

PRAKTIKUM 5

PHP dan AJAX

1. Tujuan Praktikum

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat memahami cara untuk:

- Membuat objek file XMLHttpRequest sebagai core Ajax
- Melakukan request ke server melalui AJAX menggunakan method GET
- Melakukan request ke server melalui AJAX menggunakan method POST

2. Tools

Tools yang dibutuhkan untuk melakukan praktikum ini adalah Apache web server, PHP interpreter, DBMS MySQL dan text editor.

3. Langkah-Langkah Praktikum

3.1. Praktikum 1 : Membuat objek XMLHttpRequest

Fungsi berikut digunakan untuk membuat objek XMLHttpRequest sebagai core Ajax. Buatlah fungsi tersebut dalam sebuah file dan simpan dengan nama ajax.js.

ajax.js

```
1 //fungsi untuk membuat objek XMLHttpRequest
2 function getXMLHttpRequest() {
3     if (window.XMLHttpRequest) {
4         // code for modern browsers
5         return new XMLHttpRequest();
6     } else {
7         // code for old IE browsers
8         return new ActiveXObject("Microsoft.XMLHTTP");
9     }
10 }
```

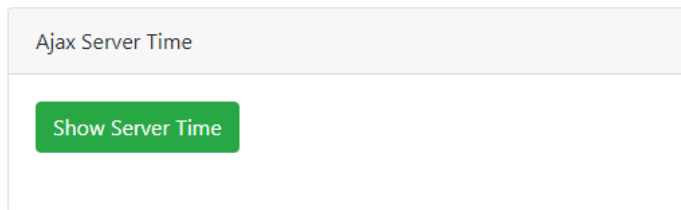
3.2. Praktikum 1 : Contoh AJAX sederhana

Berikut ini merupakan contoh sederhana penggunaan Ajax untuk melakukan request ke server untuk menampilkan waktu server.

- File show_server_time.php berisi tombol 'Show Server Time'. Jika tombol tersebut diklik, maka akan memanggil fungsi javascript get_server_time() yang melakukan request ke file get_server_time.php melalui Ajax dan menampilkan responnya di elemen dengan id="show_time".

```
1 <!--File: show_server_time.html
2 Deskripsi : menampilkan waktu server dengan ajax
3 -->
4
5 <?php include('../header.html'); ?><br>
6 <div class="card">
7     <div class="card-header">Ajax Server Time</div>
8     <div class="card-body">
9         <button class="btn btn-success" onclick="get_server_time()">Show Server Time
10        </button>
11        <div id="showtime"></div>
12    </div>
13 </div>
14 <?php include('../footer.html'); ?><br>
```

Tampilan dari file show_server_time.php adalah sebagai berikut:



2. Isi dari fungsi javascript get_server_time(). Tambahkan fungsi javascript tersebut pada file ajax.js lalu, include file ajax.js pada file show_server_time.php. Lokasi file get_server_time.php yang di-request relatif terhadap file show_server_time.php.

ajax.js

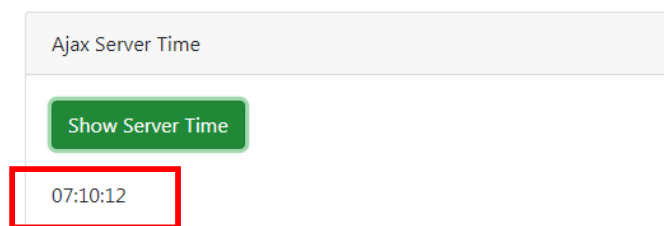
```
11 //fungsi untuk melakukan request ke get_server_time.php melalui ajax
12 function get_server_time() {
13     var xmlhttp = getXMLHttpRequest();
14     var page = 'get_server_time.php';
15     xmlhttp.open("GET", page, true);
16     xmlhttp.onreadystatechange = function() {
17         document.getElementById('showtime').innerHTML = '<img
18             src= "../images/ajax_loader.png"/>';
19         if ((xmlhttp.readyState == 4) && (xmlhttp.status == 200)){
20             document.getElementById('showtime').innerHTML = xmlhttp.responseText;
21         }
22     }
23     xmlhttp.send(null);
24 }
```

3. Isi dari file get_server_time.php yang di-request adalah sebagai berikut:

get_server_time.php

```
1 <?php
2 //menampilkan waktu server
3 echo date('H:i:s');
4 ?>
```

4. Panggil file show_server_time.php di browser, lalu klik tombol "Show Server Time", perhatikan bahwa waktu server akan ditampilkan pada halaman tersebut tanpa melakukan load keseluruhan isi halaman.



3.3. Praktikum 1 : Contoh AJAX dengan method GET

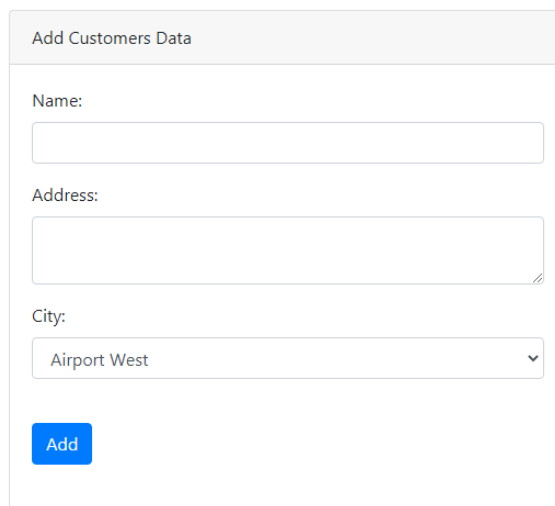
Berikut ini adalah contoh request ke server melalui Ajax menggunakan method GET untuk menyimpan data ke basis data.

1. File `add_customer.php` berisi form data customer yang terdiri atas name, address, dan city. Tombol submit diganti dengan `<button type="button" ...>` yang jika diklik akan memanggil fungsi javascript `add_customer_get()`. Fungsi tersebut selanjutnya me-request file `add_customer_get.php` melalui Ajax dengan method GET dan menampilkan responnya pada elemen dengan `id="add_response"`.

`add_customer.php`

```
11 <form method="GET" autocomplete="on">
12   <div class="form-group">
13     <label for="name">Name:</label>
14     <input type="text" class="form-control" id="name" name="name">
15   </div>
16   <div class="form-group">
17     <label for="address">Address:</label>
18     <textarea class="form-control" id="address" name="address"></textarea>
19   </div>
20   <div class="form-group">
21     <label for="city">City:</label>
22     <select name="city" id="city" class="form-control" required>
23       <option value="Airport West">Airport West</option>
24       <option value="Box Hill">Box Hill</option>
25       <option value="Yarraville">Yarraville</option>
26     </select>
27   </div><br>
28   <button type="button" class="btn btn-primary" onclick="add_customer_get()">Add
29 </form>
30 <br>
31 <div id="add_response"></div>
```

Tampilan dari file `add_customer.php` adalah sebagai berikut:



2. Isi dari fungsi javascript `add_customer_get()` adalah sebagai berikut. Tambahkan fungsi javascript tersebut pada file `ajax.js` lalu, include file `ajax.js` pada file `add_customer.php`. Lokasi file `add_customer_post.php` yang di-request relatif terhadap file `add_customer.php`. Perhatikan bahwa saat menggunakan method GET, data yang dikirimkan (isi form add customer) dilewatkan melalui URL.

ajax.js

```

25 function add_customer_get(){
26     var xmlhttp = getXMLHttpRequest();
27     //get input value
28     var name = encodeURI(document.getElementById('name').value);
29     var address = encodeURI(document.getElementById('address').value);
30     var city = encodeURI(document.getElementById('city').value);
31     //validate
32     if (name != "" && address != "" && city != ""){
33         //set url and inner
34         var url = "add_customer_get.php?name=" + name + "&address=" + address +
35             "&city=" + city;
36         //alert(url);
37         var inner = "add response";
38         //open request
39         xmlhttp.open('GET', url, true);
40         xmlhttp.onreadystatechange = function() {
41             document.getElementById(inner).innerHTML = '';
43             if ((xmlhttp.readyState == 4) && (xmlhttp.status == 200)){
44                 document.getElementById(inner).innerHTML = xmlhttp.responseText;
45             }
46             return false;
47         }
48         xmlhttp.send(null);
49     }else{
50         alert("Please fill all the fields");
51     }
52 }

```

- Isi dari file `add_customer_get.php` yang di-request adalah sebagai berikut. File tersebut menyimpan masukan isi form ke basis data, lalu menampilkan respon berupa konfirmasi berdasarkan hasil eksekusi query. Saat menggunakan method GET, data yang dikirimkan diakses melalui array superglobal `$_GET[]`.

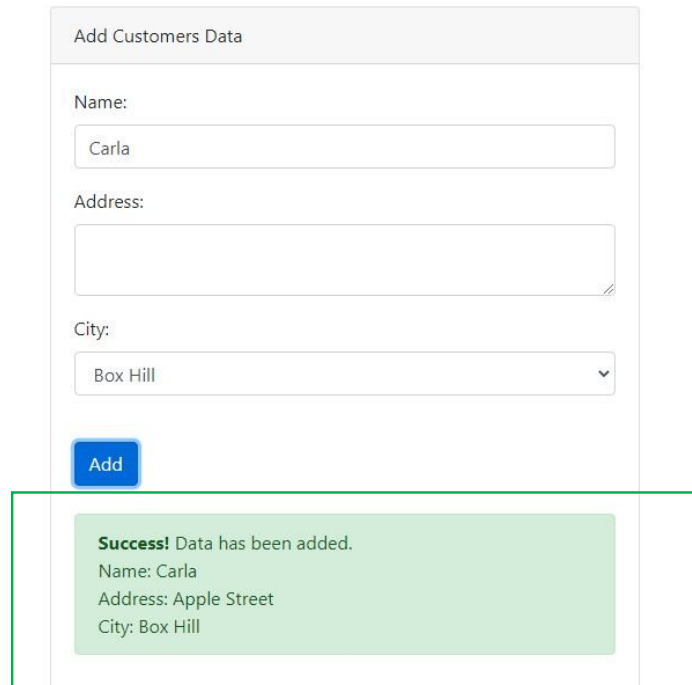
add_customer_get.php

```

1 <?php
2 require_once('../lib/db_login.php');
3 $name = $db->real_escape_string($_GET['name']);
4 $address = $db->real_escape_string($_GET['address']);
5 $city = $db->real_escape_string($_GET['city']);
6 //Assign a query
7 $query = "INSERT INTO customers (name, address, city) VALUES('".$name."','".$address."','".$city."' )";
8 $result = $db->query( $query );
9 if (!$result){
10     echo '<div class="alert alert-danger alert-dismissible">
11         <strong>Error!</strong> Could not query the database<br>'.
12         $db->error. ' <br>query = ' . $query.
13         '</div>';
14 }else{
15     echo '<div class="alert alert-success alert-dismissible">
16         <strong>Success!</strong> Data has been added.<br>
17         Name: ' . $_GET['name'] . '<br>
18         Address: ' . $_GET['address'] . '<br>
19         City: ' . $_GET['city'] . '<br>
20         </div>';
21 }
22 //close db connection
23 $db->close();
24 ?>

```

4. Panggil file `add_customer.php` di browser, lalu isi form dan klik tombol 'Add', perhatikan bahwa isi form akan disimpan ke basis data dan konfirmasi akan ditampilkan pada halaman tersebut tanpa melakukan load keseluruhan isi halaman. Tampilan file `add_customer.php` setelah pengisian form adalah sebagai berikut:



The screenshot shows a web form titled "Add Customers Data". It contains three input fields: "Name:" with the value "Carla", "Address:" which is empty, and "City:" with a dropdown menu showing "Box Hill". Below these fields is a blue "Add" button. A green box highlights a success message area at the bottom of the form, which reads: "Success! Data has been added. Name: Carla Address: Apple Street City: Box Hill".

3.4. Praktikum 1 : Contoh AJAX dengan method POST

Berikut ini adalah contoh request ke server melalui Ajax menggunakan method POST untuk menyimpan data ke basis data.

1. Ubah isi pada file `add_customer.php`, yaitu pada `<button type="button" ...>` jika diklik akan memanggil fungsi javascript `add_customer_post()`. Fungsi tersebut selanjutnya me-request file `add_customer_post.php` melalui Ajax dengan method POST dan menampilkan responnya pada elemen dengan `id="add_response"`.

`add_customer.php`

```
28 <button type="button" class="btn btn-primary" onclick="add_customer_post()">Add
    </button>
```

2. Isi dari fungsi javascript `add_customer_post()` adalah sebagai berikut. Tambahkan fungsi javascript tersebut pada file `ajax.js` lalu, include file `ajax.js` pada file `add_customer.php`. Lokasi file `add_customer_post.php` yang di-request relatif terhadap file `add_customer.php`. Perhatikan bahwa saat menggunakan method POST, URL tidak ditambahkan parameter, tetapi data yang dikirimkan (isi form add customer) dilewatkan sebagai parameter pada header HTTP.

ajax.js

```

52 function add_customer_post(){
53     var xmlhttp = getXMLHttpRequest();
54     //get input value
55     var name = encodeURI(document.getElementById('name').value);
56     var address = encodeURI(document.getElementById('address').value);
57     var city = encodeURI(document.getElementById('city').value);
58     if (name != "" && address != "" && city != ""){
59         //set url and inner
60         var url = "add_customer_post.php"; alert(url);
61         var inner = "add_response";
62         //set parameter and open request
63         var params = "name=" + name + "&address=" + address + "&city=" + city;
64         xmlhttp.open("POST", url, true);
65         xmlhttp.setRequestHeader("Content-type",
66             "application/x-www-form-urlencoded");
67         xmlhttp.onreadystatechange = function() {
68             document.getElementById(inner).innerHTML = '';
70             if ((xmlhttp.readyState == 4) && (xmlhttp.status == 200)){
71                 document.getElementById(inner).innerHTML = xmlhttp.responseText;
72                 return false;
73             }
74             xmlhttp.send(params);
75         }else{
76             alert("Please fill all the fields");
77         }
78     }
79 }

```

- Isi dari file add_customer_post.php yang di-request adalah sebagai berikut. File tersebut menyimpan masukan isi form ke basis data, lalu menampilkan respon berupa konfirmasi berdasarkan hasil eksekusi query. Saat menggunakan method POST, data yang dikirimkan diakses melalui array superglobal \$_POST[].

add_customer_post.php

```

1 <?php
2 require_once('../lib/db_login.php');
3 $name = $db->real_escape_string($_POST['name']);
4 $address = $db->real_escape_string($_POST['address']);
5 $city = $db->real_escape_string($_POST['city']);
6 //Assign a query
7 $query = " INSERT INTO customers (name, address, city) VALUES('".$name."','".$
8     $address."','".$city."' ) ";
9 $result = $db->query( $query );
10 if (!$result){
11     echo '<div class="alert alert-danger alert-dismissible">
12         <strong>Error!</strong> Could not query the database<br>'.
13         $db->error. '<br>query = '.$query.
14         '</div>';
15 }else{
16     echo '<div class="alert alert-success alert-dismissible">
17         <strong>Success!</strong> Data has been added.<br>
18         Name: '.$_POST['name'].'<br>
19         Address: '.$_POST['address'].'<br>
20         City: '.$_POST['city'].'<br>
21         </div>';
22 }
23 //close db connection
24 $db->close();
25 ?>

```


4. Panggil file `add_customer.php` di browser, lalu isi form dan klik tombol 'Add', perhatikan bahwa isi form akan disimpan ke basis data dan konfirmasi akan ditampilkan pada halaman tersebut tanpa melakukan load keseluruhan isi halaman.

3.5. Praktikum 1 : Contoh AJAX untuk menampilkan data dari basis data

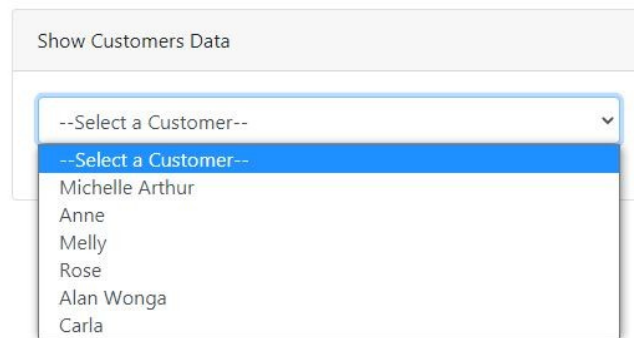
Berikut ini adalah contoh request ke server melalui Ajax menggunakan method GET untuk menampilkan detail data customer berdasarkan nama yang dipilih oleh pengguna.

1. File `show_customer.php` menampilkan dropdown list berisi nama customer yang diambil dari tabel customer. Jika pengguna mengubah pilihan pada dropdown list tersebut, maka akan dipanggil fungsi javascript `showCustomer(this.value)`. Value (berupa `customerid`) yang dipilih dilewatkan sebagai parameter fungsi. Fungsi tersebut selanjutnya me-request file `get_customer.php` melalui Ajax dengan method GET dan menampilkan responnya pada elemen dengan `id="detail_customer"`.

`show_customer.php`

```
1 <!--File      : show_customer.php
2 Deskripsi    : menampilkan data customers
3 -->
4 <?php include('../header.html') ?>
5 <br>
6 <div class="card">
7 <div class="card-header">Show Customers Data</div>
8 <div class="card-body">
9
10 <select name="customer" id="customer" class="form-control" onchange=
    "showCustomer(this.value)">
11 <option value="">--Select a Customer--</option>
12 <?php
13 require_once('../lib/db_login.php');
14 //Assign a query
15 $query = " SELECT * FROM customers ORDER BY customerid ";
16 $result = $db->query( $query );
17 if (!$result){
18     die ("Could not query the database: <br />". $db->error);
19 }
20 // Fetch and display the results
21 while ($row = $result->fetch_object()){
22     echo '<option value="'. $row->customerid. '">'. $row->name. '</option>';
23 }
24 $result->free();
25 $db->close();
26 ?>
27 </select>
28 <br>
29 <div id="detail_customer"></div>
30 </div>
31 </div>
32 <?php include('../footer.html') ?>
```

Tampilan dari file show_customer.php adalah sebagai berikut:



2. Isi dari fungsi javascript showCustomer() adalah sebagai berikut. Lokasi file get_customer.php yang di-request relatif terhadap file show_customer.php. Perhatikan bahwa saat menggunakan method GET, data yang dikirimkan (customerid) dilewatkan melalui URL. Fungsi showCustomer() memanggil fungsi callAjax() untuk melakukan request ke server melalui Ajax. Tambahkan kedua fungsi javascript tersebut pada file ajax.js lalu, include file ajax.js pada file show_customer.php.

ajax.js

```
80 function callAjax(url,inner){
81     var xmlhttp = getXMLHttpRequest();
82     xmlhttp.open('GET', url, true);
83     xmlhttp.onreadystatechange = function() {
84         document.getElementById(inner).innerHTML = '';
86         if ((xmlhttp.readyState == 4) && (xmlhttp.status == 200)){
87             document.getElementById(inner).innerHTML = xmlhttp.responseText;
88         }
89         return false;
90     }
91     xmlhttp.send(null);
92 }
93 function showCustomer(customerid){
94     var inner = 'detail_customer';
95     var url = 'get_customer.php?id=' + customerid;
96     if(customerid == ""){
97         document.getElementById(inner).innerHTML = '';
98     }else{
99         callAjax(url,inner);
100     }
101 }
```

3. Isi dari file get_customer.php yang di-request adalah sebagai berikut. File tersebut mengambil data customer dari basis data berdasarkan customerid yang dipilih, lalu menampilkan respon detail customer berdasarkan hasil eksekusi query. Saat menggunakan method GET, data yang dikirimkan diakses melalui array superglobal \$_GET[].

get_customer.php

```

1 <?php
2 require_once('../lib/db_login.php');
3 $customerid = $_GET['id'];
4 //Assign a query
5 $query = " SELECT * FROM customers where customerid=".$customerid;
6 $result = $db->query( $query );
7 if (!$result){
8     die ("Could not query the database: <br />". $db->error);
9 }
10 // Fetch and display the results
11 while ($row = $result->fetch_object()){
12     echo 'Name: '.$row->name.'<br />';
13     echo 'Address: '.$row->address.'<br />';
14     echo 'City: '.$row->city.'<br />';
15 }
16 $result->free();
17 $db->close();
18 ?>

```

4. Panggil file show_customer.php di browser, lalu pilih salah satu nama customer dari dropdown list, perhatikan bahwa detail customer akan ditampilkan pada halaman tersebut tanpa melakukan load keseluruhan isi halaman. Tampilan file show_customer.php setelah dipilih salah satu nama customer adalah sebagai berikut:

Show Customers Data

Michelle Arthur

Name: Michelle Arthur

Address: 357 North Road A

City: Yarraville

4. Tugas

1. Selesaikanlah contoh penggunaan Ajax pada sub bab praktikum 3.2 – 3.5.

| Praktikum | File yang me-request | Fungsi javascript yang dipanggil | File yang di-request |
|-----------|----------------------|----------------------------------|-----------------------|
| 3.2 | show_server_time.php | get_server_time() | get_server_time.php |
| 3.3 | add_customer.php | add_customer_get() | add_customer_get.php |
| 3.4 | add_customer.php | add_customer_post() | add_customer_post.php |
| 3.5 | show_customer.php | showcustomer(this.value) | get_customer.php |

2. Buatlah file untuk pencarian data buku berdasarkan judul yang diketik, lalu menampilkan detail buku pada halaman yang sama menggunakan Ajax.

===== Selamat Berlatih, Semoga Sukses =====