

# Praktikum 11

## Pemrograman Web

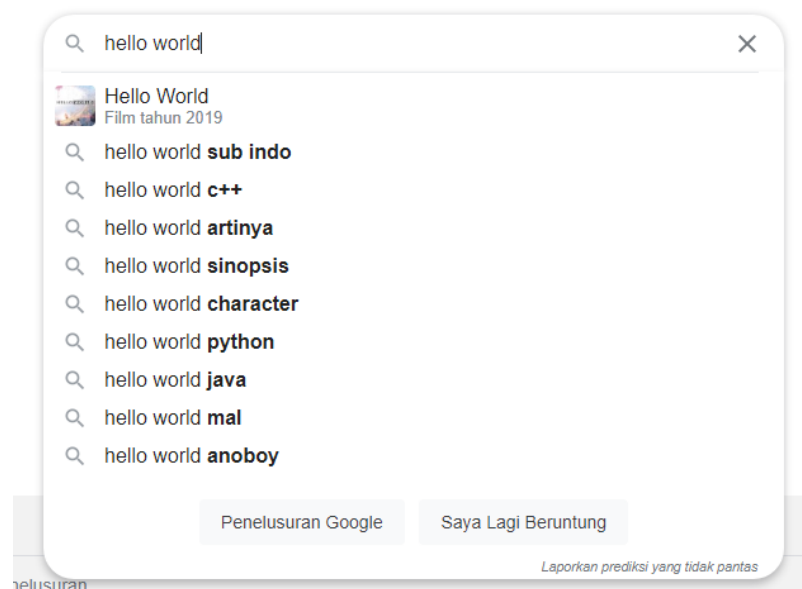
### jQuery - AJAX

Pada praktikum ke 11 ini kalian akan mempelajari mengenai pengertian dari AJAX dalam jQuery, serta konsep penerapan AJAX itu sendiri, pada praktikum kali ini kita akan mencoba mengambil data API dari suatu situs Website yang bernama open weather map.

Dalam pengambilan data pada situs open weather map nantinya kita akan menerapkan konsep jQuery dengan menggunakan AJAX. Lalu apa itu AJAX ? Dan bagaimana hubungan antara AJAX dengan jQuery itu sendiri ? mengenai hal tersebut simak penjelasan berikut,

AJAX adalah sebuah singkatan dari Asynchronous Javascript and XML dan mengacu pada sekumpulan teknis pengembangan web (web development) yang memungkinkan aplikasi web untuk bekerja secara asynchronous (tidak langsung) – memproses setiap request (permintaan) yang datang ke server di sisi background.

Contoh penggunaan AJAX sebagai berikut :



Sehingga, AJAX **bukan** merupakan suatu **bahasa pemrograman**, melainkan suatu teknik yang memungkinkan **membuat suatu website lebih interaktif**. Dengan menggunakan teknik ini suatu request & response ke server dapat dilakukan di background secara asynchronous sehingga halaman web tidak perlu **reload/refresh**.

Mengenai praktikum sebelumnya kalian pasti sudah belajar mengenai konsep penerapan jQuery, jadi kita dapat menggabungkan konsep jQuery dengan AJAX dengan tujuan untuk mengambil data API dari suatu situs website yang menyediakan API. Sehingga bentuk kode dalam menerapkan konsep AJAX dalam jQuery yaitu dengan menuliskan function **.ajax()** Serta untuk parameter yang digunakan sebagai berikut :

- **URL**, menentukan alamat URL mana yang akan direquest oleh HTTP Request (AJAX)
- **Type**, tipe request yang akan dijalankan, apakah POST atau GET
- **BeforeSend**, event yang dilakukan ketika HTTP Request akan diproses
- **Complete**, event yang dilakukan ketika HTTP Request selesai dijalankan, namun tidak melihat apakah request sukses atau gagal
- **Success**, event yang dilakukan ketika HTTP Request selesai dan berhasil dijalankan dan mempunyai nilai balik yaitu data hasil request
- **Data**, data yang dikirim ke server ketika request dan diolah di server. Data ini akan diberlakukan berdasarkan tipe yang ada, yaitu GET atau POST.

Oke, setelah mengetahui parameter dan bentuk fungsi pada AJAX dalam jQuery selanjutnya kita akan menerapkan konsep AJAX sederhana, dan melihat bentuk kode sederhana dari AJAX itu sendiri.

## Implementasi AJAX

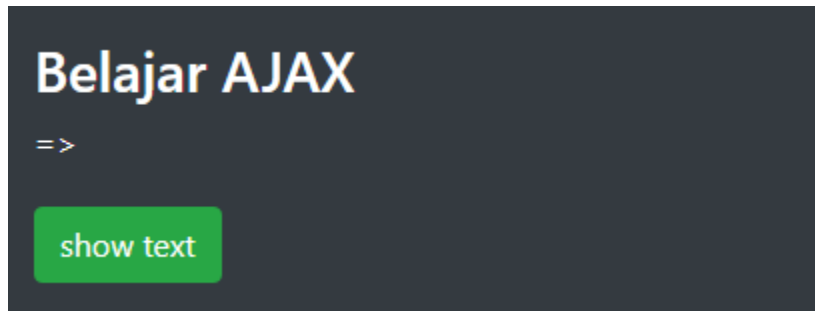
Pada sesi praktikum ini, kalian akan mengetahui bagaimana penerapan AJAX dalam jQuery, serta bagaimana bentuk penulisan kode pada AJAX.

Pertama, kalian siapkan file html baru letakan didalam folder dan beri nama folder praktikum 11, dan nama html sesuai dengan keinginan kalian. Kemudian, masukan script jquery, untuk link script bisa kunjungi tautan dibawah ini :

<https://code.jquery.com/>

Kemudian untuk yang menggunakan bootstrap jangan lupa jQuery bawaan dari bootstrap bisa dihapus diganti dengan CDN dari code.jquery.com.

Selanjutnya buat sebuah layout html seperti berikut,



Kemudian pada folder praktikum 11 didalamnya tambahkan file dengan format **txt** kemudian isikan text “Hallo ini file text” kemudian save.

Lalu copy kan script berikut :

```
<script>
    $(document).ready(function () {
        $("#btn1").click(function() {
            $.ajax({
                url: 'hello.txt',
                success: function(res) {
                    $("#result").text(res);
                }
            });
        });
    });
</script>
```

Untuk **button** dengan **id="btn1"**, dan untuk **paragraf** dengan **id="result"**.

Kemudian jalankan kode tersebut di browser. Lalu teman" pasti akan menjumpai error seperti berikut:

```
✖ Access to XMLHttpRequest at 'file:///E:/Semester%206%20(+)/Data%20Asisten%20Pemrogr
aman%20Web/praktikum9/hello.txt' from origin 'null'
has been blocked by CORS policy: Cross origin
requests are only supported for protocol schemes:
http, data, chrome, chrome-extension, chrome-
untrusted, https.
```

Hal diatas disebabkan karena AJAX hanya dapat berjalan pada suatu server tertentu, kali ini kita akan memanfaatkan server local yaitu menggunakan XAMPP, cara penerapannya yaitu pertama kalian bisa buka XAMPP Cpanel terlebih dahulu. Kemudian klik start pada bagian Apachenya. Setelah running lalu pindahkan folder praktikum 11 di dalam /htdocs, termasuk file-file HTML, gambar, css, atau file image yang ada sebelumnya dipindah ke direktori /xampp/htdocs/. serta letak htdocs sendiri terletak didalam folder XAMPP saat kalian lakukan instalasi.

Mengenai instalasi XAMPP teman” bisa mengunjungi web berikut:

<https://www.rumahweb.com/journal/install-xampp-di-windows/>

Setelah itu kalian bisa kembali ke browser kalian kemudian untuk menjalankan script yang telah kita buat melalui browser, kita isikan alamat/url nya: localhost/praktikum9/latihan.html, dimana url harus sesuai dengan nama folder dan file html kalian. Lalu tekan enter. Hasilnya menjadi seperti berikut :



Oke dengan ini kalian sudah paham mengenai konsep Ajax secara dasar.

## Akses API

kemudian selanjutnya kita akan membuat kode jQuery dengan menerapkan konsep Ajax untuk mengakses API dari open Weather map. Kalian dapat mengunjunginya pada link berikut :

<https://openweathermap.org/>

Lalu silahkan kalian bisa melakukan Sign In terlebih dahulu pada website tersebut untuk mendapatkan API Key.

Setelah membuat/register akun, kalian bisa cek API Keys dari menu Api keys atau di url [https://home.openweathermap.org/api\\_keys](https://home.openweathermap.org/api_keys) . Kemudian simpan data Key tersebut selanjutnya akan kita gunakan untuk mengambil data.

New Products Services **API keys** Billing plans Payments Block logs Marketplace My orders My profile

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name		Create key
<div>4d1d9943-499e-4166c93893b6f1ae</div>	Default	<div></div>	<div>* Name</div> <div></div> <div>Generate</div>
<div>4d38999941-333f-45881c91-9-9971</div>	Forest Wallpaper API	<div></div>	

Gunakan Key Default untuk project ini. Jika kalian ingin membuat lebih dari 1 aplikasi yang akses ke data weather ini maka kalian bisa create key baru lagi.

Pada versi gratis ini kita hanya dapat mengambil data real time pada suatu lokasi dan forecast/perkiraan 5 hari/3 jam kedepan saja. Untuk mengambil data saat ini gunakan url dibawah ini:

[api.openweathermap.org/data/2.5/weather?q={city name}&appid={your api key}](https://api.openweathermap.org/data/2.5/weather?q={city name}&appid={your api key})

Pada {city name} bisa kita ganti nama kota, misalnya Sleman dan {your api key} kita gantikan dengan key yang sudah kita dapatkan dari login sebelumnya.

Sedangkan untuk mengambil data forecast kita bisa gunakan url:

[api.openweathermap.org/data/2.5/forecast?q={city name}&appid={your api key}](https://api.openweathermap.org/data/2.5/forecast?q={city name}&appid={your api key})

Buat file html didalam folder praktikum 11, beri nama **weather.html**,  
Boleh menggunakan bootstrap, dan CDN jQuery.

Kemudian <script> untuk mendeklarasikan variabel api, key dan city dimana ketiga variabel ini akan digunakan sebagai url untuk akses API menggunakan Ajax. dengan mengcopy kode berikut kedalam file html kalian :

```

<script>
  api = 'https://api.openweathermap.org/data/2.5/';
  key = '4de4xxx';
  city = 'London';

  $(document).ready(function () {
    $.ajax({
      url: api + 'weather?q=' + city + '&appid=' + key +
'&units=metric',
      success: function (res) {
        console.log(JSON.stringify(res));
      }
    });
  });
</script>

```

Dengan menggunakan function `JSON.stringify( )` output JSON akan diubah ke string dan ditampilkan di console browser untuk mempermudah kita memahami struktur output dari API tersebut.

```

{"coord":{"lon":-0.13,"lat":51.51},"weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04n"}],
"base":"stations","main":{"temp":14.24,"feels_like":12.02,"temp_min":13.33,"temp_max":15,"pressure":1009,"humidity":87},
"visibility":10000,"wind":{"speed":4.1,"deg":60},"clouds":{"all":90},"dt":1591925645,"sys":{"type":1,"id":1414,"country":"GB",
"sunrise":1591933395,"sunset":1591993069},"timezone":3600,"id":2643743,"name":"London","cod":200}

```

Dari output tersebut kita membutuhkan formatter JSON untuk memahami struktur JSON. Gunakan online JSON formatter di <https://jsonformatter.org/> kemudian copy output tersebut dan klik tombol format/beauty.

Sebagai contoh kita akan mengambil data temperature, maka kita harus memahami struktur dari JSON tersebut dan object temperatur terletak pada object main, sehingga caranya adalah:

```

var condition = res.weather[0].main;

```

Selanjutnya kita bisa siapkan view untuk menampilkan data real-time cuaca tersebut, dapat kalian tuliskan kode HTML berikut ini

```

<div class="container p-3 my-3 text-center">
  <h2>Cuaca di <span id="kota"></span></h2>
  <p id="temp" class="display-4"></p>
  <img height="200"/>
  <h3></h3>
  <p>
    Min: <span id="min"></span> Max: <span id="max"></span>
  </p>
</div>

```

Dari kode HTML diatas kita akan memberi nilai pada beberapa elemen yang sudah ada tersebut. Berikut elemen yang perlu kita set nilainya:

1. Atribut id = temp
2. Element img
3. Element h3
4. Atribut id=min dan id=max

Berikut ini script JS untuk assign nilai yang didapat dari output API tersebut :

```

$('#kota').html(res.name);
$('#temp').html(res.main.temp+'&deg;C');
$('#min').html(res.main.temp_min+'&deg;C');
$('#max').html(res.main.temp_max+'&deg;C');

var icon = res.weather[0].icon;

$('img').attr('src','http://openweathermap.org/img/wn/'+icon+'@2x.png');
$('h3').text(res.weather[0].main+' - '+res.weather[0].description);

```

Outputnya akan menjadi seperti berikut :

**Cuaca di London**

13.19°C



**Rain - moderate rain**

Min: 12.38°C Max: 14.27°C