

# **Kuliah Pemrograman Internet**

# **KOLABORASI JQUERY +**

# **AJAX + PHP + MySQL**

I Putu Arya Dharmaadi, S.T., M.T.  
Dosen Jurusan Teknologi Informasi  
Universitas Udayana  
<http://aryadharmaadi.com>

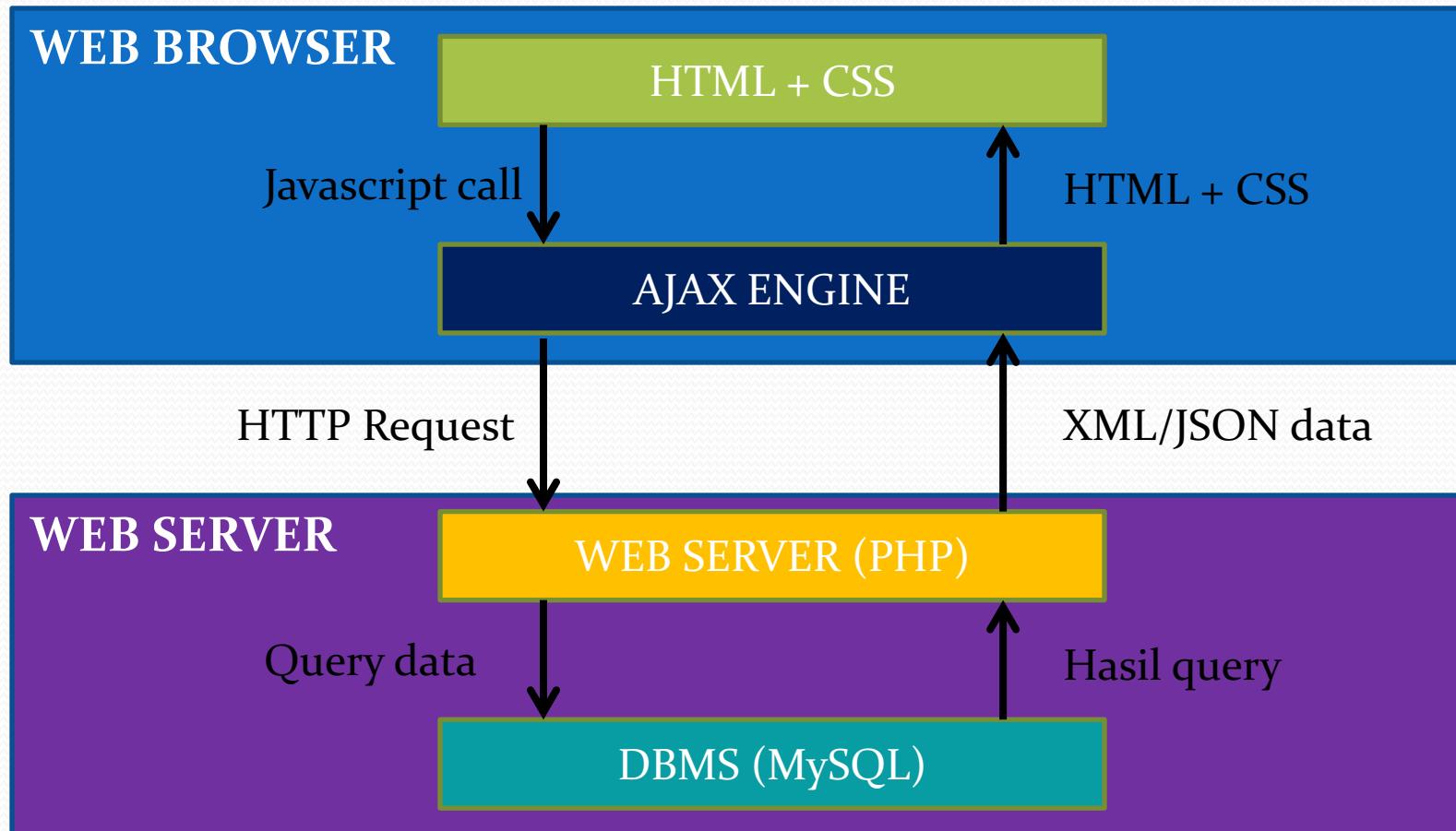




# Pendahuluan

- AJAX dimanfaatkan untuk membuat web interaktif dan dinamis tanpa perlu membuat browser melakukan refresh ketika meminta data ke server
- AJAX memungkinkan semua dapat dilakukan secara instan, tanpa perlu lagi menunggu halaman di-reload, sehingga aplikasi web mirip dengan aplikasi desktop dalam hal responsibilitas maupun interaktifitas
- Saat ini, AJAX sudah menjadi teknologi yang wajib diterapkan dalam pengembangan web modern (web 2.0)

# Hubungan antara HTML, Ajax, PHP, dan MySQL



# Penjelasan Singkat

- Pada saat dibuka, web browser masih kosong
- Kemudian browser membuka sebuah halaman sehingga browser melakukan request ke server
- Hasil request adalah script HTML, CSS, JS (termasuk Ajax Engine), dan script-script tersebut di-load ke web browser
- Selanjutnya seperti pada gambar di atas, ketika user akan melakukan suatu perintah pada aplikasi, halaman HTML tetap diam, namun di belakang layar, AJAX Engine yang melakukan http request ke server

# Antara jQuery dan AJAX

- jQuery merupakan sebuah library Javascript sehingga mempermudah manipulasi komponen di dokumen HTML, menangani event, animasi, efek, dll.
- Selain itu, jQuery mempermudah developer web untuk memproses interaksi AJAX
- Tanpa library seperti jQuery, menerapkan AJAX akan lebih sulit, panjang, dan rumit, terutama untuk pemula yang baru belajar Javascript.

# Memulai Project

- Sebagai latihan, akan dibuat sebuah aplikasi ajax berdasarkan project yang sebelumnya (aplikasi insert nama dan hobi)
- Aplikasi ajax ini akan mencari nama pada tabel ‘daftar\_hobi’ dan akan menampilkan hobinya.
- So, jalankan web server pada komputer anda
- Dari project yang sebelumnya (folder latihan), buatlah folder untuk project yang baru dengan nama ‘ajax’
- Buatlah sebuah file html, dengan nama ‘index.html’

# Bagian header file index.html

```
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Latihan AJAX</title>

    <!-- Jquery Mobile -->
    <link rel="stylesheet" href="css/jquery.mobile-1.4.5.min.css">
    <link href="css/my-style.css" rel="stylesheet">

    <!-- jQuery -->
    <script src="js/jquery.js"></script>
    <script src="js/jquery.mobile-1.4.5.min.js"></script>
    <script type='text/javascript'>
```

# Bagian body file index.html

```
<div style="margin-top: 50px; margin-bottom: 30px;" id="main" role="main">
  <div id="result" style="margin-bottom: 20px; display: none">
    <h3 class="ui-bar ui-bar-a ui-corner-all">Hasil Pencarian</h3>
    <div class="ui-body ui-body-a ui-corner-all">

      </div>
    </div>

<div id="pencarian">
  <h3 class="ui-bar ui-bar-a ui-corner-all">Form Pencarian</h3>
  <div class="ui-body ui-body-a ui-corner-all">
    <form id="form-pencarian" action="get_hobi.php" method="post">
      <label for="search-1">Masukkan Nama anda:</label>
      <input name="nama" id="nama" value="" placeholder="Cari nama hobi..." type="text">
      <input value="C A R I" data-theme="a" type="submit">
    </form>
  </div>
</div>
</div><!-- /content -->
```

# Keterangan

- Project ini memanfaatkan library jQueryMobile untuk mempercantik tampilan web dan library jQuery untuk mempermudah penggunaan AJAX
- Poin yang paling penting adalah:
  - Menyediakan element untuk menampilkan hasil pencarian, yaitu div dengan id=“result”
  - Membuat form untuk pengiriman data, dengan id=“form-pencarian” action=“get\_hobi.php” method=“post”
  - Form tersebut terdiri dari input dengan type=“text” dan input dengan type=“submit”

# Tampilan index.html

## Form Pencarian

Masukkan Nama anda:

 Cari nama . . .

**C A R I**

# Buat Script untuk AJAX

- Setelah selesai membuat script HTML, selanjutnya buat script untuk penanganan AJAX
- Script bisa ditulis di file terpisah atau ditulis di bagian head file index.html
- Dalam project ini, script ditulis di bagian head file index.html karena code-nya sedikit

# Javascript

```
<script type='text/javascript'>
$(document).ready(function (){
    $('#form-pencarian').submit(function(event) {
        event.preventDefault();

        $('#result .ui-body').html('');
        var url = $(this).attr('action');

        $.ajax({
            type: $(this).attr('method'),
            url: url,
            data: $(this).serialize(),
            dataType: "JSON",
            success: function(data){
                if (data == null){
                    $('#result .ui-body').html('Data tidak ditemukan');
                }else{
                    $('#result .ui-body').html(data[0].nama+' memiliki hobi '+data[0].hobi);
                }
                $('#result').show();
            }
        });
    });
});
</script>
```

# Keterangan

- `$(document).ready()` berfungsi untuk pengecek event, jadi apabila semua elemen pada dokumen HTML sudah di-load semua, maka script di dalam blok fungsi `$(document).ready()` akan dijalankan
- `'#form-pencarian'.submit()` berfungsi sebagai listener event submit. Jadi, apabila form pencarian diklik submit, maka script di dalam blok fungsi `'#form-pencarian'.submit()` akan dijalankan

# Keterangan (2)

- `event.preventDefault()` berfungsi untuk menghentikan pengiriman data pada form secara normal. Fungsi tersebut digunakan karena pada project ini akan menggunakan pengiriman data melalui AJAX sehingga pengiriman data secara normal harus dihentikan.
- `$.ajax()` merupakan proses pengiriman data dengan AJAX, dengan parameter:
  - **type** → method pengiriman, bisa “GET” atau “POST”
  - **url** → server tujuan pengiriman data

# Keterangan (3)

- data → data form yang akan dikirimkan ke server, bisa didapatkan dengan menggunakan *form.serialize()*. Jika ingin menulis data form secara manual, caranya adalah
  - “nama=“+\$(‘#nama’).val()”
- dataType → format data yang akan diterima dari server, bisa “XML” atau “JSON”
- success → fungsi yang akan dijalankan ketika AJAX berhasil mengirimkan data ke server dan server memberikan respon balik. Respon balik dari server akan disimpan di variabel data.

# Buat File PHP untuk Koneksi ke Database

- File PHP untuk koneksi ke database sudah dibahas pada project sebelumnya
- Untuk itu, buatlah file dengan nama *database.php* yang disalin dari project sebelumnya
- File ini hanya khusus menangani koneksi ke database, sedangkan penggunaan database (query database) akan dilakukan di file PHP yang lain dengan meng-include-kan file *database.php*

## File database.php

```
<?php  
$dbhost = 'localhost';  
$dbuser = 'root';  
$dbpass = '';  
$dbname = 'latihan';  
$db = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);  
  
// Check connection  
if ($db->connect_error) {  
    die("Connection failed: " . $db->connect_error);  
}  
?  
>
```

# Buat File PHP (Server Tujuan AJAX)

- Setelah file *database.php* selesai dibuat, sekarang buatlah file yang menjadi target pengiriman form (*get\_hobi.php*)
- File tersebut akan meng-include-kan file *database.php* agar bisa terhubung ke database
- File *get\_hobi.php* mirip dengan file project sebelumnya yang meng-query data, namun di file *get\_hobi.php* akan dibuat untuk me-return data dalam bentuk JSON

```
<?php
require "database.php";

$namaku = $_POST['nama'];

$stmt = $db->prepare("SELECT * FROM daftar_hobi WHERE nama=?");
$stmt->bind_param("s", $namaku);
$stmt->execute();

$stmt->bind_result($nama,$hobi);

$result = null;
while ($stmt->fetch()) {
    $result[] = array(
        'nama' => $nama,
        'hobi' => $hobi
    );
}
echo json_encode($result);

$stmt->close();
$db->close();
?>
```

# Keterangan Singkat

- Require “database.php” merupakan code untuk memanggil file *database.php* dan meng-include-kan semua variabel dan code-nya ke dalam *get\_hobi.php*
- Penggunaan prepare statement sama dengan project sebelumnya
- Untuk mendapatkan hasil dari prepare statement, digunakan method *bind\_result(\$nama,\$hobi)*, dimana jumlah variabel pada parameteranya *bind\_result* disesuaikan dengan jumlah kolom yang di-query dari database

# Keterangan Singkat (2)

- Untuk mendapatkan hasil per baris-nya, gunakan fungsi `fetch()`, dimana fungsi ini akan menggeser pointer dari satu row hasil data query ke row selanjutnya dan menyimpan datanya ke variabel `$nama` dan `$hobi` .
- Fungsi tersebut akan menghasilkan true jika baris yang ditunjuk memiliki data dan menghasilkan false jika baris yang ditunjuk tidak ada.
- Fungsi `json_encode` adalah fungsi untuk mengubah array menjadi format JSON sehingga mudah dibaca oleh AJAX engine

# Screen Shot Direktori Project

The screenshot shows a Windows File Explorer window with the following details:

- Path:** Computer > Local Disk (C:) > xampp > htdocs > latihan > ajax
- Search Bar:** Search ajax
- Toolbar Buttons:** Include in library, Share with, Burn, New folder
- Table Headers:** Name, Date modified, Type, Size
- Content:** A list of files and folders including 'css', 'fonts', 'js', 'database.php', 'get\_hobi.php', and 'index.html'.

Name	Date modified	Type	Size
css	13/11/2015 13:53	File folder	
fonts	13/11/2015 13:53	File folder	
js	13/11/2015 13:53	File folder	
database.php	14/11/2015 0:42	PHP File	1 KB
get_hobi.php	14/11/2015 1:25	PHP File	1 KB
index.html	14/11/2015 10:39	Firefox HTML Doc...	4 KB

# Hasil Running

## Hasil Pencarian

agung memiliki hobi bersepeda

## Form Pencarian

Masukkan Nama anda:



C A R I

# Penutup

- AJAX mendukung asynchronous sehingga halaman web dapat di-update seperlunya.
- Proses update website menjadi lebih cepat karena tidak seluruh halaman di-download.

# Next . . .

- Framework CodeIgniter

# Referensi

- <http://jquery.com>
- <http://w3school.com>
- <http://php.net/manual>