MINGGU 13 (PEMROGRAMAN BERBASIS WEB)

Javascript (BASIC session-4)

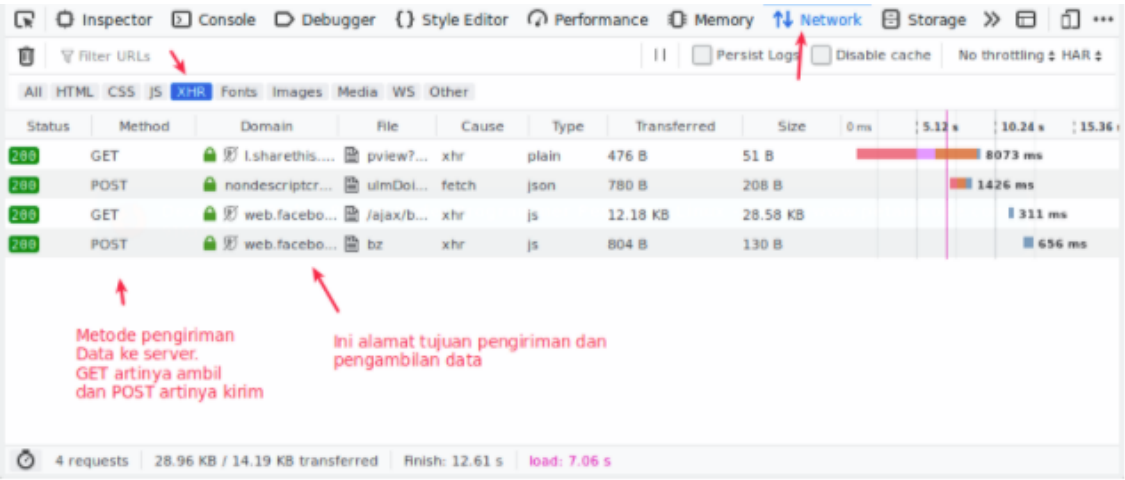
Pada pertemuan 13 ini melanjutkan praktikum javascript session-3, akan dibahas mengenai AJAX, dan Mengenal JSON dan kegunaannya pada Web Programming.

AJAX

AJAX merupakan singkatan dari **A**synchronous **J**avascript **A**nd **X**ML, diantaranya berfungsi untuk:

- Mengambil data dari server ***secara background***;
- Update tampilan web tanpa harus ***relaod browser***;
- Mengirim data ke server ***secara background***.

Secara prinsip AJAX menggunakan objek ***XMLHttpRequest*** untuk berkomunikasi dengan server, untuk melihat proses AJAX dapat melalui inspect elemen pada browser, kemudian dengan membuka tab ***Network*** dan aktifkan filter ***XHR*** *(XML HTTP Request)*.



Pada singkatan di AJAX ada kata "XML", tetapi bukan berarti hanya mendukung ***XML*** saja, karena AJAX juga mendukung format lain seperi JSON, Plain Text, maupun HTML.

Cara Penggunaan pada Javascript

Langkah-langkah menggunakan AJAX sebagai berikut:

1. Membuat Objek Ajax

```
var xhr = new XMLHttpRequest();
```

2. Menentukan fungsi handler untuk Event

```
xhr.onreadystatechange = function() { ... };
xhr.onload = function() { ... };
xhr.onerror = function() { ... };
xhr.onprogress = function() { ... };
```

3. Menentukan method dan URL

```
xhr.open("GET", url, true);
```

4. Mengirim request

```
xhr.send();
```

Praktikum 1 : buat file **ajax-1.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Belajar Dasar Ajax</title>
</head>
<body>
    <h1>Tutorial Ajax</h1>
    <div id="hasil"></div>
    <button onclick="loadContent()">Load Content</button>

    <script>
        function loadContent(){
            var xhr = new XMLHttpRequest();
            var url = "http://localhost/ajax_files/kode.json";
            xhr.onreadystatechange = function(){
                if(this.readyState == 4 && this.status == 200){
                    document.getElementById("hasil").innerHTML =
this.responseText;
                }
            };
            xhr.open("GET", url, true);
            xhr.send();
```

```
        }
    </script>
</body>
</html>
```

Catatan: *kode.json dapat diambil pada url materi : http://devel.dinustech.com/aboe/materi/data/ bisa disesuaikan untuk path dan filenya, pastikan webserver di laptop / PC sudah jalan. Praktikum AJAX hanya bisa jalan pada URL* ***http:// atau https://***

Pada contoh di atas, proses mengambil data dari file.txt dengan methode GET. kemudian hasilnya dimasukan ke dalam elemn , event yang digunakan adalah ***onreadystatechange***, pada event ini bisa mengecek *state* dan status AJAX.

```
if(this.readyState == 4 && this.status == 200){
    //...
}
```

Kode state 4 artinya ***done*** dan status 200 artinya ***sukses***. Berikut adalah daftar kode state AJAX:

Kode	State	Keterangan
0	UNSENT	Objek AAJAX sudah dibuat tapi belum memanggil method `open()`.
1	OPENED	Method `open()` sudah dipanggil.
2	HEADERS_RECEIVED	Method `send()` sudah dipanggil, dan di sini sudah tersedia header status.
3	LOADING	Downloading; sedang mendownload data.
4	DONE	Operasi AJAX selesai.

Sementara untuk status header 200 adalah status HTTP Request. Biasanya kode di atas 200 artinya ***baik*** dan di bawah 200 artinya ***buruk***.

Perhatikan kode berikut ini:

```
xhr.open("GET", url, true);
```

Terdapat tiga parameter yang diberikan kepada method ***open()***:

1. **GET** adalah metode request yang akan digunakan;
2. **url** adalah alamat URL tujuan;
3. **true** adalah untuk mengeksekusi AJAX secara *asynchronous*.

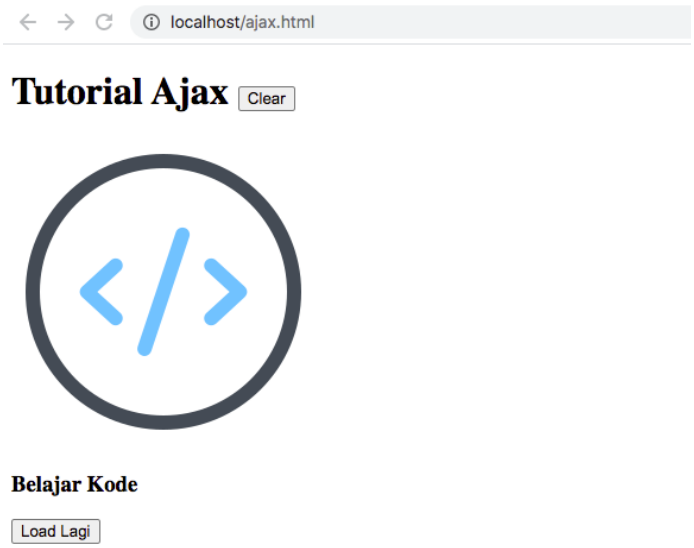
Praktikum 2 : buat file **ajax-2.html**

Modifikasi file sebelumnya menjadi seperti berikut ini:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Belajar Dasar Ajax</title>
</head>

<body>
    <h1>Tutorial Ajax <button id="btn-clear"
onclick="clearResult()">Clear</button></h1>
    <div id="hasil"></div>
    <button id="button" onclick="loadContent()">Load Content</button>

    <script>
        function loadContent() {
            var xhr = new XMLHttpRequest();
            var url = "http://localhost/ajax_files/kode.json";

            xhr.onloadstart = function () {
                document.getElementById("button").innerHTML = "Loading...";
            }

            xhr.onerror = function () {
                alert("Gagal mengambil data");
            };

            xhr.onloadend = function () {
                if (this.responseText !== "") {
                    var data = JSON.parse(this.responseText);
                    var img = document.createElement("img");
                    img.src = data.avatar_url;
                    var name = document.createElement("h3");
                    name.innerHTML = data.name;

                    document.getElementById("hasil").append(img, name);
                    document.getElementById("button").innerHTML = "Done";

                    setTimeout(function () {
                        document.getElementById("button").innerHTML = "Load
Lagi";
                    }, 3000);
                }
            };
```

```
            xhr.open("GET", url, true);
            xhr.send();
        }

        function clearResult() {
            document.getElementById("hasil").innerHTML = "";
        }
    </script>
</body>
</html>
```

Jika berhasil pada saat klik tombol load akan muncul tampilan sebagai berikut:



Mengirim data ke server ajax

Praktikum 3 : buat file **ajax-3.html**

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Belajar Dasar Ajax</title>
</head>

<body>
    <h1>Kirim Data dengan Ajax</h1>
    <form method="POST" onsubmit="return sendData()">
        <p>
            <label>Title</label>
            <input type="text" id="title" placeholder="judul artikel">
        </p>
```

```
        <p>
            <label>Isi Artikel</label><br>
            <textarea id="body" placeholder="isi artikel..." cols="50"
rows="10"></textarea>
        </p>
        <input type="submit" value="Kirim" />
    </form>

    <script>
        function sendData() {
            var xhr = new XMLHttpRequest();
            var url = "https://jsonplaceholder.typicode.com/posts";

            var data = JSON.stringify({
                title: document.getElementById("title").value,
                body: document.getElementById("body").value,
                userId: 1
            });

            xhr.open("POST", url, true);
            xhr.setRequestHeader("Content-Type",
"application/json;charset=UTF-8");
            xhr.onload = function () {
                console.log (this.responseText);
            };

            xhr.send(data);
            return false;
        }
    </script>
</body>
</html>
```

Catatan: Pengiriman data hanya untuk simulasi. Data tidak benar-benar terkirim ke server https://jsonplaceholder.typicode.com/posts.

AJAX menggunakan jQuery

JQuery adalah library Javascript yang menyederhanakan fungsi-fungsi Javascript. Pada JQuery, AJAX dapat dibuat seperti dibawah ini:

```
// load data ke elemen tertentu via AJAX
$(selector).load(URL,data,callback);

// ambil data dari server
$.get(URL,callback);

// kirim data dari Server
$.post(URL,data,callback);
```

Praktikum 4 : buat file **ajax-jquery.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Belajar AJAX dengan JQuery</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></scr
ipt>
</head>
<body>
    <h1>Load Data</h1>
    <pre id="result"></pre>

    <script>
        $("#result").load("http://localhost/ajax_files/kode.json");
    </script>
</body>
</html>
```

Dengan fungsi ***$("#result").load()***, dapat mengambil data dengan AJAX dan langsung menampilkannya pada elemen yang dipilih, fungsi JQuery ***load()*** cocok untuk mengambil bagian dari HTML untuk ditampilkan.

Praktikum 5 : buat file **ajax-jquery2.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Belajar AJAX dengan JQuery</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></scr
ipt>
</head>
<body>
    <h1>Load Data</h1>
    <img id="avatar" src="" width="100" height="100">
    <br>
    <h3 id="name"></h3>
    <p id="location"></p>


    <script>
        var url = "http://localhost/ajax_files/kode.json";
        $.get(url, function(data, status){
            $("#avatar").attr("src", data.avatar_url);
            $("#name").text(data.name);
```

```
            $("#location").text(data.location);
        });
    </script>
</body>
</html>
```

Untuk mengirim data dengan AJAX pada jQuery, caranya sama seperti mengambil data dengan $.get().

```
<script>
var url = "https://jsonplaceholder.typicode.com/posts";
var data = {
    title: "Tutorial AJAX dengan JQuery",
    body: "Ini adalah tutorial tentang AJAX",
    userId: 1
};
$.post(url, data, function(data, status){
    // data terkirim, lakukan sesuatu di sini
});
</script>
```

AJAX menggunakan fetch API

Fetch artinya mengambil. Metode *fetch* bisa jadi alternatif untuk AJAX, methode ini mulai hadir pada Javascript versi ES6.

Perbedaanya dengan **XMLHttpRequest** dan **JQuery** adalah:

- **Fetch** akan mengembalikan sebuah promise;
- Secara bawaan (default), ***fetch*** tidak akan mengirim atau menerima cookie dari server.
- **Fetch** dapat digunakan di web browser.

Berikut ini sintak dasar penggunaan ***Fetch***.

```
fetch('http://example.com/movies.json')
  .then(function(response) {
    return response.json();
  })
  .then(function(myJson) {
    console.log(JSON.stringify(myJson));
  });
```

Dan untuk pengiriman data dengan metode POST, bentuknya seperti dibawah ini:

```
fetch(url, {
        method: "POST", // *GET, POST, PUT, DELETE, etc.
        mode: "cors", // no-cors, cors, *same-origin
```

```
        cache: "no-cache", // *default, no-cache, reload, force-cache, only-
if-cached
        credentials: "same-origin", // include, *same-origin, omit
        headers: {
            "Content-Type": "application/json",
            // "Content-Type": "application/x-www-form-urlencoded",
        },
        redirect: "follow", // manual, *follow, error
        referrer: "no-referrer", // no-referrer, *client
        body: JSON.stringify(data), // body data type must match "Content-
Type" header
    })
    .then(response => response.json()); // parses JSON response into native
Javascript objects
```

Praktikum 6 : buat file **ajax-fetch.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Belajar Dasar Ajax dengan Fetch</title>
</head>
<body>
    <h1>Tutorial Ajax dengan Fetch</h1>
    <button onclick="loadContent()">Load Content</button>
    <div id="hasil"></div>

    <script>
        function loadContent(){
            var url = "https://jsonplaceholder.typicode.com/posts/";
            fetch(url).then(response => response.json())
                .then(function(data){
                    var template = data.map(post => {
                        return `
                        <h3>${post.title}</h3>
                        <p>${post.body}</p>
                        <hr>
                        `;
                    });

                    document.getElementById("hasil").innerHTML =
template.join('<br>');
                }).catch(function(e){
                    alert("gagal mengambil data");
                });
        }
    </script>
</body>
</html>
```

AJAX menggunakan AXIOS

Axios hampir sama seperti ***fetch***. **Bedanya** Axios adalah sebuah library sedangkan *fetch* adalah API yang tersedia di web browser.

```
axios.get('https://jsonplaceholder.typicode.com/posts/')
  .then(function (response) {
    // handle success
    console.log(response);
  })
  .catch(function (error) {
    // handle error
    console.log(error);
  })
  .then(function () {
    // always executed
  });
```

Praktikum 7 : buat file **ajax-axios.html**

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Tutorial AJAX dengan AXIOS</title>
    <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
</head>

<body>
    <h1>Tutorial AJAX dengan AXIOS</h1>
    <button id="btn-load" onclick="loadContent()">Load Content</button>
    <div id="result"></div>

    <script>
        function loadContent() {
            document.getElementById("btn-load").innerHTML = "loading...";
            document.getElementById("btn-load").setAttribute("disabled",
"true");
            axios.get('https://jsonplaceholder.typicode.com/posts/')
                .then(function (response) {
                    var template = response.data.map(post => {
                        return `
                        <h3>${post.title}</h3>
                        <p>${post.body}</p>
                        <hr>
                        `
                    }).join("");

                    document.getElementById("result").innerHTML = template;
                })
                .catch(function (error) {
                    // handle error
                    console.log(error);
                })
```

```
                .then(function () {
                    document.getElementById("btn-load").innerHTML = "Done";
                    document.getElementById("btn-
load").removeAttribute("disabled");
                });
        }
    </script>
</body>

</html>
```

Mengenal JSON

JSON *(JavaScript Object Notation)* adalah sebuah **format data** yang digunakan untuk **pertukaran dan penyimpanan data**.

JSON merupakan bagian *(subset)* dari Javascript. JSON bisa dibaca dengan berbagai macam bahasa pemrograman seperti C, C++, C#, Java, Javascript Perl, Python, php dan banyak lagi, hal ini membuat JSON menjadi bahasa yang ideal untuk **pertukaran data antar aplikasi** dan kalau dibandingkan dengan XML, JSON lebih sederhana dan mudah dibaca.

Sejarah singkat JSON

JSON pertamakali dipopulerkan oleh Douglas Crockford, *software engineer* yang juga terlibat pengembangan bahasa pemrograman Javascript, sebelumnya arti kata "JSON" belum ada. Orang-orang hanya mengenal Objek Javascript yang dikirim melalui jaringan, sejak meledaknya teknologi AJAX pada tahun 2000. JSON mulai diperkenalkan dan pada tahun 2001, domain json.org mulai aktif dan hingga saat ini JSON banyak digunakan di mana-mana.

Penerapan JSON dalam pemrograman

JSON biasanya digunakan sebagai format standar untuk bertukar data antar aplikasi.

Dan sebenarnya tidak hanya itu, masih ada fungsi lain dari JSON.

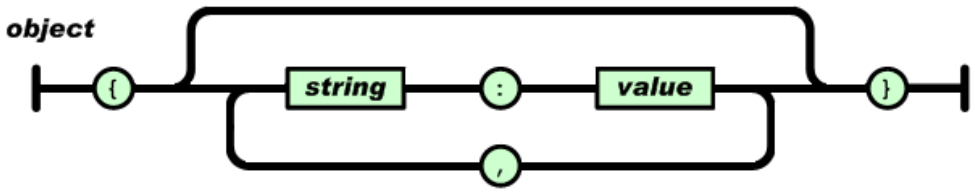
Berikut ini beberapa penerapan JSON yang sering digunakan :

- JSON sebagai format untuk bertukar data client dan server atau antar aplikasi. Contoh: RESTful API;
- JSON sebagai tempat menyimpan data, contoh: Database Mongodb;
- JSON digunakan untuk menyimpan konfigurasi project, contoh: file composer.json pada project PHP dan package.json pada Nodejs;
- JSON digunakan untuk menyimpan konfigurasi dan penyimpanan data pada Hugo;
- JSON digunakan untuk menyimpan konfigurasi project pada Nodejs;
- JSON digunakan untuk menyimpan data menifest;
- dan masih banyak lagi.

Struktur Dasar JSON

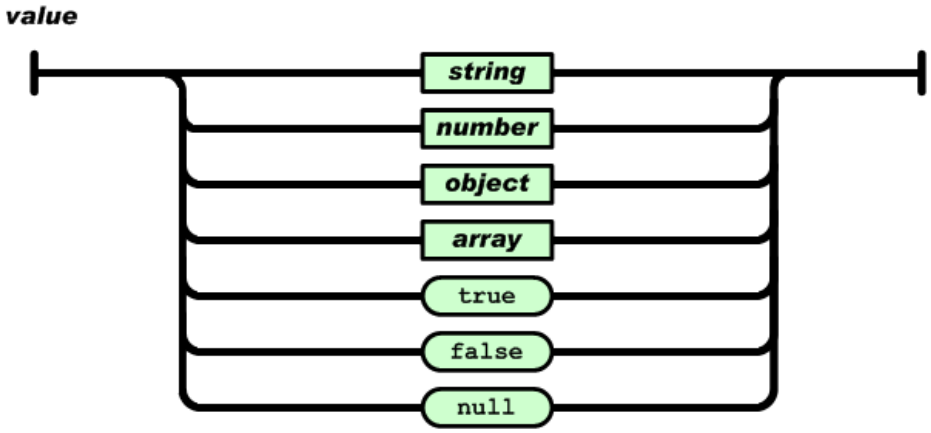
Struktur sederhana JSON:

JSON selalu dimulai dengan tanda kurung kurawal { dan ditutup }, kemudian di dalam kurung kurawal, berisi data yang format *key* dan *value*. Jika terdapat lebih dari satu data, maka dipisah dengan tanda koma dan di data terakhir tidak diberikan koma.

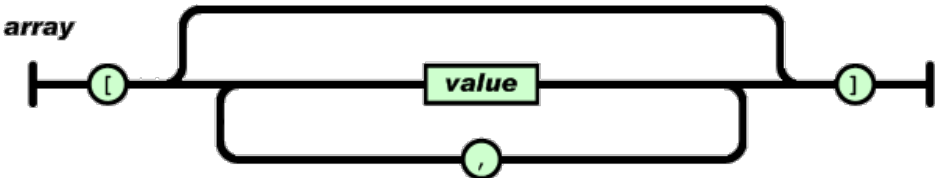


Kemudian key dan value dipisah dengan titik dua.

Type data yang didukung JSON:



Untuk array, dibuat dengan tanda kurung siku:



Contoh:

```
{
    "name": "Budi Sudarsono",
    "hobbies": ["Coding", "Blogging", "Drawing"]
}
```

Contoh Objek :

```
{
    "name": "belajarKode",
    "url": "dinus.ac.id",
    "rank": 1,
    "socialmedia": {
        "facebook": "belajarkode",
        "twitter": "belajarkode ",
        "instagram": "belajarkode ",
        "youtube": "belajarkode ",
        "github": "belajarkode "
    }
}
```

Perhatikan key socialmedia diatas, proses pemberian nilainya dengan objek.

Cara Pembuatan JSON

Setiap bahasa pemrograman memiliki cara yang berbeda-beda untuk membuat dan memakai JSON.

Pada Javascript, bisa menggunakan fungsi ***JSON.stringify()*** untuk membuat JSON dari objek Javascript, contoh:

```
// objek javascript
var person = {
    name: "Budi",
    age: 20
}

// string JSON
var jsonString = JSON.stringify(person);

// maka akan menghasilkan:
// {"name":"Budi","age":20}
```

Parsing Data JSON dengan Javascript

Praktikum 8 : Kasus JSON, buat 1 folder dengan nama json-lat, kemudian disikan file Latihan berikut ini:

- Buat file data.json

```
[
  {
      "nim":"1234",
      "nama":"Agung Suradji",
      "alamat":"Semarang",
      "jurusan":"Teknik Informatika",
      "MK":[
         {
            "mk_kode":"101",
            "nama_mk":"SISTEM BASIS DATA"
         },
         {
            "mk_kode":"102",
            "nama_mk":"ALJABAR LINIER"
         },
         {
            "mk_kode":"103",
            "nama_mk":"ANALISA ALGORITMA"
         },
         {
            "mk_kode":"104",
```

```
            "nama_mk":"KECERDASAN BUATAN"
         },
         {
            "mk_kode":"105",
            "nama_mk":"JARINGAN KOMPUTER"
         }
      ]
   },
   {
      "nim":"1236",
      "nama":"Mira Rahmawan",
      "alamat":"Pemalang",
      "jurusan":"Sistem Informasi",
      "MK":[
         {
            "mk_kode":"301",
            "nama_mk":"ANALISA PROSES BISNIS"
         },
         {
            "mk_kode":"302",
            "nama_mk":"DATABASE TERDISTRIBUSI"
         },
         {
            "mk_kode":"303",
            "nama_mk":"ETIKA PROFESI"
         },
         {
            "mk_kode":"304",
            "nama_mk":"REKAYASA WEB"
         },
         {
            "mk_kode":"305",
            "nama_mk":"AUDIT SISTEM INFORMASI"
         }
      ]
   }
]
```

- Buat file latihan-json.html

```
<!DOCTYPE html>
<html>
<head>
 <title>Mari Belajar Coding</title>
 <script type="text/javascript">
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
         // Typical action to be performed when the document is ready:
         var data=JSON.parse(xhttp.responseText);
         console.log(data);
      }
  };
  xhttp.open("GET", "data.json", true);
  xhttp.send();

 </script>
</head>
<body>
```

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

Jalankan dan lihat dari console data json akan ditampilkan menjadi data objek.

- Buat file latihan-json2.html

  Untuk bisa ditampilkan pada tampilan browser dapat menggunakan append().

```
<!DOCTYPE html>
<html>
<head>
 <title>Mari Belajar Coding</title>
 <script type="text/javascript">
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
         var data=JSON.parse(xhttp.responseText);
         data.forEach(function(element) {
      document.getElementById("demo").innerHTML +="NIM :" +
element.nim + "<br>Nama :" + element.nama + "<br>Alamat :" +
element.alamat + "<br><br>";
    });
      }
  };
  xhttp.open("GET", "data.json", true);
  xhttp.send();

 </script>
</head>
<body>
<p id="demo"></p>
</body>
</html>
```

- Buat file latihan-json-jquery.html

```
<!DOCTYPE html>
<html>
<head>
 <title>Mari Belajar Coding</title>
 <script
   src="http://code.jquery.com/jquery-3.3.1.js"
   integrity="sha256-2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60="
   crossorigin="anonymous"></script>

   <script type="text/javascript">
     $.getJSON("data.json", function(result){
       console.log(result);
     });
   </script>
</head>
<body>
</body>
</html>
```

- Buat file latihan-json-jquery2.html

```
<!DOCTYPE html>
<html>
<head>
 <title>Mari Belajar Coding</title>
 <script
   src="http://code.jquery.com/jquery-3.3.1.js"
   integrity="sha256-2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60="
   crossorigin="anonymous"></script>

   <script type="text/javascript">
     $.getJSON("data.json", function(result){
       console.log(result);
       $.each(result, function(i){
         document.getElementById("demo").innerHTML +="NIM :" +
result[i].nim + "<br>Nama :" + result[i].nama + "<br>Alamat :" +
result[i].alamat + "<br><br>";
       });
     });
   </script>
</head>
<body>
 <div id="demo"></div>
</body>
</html>
```

Repositori Assignment #10 Pertemuan 13

Push hasil latihan ke Github (repository webdas) dan kirim urlnya melalui kulino pada blok (Repositori Assignment #10 Pertemuan 13).

Untuk susunan folder dan file sebagai berikut

- repominggu13 (folder utama)
  - latihanJavascript4
    - ajax-1.html
    - dan seterusnya sesuai susunan folder Latihan diatas.