

## MODUL 11 PENGENALAN AJAX DAN JSON

---

### 11.1. Deskripsi Singkat

Aplikasi berbasis web telah mengalami perkembangan beberapa kali, dari yang pertama dikenal dengan Web 1.0, kemudian berkembang menjadi yang dikenal dengan Web 2.0 dalam era media sosial, lalu kemudian, sekarang, memasuki era Web 3.0 dengan kecerdasan artifisial (AI) dan blockchain. Yang menjadi salah satu kunci dalam perkembangan teknologi web ini adalah dimungkinkannya antar halaman web atau program berbasis web untuk saling berinteraksi secara langsung dan dinamis. Dalam modul ini, mahasiswa diperkenalkan dengan teknologi AJAX (Asynchronous Javascript and XML). Selain itu, mahasiswa juga diperkenalkan dengan JSON.

### 11.2. Tujuan Praktikum

Setelah menyelesaikan praktikum pada modul ini, mahasiswa diharapkan dapat mengenal bagaimana cara kerja dan dapat membuat aplikasi web yang dinamis menggunakan AJAX, serta dapat memahami penggunaan JSON.

### 11.3. Material Praktikum

Praktikum menggunakan editor teks seperti Notepad atau Notepad++ atau aplikasi editor teks lainnya yang tersedia untuk menulis HTML dan program PHP. Pada praktikum ini juga menggunakan web server Apache, program PHP, dan RDBMS MariaDB, yang telah disiapkan pada modul sebelumnya.

### 11.4. Kegiatan Praktikum

#### 11.4.1. Membuat Aplikasi AJAX Sederhana

Sebagai latihan pertama, kita akan coba membuat aplikasi Ajax sederhana menggunakan PHP sebagai aplikasi *backend*-nya. Kita akan membuat satu halaman web yang akan mengambil waktu server dan menampilkannya tanpa memuat ulang seluruh halaman.

1. Buka editor teks dan masukkan kode PHP berikut:

```
<?php
    echo date('H:i:s');
?>
```

Simpan dalam file bernama **telltime.php** di web server. Buka dengan web browser dan pastikan menampilkan waktu saat halaman ini dibuka seperti berikut.

18:01:26

Jika waktu tidak menunjukkan waktu yang sebenarnya, misalnya jamnya salah, ada kemungkinan zona waktu (*timezone*) tidak sesuai karena menunjukkan waktu GMT. Coba hitung apakah jam yang ditunjukkan, jika ditambah 7 sesuai dengan waktu WIB. Jika itu penyebabnya, perlu dilakukan penyesuaian zona waktu di pengaturan PHP. Pengaturan bisa dilakukan dengan menambahkan perintah berikut sebelum memanggil fungsi `date()` seperti berikut:

```
date_default_timezone_set('Asia/Jakarta'); // Zona waktu WIB
                                           // Lihat referensi
                                           // untuk zona waktu
                                           // WITA atau WIT
```

Pengaturan di atas hanya berlaku pada kode php yang dijalankan. Jika ingin mengubah pengaturan untuk semua kode PHP di komputer yang sama, maka perlu menambahkan perintah berikut pada file pengaturan PHP `php.ini`.

```
date.timezone = "Asia/Jakarta"
```

Pelajari program PHP yang anda gunakan untuk mengetahui lokasi file `php.ini`.

2. Buka editor teks dan masukkan markup HTML dan JavaScript berikut:

```
<!DOCTYPE html>
<html lang="en-ID">
  <head>
    <title>Ajax Demonstration</title>
    <meta charset="utf-8">
  </head>
  <style>
    .displaybox {
      display: table-cell;
      vertical-align: middle;
      width: 150px;
      height: 40px;
      background-color: #FFFFFF;
      border: 2px solid #000000;
      padding: 10px;
      font: 1.5em normal verdana, helvetica, arial, sans-serif;
    }
  </style>
  <script>
    var ajaxRequest;

    function ajaxResponse() {
      // check to see if we're done
      if (ajaxRequest.readyState != 4) {
        return;
      } else {
        // check to see if successful
        if (ajaxRequest.status == 200) {
```

```

        var plholder = document.getElementById("showtime");
        plholder.innerHTML = ajaxRequest.responseText;
    } else {
        alert("Request failed: " + ajaxRequest.statusText);
    }
}
}

function getServerTime() {
    ajaxRequest = new XMLHttpRequest();
    var svcUrl = "tellttime.php";
    ajaxRequest.onreadystatechange = ajaxResponse;
    ajaxRequest.open("GET", svcUrl);
    ajaxRequest.send(null);
}

// Write a function to auto-update the time

</script>
<body onload="getServerTime()">
    <h1>Ajax Demonstration</h1>
    <h2>Getting the server time without refreshing the page</h2>
    <form>
        <input type="button" value="Get Server Time"
            onclick="getServerTime()"/>
    </form><br/>
    <div id="showtime" class="displaybox"></div>
</body>
</html>

```

Simpan dalam file **showtime.html** di web server.

3. Buka **showtime.html** dengan web browser. Pastikan anda melihat kotak berisi tampilan waktu server. Tunggu beberapa waktu, lalu klik tombol "Get Server Time". Pastikan kotak waktu berubah menampilkan waktu terkini.
4. Saat ini, untuk memperbarui tampilan waktu, pengguna harus mengklik tombol "Get Server Time" secara manual. Ubah program agar dapat menampilkan waktu secara otomatis setiap detik.

Buat fungsi baru **autoUpdateTime()** di tempat yang ditunjukkan dengan komentar dalam file **showtime.html** di atas. Dalam fungsi ini, tuliskan kode JavaScript yang memanggil fungsi **getServerTime()** tiap satu detik. **Hint:** Gunakan fungsi **setInterval()** yang sudah disediakan oleh JavaScript.

Setelah itu, agar fungsi **autoUpdateTime()** otomatis dipanggil / dieksekusi ketika halaman dimuat, tambahkan perintah pemanggilan fungsi **autoUpdateTime()** pada *event handler* **onload** pada elemen **body**.

5. Simpan perubahan dan muat ulang ke dalam browser. Pastikan bahwa setelah halaman selesai dimuat dan waktu ditampilkan untuk pertama kali, waktu diperbarui setiap satu detik.
6. Saat ini, hanya ada waktu saja yang ditampilkan. Ubah kode PHP `telltime.php` di atas agar menampilkan juga tanggal seperti ini:

2020-04-18 15:25:53

Anda mungkin perlu menyesuaikan *style-sheet* di dalam markup `showtime.html` agar tanggal dan waktu dalam satu baris.

7. Simpan file `telltime.php` dan `showtime.html`, lalu muat ulang ke dalam browser. Pastikan tampilan halaman sesuai dengan yang diharapkan. Coba buka di penghujung hari dan amati bagaimana perubahan ketika berganti hari.

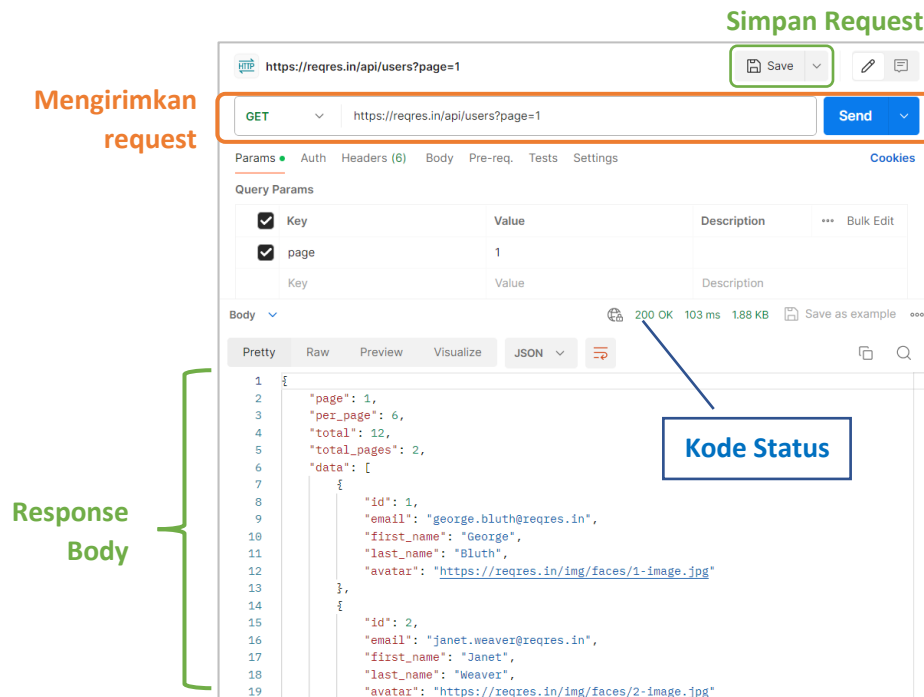
#### 11.4.2. Memproses data JSON dari API

JSON merupakan salah satu format data yang banyak digunakan saat ini, baik dalam penggunaan AJAX maupun dalam komunikasi antar web melalui API. Pada kegiatan praktikum ini, pertama-tama kita akan mencoba mengakses open API menggunakan Postman. Selanjutnya, kita akan mencoba mengakses open API menggunakan PHP dan memproses datanya agar dapat ditampilkan pada sebuah halaman web.

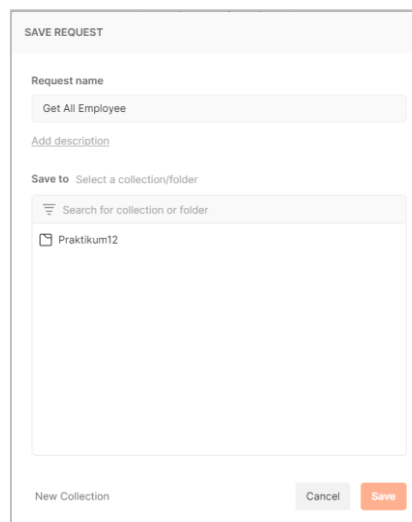
1. Berikut ini adalah web api yang akan Anda gunakan untuk belajar: <https://reqres.in/>. Kunjungi tautan tersebut dan Anda akan melihat berbagai request URL yang bisa digunakan untuk mengambil data (GET), menambahkan data (POST), mengubah data (PUT) dan juga menghapus data (DELETE).

|        |                             |   |
|--------|-----------------------------|---|
| GET    | LIST USERS                  | <pre>{   "page": 2,   "per_page": 6,   "total": 12,   "total_pages": 2,   "data": [     {       "id": 7,       "email": "michael.lawson@reqres.in",       "first_name": "Michael",       "last_name": "Lawson",       "avatar": "https://reqres.in/img/fac"     },     {       "id": 8,       "email": "lindsay.ferguson@reqres.in",       "first_name": "Lindsay",       "last_name": "Ferguson",       "avatar": "https://reqres.in/img/fac"     },     {       "id": 9,       "email": "tobias.funke@reqres.in",       "first_name": "Tobias",       "last_name": "Funke",       "avatar": "https://reqres.in/img/fac"     }   ] }</pre> |
| GET    | SINGLE USER                 |   |
| GET    | SINGLE USER NOT FOUND       |   |
| GET    | LIST <RESOURCE>             |   |
| GET    | SINGLE <RESOURCE>           |   |
| GET    | SINGLE <RESOURCE> NOT FOUND |   |
| POST   | CREATE                      |   |
| PUT    | UPDATE                      |   |
| PATCH  | UPDATE                      |   |
| DELETE | DELETE                      |   |

2. Cobalah untuk melakukan request ke service-service tersebut. Yang pertama akan kita coba adalah GET LIST USERS. Masukkan URL <https://reqres.in/api/users?page=1> untuk mendapatkan seluruh data user pada postman Anda. Pastikan method yang dipilih pada postman sudah sesuai, yaitu **GET**. Klik tombol **SEND**. Jika request berhasil maka Anda akan melihat response berupa data JSON.



3. Klik tombol **Save** jika ingin menyimpan request di atas. Simpan dengan nama **Get All Users** dalam collection **Praktikum 12**.



4. Cobalah untuk mengakses service yang kedua yaitu mendapatkan satu data user berdasarkan id user. Caranya sama dengan tahapan sebelumnya. Simpan request dengan nama **Get User ByID**. ID user merupakan angka yang terdapat pada akhir

request url. Coba modifikasi angka pada request url lalu kirimkan ulang requestnya. Apakah terdapat perbedaan pada response yang Anda terima?

5. Selain menggunakan Postman, kita juga dapat mengakses API melalui PHP menggunakan fungsi `file_get_contents()` atau library `cURL`. Kita akan mencoba fungsi `file_get_contents()` terlebih dahulu. Anda dapat mengakses API <https://reqres.in/api/users?page=1> dengan cara berikut ini.

```
$data = file_get_contents("https://reqres.in/api/users?page=1");  
var_dump($data);
```

Simpan potongan program di atas dengan nama **php11A.php**, lalu jalankan pada browser Anda. Jika berhasil, anda dapat melihat output yang ditampilkan berupa json string.

6. Agar json string tersebut dapat diproses lebih lanjut, maka kita harus menjalankan fungsi decode seperti berikut ini.

```
$parse_data = json_decode($data, true);  
echo "<br/><br/>";  
var_dump($parse_data);
```

7. Paramater `true` pada fungsi `json_decode` berfungsi untuk mengubah json string menjadi array. Jika anda menginginkan hasil decode berupa object, maka parameter `true` dapat anda hapus. Cobalah untuk menghapus parameter tersebut. Lalu jalankan kembali **php11A.php** pada browser untuk melihat perbedaan output yang dihasilkan.

8. Selanjutnya coba tampilkan data yang didapatkan dalam bentuk tabel.

9. Jika menggunakan library `cURL`, Anda dapat mengakses API yang sama dengan cara berikut ini.

```
$ch = curl_init();  
curl_setopt($ch, CURLOPT_URL, " https://reqres.in/api/users?page=1");  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);  
$output = curl_exec($ch);  
curl_close($ch);  
echo $output;  
  
$data = json_decode($output, true);  
echo "<br/><br/>";  
var_dump($data);
```

10. Anda dapat mempelajari lebih lanjut mengenai fungsi `file_get_contents()` dan library `cURL` dokumentasi di bawah ini.

<https://www.php.net/manual/en/function.file-get-contents.php>  
<https://www.php.net/manual/en/book.curl.php>

Cobalah untuk memodifikasi restClient.php sehingga dapat mengakses open API lainnya.

#### 11.4.3. Membuat Aplikasi AJAX untuk Pencarian

Pada latihan kedua ini, kita akan membuat aplikasi *suggestion* pada fitur pencarian. Kita akan menggunakan table **peminjam** yang telah dibuat pada praktikum sebelumnya. Kita akan membuat fitur pencarian dimana pengguna dapat mencari data peminjam berdasarkan nama, lalu aplikasi akan menampilkan daftar *suggestion* atau saran pencarian.

1. Buka kembali file page09B.php dari website studi kasus peminjaman komik yang telah Anda bangun, dimana halaman tersebut menampilkan seluruh data yang terdapat pada tabel peminjam.
2. Pada bagian atas tabel, tambahkan sebuah textfield yang akan digunakan untuk fitur pencarian dan sebuah elemen yang akan digunakan untuk menampilkan saran pencarian.

```
<form action="">
Cari Nama: <input type="text" id="txt1"
onkeyup="showHint(this.value)">
</form>

<p>Suggestions: <span id="txtHint"></span></p>
```

3. Pada bagian akhir program pada page09B.php, sebelum tag penutup </body>, tambahkan potongan program berikut.

```
<script src="page11A_suggestion.js"></script>
```

4. Buat sebuah file javascript dengan nama **page11A\_suggestion.js**, lalu tambahkan potongan program berikut.

```
function showHint(str) {
    if (str.length == 0) {
        //Code 4a
    }

    xhttp = new XMLHttpRequest();

    //Code 4b

    xhttp.open("GET", "page11B_gethint.php?keyword="+str, true);
    xhttp.send();
}
```

Saat pengguna mengetik karakter di kolom input txt1, fungsi "showHint()" dijalankan. Fungsi ini dipicu oleh event "onkeyup".

- a. Sebelum kita menampilkan saran pencarian, pertama-tama kita periksa terlebih dahulu apakah ada karakter yang diketikkan oleh user di kolom input txt1. Pada bagian //Code 4a, tambahkan script yang memeriksa isi textfield pencarian. Jika text field kosong (`str.length==0`), fungsi akan menghapus konten pada elemen txtHint dan keluar dari fungsi.
- b. Jika kolom input tidak kosong, maka program akan menjalankan hal-hal berikut:
  - 1) Membuat objek XMLHttpRequest.
  - 2) Membuat fungsi yang akan dieksekusi ketika respons server sudah siap yaitu menampilkan saran pencarian yang didapat dari server pada elemen txtHint.
  - 3) Mengirimkan permintaan ke file di server. Perhatikan bahwa parameter keyword ditambahkan ke URL yang berisi kata kunci pencarian pada txt1.

Perintah 1) dan 3) sudah ada pada fungsi showHint(), namun perintah 2) belum ada. Tambahkan perintah 2) tersebut pada bagian //Code 4b. Pastikan Anda memeriksa `readyState` dan `Status` dari respon server sebelum menjalankan fungsi.

5. Berikutnya, buat sebuah file php dengan nama **page11B\_gethint.php**. Tambahkan potongan script berikut lalu pahami cara kerjanya.

```
<?php
include 'dbconn.php';

try {

    //Code 6
    $keyword = $_GET["keyword"];

    // lookup all hints if query result is not empty
    $hint = "";
    if ($stmt) {
        foreach($stmt as $row) {
            if ($hint === "") {
                $hint = $row["name"];
            } else {
                $hint .= ", " . $row["name"];
            }
        }
    }
}
```



```

    }
}

// Output "no suggestion" if no hint was found or output correct values
echo $hint === "" ? "no suggestion" : $hint;

    $pdo = NULL;
}
catch (PDOException $e) {
    exit("PDO Error: ".$e->getMessage()."<br>");
}
?>

```

6. untuk mendapatkan saran pencarian, kita perlu mengambil nama-nama di tabel meetings yang sesuai dengan kata kunci yang diinput oleh user pada textfield pencarian. Tambahkan query select pada bagian //Code 6 sehingga didapatkan data dengan kolom nama yang memuat kata kunci pencarian yang diinput oleh user.
7. Sekarang, kita akan coba jalankan aplikasi pencarian kita. Buka halaman page09B.php dengan browser. Coba masukkan potongan nama yang terdapat pada tabel meetings Anda. Pastikan saran pencarian yang ditampilkan sudah sesuai dengan kata kunci yang diinput pada textfield pencarian.
8. Karena umumnya pertukaran data pada AJAX dilakukan dalam format JSON, selanjutnya kita akan modifikasi beberapa baris program di atas agar client mendapatkan data dalam bentuk JSON dari server saat melakukan request AJAX.

Kita akan melakukan sedikit modifikasi pada php11B\_gethint.php agar data yang dikembalikan ke client berupa data JSON. Ubah script bagian berikut:

```

// lookup all hints if query result is not empty
$hint = "";
if ($stmt) {
    foreach($stmt as $row) {
        if ($hint === "") {
            $hint = $row["name"];
        } else {
            $hint .= "," . $row["name"];
        }
    }
}

// Output "no suggestion" if no hint was found or output correct values
echo $hint === "" ? "no suggestion" : $hint;

```

Menjadi

```
// lookup all hints if query result is not empty
if ($stmt) {
    echo json_encode($stmt);
}
else{// Output "no suggestion" if no hint was found or output
correct values
    $response[] = array(
        'name'=>'no suggestion'
    );

    echo json_encode($response);
}
```

Fungsi `json_encode()` digunakan untuk mengkonversi hasil query dari database menjadi data dengan format JSON.

9. Jalankan ulang `page09B.php`. Seharusnya saran pencarian muncul pada elemen `txtHint` namun dalam format json. Jika saran pencarian Anda tidak muncul sama sekali pada elemen `txtHint`, coba `var_dump` hasil query Anda pada `php11B_gethint.php`. Jika hasil query masih berupa query string, Anda harus mengubahnya terlebih dahulu ke dalam format Array menggunakan fungsi `PDO fetch_all()`.

```
$pdo->query($sql)->fetchAll();
```

10. Selanjutnya, tugas Anda memodifikasi script yang Anda tulis pada `php11A_suggestion.js` agar saran pencarian yang ditampilkan pada elemen `txtHint` dalam bentuk string biasa. Tepatnya modifikasi pada bagian `//Code 4b` di dalam fungsi yang akan dieksekusi ketika respons server sudah siap. Anda harus mengkonversi `responseText` dari server menjadi format data yang dapat diproses menggunakan JavaScript. Konversi dilakukan menggunakan fungsi `JSON.parse()`. Sebagai referensi, buka kembali materi pada sesi teori atau referensi lain pada buku teks maupun internet.

### 11.5. Responsi

File-file yang harus Anda kumpulkan dari praktikum ini mencakup file dari praktikum 9 dan 10 lalu:

- dbconn.php
- page09A.php, page09B.php, page09C.php, page09C\_action.php, page09D.php, page09D\_action.php, page09E.php, page09E\_action.php, page09F.php, page09G.php, page09G\_action.php, page09H.php,.
- page10A.php, page10A\_action.php, page10B.php

ditambah dengan file-file dari praktikum ini:

- page11A\_suggestion.js
- page11B\_gethint.php

sertakan pula file-file pendukung untuk membuka website anda dengan sempurna, diantaranya:

- dump SQL dari database Anda
- css
- javascript
- folder image beserta isi filenya
- dan lain-lain

Gabungkan keseluruhan file tersebut dalam satu file zip dengan format nama <<nim>>\_modul11, contoh: 192191234\_modul11 lalu kumpulkan melalui Google Classroom.

Silakan kumpulkan sesuai batas waktu yang telah ditetapkan. Setelah batas waktu tersebut, Anda sudah TIDAK DAPAT mengumpulkan ulang atau mengumpulkan susulan file yang tertinggal dengan alasan apapun.

Seluruh kecurangan dalam bentuk apapun diberikan sanksi berupa nilai 0 untuk praktikum modul 9-11.