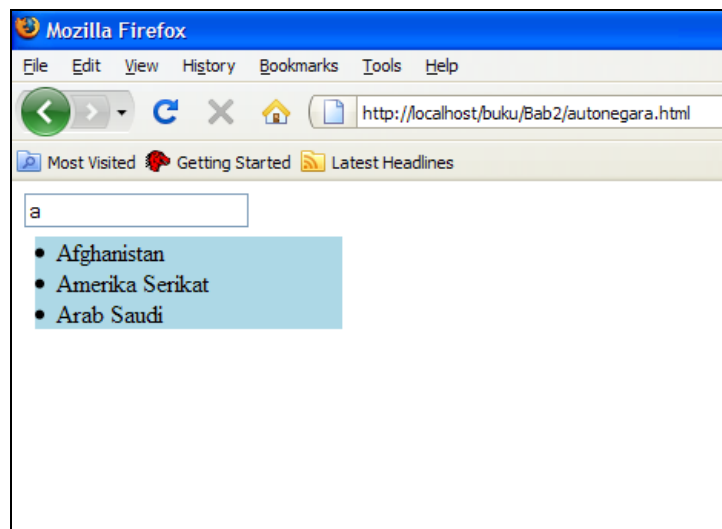
Logikanya adalah sebagai berikut :

1. Setelah kita mengetik karakter atau setelah menekan keyboard (onkeyup) di input text, maka langsung jalankan fungsi lihat()
2. Cek, apakah input text ada nilainya, jika kosong maka kita menyembunyikan kotaksugest, jika ada nilainya, maka kita buat objek ajax dan mengirimkan request ke server.
3. Apabila request yang dikirim telah komplet dan OK, maka kita tampilkan di kotaksugest.

Sekarang mari kita buat kode di sisi server (cari.php), yang berfungsi untuk melakukan query terhadap table Negara, dan menampilkan hasil query.

Kode 4. cari.php

```
<?php  
mysql_connect("localhost","root","");  
mysql_select_db("test");  
  
$kata = $_POST['q'];  
$query = mysql_query("select nama from negara where nama like '$kata%'  
limit 10");  
while($k=mysql_fetch_array($query)){  
    echo '<li onClick="isi(\''. $k[0]. '\')";'  
    style="cursor:pointer">'. $k[0]. '</li>';  
}  
?>
```



Gambar 3. Tampilan AutoComplete

3

Format Data Respon Ajax

Ketika Ajax mengirim request ke server, maka akan ada balasan dari server berupa data. Nah, data yang dikirim dari server ke ajax ini ada macam-macam. Pada contoh-contoh sebelumnya, anda telah melihat data yang dikirim berupa text atau HTML. Seperti contoh di bawah ini, program PHP memberikan data berupa HTML

```
while($k=mysql_fetch_array($query)){  
    echo '<li onClick="isi(\''. $k[0].'\')";  
    style="cursor:pointer">'. $k[0].'\</li>';  
}
```

Pada contoh tersebut, program PHP memberikan output berupa HTML dengan tag . Ada beberapa format data yang populer yang biasa digunakan untuk untuk mengirim data dari server ke client, yaitu JSON, Text, AHAH, XML dan lainnya.

JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

Beberapa orang lebih suka JSON, karena paling mudah untuk mem-parse-nya, hanya menempatkan sebuah **eval** dan selesai sudah.

JSON terbuat dari dua struktur:

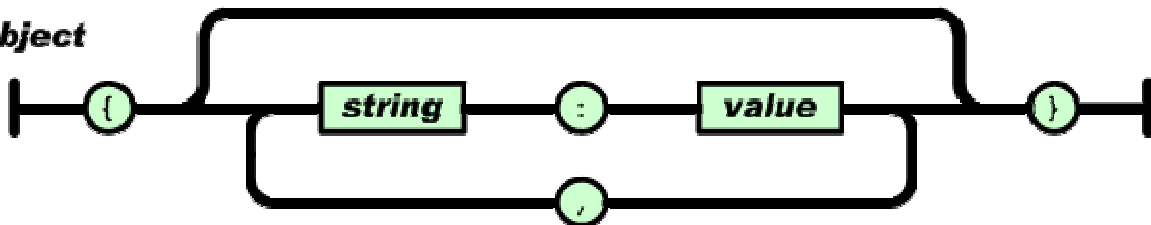
- Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (object), rekaman (record), struktur (struct), kamus (dictionary), tabel hash (hash table), daftar berkunci (keyed list), atau associative array.
- Daftar nilai terurutkan (an ordered list of values). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (array), vektor (vector), daftar (list), atau urutan (sequence).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini.

JSON menggunakan bentuk sebagai berikut:

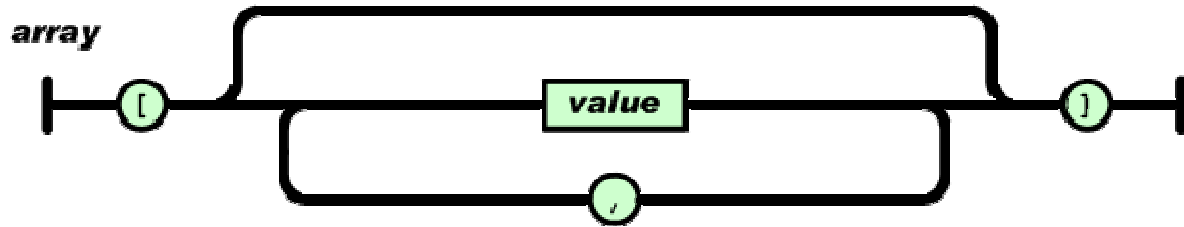
Object adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).

object



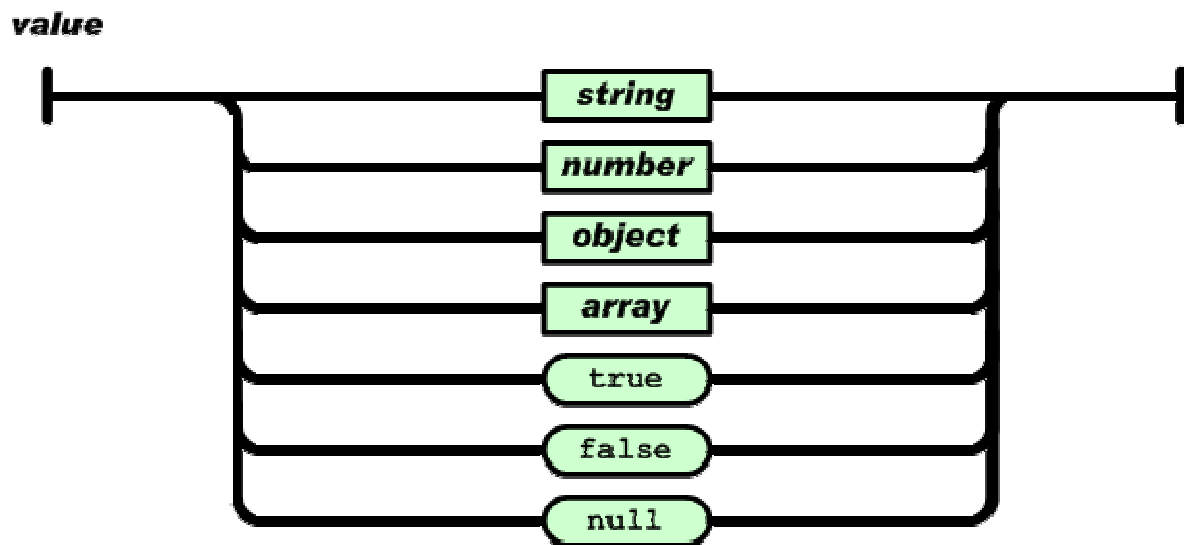
Sumber : www.json.org

Array adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).



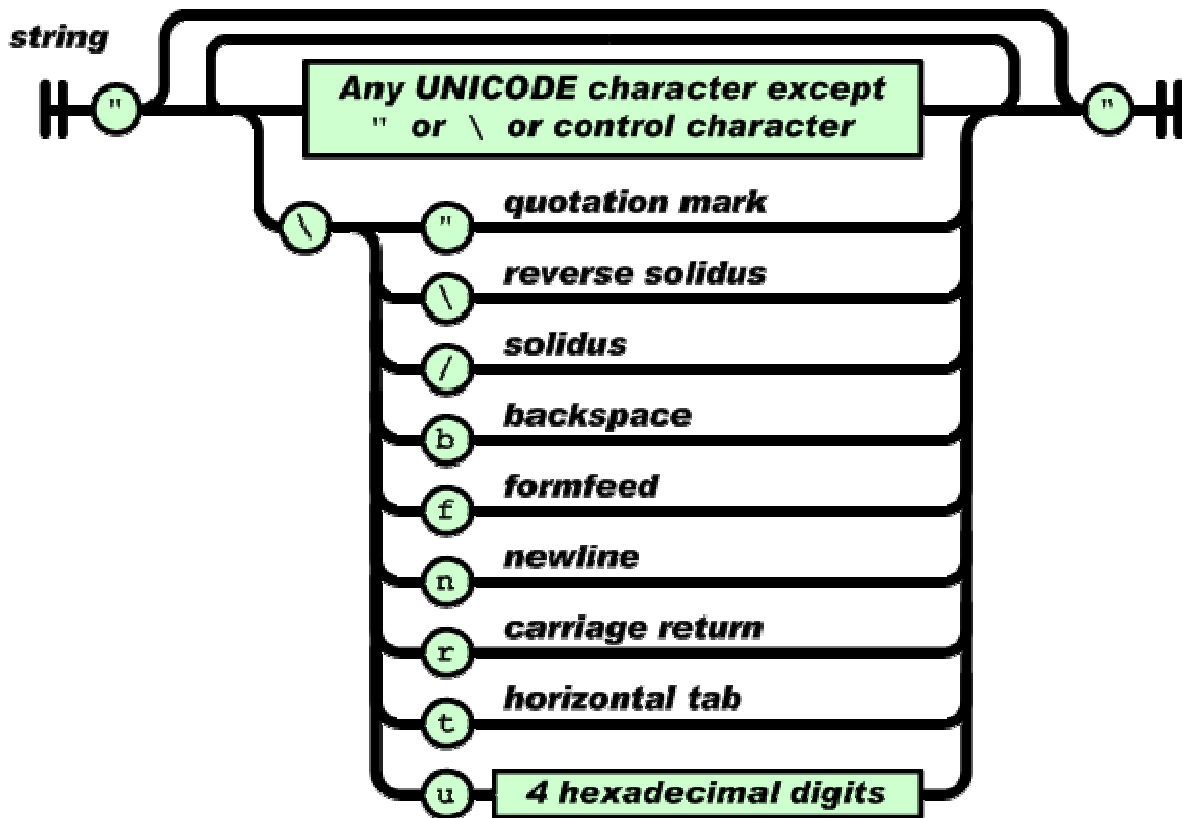
Sumber : www.json.org

Nilai (value) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau true atau false atau null, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.



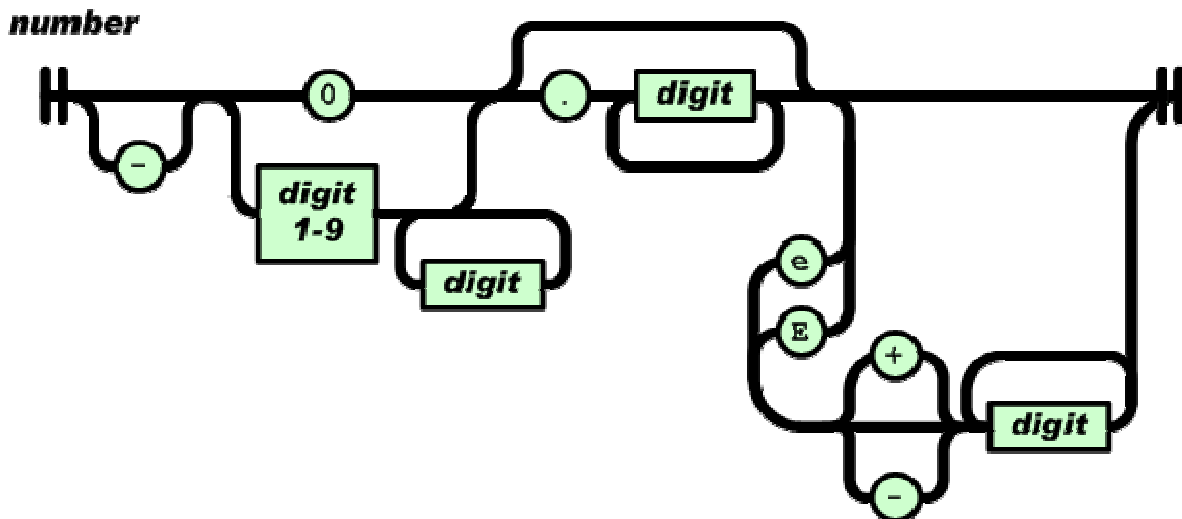
Sumber : www.json.org

String adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan backslash escapes "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.



Sumber : www.json.org

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Sumber : www.json.org

Contoh JSON

Sekarang mari kita lihat contoh-contoh bentuk JSON yang saya ambil dari json.org. Berikut bentuk XML, yang telah familiar dengan kita.

```
<menu id="file" value="File">
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()" />
    <menuitem value="Open" onclick="OpenDoc()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>
```

Jika kita ubah ke bentuk JSON adalah :

```
{ "menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      { "value": "New", "onclick": "CreateNewDoc()" },
      { "value": "Open", "onclick": "OpenDoc()" },
      { "value": "Close", "onclick": "CloseDoc()" }
    ]
  }
}
```

Bentuk XML :

```
<widget>
  <debug>on</debug>
  <window title="Sample Konfabulator Widget">
    <name>main_window</name>
    <width>500</width>
    <height>500</height>
  </window>
  <image src="Images/Sun.png" name="sun1">
    <hOffset>250</hOffset>
    <vOffset>250</vOffset>
    <alignment>center</alignment>
  </image>
  <text data="Click Here" size="36" style="bold">
    <name>text1</name>
    <hOffset>250</hOffset>
    <vOffset>100</vOffset>
    <alignment>center</alignment>
    <onMouseUp>
      sun1.opacity = (sun1.opacity / 100) * 90;
    </onMouseUp>
  </text>
</widget>
```

Bentuk JSON :

```
{ "widget": {
  "debug": "on",
  "window": {
    "title": "Sample Konfabulator Widget",
    "name": "main_window",
    "width": 500,
    "height": 500
  },
  "image": {
    "src": "Images/Sun.png",
    "name": "sun1",
    "hOffset": 250,
    "vOffset": 250,
    "alignment": "center"
  },
  "text": {
    "data": "Click Here",
    "size": 36,
    "style": "bold",
    "name": "text1",
    "hOffset": 250,
    "vOffset": 100,
    "alignment": "center",
    "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;"
  }
}
```

| XML | JSON |
|---------------------|--------------------------|
| < xx yy='nn'>< /xx> | { "xx": { "yy": "nn" } } |

Membuat JSON di sisi server

Sekarang mari kita lihat bagaimana cara membuat bentuk bentuk JSON. Dari sisi server kita akan membuat JSON menggunakan PHP, pada PHP versi 5.2 telah terdapat fungsi untuk membuat JSON, yaitu fungsi **json_encode()**.

Syntax-nya adalah sebagai berikut

```
string json_encode ( mixed $value [, int $options= 0 ] )
```

\$value, adalah nilai yang akan di encode menjadi string JSON

\$option, bisa berupa **JSON_HEX_QUOT**, **JSON_HEX_TAG**, **JSON_HEX_AMP**, **JSON_HEX_APOS**, **JSON_FORCE_OBJECT**, defaultnya adalah 0.

Contoh :

```
<?php
$arr = array ('a'=>1, 'b'=>2, 'c'=>3, 'd'=>4, 'e'=>5);

echo json_encode($arr);
?>
```

Kode di atas akan menghasilkan :

```
{ "a":1, "b":2, "c":3, "d":4, "e":5 }
```

Selain menggunakan `json_encode()`, kita juga bisa membuat JSON secara manual dengan menggunakan PHP.

Misal, kita memiliki tabel di MySQL dengan struktur sebagai berikut :

```
CREATE TABLE `message` (  
  `message_id` INT(11) NOT NULL AUTO_INCREMENT,  
  `chat_id` INT(11) NOT NULL DEFAULT '0',  
  `user_id` INT(11) NOT NULL DEFAULT '0',  
  `user_name` VARCHAR(64) DEFAULT NULL,  
  `message` TEXT,  
  `post_time` DATETIME DEFAULT NULL,  
  PRIMARY KEY (`message_id`)  
)
```

Isi dari tabel tersebut adalah sebagai berikut

| message_id | chat_id | user_id | user_name | message | post_time |
|------------|---------|---------|-----------|----------------|-----------------|
| 1 | 1 | 1 | Nurmi | Halo apa kabar | 4/28/2010 17:30 |
| 2 | 1 | 2 | Desrizal | Kabar Baik | 4/28/2010 17:30 |
| 3 | 1 | 1 | Nurmi | Lagi di mana | 4/28/2010 17:30 |
| 4 | 1 | 2 | Desrizal | Belitung | 4/28/2010 17:30 |

Dan kita ingin membuat JSON dari tabel di atas dengan bentuk seperti ini :

```
{ "messages":  
  { "pesan": [  
    { "id": "1",  
      "user": "Nurmi",  
      "text": "Halo apa kabar",  
      "time": "4/28/2010 17:30"  
    }, { "id": "2",  
      "user": "Desrizal",  
      "text": "Kabar Baik",  
      "time": "4/28/2010 17:30"  
    }, { "id": "3",  
      "user": "Nurmi",  
      "text": "Lagi di mana",  
      "time": "4/28/2010 17:30"  
    }, { "id": "4",  
      "user": "Desrizal",  
      "text": "Belitung",  
      "time": "4/28/2010 17:30"  
    }  
  ]  
  }  
}
```

Berikut kode PHP untuk membuat JSON tersebut.

Kode 5. Membuat JSON dengan PHP secara manual (phpjson.php)

```
<?php  
mysql_connect("localhost", "root", "");  
mysql_select_db("test");
```



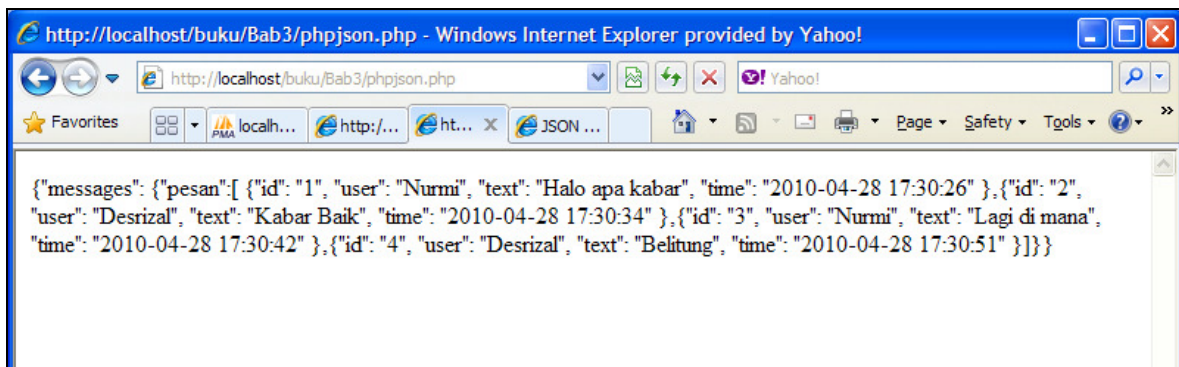
```
$json = '{"messages": {';
$query = mysql_query("select * from message");
$json .= '"pesan":[ ';
while($x =mysql_fetch_array($query)){
    $json .= '{';
    $json .= '"id": "' . $x['message_id'] . '",
            "user": "' . htmlspecialchars($x['user_name']) . '",
            "text": "' . htmlspecialchars($x['message']) . '",
            "time": "' . $x['post_time'] . '"
        },';
}

//hilangkan koma (,) di akhir
$json = substr($json,0,strlen($json)-1);

//lengkapi penutup format JSON
$json .= ']';
$json .= '}}';

echo $json;
?>
```

Pada contoh di atas, kita membuat JSON secara manual, yang penting adalah kita harus tahu penempatan setiap elemen. Jika kita lihat outputnya di browser akan seperti berikut :



Gambar 4. Tampilan phpjson.php

Mem-parse JSON di sisi client

Oke, setelah kita mengetahui bagaimana cara membuat output dengan format JSON, sekarang mari kita lihat bagaimana cara menguraikan format JSON tersebut menggunakan javascript. Untuk menguraikan /mem-parse JSON adalah dengan menggunakan fungsi `eval("(" + jsonteks + ")")`.

Selain menggunakan `eval()`, kita juga bisa menggunakan fungsi-fungsi yang telah dibuat orang lain, seperti yang ada di www.json.org.

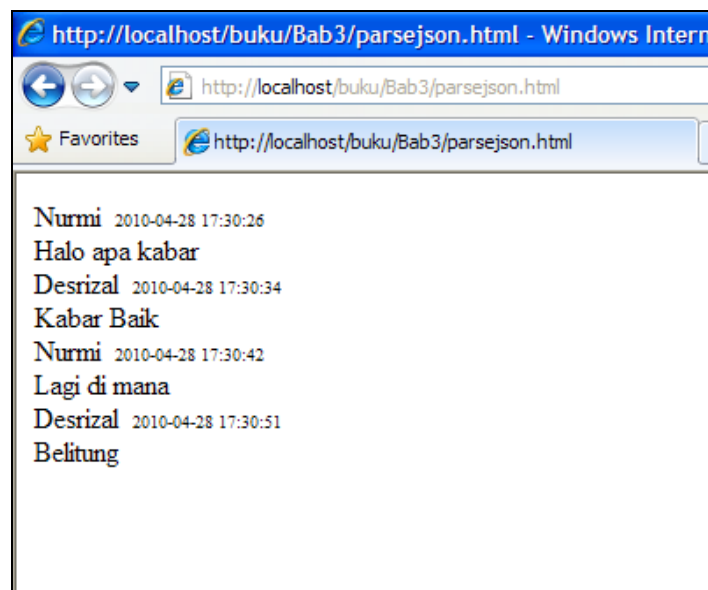
Para ahli, menyarankan untuk menghindari penggunaan `eval()`, karena kurang aman, sebaiknya gunakan fungsi-fungsi yang dibuat oleh orang lain (seperti : http://www.JSON.org/json_parse.js).

Tetapi sebagai contoh pelajaran, saya akan memperlihatkan cara mem-parse menggunakan `eval()`. Untuk menyederhanakan contoh berikut, saya tidak membuat kode yang berbau ajax, saya anggap kita sudah memiliki teks yang berformat JSON.

Kode 6. Mem-parse JSON (parsejson.html)

```
<html>
<head>
<script>
function tes(){
json = '{"messages": {"pesan":[ {"id": "1", "user": "Nurmi", "text":
"Halo apa kabar", "time": "2010-04-28 17:30:26" },{"id": "2", "user":
"Desrizal", "text": "Kabar Baik", "time": "2010-04-28 17:30:34" },{"id":
"3", "user": "Nurmi", "text": "Lagi di mana", "time": "2010-04-28
17:30:42" },{"id": "4", "user": "Desrizal", "text": "Belitung", "time":
"2010-04-28 17:30:51" } ]}}'

var teks_div = document.getElementById('divteks');
var response = eval("(" + json + ")");
for(i=0;i < response.messages.pesan.length; i++) {
    teks_div.innerHTML += response.messages.pesan[i].user;
    teks_div.innerHTML += '&nbsp;&nbsp;&nbsp;<font size=1>' +
response.messages.pesan[i].time + '</font><br />';
    teks_div.innerHTML += response.messages.pesan[i].text + '<br />';
}
}
</script>
</head>
<body onload=tes()>
<div id=divteks>
</div>
</body>
```



Gambar 5. Tampilan parsejson.html

JSON Object

Kita telah mempelajari cara mem-parse JSON menggunakan `eval()`. Kelemahan menggunakan `eval()` adalah kurang aman. Untuk itu, maka pada FF3.5+, IE8+, dan Opera 10.5+ diperkenalkan javascript built in JSON Object.

JSON Object menyediakan metode untuk mengubah objek JSON menjadi string dan JSON string menjadi objek javascript. Berikut adalah metode-metode dari JSON Object.

stringify(obj, [replacer], [space])

Berguna untuk mengubah javascript object menjadi JSON String.

Contoh :

```
var jsonobjek={"nama":"Desrizal", "umur":28, "teman":["Nurmi", "Sarah", "Albert"]}  
var jsonstr=JSON.stringify(jsonobjek)  
alert(typeof jsonstr) //string
```

parse(string, [reviver])

Berguna untuk mengubah JSON String menjadi Javascript Object

Contoh :

```
var jsonstr='{ "nama": "Desrizal", "umur": 28, "teman": ["Nurmi", "Sarah", "Albert"] }'  
var george=JSON.parse(jsonstr) //convert JSON string into object  
alert(george.age) //alerts 28
```

JSON Ajax Web Chat

Oke sekarang kita lihat contoh aplikasi ajax yang menggunakan JSON sebagai format output yang di berikan oleh server ke client.

Pertama-tama mari kita buat tabel di database, sengan struktur tabel sebagai berikut :

```
CREATE TABLE `drzchat` (  
  `nomor` int(3) NOT NULL auto_increment,  
  `nama` varchar(20) NOT NULL,  
  `pesan` varchar(200) NOT NULL,  
  `waktu` varchar(10) NOT NULL,  
  PRIMARY KEY (`nomor`)  
);
```

Kemudian mari kita buat tampilan layout HTML nya dan program Ajax di sisi client.

Kode 6. drzchat.html

```
<html>  
<head>  
<title>  
DRZ Chat 1.0  
</title>  
<script>  
var ajaxku = buatAjax();  
var tnama = 0;  
var pesanakhir = 0;
```

```
var timer;
function taruhNama(){
    if(tnama==0){
        document.getElementById("nama").disabled = "true";
        tnama = 1;
    }else{
        document.getElementById("nama").disabled = "";
        tnama = 0;
    }
    ambilPesan();
}
function buatAjax(){
    if(window.XMLHttpRequest){
        return new XMLHttpRequest();
    }else if(window.ActiveXObject){
        return new ActiveXObject("Microsoft.XMLHTTP");
    }
}

function ambilPesan(){
    namaku = document.getElementById("nama").value
    if(ajaxku.readyState == 4 || ajaxku.readyState == 0){

ajaxku.open("GET","ambilchat.php?akhir="+pesanakhir+"&nama="+namaku+"&sid
="+Math.random(),true);
        ajaxku.onreadystatechange = aturAmbilPesan;
        ajaxku.send(null);
    }
}

function aturAmbilPesan(){
    if(ajaxku.readyState == 4){
        var chat_div = document.getElementById("div_chat");
        var data = eval("(" + ajaxku.responseText + ")");
        for(i=0;i<data.messages.pesan.length;i++){
            chat_div.innerHTML += "<font
color=red>" + data.messages.pesan[i].nama + "</font> ";
            chat_div.innerHTML += "<font
size=1>" + data.messages.pesan[i].waktu + "</font> ";
            chat_div.innerHTML += " :
" + data.messages.pesan[i].teks + "<br>";
            chat_div.scrollTop = chat_div.scrollHeight;
            pesanakhir = data.messages.pesan[i].id;
        }
    }
    timer = setTimeout("ambilPesan()",1000);
}

function kirimPesan(){
    pesannya = document.getElementById("pesan").value
    namaku = document.getElementById("nama").value
    if(pesannya != "" && document.getElementById("nama").value != ""){
```

```
ajaxku.open("GET", "ambilchat.php?akhir="+pesanakhir+"&nama="+namaku+"&pesan="+pesannya+"&sid="+Math.random(), true);
    ajaxku.onreadystatechange = aturAmbilPesan;
    ajaxku.send(null);
    document.getElementById("pesan").value = "";
} else {
    alert("Nama atau pesan masih kosong");
}
}

function aturKirimPesan() {
    clearInterval(timer);
    ambilPesan();
}

function blockSubmit() {
    kirimPesan();
    return false;
}
</script>
</head>
<body>
Nama : <input type="text" name="nama" id="nama">
<input type="button" value="login" id="tmb_login" onclick="taruhNama()"><p>
<div id="div_chat" style="height: 300px; width: 500px; overflow: auto;
background-color: lightyellow; border: 1px solid #555555;">

</div>
<form onsubmit="return blockSubmit();">
Pesan : <input type="text" name="pesan" id="pesan" size=50>
<input type="button" value="kirim" onclick="kirimPesan()">
</form>
</body>
</html>
```

Pada program chat sisi client di atas, pertama-tama user harus memasukkan nama mereka di suatu teks input. Setelah kita menekan tombol login, maka segera dijalankan fungsi **ambilPesan()**. Di mana fungsi ini berfungsi untuk melakukan request ke server untuk mengambil data dari server, **ambilchat.php**.

Data yang diambil dari server berupa teks dengan format JSON sebagai berikut :

```
{ "messages":
  { "pesan": [
    { "id": "35",
      "nama": "Batman",
      "teks": "halo Nurmi",
      "waktu": "16:31" },
    { "id": "36",
      "nama": "Nurmi",
      "teks": "halo juga",
      "waktu": "16:31" },
    { "id": "37",
      "nama": "desrizal",
      "teks": "hai apa kabar",
      "waktu": "16:47" }
```

```
    ]  
  }  
}
```

Format JSON tersebut diambil dari server, yang di masukkan ke dalam variable data, kemudian kita gunakan eval untuk mengubah JSON string menjadi objek javascript

```
var data = eval("(" + ajaxku.responseText + ")");
```

Kemudian Objek JSON yang telah di-parse di atas kita masukkan ke dalam <div>

```
for (i=0; i<data.messages.pesan.length; i++) {  
  chat_div.innerHTML += "<font color=red>" + data.messages.pesan[i].nama + "</font> ";  
  chat_div.innerHTML += "<font size=1>" + data.messages.pesan[i].waktu + "</font> ";  
  chat_div.innerHTML += " : " + data.messages.pesan[i].teks + "<br>";  
  chat_div.scrollTop = chat_div.scrollHeight;  
  pesanakhir = data.messages.pesan[i].id;  
}
```

Supaya ajax hanya megambil data dari pesan yang terakhir maka kita juga mengirimkan nomor pesan id terakhir

```
pesanakhir = data.messages.pesan[i].id;
```

Supaya ajax mengambil data tiap satu detik maka kita gunakan fungsi setTimeout()

```
timer = setTimeout("ambilPesan()", 1000);
```

Biasanya kalau kita chat, kita ketik pesannya di input teks lalu kita tekan ENTER, biasanya kalau form di tekan ENTER akan otomatis ter-submit dan halaman akan bertukar. Untuk mencegah halaman ter-submit ketika tekan enter, maka kita blok menggunakan fungsi berikut :

```
function aturKirimPesan() {  
  clearInterval(timer);  
  ambilPesan();  
}
```

Fungsi di atas dipanggil ketika ada event onSubmit.

```
<form onSubmit="return blockSubmit();" >
```

Sekarang mari kita lihat kode PHP di sisi server, di mana di program PHP ini kita membuat output data berformat JSON

Kode 7. ambilchat.php

```
<?php  
mysql_connect("localhost", "root", "");  
mysql_select_db("test");  
  
$nama = $_GET['nama'];  
$pesan = $_GET['pesan'];  
$waktu = date("H:i");  
$akhir = $_GET['akhir'];  
  
$json = '{"messages": {';  
if ($akhir == 0) {  
  $nomor = mysql_query("select nomor from drzchat order by nomor desc  
limit 1");  
  $n = mysql_fetch_array($nomor);  
  $no = $n['nomor'] + 1;
```

```
$json .= '"pesan":[ {';
$json .= '"id":"' . $no . '",
        "nama": "Admin",
        "teks": "Selamat datang di chatting room",
        "waktu": "' . $waktu . '"
    }]';

$masuk = mysql_query("insert into drzchat values(null, 'Admin', '$nama
bergabung dalam chat', '$waktu')");
}else{
    if($pesan){
        $masuk = mysql_query("insert into drzchat
values(null, '$nama', '$pesan', '$waktu')");
    }

    $query = mysql_query("select * from drzchat where nomor > $akhir");
    $json .= '"pesan":[ ' ;
    while($x = mysql_fetch_array($query)){
        $json .= '{';
        $json .= '"id":"' . $x['nomor'] . '",
                "nama": "' . htmlspecialchars($x['nama']) . '",
                "teks": "' . htmlspecialchars($x['pesan']) . '",
                "waktu": "' . $x['waktu'] . '"
            },';
    }
    $json = substr($json, 0, strlen($json)-1);
    $json .= ']';
}

$json .= '}}';
echo $json;
?>
```

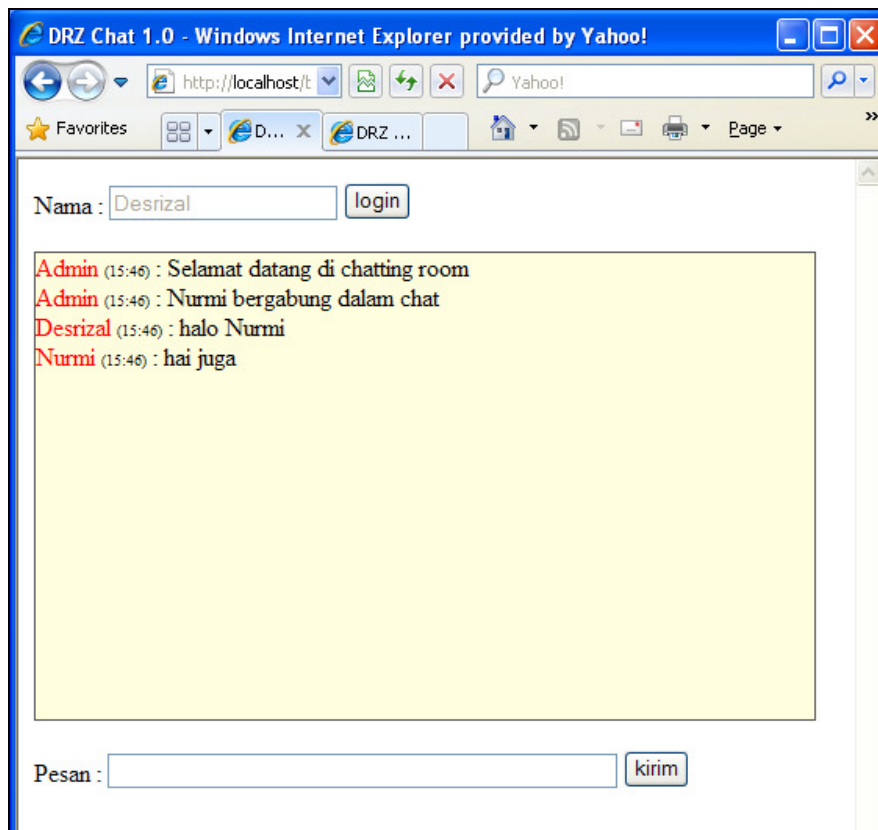
Logika dari program di atas adalah :

1. Lakukan koneksi ke server MySQL
2. Ambil nilai-nilai parameter yang di berikan oleh client dengan metode GET
3. Cek apakah user pertama kali login dengan melihat nilai variable \$akhir, jika bernilai 0 artinya adalah user chat pertama kali, lalu kita kirim data berupa string berformat JSON, ucapan selamat data dari admin, lalu kita masukkan ke dalam database untuk supaya user lain mengetahui ada user lain yang login
4. Apabila nilai \$akhir besar dari 0, artinya user bukan masuk untuk pertama kalinya, berikan data-data dari database setelah yang terakhir dia dapat, yang kita dapat dari variable \$akhir

```
$query = mysql_query("select * from drzchat where nomor > $akhir");
```

5. Ouput yang kita buat dengan format json, untuk mengirim data ke client cukup melakukan `echo $json;`.

Berikut adalah tampilan dari aplikasi chat kita.



Gambar 7. Tampilan Aplikasi Chat

AHAH

AHAH (Asynchronous HTML over HTTP) adalah format data dari respon ajax. Format datanya langsung berupa HTML, jadi kita langsung bisa menampilkan respon tersebut.

```
ElementHTML.innerHTML = objekAjax.responseText;
```

Pada contoh berikut kita akan membuat tab menggunakan Ajax, di mana url yang direquest bukan file PHP seperti biasanya, tetapi langsung berupa file HTML, yang berisi kode-kode HTML yang merupakan konten suatu halaman.

Tampilan tab yang akan kita buat adalah sebagai berikut :



Gambar 8. Tampilan Ajax Tab

Pertama-tama untuk membuat tab seperti di atas kita buat dengan HTML

```
<li class="tabs" id="tab1">Tab 1</li>
<li class="tabs" id="tab2">Tab 2</li>
<li class="tabs" id="tab3">Tab 3</li>
<li class="tabs" id="tab4">Tab 4</li>
<div id="content" style="overflow:auto">

</div>
```

Kemudian untuk mengatur posisi masing-masing tab, kita gunakan CSS

```
body {
    font-family:Arial, Helvetica, sans-serif;
    font-size:12px;
}
.tabs {
    list-style-type: none;
    width:50px;
    margin-right:5px;
    padding:4px;
    text-align:center;
    float:left;
    cursor:pointer;
    border:1px solid #ccc;
    border-bottom:0;
    background: white url(shade.gif) top left repeat-x;
}
#content {
    width:500px;
    height: 250;
    clear:both;
    border:1px solid #ccc;
    padding : 3px
}
```

Sekarang mari kita lihat kode javascript dan kode keseluruhan halaman tab ajax tersebut.

Kode 8. ajaxtab.html

```
<html>
<head>
<title>Ajax Tab</title>
<script type="text/javascript">
var ajaxku = buatAjax();
function init (){
var tabs = document.getElementsByClassName('tabs');
    for (var i = 0; i < tabs.length; i++) {
        tabs[i].onclick = function () {
            ambilTabData(this.id);
        }
    }
}
function ambilTabData(id) {
    var url;
    switch(id){
        case 'tab1' : url = "halaman1.html";break;
        case 'tab2' : url = "halaman2.html";break;
        case 'tab3' : url = "halaman3.html";break;
        case 'tab4' : url = "halaman4.html";break;
    }
    ajaxku.open("GET",url,true);
    ajaxku.onreadystatechange = tampilkanHalaman;
    ajaxku.send(null);
}

function buatAjax(){
    if(window.XMLHttpRequest){
        return new XMLHttpRequest();
    }else if(window.ActiveXObject){
        return new ActiveXObject("Microsoft.XMLHTTP");
    }
}

function tampilkanHalaman(){
    if(ajaxku.readyState == 4){
        var halaman = ajaxku.responseText;
        document.getElementById("content").innerHTML = halaman;
    }else{
        document.getElementById("content").innerHTML = "Loading....";
    }
}
</script>
<style type="text/css">
body {
    font-family:Arial, Helvetica, sans-serif;
    font-size:12px;
}

.tabs {
    list-style-type: none;
```

```
width:50px;
margin-right:5px;
padding:4px;
text-align:center;
float:left;
cursor:pointer;
border:1px solid #ccc;
border-bottom:0;
background: white url(shade.gif) top left repeat-x;
}
#content {
width:500px;
height: 250;
clear:both;
border:1px solid #ccc;
padding : 3px
}
</style>
</head>
<body onload=init()>
<li class="tabs" id="tab1">Tab 1</li>
<li class="tabs" id="tab2">Tab 2</li>
<li class="tabs" id="tab3">Tab 3</li>
<li class="tabs" id="tab4">Tab 4</li>
<div id="content" style="overflow:auto">

</div>

</body>
</html>
```

Dari kode di atas, awal-awal, ketika seluruh body telah di load, akan menjalankan fungsi `init()`. Fungsi ini berguna untuk menentukan hal apa yang akan dilakukan jika elemen `` dengan `class="tabs"` di click oleh user, yaitu menjalankan fungsi `ambilTabData(id)`.

Fungsi `ambilTabData(id)` berguna untuk menentukan halaman apa yang akan di-load untuk masing-masing tab. Kemudian halaman yang telah ditentukan diambil dari server dengan metode `open` dan `send`.

Isi dari halaman HTML yang diambil dari server kita tampilkan ke dalam elemen `<div>`

```
var halaman = ajaxku.responseText;
document.getElementById("content").innerHTML = halaman;
```

Berikut adalah contoh isi dari halaman HTML yang diambil dari server

Kode 9. halaman1.html

```
<font color=red>
Ini adalah isi dari halaman "halaman1.html"<br />
Ini adalah isi dari halaman "halaman1.html"<br />
Ini adalah isi dari halaman "halaman1.html"<br />
</font>
```

Untuk isi dari halaman2.html dan lainnya hamper sama aja, content yang berupa dari koda-kode HTML.

XML

XML adalah format data yang amat terkenal, sudah didukung hampir semua bahasa pemrograman untuk meng-generate dan mem-parse XML. Berikut adalah contoh pemrograman ajax, di mana format respon ajax berupa XML.

Misal kita memiliki data XML sebagai berikut.

Kode 10. data.xml

```
<books>
  <book>
    <author>Desrizal</author>
    <title>Action Script Advanced</title>
  </book>
  <book>
    <author>Nurmi Yulita Rahmi</author>
    <title>Hacking exposed</title>
  </book>
  <book>
    <author>Batman</author>
    <title>PHP Hacking</title>
  </book>
</books>
```

Kemudian di sisi client, kita punya kode ajax sebagai berikut :

Kode 11. ajaxxml.html

```
<html>
<head>
<script>
var drz = buatAjax();
var url = "data.xml";

function buatAjax(){
  if(window.XMLHttpRequest){
    return new XMLHttpRequest();
  }else if(window.ActiveXObject){
    return new ActiveXObject("Microsoft.XMLHTTP");
  }
}

function ambilXMLDoc(){
  drz.open("GET",url+"?sid="+Math.random(),true);
  drz.onreadystatechange = prosesData;
  drz.send(null);
}

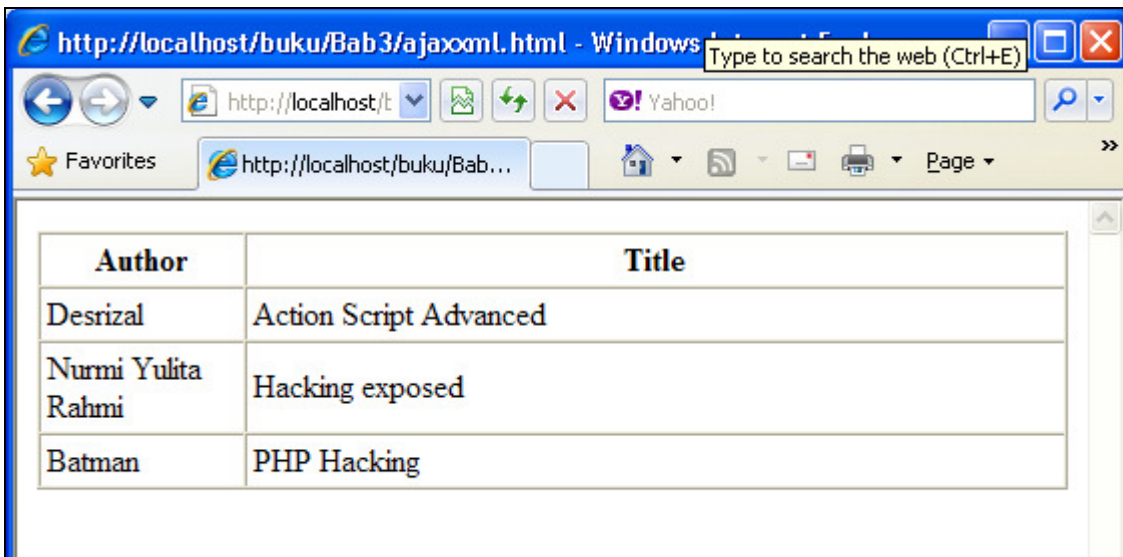
function prosesData(){
  if(drz.readyState == 4){
    var dtable = document.getElementById( 'tabel' );
    var nl = drz.responseXML.getElementsByTagName( 'book' );
    for( var i = 0; i < nl.length; i++ ) {
      var nli = nl.item( i );
      var elAuthor = nli.getElementsByTagName( 'author' );
      var author = elAuthor.item(0).firstChild.nodeValue;
```

```
var elTitle = nli.getElementsByTagName( 'title' );
var title = elTitle.item(0).firstChild.nodeValue;
var elTr = dtable.insertRow( -1 );
var elAuthorTd = elTr.insertCell( -1 );
elAuthorTd.innerHTML = author;
var elTitleTd = elTr.insertCell( -1 );
elTitleTd.innerHTML = title;
    }
}
}

</script>
</head>
<body onload=ambilXMLDoc()>
<table cellspacing="0" cellpadding="3" border=1>
<tbody id="tabel">
<tr>
    <th width="20%">Author</th>
    <th width="80%">Title</th>
</tr>
</tbody>
</table>

</body>
</html>
```

Berikut tampilan jika ajaxxml.html di jalankan di browser



Gambar 9. Tampilan ajaxxml.html

4

Pengenalan jQuery

Apa itu jQuery?

jQuery adalah javascript library, jQuery mempunyai semboyan “write less, do more”. jQuery dirancang untuk memperingkas kode-kode javascript. JQuery adalah javascript library yang cepat dan ringan untuk menangani dokumen HTML, menangani event, membuat animasi dan interaksi ajax. JQuery dirancang untuk mengubah cara anda menulis javascript.

Sebelum anda memulai mempelajari jQuery, anda harus mempunyai pengetahuan dasar mengenai HTML, CSS dan Javascript.

Library jQuery mempunyai kemampuan :

- Kemudahan mengakses elemen-elemen HTML
- Memanipulasi elemen HTML
- Memanipulasi CSS
- Penanganan event HTML
- Efek-efek javascript dan animasi
- Modifikasi HTML DOM
- AJAX
- Menyederhanakan kode javascript lainnya

Kemudian untuk memulai mempelajari jQuery, anda harus mendownload **jquery.js** dari situs <http://www.jquery.com>. Setiap anda menulis kode javascript dengan menggunakan jquery, jangan lupa untuk memasukan file **jquery.js** kedalam kode javascript anda.

```
<script type="text/javascript" src="jquery.js"></script>
```

Oke, sekarang mari kita lihat contoh sederhana pemrograman dengan menggunakan jquery.

Kode 12. hello world jquery

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $(".tombol1").click(function(){
        $("p").hide(1000);
    });
    $(".tombol2").click(function(){
        $("p").show(1000);
    });
});
</script>
</head>
<body>
<p>Hello World!</p>
<button class="tombol1">Sembunyikan</button>
<button class="tombol2">Tampilkan</button>
</body>
</html>
```

Sintaks jQuery

Sintaks jquery biasanya dibuat untuk memilih elemen-elemen HTML dan melakukan aksi terhadap elemen yang dipilih.

Sintaks :

```
$(selector).action()
```

- Tanda dollar, untuk mendefinisikan jQuery
- (selector), untuk menunjukkan elemen yang dipilih atau dituju
- action(), adalah jQuery action yang akan dilakukan terhadap elemen yang dipilih.

Contoh :

```
$(this).hide() - menyembunyikan elemen saat ini
```

```
$("p").hide() - menyembunyikan semua paragraf atau konten dari tag <p>
```

```
$(".test").hide() - menyembunyikan elemen yang mempunya class="test"
```

```
$("#test").show() - menampilkan elemen yang mempunyai id="test"
```

Karena hampir segala sesuatu yang kita lakukan bila menggunakan jQuery membaca atau memanipulasi document object model (DOM), kita perlu memastikan bahwa kita mulai menambahkan event segera setelah DOM siap.

Untuk melakukan hal ini, kita menambahkan kode ready event untuk dokumen.

```
$(document).ready(function(){  
    //kode anda di sini  
});
```

Kode di atas berarti kita ingin kode dijalankan apabila halaman HTML telah di-load semuanya. Atau dengan kode javascript biasanya seperti ini :

```
window.onload = function(){ //kode anda di sini }
```

Sekarang mari kita lihat pada contoh **Kode 12**.

```
$(".tomboll").click(function(){  
    $("p").hide(1000);  
});
```

Kode `$(".tomboll")` adalah jQuery selektor. Di mana kita memilih elemen yang mempunyai class="tombol1" untuk kita lakukan sesuatu. `$` sendiri adalah alias untuk jQuery class. Oleh karena `$()` untuk membuat objek jQuery. Kemudian kita tambahkan fungsi `click()`. Ini berguna untuk memberikan event onclick untuk elemen yang kita pilih tadi (dalam hal ini adalah elemen yang mempunya class='tombol1'). Dan kemudian melaksanakan fungsi yang diberikan apabila event onclick terjadi. Jadi artinya apabila elemen dengan class="tombol1" diklik maka lakukan fungsi `$("p").hide(1000);`

Fungsi `hide()` dan `show()` adalah fungsi built in dari jQuery, nanti akan kita lihat lebih lanjut fungsi-fungsi built in dari jQuery.

jQuery Selectors

Selectors memungkinkan Anda untuk memanipulasi elemen HTML sebagai kelompok atau sebagai elemen tunggal. Pada penjelasan sebelumnya kita telah melihat contoh cara memilih elemen HTML menggunakan jQuery.

jQuery element selectors dan attribute selectors memungkinkan Anda untuk memilih elemen HTML (atau kelompok elemen) dengan nama tag, nama atribut atau konten. Selectors memungkinkan Anda untuk memanipulasi elemen HTML sebagai kelompok atau sebagai elemen tunggal.

jQuery Element Selectors

jQuery mirip CSS dalam hal memilih elemen HTML.

`$("p")` memilih semua elemen `<p>`

`$("p.intro")` memilih semua elemen `<p>` yang mempunyai `class="intro"`.

`$("p#demo")` memilih elemen `<p>` yang mempunyai `id="demo"`.

jQuery Attribute Selectors

jQuery mirip XPath dalam hal memilih elemen berdasarkan atribut yang ada.

`$(" [href] ")` memilih semua elemen dengan atribut href.

`$(" [href='#'] ")` memilih semua elemen dengan atribut href bernilai="#".

`$(" [href!='#'] ")` memilih semua elemen dengan atribut href dengan nilai bukan sama dengan "#".

`$(" [href$='.jpg'] ")` memilih semua elemen dengan atribut href yang mengandung ".jpg".

Berikut tabel daftar jQuery Selector.

| Selector | Contoh | Yang dipilih |
|-----------------|--|--|
| * | <code> \$("*")</code> | Semua elemen |
| #id | <code> \$(" #lastname")</code> | Elemen yang mempunyai <code>id=lastname</code> |
| .class | <code> \$(" .intro")</code> | Semua elemen yang mempunyai <code>class="intro"</code> |
| element | <code> \$(" p")</code> | Semua elemen <code><p></code> |
| .class.class | <code> \$(" .intro.demo")</code> | Semua elemen yang mempunyai <code>class=intro</code> dan <code>class=demo</code> |
| :first | <code> \$(" p:first")</code> | Elemen <code><p></code> yang pertama |
| :last | <code> \$(" p:last")</code> | Elemen <code><p></code> yang terakhir |
| :even | <code> \$(" tr:even")</code> | Semua elemen <code><tr></code> yang genap |
| :odd | <code> \$(" tr:odd")</code> | Semua elemen <code><tr></code> yang ganjil |
| :eq(index) | <code> \$(" ul li:eq(3)")</code> | Elemen ke-empat dari suatu list (index starts at 0) |
| :gt(no) | <code> \$(" ul li:gt(3)")</code> | Elemen-elemen dari suatu list dengan index lebih besar dari 3 |
| :lt(no) | <code> \$(" ul li:lt(3)")</code> | Elemen-elemen dari suatu list dengan index kecil dari 3 |
| :not(selector) | <code> \$(" input:not(:empty)")</code> | Semua input elemen yang tidak kosong |
| :header | <code> \$(" :header")</code> | Semua elemen header <code><h1><h2>...</code> |
| :animated | | Semua elemen animasi |
| :contains(text) | <code> \$(" :contains('W3Schools')")</code> | Semua elemen yang mengandung teks 'W3Schools' |
| :empty | <code> \$(" :empty")</code> | Semua elemen yang tidak mempunyai child (elements) nodes |
| :hidden | <code> \$(" p:hidden")</code> | Semua elemen <code><p></code> yang tersembunyi |

| | | |
|---------------------|-----------------------|--|
| :visible | \$("table:visible") | Semua tabel yang visible |
| s1,s2,s3 | \$("th,td,.intro") | Semua elemen yang cocok dengan th, td, .intro |
| [attribute] | \$("[href]") | Semua elemen yang mempunyai atribut href |
| [attribute=value] | \$("[href='#']") | Semua elemen yang mempunyai atribut href bernilai="#" |
| [attribute!=value] | \$("[href!='#']") | Semua elemen yang mempunyai atribut href bernilai <>"#" |
| [attribute\$=value] | \$("[href\$='.jpg']") | Semua elemen yang mempunyai atribut href yang nilainya mengandung ".jpg" |
| :input | \$(":input") | Semua elemen <input> |
| :text | \$(":text") | Semua elemen <input> dengan type="text" |
| :password | \$(":password") | Semua elemen <input> dengan type="password" |
| :radio | \$(":radio") | Semua elemen <input> dengan type="radio" |
| :checkbox | \$(":checkbox") | Semua elemen <input> dengan type="checkbox" |
| :submit | \$(":submit") | Semua elemen <input> dengan type="submit" |
| :reset | \$(":reset") | Semua elemen <input> dengan type="reset" |
| :button | \$(":button") | Semua elemen <input> dengan type="button" |
| :image | \$(":image") | Semua elemen <input> dengan type="image" |
| :file | \$(":file") | Semua elemen <input> dengan type="file" |
| :enabled | \$(":enabled") | Semua elemen <input> yang enabled |
| :disabled | \$(":disabled") | Semua elemen <input> yang disabled |
| :selected | \$(":selected") | Semua elemen <input> yang terseleksi |
| :checked | \$(":checked") | Semua elemen <input> yang ter-check |

jQuery Events

Salah satu kemampuan utama jquery adalah menangani event. Dalam pemograman jquery, biasanya kode-kode pemograman diletakkan di dalam penanganan event yang terjadi pada suatu atau kelompok elemen yang dipilih.

Hampir-hampir mirip dengan Visual Basic, biasanya suatu kode dijalankan apabila ada sesuatu yang terjadi (event) pada suatu elemen. Misalnya, kalau ada tombol di klik, maka action atau kode apa yang dijalankan, apabila ada combobox dipilih, kode apa yang dijalankan, pada contoh jquery sebelumnya :

```
$(".tombol1").click(function(){  
    $("p").hide(1000);  
});
```

Kode di atas berarti apabila terjadi event mengklik elemen yang mempunyai class='tombol1', maka lakukan fungsi hide() terhadap semua element <p>.

Berikut daftar event-event yang dapat terjadi dari elemen HTML.

Event function

`$(document).ready(function)`

`$(selector).blur(function)`

`$(selector).change(function)`

`$(selector).click(function)`

`$(selector).dblclick(function)`

`$(selector).error(function)`

`$(selector).focus(function)`

`$(selector).keydown(function)`

`$(selector).keypress(function)`

`$(selector).keyup(function)`

`$(selector).load(function)`

`$(selector).mousedown(function)`

`$(selector).mouseenter(function)`

`$(selector).mouseleave(function)`

`$(selector).mousemove(function)`

`$(selector).mouseout(function)`

`$(selector).mouseover(function)`

`$(selector).mouseup(function)`

`$(selector).resize(function)`

`$(selector).scroll(function)`

`$(selector).select(function)`

`$(selector).submit(function)`

`$(selector).unload(function)`

Efek-Efek dengan jQuery

Salah satu kemampuan jquery adalah, adanya fungsi-fungsi efek yang siap pakai. Biasanya untuk membuat efek memudar di javascript, kita harus membuat kode yang lumayan panjang. Tapi dengan menggunakan jquery cukup menggunakan fungsi `$(selector).fadeIn()`

Berikut adalah efek-efek siap pakai yang disediakan oleh jquery.

jQuery show() Effect

Berguna untuk menampilkan elemen yang tersembunyi. Untuk mengatur elemen yang tersembunyi melalui CSS adalah `display:none` (bukan `visibility:hidden`).

Sintaks :

```
$(selector).show(speed, callback)
```

| Parameter | Deskripsi |
|-----------|--|
| speed | Opsional. Menentukan kecepatan elemen muncul dari hidden ke visible. Defaultnya adalah 0. Nilainya bisa berupa: <ul style="list-style-type: none">• milliseconds (contoh : 1500)• "slow"• "normal"• "fast" |
| callback | Opsional. Suatu fungsi yang akan dijalankann apabila efek show selesai dijalankan. |

Contoh :

Kode 13. Contoh show()

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $(".tomboll").click(function(){
        $("#p").show(1000,tampilkanAlert);
    });
});
function tampilkanAlert(){
    alert("Paragraf sekarang muncul");
}
</script>
</head>
<body>
<p style="display:none">Ini adalah paragraph tersembunyi.</p>
<button class="tomboll">Show</button>
</body>
</html>
```

jQuery hide() Effect

Berfungsi untuk menyembunyikan elemen yang dipilih.

Sintaks :

```
$(selector).hide(speed,callback)
```

Untuk parameter speed dan callback adalah sama dengan show() effect.

Contoh :

```
$(".tomboll").click(function(){
    $("#p").hide();
});
```

jQuery toggle() Effect

Adalah gabungan fungsi hide dan show. Jadi toggle() berfungsi menampilkan yang tersembunyi, menyembunyikan yang tampak.

Sintaks :

```
$(selector).toggle(speed,callback,switch)
```

| Parameter | Deskripsi |
|-----------|--|
| speed | Opsional. Menentukan kecepatan elemen muncul dari hidden ke visible. Defaultnya adalah 0. Nilainya bisa berupa: <ul style="list-style-type: none">• milliseconds (contoh : 1500)• "slow"• "normal"• "fast" |

| | |
|----------|---|
| callback | Opsional. Suatu fungsi yang akan dijalankan apabila efek show selesai dijalankan. |
| switch | Opsional. Bernilai Boolean <ul style="list-style-type: none">• True, hanya untuk menampilkan semua elemen• False, hanya menyembunyikan semua elemen <p>Jika parameter ini diset, parameter speed dan callback parameters tidak bisa digunakan.</p> |

Contoh :

Kode 14. Contoh toggle()

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function() {
    $(".tomboll1").click(function() {
        $("p").toggle(true);
    });
});
</script>
</head>
<body>
<p>Ini adalah paragraf</p>
<p style="display:none">Ini adalah paragraf lainnya yang belum muncul</p>
<p>Jika bernilai true hanya berfungsi menampilkan, kalau false hanya menyembunyikan</p>
<button class="tomboll1">Tampilkan semua elemen</button>
</body>
</html>
```

jQuery slideDown() Effect

Menampilkan elemen yang tersembunyi, secara efek sliding.

Sintaks :

```
$(selector).slideDown(speed, callback)
```

Untuk parameter speed dan callback adalah sama dengan fungsi show().

Contoh:

```
$(".tomboll1").click(function() {
    $("p").slideDown();
});
```

jQuery slideUp() Effect

Menyembunyikan elemen secara efek sliding.

Sintaks :

```
$(selector).slideUp(speed, callback)
```

Untuk parameter speed dan callback adalah sama dengan fungsi show().

Contoh:

```
$("#tombol1").click(function(){  
    $("#p").slideUp();  
});
```

jQuery slideToggle() Effect

Gabungan antara slideDown() dan slideUp(). Menyembunyikan elemen jika dalam keadaan visible, menampilkan elemen jika dalam keadaan hidden.

Sintaks :

```
$(selector).slideToggle(speed,callback)
```

Untuk parameter speed dan callback adalah sama dengan fungsi show().

Contoh:

Kode 15. Contoh slideToggle()

```
<html>  
<head>  
<script type="text/javascript" src="jquery.js"></script>  
<script type="text/javascript">  
$(document).ready(function(){  
    $("#contact").click(function(){  
        $("#contact-box").slideToggle("slow");  
    });  
});  
</script>  
</head>  
<body>  
  
<p><span id="contact" style="cursor:hand;background-color:#ababab;padding:3;font-family:Verdana">Contact</span></p>  
<div id="contact-box"  
style="background:#98bf21,height:200px;width:300px;position:relative;display:none;padding:10">  
<form>  
Nama : <input type=text><p>  
Email : <input type=text><p>  
Komentar :<textarea rows=5></textarea><p>  
<input type=submit value=kirim>  
</div>  
<p>  
jQuery adalah javascript library, jQuery mempunyai semboyan "write less, do more".  
jQuery dirancang untuk memperingkas kode-kode javascript. JQuery adalah javascript library yang cepat dan ringan untuk menangani dokumen HTML, menangani event, membuat animasi dan interaksi ajax. JQuery dirancang untuk mengubah cara anda menulis javascript.  
<p>Sebelum anda memulai mempelajari jQuery, anda harus mempunyai pengetahuan dasar mengenai HTML, CSS dan Javascript.  
</body>  
</html>
```

jQuery fadeIn() Effect

Menampilkan elemen yang dipilih jika tersembunyi, secara efek memudar.

Sintaks :

```
$(selector).fadeIn(speed, callback)
```

Untuk parameter speed dan callback adalah sama dengan fungsi show().

Contoh:

```
$(".tombol1").click(function(){
    $("p").fadeIn();
});
```

jQuery fadeOut() Effect

Menyembunyikan elemen yang dipilih, secara efek memudar.

Sintaks :

```
$(selector).fadeOut(speed, callback)
```

Untuk parameter speed dan callback adalah sama dengan fungsi show().

Contoh:

Kode 16. Contoh fadeOut()

```
<html>
<head>
<style>
#box
{
background-color:#96BC43;
border:solid 3px #333333;
height:160px;
margin-top:30px;
}
</style>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(function(){
    $('.fadeOut_box').click(function(){
        $('#box').fadeOut("slow");
    });

    $('.fadeIn_box').click(function(){
        $('#box').fadeIn("slow");
    });
});
</script>
<body>
<a href="#" class="fadeOut_box">fadeOut()</a>
<a href="#" class="fadeIn_box">fadeIn()</a>
<div id="box"></div>
</body>
</html>
```

jQuery fadeTo() Effect

Mengatur tingkat keopudaran elemen terpilih menuju tingkat opacity yang ditentukan.

Sintaks :

```
$(selector).fadeTo(speed,opacity,callback)
```

Untuk parameter speed dan callback adalah sama dengan fungsi show().

Untuk parameter opacity bisa bernilai 0 sampai 1.

Contoh :

```
$("#tombol1").click(function(){  
    $("#p").fadeTo(1000,0.6);  
});
```

jQuery animate() Effect

Mengubah suatu elemen dari satu keadaan ke keadaan yang lainnya. Keadaan yang diubah ini berdasarkan CSS.

Nilai properti CSS yang berubah secara berangsur-angsur, hal ini untuk menciptakan efek animasi. Nilai properti CSS yang bisa diubah adalah nilai bertipe angka, baik satuannya pixel atau persen %. Untuk tipe string tidak bisa dianimasikan.

Sintaks :

```
$(selector).animate(styles,speed,easing,callback)
```

| Parameter | Deskripsi |
|-----------|---|
| styles | Wajib. Menentukan properti CSS dan nilainya yang akan di-animasi. |

Properti CSS yang bisa di-animasi :

- backgroundPosition
- borderWidth
- borderBottomWidth
- borderLeftWidth
- borderRightWidth
- borderTopWidth
- borderSpacing
- margin
- marginBottom
- marginLeft
- marginRight
- marginTop
- outlineWidth
- padding
- paddingBottom
- paddingLeft
- paddingRight
- paddingTop
- height
- width

- maxHeight
- maxWidth
- minHeight
- maxwidth
- font
- fontSize
- bottom
- left
- right
- top
- letterSpacing
- wordSpacing
- lineHeight
- textIndent

speed Opsional. Menentukan kecepatan elemen muncul dari hidden ke visible. Defaultnya adalah 0.

Nilainya bisa berupa:

- milliseconds (contoh : 1500)
- "slow"
- "normal"
- "fast"

easing Opsional. Menentukan fungsi easing yang diset pada speed animasi. Built in fungsi easing adalah:

- swing
- linear

callback Opsional. Suatu fungsi yang akan dijalankan apabila efek show selesai dijalankan.

Contoh :

Kode 17. Contoh animate()

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
  $(".tombol1").click(function(){
    $("#box").animate({height:"300px"});
  });
  $(".tombol2").click(function(){
    $("#box").animate({height:"100px"});
  });
});
</script>
```

```
</head>
<body>
<div id="box"
style="background:#98bf21;height:100px;width:100px;margin:6px;">
</div>
<button class="tombol1">Animasi</button>
<button class="tombol2">Reset</button>
</body>
</html>
```

jQuery stop() Effect

Menghentikan animasi yang sedang berjalan.

Sintaks :

```
$(selector).stop(stopAll,goToEnd)
```

| Parameter | Deskripsi |
|-----------|---|
| stopAll | Opsional. Bernilai boolean, menentukan apakah menghentikan semua animasi, termasuk yang ngantri untuk dijalankan pada elemen yang dipilih atau tidak. |
| goToEnd | Opsional. Bernilai Boolean, menentukan apakah animasi yang sedang jalan dikompletkan atau tidak. |

Parameter ini hanya bisa digunakan jika parameter stopAll di-set.

Contoh :

Kode 18. Contoh stop()

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function() {
  $("#start").click(function() {
    $("#box").animate({height:300}, "slow");
    $("#box").animate({width:300}, "slow");
    $("#box").animate({height:100}, "slow");
    $("#box").animate({width:100}, "slow");
  });
  $("#stop").click(function() {
    $("#box").stop(false,true);
  });
});
</script>
</head>

<body>

<p><button id="start">Start Animasi</button><button id="stop">Stop
Animasi</button></p>

<div id="box"
style="background:#ababab;height:100px;width:100px;position:relative">
```

```
</div>  
</body>  
</html>
```

\$(selector).delay()

Berguna untuk men-set delay untuk semua fungsi yang masih ngantri untuk elemen yang dipilih.

Sintaks :

```
$(selector).delay()
```

\$(selector).dequeue()

Menjalankan fungsi berikutnya yang lagi ngantri untuk elemen yang dipilih.

Sintaks :

```
$(selector).dequeue()
```

\$(selector).queue()

Menampilkan fungsi yang ngantri untuk elemen terpilih.

Sintaks :

```
$(selector).queue()
```

jQuery clearQueue() Effect

Menghentikan semua fungsi yang lagi ngantri dan belum dijalankan.

Sintaks :

```
$(selector).stop(queueName)
```

Parameter *queueName* adalah nama antrian yang akan dihentikan.

Contoh :

```
$("#stop").click(function(){  
    $("#box").clearQueue();  
});
```

Manipulasi HTML dengan jQuery

jQuery mempunyai kemampuan yang powerfull dalam hal memanipulasi, mengambil atau menambah konten, dan sebagainya terhadap HTML.

html()

Untuk me-set konten (biasanya kita sebut innerHTML) dari elemen HTML yang dipilih kita gunakan `$(selector).html(content)`. Apabila parameter *content* tidak kita tentukan maka fungsi `html()` berguna untuk mendapatkan konten dari HTML.

Contoh :

Kode 19. Contoh html()

```
<html>  
<head>  
<script type="text/javascript" src="jquery.js"></script>  
<script type="text/javascript">
```

```
$(function(){
    $("#lihat").click(function(){
        var isinya = $("#paragraf").html();
        alert(isinya);
    });
    $("#isikan").click(function(){
        var isilagi = $("#isi").val();
        alert(isilagi);
        $("#paragraf").html(isilagi);
    });
});
</script>
<body>
<button id=lihat>Lihat</button>
<p id=paragraf>
jQuery mempunyai kemampuan yang powerfull dalam hal memanipulasi,
mengambil atau menambah konten, dan sebagainya terhadap HTML
</p>
Teks :
<input type=text id=isi>
<p>
<button id=isikan>Isikan</button>
</body>
</html>
```

val()

Pada contoh di atas kita lihat ada kode

```
var isilagi = $("#isi").val();
```

Kode ini berarti kita mengambil nilai dari `<input type=text id=isi>` dan memasukkannya ke variabel `isilagi`. Sama dengan fungsi `html()`, apabila kita masukkan nilai dari `val()`, maka artinya kita me-set nilai dari elemen yang dipilih. Biasanya `val()` digunakan pada elemen-elemen HTML `<input>`, contoh :

```
$("#isi").val("hallo apa kabar");
```

Kode di atas berarti kita me-set nilai `<input type=text id=isi>` dengan "hallo apa kabar".

attr()

Dengan menggunakan fungsi `attr()`, ini membuat mudah bagi kita untuk mendapatkan nilai dari suatu properti elemen HTML yang kita pilih.

Sintaks :

```
$(selector).attr(properties,nilai);
```

Parameter `properties` adalah nama properti yang ingin kita ambil atau set. Contoh properti misalnya : `id`, `class`, `title`, `src`, `href` dan sebagainya.

Parameter `nilai`, apabila kita isi artinya kita me-set nilai properti yang kita tetukan, apabila kosong artinya kita mengambil nilai properti yang kita tetukan.

Contoh :

Kode 20. Contoh attr()

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(function(){
    $('.link').click(function(){
        var id=$(this).attr("id");
        var class=$(this).attr("class");
        var href=$(this).attr("href");
        alert(id);
        alert(class);
        alert(href);
    });
});
</script>
<a href="http://www.desrizal.com" class="link" id="13">link</a>
</body>
</html>
```

addClass()

Berguna untuk menambahkan atau mengubah class untuk elemen yang dipilih.

Sintaks :

```
$(selector).addClass(namakelas)
```

Contoh :

Kode 21. Contoh addClass()

```
<html>
<head>
<style>
.besar
{
font-size:106px;
}
.kecil
{
font-size:12px;
}
</style>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(function(){
    $('.ubahclass').click(function(){
        $(".besar").addClass("kecil");
    });
});
</script>
<a href="#" class="ubahclass" >Klik </a>
<div class="besar">www.desrizal.com</div>
</body>
</html>
```

Ada banyak fungsi-fungsi lainnya untuk memanipulasi HTML :

| Manipulasi | Description |
|--|---|
| <code>\$(selector).html(content)</code> | Set konten (inner HTML) dari elemen yang dipilih |
| <code>\$(selector).text(text)</code> | Sama seperti <code>html()</code> , tapi tag akan di-escape |
| <code>\$(selector).attr(attr,value)</code> | Set atribut dan nilainya dari elemen yang dipilih |
| <code>\$(selector).val(value)</code> | Set nilai dari elemen yang dipilih |
| Getting Contents | |
| <code>\$(selector).html()</code> | Mengambil konten (inner HTML) dari elemen yang dipilih |
| <code>\$(selector).text()</code> | Mengambil konten teks dari elemen yang dipilih |
| <code>\$(selector).attr(attr)</code> | Mengambil nilai dari suatu atribut elemen yang dipilih. |
| <code>\$(selector).val()</code> | Mengambil nilai/value dari elemen yang dipilih. |
| Adding Content | |
| <code>\$(selector).after(content)</code> | Menambah konten setelah elemen dipilih. |
| <code>\$(selector).before(content)</code> | Menambah konten sebelum elemen dipilih. |
| <code>\$(selector).insertAfter(selector)</code> | Menambah elemen yang ditentukan setelah elemen yang dipilih. |
| <code>\$(selector).insertBefore(selector)</code> | Menambah elemen yang ditentukan sebelum elemen yang dipilih. |
| Manipulate CSS | |
| <code>\$(selector).addClass(content)</code> | Menambah atau mengubah class elemen yang dipilih. |
| <code>\$(selector).removeClass(content)</code> | Menghilangkan class dari elemen yang dipilih |
| <code>\$(selector).toggleClass(content)</code> | Toggle antara penambahan dan penghilangan class dari elemen yang dipilih. |
| <code>\$(selector).hasClass(content)</code> | Cek apakah elemen memiliki class. |
| Adding Inner Content | |
| <code>\$(selector).append(content)</code> | Menambahkan konten ke dalam konten dari elemen yang dipilih |
| <code>\$(selector).prepend(content)</code> | "Prepend" konten ke dalam konten dari elemen yang dipilih |
| <code>\$(content).appendTo(selector)</code> | Menambahkan elemen ke dalam konten dari elemen yang dipilih. |
| <code>\$(content).prependTo(selector)</code> | "Prepend" elemen ke dalam konten dari elemen yang dipilih. |
| Wrapping | |
| <code>\$(selector).wrap(content)</code> | Wrap elemen yang dipilih dalam sebuah konten |
| <code>\$(selector).wrapAll(content)</code> | Wrap semua elemen yang dipilih menjadi satu konten |
| <code>\$(selector).wrapInner(content)</code> | Wrap konten inner child yang dipilih |
| <code>\$(selector).unwrap()</code> | Remove dan replace parents dari elemen yang dipilih |
| Copy, Replace, Remove | |
| <code>\$(content).replaceAll(selector)</code> | Menukar elemen yang dipilih dengan elemen yang ditentukan |
| <code>\$(selector).replaceWith(content)</code> | Menukar elemen yang dipilih dengan konten yang baru |

| | |
|--|---|
| <code>\$(selector).empty()</code> | Remove semua konten dan child elements dari elemen yang dipilih |
| <code>\$(selector).remove()</code> | Remove elemen yang dipilih |
| <code>\$(selector).removeAttr(attr)</code> | Remove atribut yang ditentukan dari elemen yang dipilih |
| <code>\$(selector).clone()</code> | Clone elemen yang dipilih |
| <code>\$(selector).detach()</code> | Remove elemen yang dipilih dari DOM |

Manipulasi CSS dengan jQuery

Manipulasi properti CSS

jQuery juga mempunyai kemampuan dalam hal memanipulasi CSS. Ada 3 fungsi utama dalam jQuery untuk melakukan manipulasi

- `$(selector).css(name, value)`
- `$(selector).css({properties})`
- `$(selector).css(name)`

Fungsi `css()` berguna untuk mendapatkan atau set sebuah properti CSS atau lebih untuk elemen yang dipilih.

Jika parameter **name** dan **value** diisi, artinya kita me-set nilai dari properti CSS.

Untuk me-set nilai-nilai untuk properti CSS lebih dari satu, bisa gunakan **{properties}**

Jika hanya untuk mendapat nilai dari properti CSS yang diinginkan dari elemen yang dipilih cukup isi parameter **name** saja.

Contoh :

Kode 22. Contoh css()

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $("#tombol1").click(function(){
        $(".p1").css("color", "red");
        $(".p2").css({"color": "white", "background-
color": "#ff8954", "font-family": "Arial", "font-
size": "20px", "padding": "5px"});
    });

    $("#tombol2").click(function(){
        var nilai = $(".p1").css("color");
        alert(nilai);
    });
});
</script>
</head>

<body>
<p class="p1">Paragraf ini cuma berubah satu properti CSS</p>
```

```
<p class="p2">Paragraf ini cuma berubah banyak properti CSS</p>
<button id="tombol1">ubah CSS paragraf</button>
<button id="tombol2">Ambil nilai CSS paragraf 1</button>
</body>
</html>
```

Manipulasi CSS Size

Untuk memanipulasi ukuran dari elemen yang dipilih (tinggi dan lebar), maka ada dua fungsi untuk hal ini, yaitu :

- `$(selector).height(value)`
- `$(selector).width(value)`

Jika anda mengisi parameter value, maka artinya anda me-set nilai height atau width dari elemen yang anda pilih.

Jika anda mengosongkan parameter value, maka artinya anda mengambil nilai height atau width dari elemen yang dipilih.

Contoh :

Kode 23. Contoh CSS size

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $("#tombol1").click(function(){
        $("#kotak").width("300px");
        $("#kotak").height("300px");
    });
    $("#tombol2").click(function(){
        var nilai = $("#kotak").width();
        alert(nilai);
    });
});
</script>
</head>

<body>
<div id="kotak" style="background-color: pink; width: 100px; height: 100px;">
Hello
</div>
<p>
<button id="tombol1">ubah ukuran kotak</button>
<button id="tombol2">Ambil nilai ukuran kotak</button>
</body>
</html>
```


Manipulasi Posisi

Untuk melakukan manipulasi posisi dari suatu elemen HTML, berikut beberapa fungsi jQuery untuk melakukannya.

offset()

Untuk mendapatkan atau me-set offset dari elemen yang dipilih. Relatif terhadap dokumen.

Sintaks :

```
$(selector).offset(coordinates)
```

Parameter `coordinates` bisa berupa koordinat dari elemen yang dipilih, contoh :
`offset({top:100,left:0})`

Jika kita mengosongkan parameter `coordinates`, artinya kita mengambil nilai offset dari elemen yang kita pilih.

Fungsi `offset()` akan menghasilkan objek yang mempunyai 2 properti yaitu `top` dan `left`.

Contoh :

Kode 24. Contoh offset()

```
<html>
<head>
<style>
#div1{
background-color : yellow;
width : 100px;
height : 100px;
position : relative;
left : 300;
}
</style>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $("#tombol").click(function(){
        posisi=new Object();
        posisi.left=$("#x").val();
        posisi.top=$("#y").val();
        $("#div1").offset(posisi);
        pos = $("#div1").offset();
        alert("x="+pos.left+",y="+pos.top);
    });
});
</script>
</head>
<body>
<div id=div1>
Hello World!
</div>
x = <input type=text size=4 id=x><br>
y = <input type=text size=4 id=y><br>
<button id=tombol>Set Offset</button>
</body>
</html>
```

offsetParent()

Untuk mendapatkan element parent terdekat dengan posisi yang telah ditentukan.

Sintaks.

```
$(selector).offset()
```

Contoh :

Kode 25. Contoh offsetParent()

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $("button").click(function(){
        $("p").offsetParent().css("background-color", "green");
    });
});
</script>
</head>
<body>
<div style="width:70%;position:absolute;left:100px;top:100px">
<div style="margin:50px;background-color:yellow">
<p>Klik tombol untuk set background color untuk elemen parent terdekat
dari paragraf ini</p>
<div>
</div>
<button>ubah warna latar</button>
</body>
</html>
```

position()

Hampir sama dengan offset(), yaitu untuk mendapatkan atau menentukan posisi dari elemen, tetapi relatif terhadap parent.

Fungsi position() akan menghasilkan objek yang mempunyai 2 properti yaitu top dan left.

Sintaks :

```
$(selector).position()
```

Contoh :

```
$(".tombol").click(function(){
    x=$("#p").position();
    $("#div1").text(x.left);
    $("#div2").text(x.top);
});
```

scrollLeft()

Untuk mendapatkan atau menentukan scroll left offset dari elemen yang dipilih.

Sintaks :

```
$(selector).scrollLeft(offset)
```

Parameter offset adalah nilai dalam pixel elemen akan di scroll dari posisi kiri elemen tersebut.

Jika parameter offset dikosongkan, maka artinya kita mengambil nilai offset dari scroll left elemen tersebut.

Contoh :

```
$(".tombol").click(function(){
    $(".div").scrollLeft(300);
});
```

scrollTop()

Untuk mendapatkan atau menentukan scroll top offset dari elemen yang dipilih.

Sintaks :

```
$(selector).scrollTop(offset)
```

Parameter offset adalah nilai dalam pixel elemen akan di scroll dari posisi atas elemen tersebut.

Jika parameter offset dikosongkan, maka artinya kita mengambil nilai offset dari scroll Top elemen tersebut.

Contoh :

```
$(".tombol").click(function(){
    $(".div").scrollTop(300);
});
```

Web Tab Scroll Content

Sekarang mari kita lihat contoh website, menggunakan jquery. Di mana web ini menggunakan tab-tab, yang me-load konten dengan melakukan animasi scroll terhadap konten yang kita sembunyikan.

Kode 26. Contoh scrollTop()

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script>
$(document).ready(function(){
    $("#home").click(function(){
        $("#konten").animate({ scrollTop: 0},"slow");
    });
    $("#about").click(function(){
        $("#konten").animate({ scrollTop: 500},"slow");
    });
    $("#contact").click(function(){
        $("#konten").animate({ scrollTop: 1000},"slow");
    });
});
</script>
<style>
#konten{
    height : 500px;
    overflow: hidden;
    background-color : yellow;
}
```

```
}
.box{
  height : 500px;
  margin:0;
  padding : 0;
}
.link{
  cursor : pointer;
  background-color : lightyellow;
  padding : 3;
}
body{
  overflow:hidden;
}
#home{
  background-color : green;
}
#about{
  background-color : red;
}
#contact{
  background-color : blue;
}
.aboutme{
  background-color : black;
  color : white;
  font-size : 26pt;
  font-family : garamond;
}
</style>
</head>
<body>
<span class=link id=home>Home</span>
<span class=link id=about>About</span>
<span class=link id=contact>Contact</span>
<div id=konten>
  <div id=home class=box>
    <br><b>Home</b><p>
    <center>
    <font size=7>Selamat Datang</font></center>
  </div>
  <div id=about class=box>
    <br><b>About Me</b>
    <p>
    <span class=aboutme>Hi,namaku Desrizal</span>
    </p>
    <p>
    <span class=aboutme>Lahir di Pekanbaru tahun 1981</span></p>
    <p>
    <span class=aboutme>Punya hobi otak atik komputer</span></p>
  </div>
  <div id=contact class=box>
    <br><b>Contact</b>
  </div>
</div>
```

```
<form method="POST" action="--WEBBOT-SELF--" style="padding:10">
<p>
Nama :<br>
<input type="text" name="T1" size="20"><br>
Emai : <br>
<input type="text" name="T2" size="20"><br>
Komentar :<br>
<textarea rows="6" name="S1" cols="24"></textarea><br>
<input type="submit" value="Submit" name="B1"><input type="reset"
value="Reset" name="B2"></p>
</form>
</div>
</body>
</html>
```

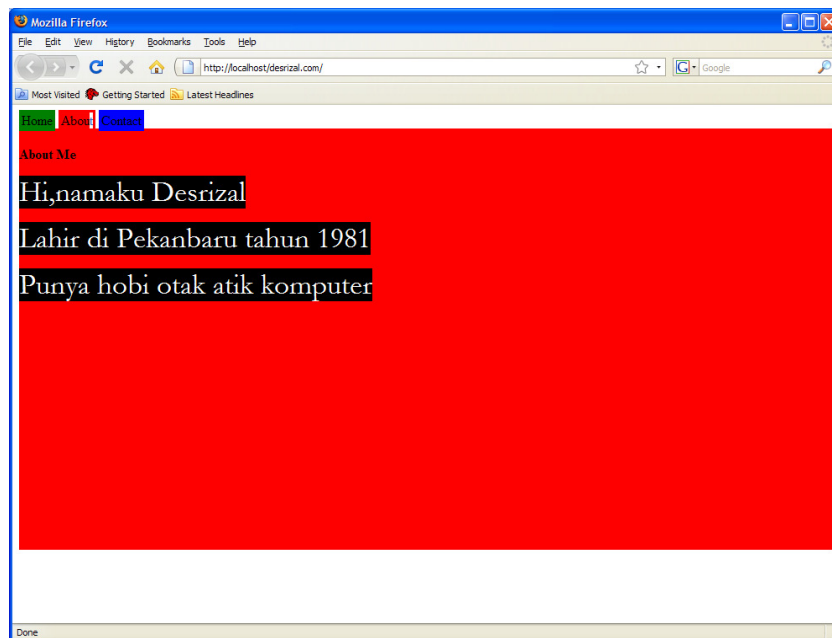
Untuk menghilangkan scroll di <div id=kontent> kita menggunakan CSS

```
overflow:hidden;
```

Ketika kita mengklik salah satu tab, maka kita membuat animasi scrolling menggunakan kode :

```
$("#kontent").animate({ scrollTop: 500}, "slow");
```

Hasil dari kode di atas adalah sebagai berikut :



Gambar 10. Tampilan website Tab scroll content

5

jQuery AJAX

Pada Bab sebelumnya kita telah mengenal apa dan bagaimana jquery. Saatnya kita mempelajari lebih lanjut bagaimana membuat aplikasi Ajax menggunakan jQuery. Library jQuery mempunyai fungsi-fungsi khusus untuk pembuatan Ajax.

Dengan menggunakan jQuery, kita dapat melakukan request data TXT, HTML, XML bahkan JSON dari server menggunakan metode GET atau POST.

Fungsi-Fungsi AJAX jQuery

Berikut adalah fungsi-fungsi yang disediakan oleh jQuery yang berkaitan dengan Ajax.

load()

Mengambil data dari server dan menemukannya ke dalam elemen HTML.

Sintaks :

```
$(selector).load(url,data,callback)
```

Parameter **url** adalah alamat atau nama file di server yang akan kita ambil atau panggil

Parameter **data** bersifat opsional, adalah pasangan key dan value yang dikirim ke server.

Parameter **callback** bersifat opsional, adalah fungsi yang dieksekusi jika data diambil.

Contoh :

Misal kita punya file **data.txt**, yang isinya :

```
Hai ini adalah konten dari file data.txt
```

Lalu kita buat kode ajax untuk me-load konten dari data.txt

Kode 27. contohload.html

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function() {
    $("#tombol").click(function() {
        $('#teks').load('data.txt');
    });
});
</script>
</head>
<body>
<div id="teks">

</div>
<button id="tombol" type="button">Klik Ini</button>
</body>
</html>
```

Dari contoh di atas, apabila kita mengklik tombol Klik Ini, maka akan segera menjalankan fungsi load file **data.txt** dan memasukkannya ke dalam innerHTML `<div id="teks">`.

Sekarang mari kita lihat contoh jika memanggil file berupa PHP dan mengirim kan beberapa data atau parameter untuk diolah di sisi server.

Kode 28. contohload2.html

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $("#tombol").click(function(){

$('#teks').load("data.php","nama=desrizal&email=drz@desrizal.com",beritahu());
    });
});

function beritahu(){
    alert("sudah diambil datanya");
}
</script>
</head>
<body>
<div id="teks">

</div>
<button id="tombol" type="button">Click Me</button>
</body>
</html>
```

Pada contoh di atas kita melakukan request ke server untuk memanggil file PHP data.php dan mengirimkan parameter-parameter dan nilainya dengan metode GET. Kemudian meng-eksekusi fungsi beritahu().

Kode 29. data.php

```
<?php
echo "Anda mengirim parameter nama dengan nilai :
<b>".$_GET['nama']. "</b> <p>";
echo "Email : <b>".$_GET['email']. "</b>.";
?>
```


jQuery.ajax()

Sintaks :

`$.ajax(settingan)`

Fungsi : untuk melakukan asynchronous HTTP (Ajax) request.

Berikut adalah beberapa **settingan** penting untuk fungsi `$.ajax()`

`type`

Yaitu tipe metode request data, yaitu POST atau GET, defaultnya adalah GET

`url`

String yang merupakan URL atau file di server, tujuan kita mengirim request

`data`

Data yang ingin kita kirim ke server untuk diproses, cara penulisannya

"data1=value1&data2=value2"

`cache`

Bisa bernilai true atau false. Jika bernilai false, data yang direquest tidak akan dicache oleh browser

`success`

Fungsi yang kita jalankan jika request telah sukses dijalankan

Untuk settingan lainnya bisa dilihat di <http://api.jquery.com/category/ajax/>

Contoh :

```
$.ajax({
  url: "proses.php",
  data: "nama=desrizal&email=desrizal@yahoo.com&sex=Male",
  cache: false,
  success: function(msg){
    $("#teks").html(msg);
  }
});
```

Aplikasi Input, Update dan Delete

Sekarang mari kita buat aplikasi sederhana menggunakan jquery dan PHP untuk pemrosesan di sisi server. Pada contoh-contoh yang akan saya berikan, kita akan banyak menggunakan fungsi-fungsi jquery seperti `val()`, `html()`, `show()`, `hide()` dan lainnya. Untuk tutorial mengenai fungsi tersebut, silahkan baca di [pengenalan jquery](#). Pada contoh berikut kita akan membuat aplikasi data karyawan

Struktur Tabel datakaryawan di MySQL

Pertama mari kita buat dulu tabel datakaryawan di database MySQL

```
CREATE TABLE `datakaryawan` (  
  `nik` varchar(10) NOT NULL,  
  `nama` varchar(50) NOT NULL,  
  `email` varchar(50) NOT NULL,  
  `alamat` varchar(200) NOT NULL,  
  PRIMARY KEY (`nik`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
INSERT INTO `datakaryawan` (`nik`, `nama`, `email`, `alamat`) VALUES  
( 'FI786578', 'Bagaskara Antara', 'bagas@yahoo.com', 'Jakarta'),  
( 'FI885245', 'Budix', 'budi@gmail.com', 'USA'),  
( 'FI889456', 'Desrizal', 'drz@desrizal.com', 'Tembagapura');
```

Kode HTML dan jQuery

Mari kita mulai membuat kode HTML dan jQuery, buat file html, kita beri nama **index.html**.

```
<html>  
<head><title>Ajax jQuery by Desrizal</title>  
<script type="text/javascript" src="jquery-1.4.3.min.js">  
</script>  
<script>  
var nik;  
var nama;  
var email;  
var alamat;  
var datanya;  
$(document).ready(function(){  
  //meload option NIK dari database  
  $("#nik").load("proses.php","op=ambiloption");  
  
  //jika ada event onchange ambil data dari database  
  $("#nik").change(function(){  
    //ambil nilai nik dari form  
    nik = $("#nik").val();  
  
    //tampilkan status loading dan animasinya  
    $("#status").html("Loading...");  
    $("#loading").show();  
  
    //lakukan pengiriman dan pengambilan data  
    $.ajax({  
      url: "proses.php",  
      data: "op=ambildata&nik="+nik,  
      cache: false,  
      success: function(msg){  
        //karna di server pembatas setiap data adalah |  
        //maka kita split dan akan membentuk array  
        data = msg.split("|");  
  
        //masukkan ke masing-masing textfield
```

```
        $("#nama").val(data[0]);
        $("#email").val(data[1]);
        $("#alamat").val(data[2]);
        //hilangkan status dan animasi loading
        $("#status").html("");
        $("#loading").hide();
    }
    });
});

//jika tombol update diclick
$("#tupdate").click(function(){
    //ambil nilai-nilai dari masing-masing input
    nik = $("#nik").val();
    if(nik=="Pilih NIK"){
        alert("Pilih dulu NIK");
        exit();
    }
    nama = $("#nama").val();
    email = $("#email").val();
    alamat = $("#alamat").val();
    datanya = "&nik="+nik+"&nama="+nama+"&email="+email;
    datanya = datanya+"&alamat="+alamat;
    //tampilkan status Updating dan animasinya
    $("#status").html("Lagi diupdate...");
    $("#loading").show();
    $.ajax({
        url: "proses.php",
        data: "op=update"+datanya,
        cache: false,
        success: function(msg){
            if(msg=="sukses"){
                $("#status").html("Update Berhasil...");
            }else{
                $("#status").html("ERROR..");
            }
            $("#loading").hide();
        }
    });
});

//jika tombol DEL diklik
$("#tdelete").click(function(){
    nik = $("#nik").val();
    if(nik=="Pilih NIK"){
        alert("Pilih dulu NIK");
        exit();
    }
    $("#status").html("Lagi didetele...");
    $("#loading").show();
    $.ajax({
        url: "proses.php",
        data: "op=delete&nik="+nik,
        cache: false,
```

```
success: function(msg){
    if(msg=="sukses"){
        $("#status").html("Delete Berhasil...");
    }else{
        $("#status").html("ERROR..");
    }
    $("#nama").val("");
    $("#email").val("");
    $("#alamat").val("");
    $("#loading").hide();
    $("#nik").load("proses.php","op=ambiloption");
}
});
});

//jika link Tambah Data Karyawan diklik
$("#formtambah").click(function(){
    $("#formnik").show();
    $("#nik2").val("");
    $("#nama").val("");
    $("#email").val("");
    $("#alamat").val("");
});

//jika tombol TAMBAH diklik
$("#ttambah").click(function(){
    //ambil nilai-nilai dari masing-masing input
    nik = $("#nik2").val();
    if(nik==""){
        alert("NIK belum diisi\nKlik Tambah Data Karyawan");
        exit();
    }
    nama = $("#nama").val();
    email = $("#email").val();
    alamat = $("#alamat").val();
    datanya = "&nik="+nik+"&nama="+nama+"&email="+email;
    datanya = datanya+"&alamat="+alamat;
    $("#status").html("Lagi ditambah...");
    $("#loading").show();
    $.ajax({
        url: "proses.php",
        data: "op=tambah"+datanya,
        cache: false,
        success: function(msg){
            if(msg=="sukses"){
                $("#status").html("Berhasil ditambah...");
            }else{
                $("#status").html("ERROR..");
            }
            $("#loading").hide();
            $("#nik").load("proses.php","op=ambiloption");
            $("#formnik").hide();
            $("#nik2").val("");
        }
    })
}
```

```
        });  
    });  
});  
</script>  
</head>  
<body>  
Nomor Induk Karyawan :  
<select id="nik"></select>  
<br>  
<a id="formtambah" style="cursor:pointer;color:red">  
<u>Tambah Data Karyawan</u></a>  
<p style="display:none" id="formnik">  
NIK :<br>  
<input type="text" id="nik2">  
</p>  
<p>  
Nama :<br>  
<input type="text" id="nama"><p>  
Email :<br>  
<input type="text" id="email"><p>  
Alamat :<br>  
<input type="text" id="alamat" size="30"><p>  
<button id="tupdate">UPDATE</button>  
<button id="tdelete">DEL</button>  
<button id="ttambah">TAMBAH</button>  
<br>  
<span id="status"></span>  
  
</body>  
</html>
```

Kode PHP, Pemrosesan Di Sisi Server

Pada kode jquery di atas, ajax melakukan request ke file proses.php

```
<?php  
mysql_connect("localhost","root","");  
mysql_select_db("test");  
  
$op = $_GET['op'];  
  
if($op == "ambiloption"){  
    $option = mysql_query("SELECT nik FROM datakaryawan");  
    echo "<option>Pilih NIK</option>\n";  
    while($op = mysql_fetch_array($option)){  
        echo "<option>".$op['nik']."</option>\n";  
    }  
}else if($op == "ambildata"){  
    $nik = $_GET['nik'];  
    $data = mysql_query("SELECT * FROM datakaryawan WHERE nik='$nik'");  
    $d = mysql_fetch_array($data);  
    echo $d['nama']."|".$d['email']."|".$d['alamat'];
```

```
}else if($op == "update"){
    $nik = $_GET['nik'];
    $nama = htmlspecialchars($_GET['nama']);
    $email = htmlspecialchars($_GET['email']);
    $alamat = htmlspecialchars($_GET['alamat']);
    $update = mysql_query("UPDATE datakaryawan
        SET nama='$nama',
            email='$email',
            alamat='$alamat'
        WHERE nik='$nik'");
    if($update){
        echo "sukses";
    }else{
        echo "error";
    }
}
}
}else if($op == "delete"){
    $nik = $_GET['nik'];
    $del = mysql_query("DELETE FROM datakaryawan WHERE nik='$nik'");
    if($del){
        echo "sukses";
    }else{
        echo "error";
    }
}
}
}else if($op == "tambah"){
    $nik = $_GET['nik'];
    $nama = htmlspecialchars($_GET['nama']);
    $email = htmlspecialchars($_GET['email']);
    $alamat = htmlspecialchars($_GET['alamat']);
    $tambah = mysql_query("INSERT INTO datakaryawan
        VALUES('$nik', '$nama', '$email', '$alamat')");
    if($tambah){
        echo "sukses";
    }else{
        echo "ERROR";
    }
}
}
?>
```