

Entri, Edit, Delete, Tampil dengan PHP dan Ajax JQuery

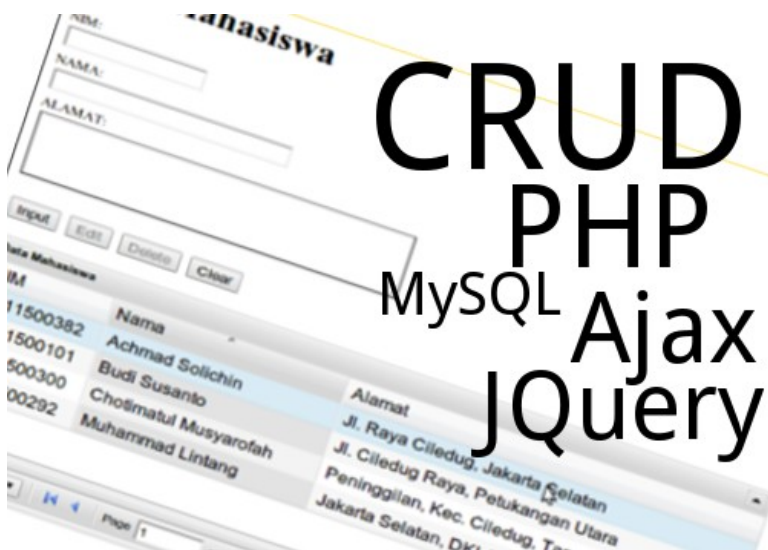
Tutorial ini ditulis oleh **Achmad Solichin**
dan sudah dipublikasikan di <http://achmatim.net> pada tanggal 2 September 2012

Dalam setiap kesempatan membahas mengenai aplikasi berbasis web (*web application*), baik di dalam sesi mengajar matakuliah Pemrograman Web di **Universitas Budi Luhur**¹ maupun sesi training terkait web, saya selalu menekankan bahwa hanya ada 4 (empat) proses dasar dalam aplikasi berbasis web. Aplikasi web yang sederhana hingga kompleks selalu berkaitan dengannya. Keempat proses tersebut adalah proses entri (input), ubah (edit), hapus (delete) dan tampil (*show*). Istilah lain dari

keempat proses tersebut adalah **CRUD (Create, Read, Update and Delete)**. Jadi kuasailah keempatnya maka selebihnya hanya variasi dari keempat proses tersebut. Sebagai contoh dalam proses update status di situs jejaring sosial facebook merupakan proses Entri atau Insert ke database, saat memperbaiki foto profil adalah proses edit (update) dan saat kita menghapus salah satu teman aalah proses delete.

Mengingat pentingnya proses tersebut, saya memberikan contoh khusus di bab terakhir dari buku **Pemrograman Web dengan PHP & MySQL**² berupa proses dasar entri, edit, delete dan tampil dengan PHP dan MySQL. Sebelum melanjutkan tutorial ini, tidak ada salahnya jika Anda mencoba contoh tersebut, terutama bagi Anda yang belum pernah mencoba program serupa. Saya juga pernah mempublikasikan artikel entri, edit, delete dan tampil PHP yang memanfaatkan Macromedia Dreamweaver.

Lalu apa yang akan dipelajari di dalam tutorial ini? Apa yang disampaikan dalam tutorial ini kurang lebih sama dengan tutorial saya sebelumnya, yaitu proses entri, edit, delete dan tampil data dengan PHP dan MySQL. Namun kali ini, saya ingin menambahkan konsep Ajax



1 <http://budiluhur.ac.id>

2 <http://achmatim.net/buku-gratis/pemrograman-web-dengan-php-dan-mysql/>

(Asynchronous Javascript and XML) dengan menggunakan library JQuery. Dengan konsep Ajax, maka proses entri, edit, delete dan tampil dapat dilakukan dengan lebih menarik.

Tujuan atau Rancangan Akhir Program

Untuk memberikan gambaran yang lebih jelas mengenai hasil akhir contoh program yang dibahas dalam tutorial ini, berikut beberapa hal penting:

- Untuk memudahkan pemahaman, program hanya menangani 1 (satu) tabel MySQL yaitu tabel "mahasiswa" yang terdiri dari field NIM, NAMA dan ALAMAT.
- Program hanya terdiri dari 1 (satu) file utama (tidak termasuk library yang digunakan)
- Program menggunakan library Ajax **JQuery**.
- Untuk penanganan form, digunakan plugin **JQuery Form**.
- Untuk validasi form, digunakan plugin **JQuery Validation**.
- Untuk menampilkan data dalam bentuk tabel, digunakan plugin **JQuery Flexigrid**
- Proses EDIT dan DELETE dilakukan dengan memilih dan klik pada data yang akan diedit/didelete.

Langsung aja kita mulai **step by step** untuk membuatnya.

1. Siapkan Database dan Tabel

Langkah pertama tentu persiapkan database dan tabel yang diperlukan. Seperti sudah disebutkan di rancangan akhir program di atas bahwa kita akan menyederhanakan struktur data yang akan dibuat. Dalam hal ini, kita akan membuat database dengan nama "**demo**" dan tabel dengan nama "**mahasiswa**". Berikut ini struktur dari tabel mahasiswa yang akan dibuat:

FIELD	TYPE	LENGTH	DESCRIPTION
nim	VARCHAR	10	NIM Mahasiswa
nama	VARCHAR	30	NAMA Mahasiswa
alamat	TEXT	-	Alamat Mahasiswa

Dan berikut ini perintah SQL untuk membuat database dan tabel di atas. Perintah dapat ditulis dan dijalankan di MySQL client kesukaan kita.

Perintah SQL untuk CREATE DATABASE dan CREATE Tabel

```
1 CREATE DATABASE demo;  
2 CREATE TABLE mahasiswa (  
3     nim VARCHAR(10) NOT NULL,  
4     nama VARCHAR(30) NOT NULL,  
5     alamat TEXT,  
6     PRIMARY KEY(nim)  
7 );
```

2. Buat Halaman beserta Form Inputan

Bagi Anda yang sudah belajar perintah-perintah HTML (*Hypertext Mark up Language*) tentunya tidak akan kesulitan dalam membuat form inputan untuk tabel mahasiswa di atas. Bentuk dan tampilan dari form inputan dapat disesuaikan sesuai keinginan Anda, namun yang penting untuk diperhatikan bahwa pastikan setiap inputan harus memiliki atribut **name** dan **id** agar dapat diakses dengan mudah melalui JQuery nantinya.

Berikut ini *source code* HTML awal yang menampilkan form inputan data mahasiswa. Simpan program berikut ini sebagai file "index.php".

Perintah HTML untuk Membuat Form Inputan

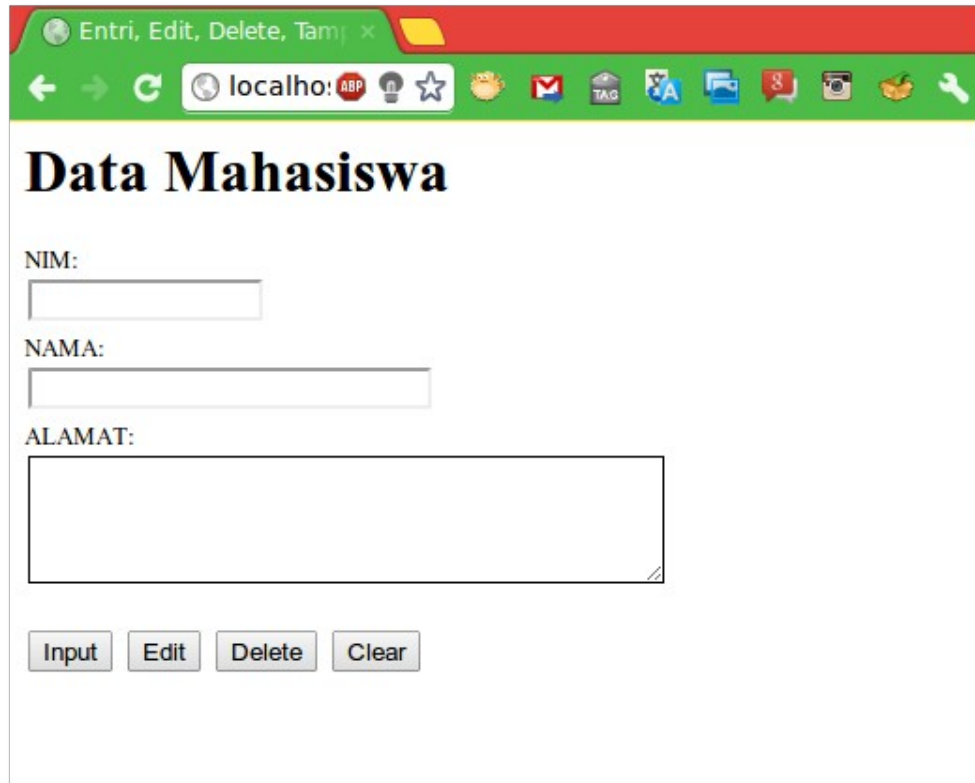
```

1 <html>
2   <head>
3     <title>Entri, Edit, Delete, Tampil Data dengan PHP dan
Ajax</title>
4     <style type="text/css">
5       .labelfrm {
6         display:block;
7         font-size:small;
8         margin-top:5px;
9       }
10      .error {
11        font-size:small;
12        color:red;
13      }
14    </style>
15  </head>
16  <body>
17    <h1>Data Mahasiswa</h1>
18    <form action="" method="post" id="frm">
19      <label for="nim" class="labelfrm">NIM: </label>
20      <input type="text" name="nim" id="nim" maxlength="10"
class="required" size="15"/>
21      <label for="nama" class="labelfrm">NAMA: </label>
22      <input type="text" name="nama" id="nama" size="30"
class="required"/>
23
24      <label for="alamat" class="labelfrm">ALAMAT: </label>
25      <textarea name="alamat" id="alamat" cols="40" rows="4"
class="required"></textarea>
26
27      <label for="submit" class="labelfrm">&nbsp;</label>
28      <input type="submit" name="Input" value="Input"
id="input"/>
29      <input type="submit" name="Edit" value="Edit"

```

```
30 id="edit"/>
    <input type="submit" name="Delete" value="Delete"
31 id="delete"/>
    <input type="reset" name="Clear" value="Clear"
32 id="clear"/>
    </form>
33 </body>
34 </html>
```

Pada program di atas terlihat bahwa terdapat satu buah form dengan id **frm**, yang memiliki 3 (tiga) buah inputan text dan textarea serta 4 (empat) buah tombol yaitu **Input**, **Edit**, **Delete** dan **Clear**. Masing-masing inputan diberi nama dan ID. Untuk inputan NIM, NAMA dan ALAMAT juga ditambahkan atribut class dengan value "required" yang berarti inputan harus diisi (menggunakan plugin JQuery Validate). Berikut ini tampilan dari program sederhana di atas.



The screenshot shows a web browser window with the title 'Entri, Edit, Delete, Tampil'. The address bar shows 'localhost:8080'. The main content area has a heading 'Data Mahasiswa'. Below the heading are three input fields: 'NIM:', 'NAMA:', and 'ALAMAT:'. Each field is followed by a text input box. Below the input boxes are four buttons: 'Input', 'Edit', 'Delete', and 'Clear'.

Gambar 1: Tampilan Form Inputan Data Mahasiswa

3. Koneksikan ke Database MySQL

Karena program dirancang untuk berhubungan dengan database MySQL, maka terlebih dahulu harus dikoneksikan antara PHP dengan MySQL. Untuk melakukan koneksi ke MySQL, dalam tutorial ini dicontohkan dengan cara yang paling mudah. Kita akan menggunakan fungsi **mysql_connect()** dan **mysql_select_db()** dari PHP. Pastikan Anda mengetahui informasi letak hostname, username, password untuk masuk ke server MySQL dan juga nama database yang akan digunakan sudah disiapkan.

Berikut ini potongan program untuk melakukan koneksi. Letakkan di awal file index.php yang telah dibuat di langkah sebelumnya atau dapat juga dibuat di file khusus dan di-include-kan.

Potongan Program untuk Koneksi PHP dan MySQL

```

1 <?php
2 /* koneksi ke db */
3 mysql_connect("localhost", "root", "qwerty") or die(mysql_error());
4 mysql_select_db("demo") or die(mysql_error());
5 /* akhir koneksi db */
6 ?>

```

4. Program Input ke MySQL

Setelah program PHP terkoneksi dengan MySQL, selanjutnya dibuat program untuk proses input (insert) ke database. Secara umum, untuk membuat program input ke database, kita ambil dulu data yang dientri oleh user sesuai dengan method dari form. Untuk me-refresh kembali mengenai penanganan form di PHP, Anda dapat mempelajari Bab 4 dari buku **Pemrograman Web dengan PHP & MySQL**³ yang dapat didownload di website saya. Setelah data dari form diambil, susun string perintah query untuk INSERT ke database. Bagaimana bentuk perintah INSERT dan contoh penggunaannya juga dapat dipelajari di buku saya tersebut, terutama di Bab 12 dan 13.

Kurang lebih berikut ini, potongan program untuk input / entri ke MySQL. Anda dapat meletakkannya setelah baris perintah koneksi database.

Potongan Program untuk Entri Data ke Database

```

1 /* penanganan form */
2 if (isset($_POST['Input'])) {
3     $nim      = strip_tags($_POST['nim']);
4     $nama     = strip_tags($_POST['nama']);
5     $alamat   = strip_tags($_POST['alamat']);
6
7     //input ke db

```

3 <http://achmatim.net/buku-gratis/pemrograman-web-dengan-php-dan-mysql/>

```
8      $query = sprintf("INSERT INTO mahasiswa VALUES('%s', '%s',  
9      '%s')",  
10      mysql_escape_string($nim),  
11      mysql_escape_string($nama),  
12      mysql_escape_string($alamat)  
13      );  
14      $sql = mysql_query($query);  
15      if ($sql) {  
16          echo "Data berhasil disimpan";  
17      } else {  
18          echo "Data gagal disimpan ";  
19          echo mysql_error();  
20      }
```

Berikut ini sedikit penjelasan dari potongan program diatas:

- Baris 2: memastikan tombol "Input" ditekan oleh user menggunakan fungsi isset() yang akan bernilai TRUE jika suatu variabel telah terbentuk (jika user menekan suatu tombol maka variabel akan terbentuk)
- Baris 3-5: proses pengambilan data dari form. Fungsi strip_tags() digunakan untuk memastikan bahwa inputan tidak mengandung tag-tag HTML atau javascript yang mungkin "berbahaya".
- Baris 8-12: pembentukan perintah untuk INSERT ke database. Fungsi mysql_escape_string() digunakan untuk menambahkan escape (backslashes, \) jika terdapat karakter khusus seperti tanda kutip dan dollar. Hal tersebut dilakukan sebagai salah satu cara mencegah praktek SQL Injection.
- Baris 13: eksekusi perintah MySQL untuk mengentri data ke database menggunakan fungsi mysql_query().
- Baris 14-19: notifikasi apakah proses berhasil atau gagal.

Pada tahap ini, sebaiknya Anda coba program dengan menginput suatu data melalui form. Pastikan tidak ada error dan data sudah masuk ke database (untuk melihat data, langsung gunakan tools MySQL yang Anda punya).

5. Sertakan Library JQuery dan Plugin JQuery Form

Hingga langkah ke-4 di atas, kita masih belum menggunakan konsep Ajax (Asynchronous Javascript and XML). Apa itu Ajax? Secara singkat, ajax merupakan konsep pengembangan web yang memungkinkan antara client dan server dapat berkomunikasi secara asynchronous, suatu proses bisa saja berjalan di belakang halaman. Contoh penerapannya saya rasa dengan mudah kita temui di banyak web, seperti pada situs Facebook, saat kita meng-update status, hanya bagian status aja yang berubah sedangkan keseluruhan halaman tidak berubah (*refresh*). Lebih lengkap mengenai Ajax, dapat Anda cari artikelnya di Google.

Pada tutorial ini, digunakan library utama **JQuery** yang dapat didownload di situs <http://code.jquery.com>. Sedangkan untuk penanganan form dengan teknologi Ajax, digunakan plugin **JQuery Form** yang dapat didownload di situs <http://malsup.com/jquery/form/>. Dengan plugin ini, kita dapat mengubah proses penanganan form secara klasik seperti pada langkah ke-4 di atas, menjadi proses penanganan form dengan “gaya” Ajax. Saya memilih plugin ini dengan alasan kemudahan dalam penerapannya.

Untuk menyertakan library JQuery dan JQuery Form ke dalam halaman yang sudah kita buat sebelumnya, tambahkan dua baris perintah berikut ini di antara tag <head> (diasumsikan bahwa kedua plugin tersimpan di folder /libs).

Potongan Program untuk Menyertakan Library JQuery dan Plugin JQuery Form

```
1 <script type="text/javascript" src="libs/jquery.min.js"></script>
2 <script type="text/javascript" src="libs/jquery.form.js"></script>
```

Dan setelah library disertakan, kita dapat memanggil plugin JQuery Form untuk mengaktifkan proses Ajax di dalam form. Pemanggilannya sangat mudah, cukup atur opsi / pilihan dan aktifkan berdasarkan selector dari form yang akan dipanggil. Berikut ini program lengkap yang sudah ditambahkan pemanggilan plugin JQuery Form.

Program Lengkap Setelah Ditambahkan Library JQuery dan Plugin JQuery Form

```
1 <?php
2 /* koneksi ke db */
3 mysql_connect("localhost", "root", "qwerty") or die(mysql_error());
4 mysql_select_db("demo") or die(mysql_error());
5 /* akhir koneksi db */
6
7 /* penanganan form */
8 if (isset($_POST['Input'])) {
9     $nim      = strip_tags($_POST['nim']);
10    $nama     = strip_tags($_POST['nama']);
11    $alamat  = strip_tags($_POST['alamat']);
12
13    //input ke db
14    $query= sprintf("INSERT INTO mahasiswa VALUES('%s', '%s', '%s')",
15                    mysql_escape_string($nim),
16                    mysql_escape_string($nama),
17                    mysql_escape_string($alamat)
18                );
19    $sql = mysql_query($query);
20    $pesan = "";
21    if ($sql) {
22        $pesan = "Data berhasil disimpan";
```

```

23     } else {
24         $pesan = "Data gagal disimpan ";
25         $pesan .= mysql_error();
26     }
27     $response = array('pesan'=>$pesan, 'data'=>$_POST);
28     echo json_encode($response);
29     exit;
30 }
31 ?>
32 <html>
33     <head>
34         <title>Entri, Edit, Delete, Tampil Data dengan PHP dan
35 Ajax</title>
36         <style type="text/css">
37             .labelfrm {
38                 display:block;
39                 font-size:small;
40                 margin-top:5px;
41             }
42             .error {
43                 font-size:small;
44                 color:red;
45             }
46         </style>
47         <script type="text/javascript"
48 src="libs/jquery.min.js"></script>
49         <script type="text/javascript"
50 src="libs/jquery.form.js"></script>
51         <script type="text/javascript">
52             $(document).ready(function() {
53                 //aktifkan ajax di form
54                 var options = {
55                     success      : showResponse,
56                     resetForm   : true,
57                     clearForm   : true,
58                     dataType    : 'json'
59                 };
60                 $('#frm').ajaxForm(options);
61             });
62             function showResponse(responseText, statusText) {
63                 var data = responseText['data'];
64                 var pesan = responseText['pesan'];
65                 alert(pesan);
66             }
67         </script>
68     </head>
69     <body>
70         <h1>Data Mahasiswa</h1>

```



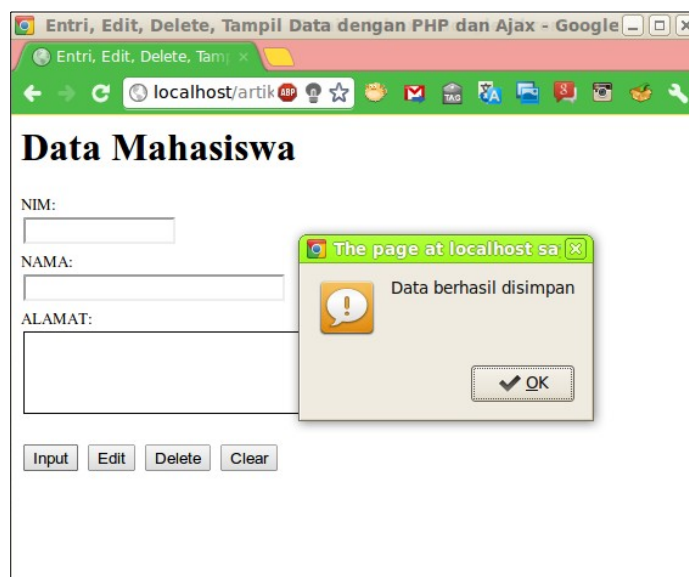
```

65         <form action="" method="post" id="frm">
66             <label for="nim" class="labelfrm">NIM: </label>
67             <input type="text" name="nim" id="nim" maxlength="10"
class="required" size="15"/>
68
69             <label for="nama" class="labelfrm">NAMA: </label>
70             <input type="text" name="nama" id="nama" size="30"
class="required"/>
71
72             <label for="alamat" class="labelfrm">ALAMAT: </label>
73             <textarea name="alamat" id="alamat" cols="40" rows="4"
class="required"></textarea>
74
75             <label for="submit" class="labelfrm">&nbsp;</label>
76             <input type="submit" name="Input" value="Input"
id="input"/>
77             <input type="submit" name="Edit" value="Edit"
id="edit"/>
78             <input type="submit" name="Delete" value="Delete"
id="delete"/>
79             <input type="reset" name="Clear" value="Clear"
id="clear"/>
80         </form>
81     </body>
82 </html>

```

Setelah library JQuery dan plugin JQuery Form disertakan (baris 42 dan 43), pada baris 45-55 diterapkan plugin JQuery Form pada form inputan (id form = #frm) dengan beberapa pengaturan (opsi). Opsi pada baris 48 berarti pada saat form inputan berhasil disubmit maka akan otomatis memanggil fungsi **showResponse()** yang didefinisikan pada baris 56-60. Opsi baris 49 dan 50 membuat form kembali ke posisi awal (reset) setelah proses submit terjadi, sedangkan opsi pada baris 51 merupakan pengaturan data yang dikirimkan secara asynchronous dalam bentuk JSON (Javascript Object Notation).

Pada saat form disubmit (ditekan tombol Input), data inputan tetap akan dikirim dan diproses oleh PHP melalui action yang telah ditentukan (baris 8-30). Namun pengiriman data tidak akan me-refresh halaman karena dilakukan di belakang halaman (asynchronous). Setelah proses penyimpanan data ke database berhasil dilakukan selanjutnya PHP akan mengirimkan "pesan" dalam bentuk JSON ke javascript. Perhatikan baris 27-29, yang mengubah variabel array berisi pesan menjadi format JSON menggunakan fungsi **json_encode()** dari PHP, serta mengirimkannya ke fungsi javascript **showResponse()** di baris 56-60. Hasil dari program di atas, setelah ditambahkan plugin JQuery Form tampak pada Gambar 2.



Gambar 2: Proses Entri setelah ditambahkan plugin JQuery Form

6. Validasi dengan plugin JQuery Validation

Seperti sudah pernah dijelaskan dan dicontohkan di tutorial saya berjudul **Membuat Validasi Form dengan JQuery Validation**⁴, selanjutnya ditambahkan validasi form inputan dengan aturan semua field inputan yang harus diisi (required) dan khusus untuk inputan NIM hanya boleh diisi 10 digit angka (tidak boleh diisi huruf). Berikut ini source code program setelah ditambahkan plugin JQuery Validation.

Program Lengkap Setelah Ditambahkan Library JQuery dan Plugin JQuery Form

```

1  <?php
2  /* koneksi ke db */
3  mysql_connect("localhost", "root", "qwerty") or die(mysql_error());
4  mysql_select_db("demo") or die(mysql_error());
5  /* akhir koneksi db */
6
7  /* penanganan form */
8  if (isset($_POST['Input'])) {
9      $nim      = strip_tags($_POST['nim']);
10     $nama     = strip_tags($_POST['nama']);
11     $alamat    = strip_tags($_POST['alamat']);
12
13     //input ke db
14     $query= sprintf("INSERT INTO mahasiswa VALUES('%s', '%s', '%s')",
15                     mysql_escape_string($nim),
16                     mysql_escape_string($nama),

```

4 <http://achmatim.net/2012/05/22/membuat-validasi-form-dengan-jquery-validation/>

```

17         mysql_escape_string($alamat)
18     );
19     $sql = mysql_query($query);
20     $pesan = "";
21     if ($sql) {
22         $pesan = "Data berhasil disimpan";
23     } else {
24         $pesan = "Data gagal disimpan ";
25         $pesan .= mysql_error();
26     }
27     $response = array('pesan'=>$pesan, 'data'=>$_POST);
28     echo json_encode($response);
29     exit;
30 }
31 ?>
32 <html>
33     <head>
34         <title>Entri, Edit, Delete, Tampil Data dengan PHP dan
35 Ajax</title>
36         <style type="text/css">
37             .labelfrm {
38                 display:block;
39                 font-size:small;
40                 margin-top:5px;
41             }
42             .error {
43                 font-size:small;
44                 color:red;
45             }
46         </style>
47         <script type="text/javascript"
48 src="libs/jquery.min.js"></script>
49         <script type="text/javascript"
50 src="libs/jquery.form.js"></script>
51         <script type="text/javascript"
52 src="libs/jquery.validate.min.js"></script>
53         <script type="text/javascript">
54             $(document).ready(function() {
55                 //aktifkan ajax di form
56                 var options = {
57                     success      : showResponse,
58                     beforeSend: function(){
59                         return $("#frm").valid();
60                     },
61                     resetForm : true,
62                     clearForm : true,
63                     dataType  : 'json'
64                 };
65                 $('#frm').ajaxForm(options);

```

```

58
59         //validasi form dgn jquery validate
60         $('#frm').validate({
61             rules: {
62                 nim : {
63                     digits: true,
64                     minlength:10,
65                     maxlength:10
66                 }
67             },
68             messages: {
69                 nim: {
70                     required: "Kolom nim harus diisi",
71                     minlength: "Kolom nim harus terdiri
dari 10 digit",
72                     maxlength: "Kolom nim harus terdiri
dari 10 digit",
73                     digits: "NIM harus berupa angka"
74                 },
75                 nama: {
76                     required: "Nama harus diisi dengan
benar"
77                 }
78             }
79         });
80     });
81     function showResponse(responseText, statusText) {
82         var data = responseText['data'];
83         var pesan = responseText['pesan'];
84         alert(pesan);
85     }
86     </script>
87 </head>
88 <body>
89     <h1>Data Mahasiswa</h1>
90     <form action="" method="post" id="frm">
91         <label for="nim" class="labelfrm">NIM: </label>
92         <input type="text" name="nim" id="nim" maxlength="10"
class="required" size="15"/>
93
94         <label for="nama" class="labelfrm">NAMA: </label>
95         <input type="text" name="nama" id="nama" size="30"
class="required"/>
96
97         <label for="alamat" class="labelfrm">ALAMAT: </label>
98         <textarea name="alamat" id="alamat" cols="40" rows="4"
class="required"></textarea>
99
100        <label for="submit" class="labelfrm">&nbsp;</label>

```

```

101         <input type="submit" name="Input" value="Input"
id="input"/>
102         <input type="submit" name="Edit" value="Edit"
id="edit"/>
103         <input type="submit" name="Delete" value="Delete"
id="delete"/>
104         <input type="reset" name="Clear" value="Clear"
id="clear"/>
105     </form>
106 </body>
107 </html>

```

Untuk menerapkan validasi di form inputan, pertama-tama sertakan plugin JQuery Validate seperti terlihat pada baris 44 program di atas. Selanjutnya terapkan rule yang sudah ditetapkan pada form sesuai dengan contoh pada baris 59 hingga 79. Dan untuk mengintegrasikan plugin JQuery Validation dengan JQuery Form yang sudah diterapkan sebelumnya, tambahkan opsi **beforeSubmit** seperti pada baris 50-52 program di atas.

6. Tampilkan Data dalam Bentuk Tabel dengan Flexigrid

Untuk menampilkan data pada dasarnya banyak pilihan bentuk, baik dalam bentuk list maupun tabel. Kali ini kita akan memanfaatkan plugin JQuery untuk menampilkan data dalam bentuk tabel, yaitu **Flexigrid**⁵. Kelebihan utama dari plugin ini adalah kemudahan dan fleksibilitas dalam penerapannya. Kita tidak perlu membuat tabel dengan perintah HTML, cukup dengan menambahkan tag <table> dengan id tertentu, selebihnya plugin Flexigrid yang akan mengatur isinya. Kolom-kolom tabel dapat diatur dengan fleksibel sesuai kebutuhan, semua dilakukan melalui perintah Javascript. Sementara itu, data yang akan ditampilkan dalam tabel dapat berasal dari berbagai bentuk data, antara lain Array Javascript, XML, JSON maupun plain-text. Dari sisi source data tentunya dapat berasal dari halaman yang sama maupun halaman lain. Kelebihan lainnya adalah kolom dapat digeser, diurutkan dan diatur ukurannya *on-the-fly*, langsung di browser. Selain itu juga sudah tersedia halaman (*paging*) dan pencarian data (*searching*) berdasarkan kolom tertentu.

Untuk menampilkan data dengan Flexigrid cukup dengan 4 (empat) langkah sederhana, yaitu (1) sertakan library (javascript & CSS) yang diperlukan, (2) siapkan obyek tabel dengan tag HTML <table>, (3) atur kolom yang akan ditampilkan dan (4) siapkan potongan program untuk menangani data.

Pada langkah pertama, kita sertakan library yang diperlukan seperti terlihat pada potongan program di bawah ini. Terdapat satu file CSS yang akan mengatur tampilan tabel flexigrid (baris 1) dan terdapat library utama flexigrid (baris 3). Selain itu, pada versi terakhir flexigrid harus disertakan plugin **JQuery Cookie**⁶ buatan Klaus Hartl (baris 2).

⁵ <http://flexigrid.info/>

⁶ <https://github.com/carhartl/jquery-cookie>

Potongan Program untuk Menyertakan Library Flexigrid

```

1 <link rel="stylesheet" type="text/css"
  href="libs/flexigrid/css/flexigrid.css">
2 <script type="text/javascript" src="libs/jquery.cookie.js"></script>
3 <script type="text/javascript" src="libs/flexigrid/js/flexigrid.js"></script>

```

Langkah kedua, siapkan obyek tabel beserta atribut id-nya pada tempat dimana kita akan menampilkan data. Gunakan perintah sederhana sebagai berikut:

Potongan Program untuk Menyiapkan Obyek Tabel

```

1 <table id="flex1" style="display:none"></table>

```

Langkah ketiga adalah mengatur kolom-kolom yang akan ditampilkan beserta beberapa opsi atau *properties* dari tabel flexigrid. Langsung saja kita lihat contoh potongan program di bawah ini.

Potongan Program untuk Mengatur Kolom dan Opsi Tabel Flexigrid

```

1 //flexigrid handling
2 $('#flex1').flexigrid
3 (
4     {
5         url: 'index.php?action=getdata',
6         dataType: 'json',
7
8         colModel : [
9             {display: 'NIM', name : 'nim', width : 100, sortable :
10              true, align: 'left'},
11             {display: 'Nama', name : 'nama', width : 200, sortable :
12              true, align: 'left'},
13             {display: 'Alamat', name : 'alamat', width : 400,
14              sortable : true, align: 'left'}
15         ],
16         searchitems : [
17             {display: 'NIM', name : 'nim'},
18             {display: 'Nama', name : 'nama', isdefault: true}
19         ],
20
21         sortname: 'nama', //default urut kolom nama
22         sortorder: 'asc', //default urut ascending
23         usepager: true, //tambahkan paging
24         title: 'Data Mahasiswa', //judul flexigrid
25         useRp: true,
26         rp: 15,
27         showTableToggleBtn: true,
28         width: 700, //lebar tabel
29         height: 400 //tinggi tabel
30     }
31 )

```

```
27     }
28 );
```

Sedikit penjelasan mengenai potongan program di atas.

- Baris 2: Pemanggilan library flexigrid diawali dengan memanggil fungsi **flexigrid()** dan diterapkan pada tabel dengan id **#flex1** sesuai dengan tabel yang sudah dibuat di langkah sebelumnya.
- Baris 5-6: Opsi **url** dapat diisi dengan alamat source data yang akan ditampilkan di tabel. Alamat source data dapat berasal dari halaman yang sama seperti pada contoh di atas, dapat juga dari halaman berbeda. Format datanya harus sesuai dengan opsi yang dipilih pada baris 6 yaitu **dataType** (dalam hal ini menggunakan JSON).
- Baris 8-12: Pengaturan kolom-kolom yang akan ditampilkan di tabel. Urutan dari kolom harus sesuai dengan urutan data source-nya, jika tidak isi data bisa saja tidak sesuai. Pada setiap kolom terdapat beberapa pengaturan seperti judul kolom (*display*), nama kolom (*name*) sebagai identitas kolom, lebar kolom (*width*), dapat diurutkan atau tidak (*sortable*) dan perataan kolom (*align*). Pada contoh di atas, terdapat 3 kolom yaitu **nim**, **nama** dan **alamat**.
- Baris 13-16: Menambahkan fitur pencarian dimana pencarian dapat dilakukan berdasarkan kolom **nim** dan **nama**.
- Baris 18-27: Beberapa pengaturan lainnya yang dapat dilakukan seperti pengaturan pengurutan, pengaturan halaman (paging) serta lebar dan tinggi tabel.

Langkah keempat atau yang terakhir untuk menampilkan data ke flexigrid adalah menyiapkan potongan program PHP untuk menangani data. Jika data berasal dari database maka tugas bagian ini adalah melakukan pengambilan data (SELECT) untuk selanjutnya diformat sesuai dengan standar flexigrid dan dikirimkan ke library flexigrid untuk ditampilkan. Dalam contoh ini, data diformat dalam bentuk JSON.

Berikut ini potongan program PHP untuk menangani data flexigrid.

Potongan Program PHP untuk Menangani Data yang Akan Tampil di Flexigrid

```
1  if (isset($_GET['action']) && $_GET['action'] == 'getdata') {
2      $page = (isset($_POST['page']))?$_POST['page']: 1;
3      $rp = (isset($_POST['rp']))?$_POST['rp'] : 10;
4      $sortname = (isset($_POST['sortname']))? $_POST['sortname'] :
'nama';
5      $sortorder = (isset($_POST['sortorder']))? $_POST['sortorder'] :
'asc';
6
7      $sort = "ORDER BY $sortname $sortorder";
8      $start = (($page-1) * $rp);
9      $limit = "LIMIT $start, $rp";
10
11     $query = (isset($_POST['query']))? $_POST['query'] : '';
```

```

12     $qtype = (isset($_POST['qtype']))? $_POST['qtype'] : '';
13
14     $where = "";
15     if ($query) $where .= "WHERE $qtype LIKE '%$query%' ";
16
17     $query = "SELECT nim, nama, alamat ";
18     $query_from = " FROM mahasiswa ";
19
20     $query .= $query_from . " $where $sort $limit";
21
22     $query_total = "SELECT COUNT(*)". $query_from." ".$where;
23
24     $sql = mysql_query($query) or die($query);
25     $sql_total = mysql_query($query_total) or die($query_total);
26     $total = mysql_fetch_row($sql_total);
27     $data = $_POST;
28     $data['total'] = $total[0];
29     $datax = array();
30     $datax_r = array();
31     while ($row = mysql_fetch_row($sql)) {
32         $rows['id'] = $row[0];
33         $datax['cell'] = $row;
34         array_push($datax_r, $datax);
35     }
36     $data['rows'] = $datax_r;
37     echo json_encode($data);
38     exit;
39 }

```

Gambar 3 berikut ini tampilan halaman setelah ditambahkan tabel flexigrid.

Data Mahasiswa

NIM:

NAMA:

ALAMAT:

NIM	Nama	Alamat
0111500382	Achmad Solichin	Jl. Raya Ciledug, Jakarta Selatan
1011500101	Budi Susanto	Jl. Ciledug Raya, Petukangan Utara
0411500300	Chotimatul Musyarofah	Peninggihan, Kec. Ciledug, Tangerang, Banten, Indonesia
0911500292	Muhammad Lintang	Jakarta Selatan, DKI Jakarta

Page 1 of 1 | Displaying 1 to 4 of 4 items

Gambar 3: Tampilan Flexigrid

7. Tampilkan Data di Form saat Baris Fexigrid Dipilih

Data sudah ditampilkan di dalam grid, selanjutnya akan dilakukan proses Edit dan Delete. Untuk keperluan proses tersebut, setiap baris data di dalam flexigrid dipilih akan ditampilkan di form yang sudah dibuat. Jadi untuk sebelum melakukan proses EDIT dan DELETE, perlu memilih data yang akan diproses dengan mengklik pada baris data di flexigrid.

Untuk menambahkan action saat baris flexigrid diklik, cukup menambahkan atribut **'process'** pada pendefinisian **colModel**. Perhatikan contoh sebagai berikut dimana **doaction** merupakan fungsi javascript yang akan dipanggil saat suatu baris diklik / dipilih.

Potongan Program untuk Menambahkan Action pada Baris Flexigrid

```

1 colModel : [
2     {display: 'NIM', name : 'nim', width : 100, sortable : true,
3       align: 'left', process: doaction},
4     {display: 'Nama', name : 'nama', width : 200, sortable : true,
5       align: 'left', process: doaction},
6     {display: 'Alamat', name : 'alamat', width : 400, sortable :
7       true, align: 'left', process: doaction}
8 ]
  
```

5],

Dan berikut ini contoh fungsi **doaction** yang akan menangani proses saat baris data di flexigrid dipilih.

Fungsi Javascript doaction() untuk Menangani Proses Klik Baris Flexigrid

```

1 function doaction( celDiv, id ) {
2     $( celDiv ).click( function() {
3         var nim = $
4         (this).parent().parent().children('td').eq(0).text();
5         $.getJSON ('index.php',{action:'get_mhs',nim:nim}, function
6         (json) {
7             $('#nim').val(json.nim);
8             $('#nama').val(json.nama);
9             $('#alamat').val(json.alamat);
10            });
11            $('#nim').attr('readonly','readonly');
12            $('#input').attr('disabled','disabled');
13            $('#edit, #delete').removeAttr('disabled');
14        });
15    });
16 }
```

Perhatikan fungsi **doaction** di atas. Pertama fungsi akan melakukan pengecekan apakah terdapat suatu cell (kolom baris) data diklik (lihat baris 2). Jika terdapat cell yang diklik, maka ambil nilai dari cell pada kolom pertama baris tersebut, yaitu cell yang berisi NIM sebagai primary key-nya (lihat baris 3). Tentunya baris 3 harus disesuaikan dengan struktur data yang ada, jika lebih dari satu primary key, ya tinggal tambahkan variabel lainnya. Lanjut, setelah NIM dari baris didapatkan, kita perlu mendapatkan data lengkap dari database sehingga nim tersebut dikirimkan ke halaman **index.php** secara *asynchronous* (baris 4). Dengan perintah baris 4, seolah-olah kita mengakses browser dengan alamat http://namaserver/index.php?action=get_mhs&nim=nim, namun dilakukan oleh Javascript sehingga tidak kelihatan secara kasat mata. Sementara itu, karena memanfaatkan fungsi **\$.getJSON**, sehingga balikan data harus dalam format JSON. Pada baris 5-7, data balikan yang berupa JSON ditampilkan di form melalui fungsi **\$.val()** dari JQuery. Untuk melengkapi proses, pada baris 9-12 diatur agar form inputan NIM tidak dapat diedit (readonly), aktifkan tombol EDIT dan DELETE, serta disable tombol INPUT.

Berikut ini potongan program PHP untuk mengambil data berdasarkan NIM yang dipilih. Data dikembalikan dalam format JSON (Javascript Object Notation) seperti terlihat pada baris 6.

Potongan Program PHP untuk Mengambil Data Lengkap dari Database

```

1  if (isset($_GET['action']) && $_GET['action'] == 'get_mhs') {
2      $nim = $_GET['nim'];
3      $query = "SELECT * FROM mahasiswa WHERE nim='$nim'";
4      $sql = mysql_query($query);
5      $row = mysql_fetch_assoc($sql);
6      echo json_encode ($row);
7      exit;
8  }

```

8. Proses Edit dan Delete

Setelah data dipilih dan ditampilkan di dalam form inputan, dapat dengan mudah dilakukan proses EDIT dan DELETE. Seperti halnya proses INPUT, pada dasarnya proses EDIT dan DELETE akan mengambil data dari form dan menjalankan perintah query UPDATE atau DELETE ke tabel. Karena form sudah diaktifkan sebagai form Ajax (lihat kembali langkah 5) dan juga sudah otomatis dilakukan validasi (lihat langkah 6), maka kita tinggal mendefinisikan potongan program PHP untuk EDIT dan DELETE. Lihat potongan program di bawah ini.

Potongan Program PHP untuk Menangani Data yang Akan Tampil di Flexigrid

```

1  if (isset($_POST['Edit'])) {
2      $nim      = strip_tags($_POST['nim']);
3      $nama      = strip_tags($_POST['nama']);
4      $alamat = strip_tags($_POST['alamat']);
5
6      //update db
7      $query = sprintf("UPDATE mahasiswa SET nama='%s', alamat='%s'
WHERE nim='%s'",
8          mysql_escape_string($nama),
9          mysql_escape_string($alamat),
10         mysql_escape_string($nim)
11     );
12     $sql = mysql_query($query);
13     $pesan = "";
14     if ($sql) {
15         $pesan = "Data berhasil disimpan";
16     } else {
17         $pesan = "Data gagal disimpan ";
18         $pesan .= mysql_error();
19     }
20     $response = array('pesan'=>$pesan, 'data'=>$_POST);
21     echo json_encode($response);
22     exit;
23 } else if (isset($_POST['Delete'])) {
24     $nim      = strip_tags($_POST['nim']);
25

```

```
26 //delete data
27 $query = sprintf("DELETE FROM mahasiswa WHERE nim='%s'",
28                 mysql_escape_string($nim)
29                 );
30 $sql = mysql_query($query);
31 $pesan = "";
32 if ($sql) {
33     $pesan = "Data berhasil dihapus";
34 } else {
35     $pesan = "Data gagal dihapus ";
36     $pesan .= mysql_error();
37 }
38 $response = array('pesan'=>$pesan, 'data'=>$_POST);
39 echo json_encode($response);
40 exit;
41 }
```

9. Program Lengkap Ada di sini!

Bagian demi bagian sudah dibahas, semoga cukup jelas dan dapat dipahami. Di bawah ini program lengkap dan hasil akhir dari tutorial kita. Anda dapat mencobanya di komputer Anda, tentunya dengan menyertakan library yang diperlukan. Di bawah ini juga tersedia tautan untuk mencoba contoh program ini secara langsung (live demo) dan juga tautan untuk mendownload program lengkap beserta library yang diperlukan.

Program Lengkap Entri, Edit, Delete, Tampil dengan PHP dan Ajax JQuery

```
1 <?php
2 /* koneksi ke db */
3 mysql_connect("localhost", "root", "qwerty") or die(mysql_error());
4 mysql_select_db("demo") or die(mysql_error());
5 /* akhir koneksi db */
6
7 /* penanganan form */
8 if (isset($_POST['Input'])) {
9     $nim      = strip_tags($_POST['nim']);
10    $nama     = strip_tags($_POST['nama']);
11    $alamat  = strip_tags($_POST['alamat']);
12
13    //input ke db
14    $query = sprintf("INSERT INTO mahasiswa VALUES('%s', '%s', '%s'",
15                    mysql_escape_string($nim),
16                    mysql_escape_string($nama),
17                    mysql_escape_string($alamat)
18                    );
19    $sql = mysql_query($query);
20    $pesan = "";
21    if ($sql) {
22        $pesan = "Data berhasil disimpan";
23    } else {
```

```

24         $pesan = "Data gagal disimpan ";
25         $pesan .= mysql_error();
26     }
27     $response = array('pesan'=>$pesan, 'data'=>$_POST);
28     echo json_encode($response);
29     exit;
30 } else if (isset($_POST['Edit'])) {
31     $nim        = strip_tags($_POST['nim']);
32     $nama       = strip_tags($_POST['nama']);
33     $alamat     = strip_tags($_POST['alamat']);
34
35     //update data
36     $query = sprintf("UPDATE mahasiswa SET nama='%s', alamat='%s' WHERE
nim='%s'",
37                     mysql_escape_string($nama),
38                     mysql_escape_string($alamat),
39                     mysql_escape_string($nim)
40                 );
41     $sql = mysql_query($query);
42     $pesan = "";
43     if ($sql) {
44         $pesan = "Data berhasil disimpan";
45     } else {
46         $pesan = "Data gagal disimpan ";
47         $pesan .= mysql_error();
48     }
49     $response = array('pesan'=>$pesan, 'data'=>$_POST);
50     echo json_encode($response);
51     exit;
52 } else if (isset($_POST['Delete'])) {
53     $nim        = strip_tags($_POST['nim']);
54
55     //delete data
56     $query = sprintf("DELETE FROM mahasiswa WHERE nim='%s'",
57                     mysql_escape_string($nim)
58                 );
59     $sql = mysql_query($query);
60     $pesan = "";
61     if ($sql) {
62         $pesan = "Data berhasil dihapus";
63     } else {
64         $pesan = "Data gagal dihapus ";
65         $pesan .= mysql_error();
66     }
67     $response = array('pesan'=>$pesan, 'data'=>$_POST);
68     echo json_encode($response);
69     exit;
70 } else if (isset($_GET['action']) && $_GET['action'] == 'getdata') {
71
72     $page = (isset($_POST['page']))?$_POST['page']: 1;
73     $rp = (isset($_POST['rp']))?$_POST['rp'] : 10;
74     $sortname = (isset($_POST['sortname']))? $_POST['sortname'] : 'nama';
75     $sortorder = (isset($_POST['sortorder']))? $_POST['sortorder'] : 'asc';
76

```

```

77     $sort = "ORDER BY $sortname $sortorder";
78     $start = (($page-1) * $rp);
79     $limit = "LIMIT $start, $rp";
80
81     $query = (isset($_POST['query']))? $_POST['query'] : '';
82     $qtype = (isset($_POST['qtype']))? $_POST['qtype'] : '';
83
84     $where = "";
85     if ($query) $where .= "WHERE $qtype LIKE '%$query%' ";
86
87     $query = "SELECT nim, nama, alamat ";
88     $query_from = " FROM mahasiswa ";
89
90     $query .= $query_from . " $where $sort $limit";
91
92     $query_total = "SELECT COUNT(*)". $query_from." ".$where;
93
94     $sql = mysql_query($query) or die($query);
95     $sql_total = mysql_query($query_total) or die($query_total);
96     $total = mysql_fetch_row($sql_total);
97     $data = $_POST;
98     $data['total'] = $total[0];
99     $datax = array();
100    $datax_r = array();
101    while ($row = mysql_fetch_row($sql)) {
102        $rows['id'] = $row[0];
103        $datax['cell'] = $row;
104        array_push($datax_r, $datax);
105    }
106    $data['rows'] = $datax_r;
107    echo json_encode($data);
108    exit;
109 } else if (isset($_GET['action']) && $_GET['action'] == 'get_mhs') {
110     $nim = $_GET['nim'];
111     $query = "SELECT * FROM mahasiswa WHERE nim='$nim'";
112     $sql = mysql_query($query);
113     $row = mysql_fetch_assoc($sql);
114     echo json_encode ($row);
115     exit;
116 }
117 ?>
118 <html>
119     <head>
120         <title>Entri, Edit, Delete, Tampil Data dengan PHP dan
121 Ajax</title>
122         <style type="text/css">
123             .labelfrm {
124                 display:block;
125                 font-size:small;
126                 margin-top:5px;
127             }
128             .error {
129                 font-size:small;
130                 color:red;

```

```

130         }
131     </style>
132     <script type="text/javascript" src="libs/jquery.min.js"></script>
133     <script type="text/javascript"
src="libs/jquery.form.js"></script>
134     <script type="text/javascript"
src="libs/jquery.validate.min.js"></script>
135     <link rel="stylesheet" type="text/css"
href="libs/flexigrid/css/flexigrid.css">
136     <script type="text/javascript"
src="libs/jquery.cookie.js"></script>
137     <script type="text/javascript"
src="libs/flexigrid/js/flexigrid.js"></script>
138     <script type="text/javascript">
139     $(document).ready(function() {
140         resetForm();
141         //aktifkan ajax di form
142         var options = {
143             success      : showResponse,
144             beforeSend:  function(){
145                 return $("#frm").valid();
146             },
147             resetForm : true,
148             clearForm : true,
149             dataType  : 'json'
150         };
151         $('#frm').ajaxForm(options);
152
153         //validasi form dgn jquery validate
154         $('#frm').validate({
155             rules: {
156                 nim : {
157                     digits: true,
158                     minlength:10,
159                     maxlength:10
160                 }
161             },
162             messages: {
163                 nim: {
164                     required: "Kolom nim harus diisi",
165                     minlength: "Kolom nim harus terdiri dari
10 digit",
166                     maxlength: "Kolom nim harus terdiri dari
10 digit",
167                     digits: "NIM harus berupa angka"
168                 },
169                 nama: {
170                     required: "Nama harus diisi dengan benar"
171                 }
172             }
173         });
174
175         //flexigrid handling
176         $('#flex1').flexigrid

```

```

177         (
178             {
179                 url: 'index.php?action=getdata',
180                 dataType: 'json',
181
182                 colModel : [
183                     {display: 'NIM', name : 'nim', width : 100,
184 sortable : true, align: 'left', process: doaction},
185                     {display: 'Nama', name : 'nama', width : 200,
186 sortable : true, align: 'left', process: doaction},
187                     {display: 'Alamat', name : 'alamat', width :
188 400, sortable : true, align: 'left', process: doaction}
189                 ],
190                 searchitems : [
191                     {display: 'NIM', name : 'nim'},
192                     {display: 'Nama', name : 'nama', isdefault:
193 true}
194                 ],
195                 sortname: 'nama',
196                 sortorder: 'asc',
197                 usepager: true,
198                 title: 'Data Mahasiswa',
199                 useRp: true,
200                 rp: 15,
201                 width: 700,
202                 height: 400
203             }
204         );
205
206     });
207     function doaction( celDiv, id ) {
208         $( celDiv ).click( function() {
209             var nim = $
210 (this).parent().parent().children('td').eq(0).text();
211             $.getJSON ('index.php',{action:'get_mhs',nim:nim},
212 function (json) {
213                 $('#nim').val(json.nim);
214                 $('#nama').val(json.nama);
215                 $('#alamat').val(json.alamat);
216             });
217             $('#nim').attr('readonly','readonly');
218             $('#input').attr('disabled','disabled');
219             $('#edit, #delete').removeAttr('disabled');
220         });
221     }
222     function showResponse(responseText, statusText) {
223         var data = responseText['data'];
224         var pesan = responseText['pesan'];
225         alert(pesan);
226         resetForm();
227         $('#flex1').flexReload();
228     }
229     function resetForm() {

```



```

225         $('#input').removeAttr('disabled');
226         $('#edit, #delete').attr('disabled', 'disabled');
227         $('#nim').removeAttr('readonly');
228     }
229     </script>
230 </head>
231 <body>
232     <h1>Data Mahasiswa</h1>
233     <form action="" method="post" id="frm" onReset="resetForm()">
234         <label for="nim" class="labelfrm">NIM: </label>
235         <input type="text" name="nim" id="nim" maxlength="10"
class="required" size="15"/>
236
237         <label for="nama" class="labelfrm">NAMA: </label>
238         <input type="text" name="nama" id="nama" size="30"
class="required"/>
239
240         <label for="alamat" class="labelfrm">ALAMAT: </label>
241         <textarea name="alamat" id="alamat" cols="40" rows="4"
class="required"></textarea>
242
243         <label for="submit" class="labelfrm">&nbsp;</label>
244         <input type="submit" name="Input" value="Input"
id="input"/>
245         <input type="submit" name="Edit" value="Edit" id="edit"/>
246         <input type="submit" name="Delete" value="Delete"
id="delete"/>
247         <input type="reset" name="Clear" value="Clear" id="clear"/>
248     </form>
249
250     <table id="flex1" style="display:none"></table>
251 </body>
252 </html>

```

Kesimpulan

Berdasarkan penjelasan di atas, dapat disimpulkan bahwa penerapan konsep Ajax di dalam program PHP dapat dilakukan dengan mudah dengan memanfaatkan library JQuery. Penggunaan JQuery juga dapat menghasilkan tampilan yang baik serta membuat pengguna lebih mudah dan nyaman dalam menggunakan aplikasi kita. Sebagai contoh dalam tutorial ini, untuk melakukan proses Entri, Edit, Delete dan Tampil Data hanya disajikan satu halaman serta semua proses tidak memerlukan refresh keseluruhan halaman. Proses terjadi secara asynchronous di “belakang” halaman. Dengan demikian, penerapan konsep Ajax akan membuat aplikasi lebih kaya dan menarik (Rich Internet Application).

Semoga tutorial ini bermanfaat untuk kita semua dan mari kobarkan terus semangat berbagi!

Demo dan Download

- Tutorial ini juga dipublikasikan di <http://achmatim.net/2012/09/02/entri-edit-delete-tampil-dengan-php-dan-ajax-jquery/>
- Demo program akhir dari tutorial ini dapat dilihat di http://achmatim.net/_demo/crud-php-ajax/
- Download source code program lengkap beserta library-nya di alamat http://achmatim.net/_downloads/crud-php-ajax.zip

Tentang Penulis



Achmad Solichin. Adalah Lulusan Teknik Informatika, [Fakultas Teknologi Informasi, Universitas Budi Luhur](#), Jakarta (S1, 2005) dan [Magister Teknologi Informasi, Universitas Indonesia](#) (S2, 2010). Saat ini sedang menempuh program Doktorat dalam bidang Ilmu Komputer di [Universitas Gadjah Mada](#), Jogjakarta. Kegiatan sehari-hari adalah sebagai Dosen di Universitas Budi Luhur (<http://www.budiluhur.ac.id>). Kegiatan lain aktif sebagai programmer, web developer, system analyst, konsultan dan memberikan pelatihan di berbagai bidang komputer serta membuat tutorial-tutorial praktis di bidang komputer. Penulis memiliki situs utama di <http://achmatim.net> yang berisi berbagai tutorial praktis di bidang komputer serta menyediakan buku gratis komputer. Penulis dapat dihubungi melalui email di achmad.solichin@budiluhur.ac.id dan achmatim@gmail.com, YM [achmatim](#), Facebook [achmatim](#) dan Twitter @[achmatim](#).

Lisensi Dokumen

Seluruh isi dalam dokumen ini dapat digunakan, dimodifikasi dan disebarluaskan secara bebas untuk tujuan pendidikan, pembelajaran dan bukan komersial (*non profit*), dengan syarat tidak menghilangkan, menghapus atau mengubah atribut penulis dokumen ini dan pernyataan dalam lisensi dokumen yang disertakan di setiap dokumen. Tidak diperbolehkan mengkomersialkan tutorial ini kecuali mendapatkan ijin terlebih dahulu dari penulis.