**TUGAS-08**

**WEB PROGRAMMING 2**

**RESUME JSON**

****

**NAMA DOSEN : M. MUZAQI, S.Kom**

**Disusun Oleh :**

**NAMA : AINUR ROFIKA (17090047)**

**MURNI KUNIASARI (17090051)**

**KUKUH AJI SANTOSO (17090059)**

**KELAS : 5 A**

**D4 TEKNIK INFORMATIKA**

**POLITEKNIK HARAPAN BERSAMA TEGAL**

**2019/2020**

**JSON** — singkatan untuk **JavaScript Object Notation** — adalah sebuah format untuk berbagi data. Seperti dapat kita lihat dari namanya, JSON diturunkan dari bahasa pemrograman javaScript, akan tetapi format ini tersedia bagi banyak bahasa lain termasuk Python, Ruby, PHP, dan Java. JSON biasanya dilafalkan seperti nama "Jason."

JSON menggunakan ekstensi .json saat ia berdiri sendiri. Saat didefinisikan di dalam format file lain (seperti di dalam .html), ia dapat tampil didalam tanda petik sebagai JSON string, atau ia dapat dimasukkan kedalam sebuah variabel. Format ini sangat mudah untuk ditransfer antar server web dengan klien atau browser.

Karena sangat mudah dibaca dan ringan, JSON memberikan alternatif lebih baik dari XML adn membutuhkan formatting yang tidak banyak. Panduan ini akan membantu pembaca untuk memahami apa itu JSON, bagaimana menggunakan data di file JSON, serta struktur dan sintaks dari format ini.

Untuk lebih memahami struktur data JSON, alangkah baiknya anda menginstall JSON viewer seperti plugin JSON viewer di Notepad++. Anda bisa menginstallnya lewat plugin manager. Berikut ini adalah struktur data JSON yang ditampilkan dengan JSON Viewer.

JSON terbuat dari dua struktur utama, yaitu:

* Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau [*associative array*](http://en.wikipedia.org/wiki/Associative_array).
* Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini.

## Penggunaan JSON Sebagai Objek pada JavaScript

Format yang digunakan **JSON** secara sintaks indentik aatau memiliki kesamaan dengan code yang digunakan JavaScript untuk membuat sebuah objek.

Karena kesamaan inilah dibandingkan dengan menggunakan parser (seperti yang dilakukan XML) sebuah program JavaScript dapat menggunakan function JavaScipt Standard untuk mengubah data pada **JSON** menjadi sebuah objek asli JavaScript. Hal ini merupakan salah satu faktor yang menjadikan **JSON** sebagai alternatif yang lebih baik dibandingkan XML sebagai bahasa format pernyimpanan dan pertukaran data.

## Keunggulan JSON

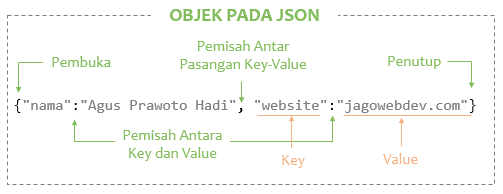
**JSON** memiliki beberapa keunggulan dibandingkan dengan XML terutama dapat penggunaan apikasi AJAX. Beberapa kelebihan **JSON** antara lain :

* **JSON** lebih ringkas, cepat, dan mudah.
* **JSON** tidak menggunakan tag penutup
* **JSON** lebih cepat untuk dibaca dan ditulis
* **JSON** dapat menggunakan array
* **JSON** tidak perlu menggunakan parser khusus untuk diubah menjadi sebuah objek, melainkan hanya perlu menggunakan function default dari JavaScript.

### Tipe Data JSON

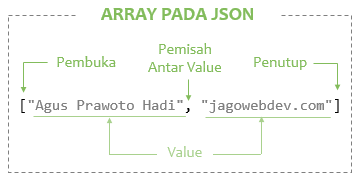
Maksud tipe data disini adalah jenis data yang dapat digunakan di dalam JSON, adapun tipe data yang dapat digunakan adalah sebagai berikut:

1. **Objek**. Objek merupakan kumpulan pasangan key dan value yang diapit dengan kurung buka dan kurung tutup (seperti penjelasan sebelumnya). Karena objek ini mencerminkan associative array, maka direkomendasikan agar tidak menggunakan key dengan nilai sama (key sebaiknya unik). Objek ini boleh bernilai kosong dan cukup ditulis {}

[](https://jagowebdev.com/wp-content/uploads/2018/01/Memahami-JSON-Objek-Pada-JSON.png)

Objek Pada JSON

1. **Array**.  Sederetan value yang diapit oleh kurung siku (seperti ilustrasi dibawah). Antara value yang satu dengan yang lain dipisahkan tanda koma. array boleh bernilai kosong dan cukup ditulis []

[](https://jagowebdev.com/wp-content/uploads/2018/01/Array-Pada-JSON.png)

Array Pada JSON

1. **Number**. Number berupa digit 0-9 dan ditulis apa adanya tanpa perlu menggunakan tanda kutip. Number dapat berupa bilangan bulat maupun desimal. Untuk desimal, karakter yang digunakan adalah titik bukan koma. Misal untuk menuliskan dua koma lima, bentuk angkanya adalah 2.5 bukan 2,5
2. **Boelan**. Boelan berupa teks true dan false tanpa tanda kutip.
3. **Null**. Null berupa teks null dan ditulis langsung tanpa tanda kutip.
4. **String**. String berupa semua karakter Unicode yang artinya semua karakter yang ada di dunia ini. String ini merupakan tipe data yang **paling sering digunakan**, dalam penulisannya, string **harus** diapit dengan dua tanda kutip ( ” ) tidak boleh menggunakan satu tanda kutip ( ‘ ) atau tidak menggunakan tanda kutip sama sekali.

**Membuat JSON di sisi server**

Membuat Table dengan nama message di mysql

CREATE TABLE `message` ( `message\_id` INT(11) NOT NULL AUTO\_INCREMENT, `chat\_id` INT(11) NOT NULL DEFAULT '0', `user\_id` INT(11) NOT NULL DEFAULT '0', `user\_name` VARCHAR(64) DEFAULT NULL, `message` TEXT, `post\_time` DATETIME DEFAULT NULL, PRIMARY KEY (`message\_id`) )

Membuat JSON dengan PHP secara manual (phpjson.php)

<?php

$con = new PDO("mysql:dbname=simple\_chat;host=localhost", "root", "");

$json = '{"message": {';

$query = $con->query("select \* from message");

$json .= '"pesan": [ ';

foreach ($query->fetchAll(PDO::FETCH\_ASSOC) as $x) {

$json .= '{';

$json .= '"id": "' . $x['message\_id'] . '",

"user": "' . htmlspecialchars($x['user\_name']) . '",

"text": "' . htmlspecialchars($x['message']) . '",

"time": "' . $x['post\_time'] . '"

},';

}

// menghilangkan koma di akhir

$json = substr($json, 0, strlen($json) - 1);

// var\_dump($json);

// lengkapi penutup format json

$json .= ']';

$json .= '}}';

echo $json;

Mem-parse JSON di sisi client

Sekarang kita lihat bagaimana cara membuat format JSON tersebut menggunakan javascript. Untuk mengetahuinya /mem-parse JSON adalah dengan menggunakan fungsi eval("(" + jsonteks + ")"). Selain menggunakan eval(),bias mencoba menggunakan fungsi-fungsi yang telah dibuat orang lain, seperti yang ada di www.json.org.

Menyarankan dalam menghindari penggunaan eval(), karena kurang aman, sebaiknya gunakan fungsi-fungsi yang dibuat oleh orang lain (seperti : http://www.JSON.org/json\_parse.js).

Coba buat file json seperti di bawah ini:

<html>

<head>

<script>

function tes() {

json = '{"message": {"pesan": [ {"id": "1", "user": "Nurmi", "text": "Halo apa kabar", "time": "2019-11-13 13:56:05" },{"id": "2", "user": "desrizal", "text": "kabar baik", "time": "2019-11-13 00:00:00" },{"id": "3", "user": "nurmi", "text": "lagi dimana", "time": "2019-11-13 00:00:00" },{"id": "4", "user": "desrizal", "text": "belitung", "time": "2019-11-13 00:00:00" }]}}'

var teks\_div = document.getElementById('divteks');

var response = eval("(" +json+")");

for(i =0; i< response.message.pesan.length; i++){

teks\_div.innerHTML += response.message.pesan[i].user;

teks\_div.innerHTML += '&nbsp;&nbsp;<font size=1>' +

response.message.pesan[i].time + '</font><br />';

teks\_div.innerHTML += response.message.pesan[i].text + '<br />';

}

}

</script>

</head>

<body onload=tes()>

<div id="divteks"></div>

</body>

</html>

JSON Object

Kita telah mempelajari cara mem-parse JSON menggunakan eval(). Kelemahan menggunakan eval() adalah kurang aman. Untuk itu, maka pada FF3.5+, IE8+, dan Opera 10.5+ diperkenalkan javascript built in JSON Object.

JSON Object menyediakan metode untuk mengubah objek JSON menjadi string dan JSON string menjadi objek javascript. Berikut adalah metode-metode dari JSON Object. stringify(obj, [replacer], [space])

Berguna untuk mengubah javascript object menjadi JSON String.

Contoh :

var jsonobjek={"nama":"Desrizal", "umur":28, "teman":["Nurmi", "Sarah", "Albert"]} var jsonstr=JSON.stringify(jsonobjek) alert(typeof jsonstr) //string parse(string, [reviver])

Berguna untuk mengubah JSON String menjadi Javascript Object

Contoh :

var jsonstr='{"nama":"Desrizal", "umur":28, "teman":["Nurmi", "Sarah", "Albert"]}' var george=JSON.parse(jsonstr) //convert JSON string into object alert(george.age) //alerts 28

JSON Ajax Web Chat

Oke sekarag kita lihat contoh aplikasi ajax yang menggunakan JSON sebagai format ouput yang di berikan oleh server ke client.

Pertama-tama mari kita buat tabel di database, sengan struktur tabel sebagai berikut :

CREATE TABLE IF NOT EXISTS `drzchat` (

`nomor` int(3) NOT NULL AUTO\_INCREMENT,

`nama` varchar(20) NOT NULL,

`pesan` varchar(200) NOT NULL,

`waktu` varchar(10) NOT NULL,

PRIMARY KEY (`nomor`)

) ENGINE=InnoDB AUTO\_INCREMENT=135 DEFAULT CHARSET=latin1;

Kemudian buat tampilan layout HTML nya dan program Ajax di sisi client.

Membuat file html dengan nama drzchat.html

<html>

<head>

<title>Chat</title>

<script>

var ajaxku = buatAjax();

var tnama = 0;

var pesanakhir = 0;

var timer;

function taruhNama() {

if (tnama == 0) {

document.getElementById("@$nama").disabled = "true";

tnama = 1;

} else {

document.getElementById("@$nama").disabled = "";

tnama = 0;

}

ambilPesan();

}

function buatAjax() {

if (window.XMLHttpRequest) {

return new XMLHttpRequest();

} else if (window.ActiveXObject) {

return new ActiveXObject("Microsoft.XMLHTTP");

}

}

function ambilPesan() {

namaku = document.getElementById("@$nama").value

if (ajaxku.readyState == 4 || ajaxku.readyState == 0) {

ajaxku.open("GET", "ambilchat.php?akhir=" + pesanakhir + "&nama=" + namaku + "&sid=" + Math.random(), true);

ajaxku.onreadystatechange = aturAmbilPesan;

ajaxku.send(null);

}

}

function aturAmbilPesan() {

if (ajaxku.readyState == 4) {

var chat\_div = document.getElementById("div\_chat");

var data = eval("(" + ajaxku.responseText + ")");

for (i = 0; i < data.messages.pesan.length; i++) {

chat\_div.innerHTML += "<font color=red>" + data.messages.pesan[i].nama + "</font>";

chat\_div.innerHTML += "<font size=1>"+data.messages.pesan[i].waktu + "</font>";

chat\_div.innerHTML += " : "+data.messages.pesan[i].teks + "<br>";

chat\_div.scrollTop = chat\_div.scrollHeight;

pesanakhir = data.messages.pesan[i].id;

}

}

timer = setTimeout("ambilPesan()",1000);

}

function kirimPesan() {

pesannya = document.getElementById('pesan').value;

namaku = document.getElementById('nama').value;

if (pesannya != "" && document.getElementById('nama').value != "") {

ajaxku.open('GET', `ambilchat.php?akhir=${pesanakhir}&nama=${namaku}&pesan=${pesannya}&sid=${Math.random()}`, true);

ajaxku.onreadystatechange = aturAmbilPesan;

ajaxku.send(null);

document.getElementById('pesan').value = "";

} else {

alert("Nama atau pesan masih kosong");

}

}

function aturKirimPesan() {

clearInterval(timer);

ambilPesan();

}

function blockSubmit() {

kirimPesan();

return false;

}

</script>

</head>

<body>

Nama : <input type="text" name="name" id="nama">

<input type="button" value="login" id="tmbl\_login" onclick="taruhNama()">

<p>

<div id="div\_chat" style="height: 300px; width: 500px; overflow: auto; background-color: yellow; border: 1px solid #555555;">

</div><br>

<form onSubmit="return blockSubmit();">

pesan : <input type="text" name="pesan" id="pesan" size="50">

<input type="button" value="kirim" onclick="kirimPesan()">

</form>

</p>

</body>

</html>

Pada program chat sisi client di atas, pertama-tama user harus memasukkan nama mereka di suatu teks input. Setelah kita menekan tombol login, maka segera dijalankan fungsi ambilPesan() ambilPesan() ambilPesan() ambilPesan(). Di mana fungsi ini berfungsi untuk melakukan request ke server untuk mengambil data dari server, ambilchat.php.

Data yang diambil dari server berupa teks dengan format JSON sebagai berikut :

Kemudian buat file php dengan nama ambilchat.php

<?php

date\_default\_timezone\_set("Asia/Jakarta");

$con = new PDO("mysql:dbname=simple\_chat;host=localhost", "root", "");

if (isset($\_GET['nama'])) {

@$nama = $\_GET['nama'];

}

if (isset($\_GET['pesan'])) {

$pesan = $\_GET['pesan'];

}

$waktu = date("H:m");

if (isset($\_GET['akhir'])) {

@$akhir = $\_GET['akhir'];

}

$json = '{"messages": {';

if (@$akhir == 0) {

$nomor = $con->query("select nomor from drzchat order by nomor desc limit 1");

$nomor->execute();

$n = $nomor->fetch(PDO::FETCH\_ASSOC);

$no = $n['nomor'] + 1;

$json .= '"pesan":[ {';

$json .= '"id":"' . $no . '",

"nama": "' . @$nama . '",

"teks": "Selamat datang di chatting room",

"waktu": "' . $waktu . '"

}]';

$masuk = $con->query("insert into drzchat values(null, 'nama', 'nama bergabung dalam chat', '$waktu')");

} else {

if (isset($pesan)) {

$masuk = $con->query("insert into drzchat values(null, '$nama','$pesan','$waktu')");

}

$query = $con->query("select \* from drzchat where nomor > $akhir");

$json .= '"pesan":[ ';

foreach ($query->fetchAll(PDO::FETCH\_ASSOC) as $x) {

$json .= '{';

$json .= '"id": "' . $x['nomor'] . '",

"nama": "' . htmlspecialchars($x['nama']) . '",

"teks": "' . htmlspecialchars($x['pesan']) . '",

"waktu": "' . $x['waktu'] . '"

},';

}

$json = substr($json, 0, strlen($json) - 1);

$json .= ']';

}

$json .= '}}';

echo $json;