

IV. JSON dan XSS

Setiap string yang berhubungan dengan javascript dan HTML maka akan tidak lepas dari pembahasan tentang XSS, demikian juga JSON. XSS ini terjadi salah satunya ketika script HTML terdapat element `<script>` yang berasal dari **sumber tidak terpercaya** (seperti form input dimana pengunjung bebas memasukkan data).

Element script tersebut tersebut mengandung kode javascript yang akan mengekspose data pribadi user kemudian mengirim data tersebut ke pembuat script.

Terkait dengan JSON, pembuatan JSON dengan fungsi `json_encode()` **sudah aman**, karena (seperti telah disampaikan sebelumnya) fungsi tersebut sudah meng-escape slash (`<\script>`) – mulai PHP versi 5.4.0 -, sehingga membuat kode javascript di dalam JSON tidak berjalan.

Terlebih lagi, perlindungan terhadap XSS sebenarnya tidak dari sisi JSON nya melainkan pada sisi inputan user, dimana ketika ada inputan dari luar yang akan langsung dieksekusi atau disimpan ke database, data inputan tersebut harus di sanitasi terlebih dahulu, misal menggunakan fungsi `htmlspecialchars()` sehingga misal ketika dibuat menjadi JSON atau format lain, data tersebut sudah aman.

Apakah perlu HEX?

Pembahasan yang sering muncul adalah, jika json kita mengandung tag HTML, atau tag `<script>`, perlukah kita menggunakan opsi `JSON_HEX_XXX` untuk mengubah elemen HTML menjadi hex? seperti berikut ini:

```
$array = array('html' => '</script><script>document.cookie</script>');  
$json = json_encode($array, JSON_HEX_TAG | JSON_HEX_QUOT | JSON_HEX_APOS |  
JSON_HEX_AMP);
```

Sekali lagi bahwa fungsi `json_encode` ini sudah aman sehingga penggunaan argumen `JSON_HEX_XXX` tidak diperlukan lagi terlebih jika sumber data sudah terpercaya, misal dari server kita sendiri dan kita sendiri yang membuat data tersebut.

V. Penutup

PHP menyediakan tools bawaan yang komplet untuk bekerja dengan JSON, hal yang perlu diperhatikan adalah versi PHP yang kita gunakan, karena pada PHP versi sebelum 5.6, terdapat beberapa bug yang merepotkan.

Demikian pembahasan mengenai JSON Pada PHP semoga bermanfaat.

Pengertian JSON <https://cahbiyen.blogspot.com/2015/09/pengertian-json.html>

{JSON}

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. **Sumber**

Data dari JSON bisa berupa single data yang terdiri dari key dan value atau bisa disebut juga dengan **JSON Object** dan bisa juga kumpulan dari key dan value atau lebih sering di sebut dengan **JSON Array** atau sekumpulan dari JSON Object.

JSON Object merupakan pasangan antara **key** dan **value** yang berada dalam satu kurung kurawa {} format penulisanya yaitu {**key:value**} atau bisa juga {**key1:value1, key2:value2**} lihat contoh di bawah ini

```
123456// JSON Object
{'id':1}

// atau bisa juga
{'id':1, 'nama':'Mochamad Rohmat'}
```

JSON Array merupakan tempat penampungan dari **JSON Object**, jadi bisa terdiri dari beberapa **JSON Object** format penulisanya yaitu dengan menggunakan kurung siku [], seperti [{**key:value**}] atau bisa juga [{**key1:value1, key2:value2**},{**key1:value1, key2:value2**}] lihat contoh di bawah ini

```
123456// JSON Array
[{'id':1}]

// atau bisa juga
[{'id':1, 'nama':'Mochamad Rohmat'},{'id':2, 'nama':'Heni Winangsih'}]
```

Ok, sampai sini dlu tutorial kali ini, selamat mencoba dan semoga bisa bermanfaat

Cara Menggunakan JSON Di PHP

<https://cahbiyen.blogspot.com/2015/09/cara-menggunakan-json-di-php.html>

kamariin saya sudah membahas cara menggunakan JSON pada javascript sekarang saya akan bahas cara menggunakan JSON pada PHP, JSON selain bisa di baca dalam javascript, JSON juga bisa di baca dengan PHP, hanya kita harus merubah format JSON tersebut ke dalam bentuk **Array** atau **Array Object**, untuk dapat menggunakan format JSON di PHP kita tidak perlu menambahkan modul ataupun plugin karena PHP sudah mendukung format JSON dengan menggunakan fungsi dari **json_encode** dan **json_decode**.

Fungsi dari **json_encode** sendiri yaitu merubah bentuk Array menjadi JSON sedangkan kalau **json_decode** adalah kebalikan dari **json_encode** yaitu merubah format JSON ke bentuk Array, coba lihat contoh di bawah ini

```
12345678910111213141516171819202122232425<!DOCTYPE html>
<html>
<head>
<title>BELAJAR JSON</title>
</head>
<body>
<?php
    // bentuk Array
    $array = array('id'=>1, 'nama'=>'Mochamad Rohmat');
    print_r($array);
    // makan akan menghasilkan : Array ( [id] => 1 [nama] => Mochamad Rohmat )

    // bentuk JSON dari variable $array
    $json = json_encode($array);
    echo $json;
    // maka akan menghasilkan : {"id":1,"nama":"Mochamad Rohmat"}

    // jika format JSON mau di rubah lagi ke format Array maka menggunakan fungsi
    dari json_decode
    $array = json_decode($json);
    print_r($array);
    // maka akan menghasilkan : stdClass Object ( [id] => 1 [nama] => Mochamad
    Rohmat )
    ?>
</body>
</html>
```

contoh json diatas yaitu contoh dari JSON Object karena hanya memiliki 1 data, lalu bagaimana cara membaca format JSON di PHP, perhatikan kode dibawah

```

1234567891011121314151617181920212223<!DOCTYPE html>
<html>
<head>
<title>BELAJAR JSON</title>
</head>
<body>
<?php
    // ketika dalam bentuk array
    $array = array('id'=>1, 'nama'=>'Mochamad Rohmat');
    echo $array['nama'];
    // maka akan menghasilkan : Mochamad Rohmat karena kita mencetak array dari key
    nama

    // convert Array ke format JSON
    $json = json_encode($array);

    // jika anda ingin membaca format JSON di PHP maka JSON harus di convert ke
    Array Object dengan menggunakan json_decode
    $array = json_decode($json);
    echo $array->nama;
    // maka akan menghasilkan : Mochamad Rohmat
?>
</body>
</html>

```

kalau tadi kita membaca format JSON Object agar bisa dibaca di PHP, bagaimana dengan JSON Array, perhatikan kode dibawah, disini saya menggunakan data dari database untuk membuat JSON Array.

```

123456789101112131415161718192021222324252627282930313233343536373839404142434445
46474849505152535455565758596061626364<!DOCTYPE html>
<html>
<head>
<title>BELAJAR JSON</title>
</head>
<body>
<?php
    // membuat koneksi ke database
    $host = 'localhost';
    $user = 'root';
    $pass = '';
    $db = 'belajar_json_php';

    $conn = mysql_connect($host, $user, $pass);
    mysql_select_db($db, $conn);

    // menampilkan data dari database
    $sql = 'SELECT * FROM users';
    $rs = mysql_query($sql);

    $i = 0;
    while($row = mysql_fetch_array($rs)){

```

```

    $datas[$i]['id'] = $row['id'];
    $datas[$i]['nama'] = $row['nama'];
    $datas[$i]['alamat'] = $row['alamat'];
    $i++; }

// jika anda cetak dengan print_r($datas) makan akan menghasilkan array seperti
ini
echo 'Ini hasil dari array : ';
print_r($datas);
echo '<hr />';
// Array ( [0] => Array ( [id] => 1 [nama] => Rohmat [alamat] => Jln. Pantura
ciasem tengah ) [1] => Array ( [id] => 2 [nama] => Mimin [alamat] => Jlan. Krajan
timur ) [2] => Array ( [id] => 3 [nama] => Maman [alamat] => Jlan. Krajan barat )
[3] => Array ( [id] => 4 [nama] => Roni [alamat] => Jlan. Mekar sari indah ) )

// jika datas di rubah menjadi format JSON dengan menggunakan json_encode makan
akah menghasilkan JSON Array seperti ini
$json = json_encode($datas);
echo 'Ini hasil dari JSON Array : ';
echo $json;
echo '<hr />';
// [{"id":"1","nama":"Rohmat","alamat":"Jln. Pantura ciasem
tengah"}, {"id":"2","nama":"Mimin","alamat":"Jlan. Krajan
timur"}, {"id":"3","nama":"Maman","alamat":"Jlan. Krajan
barat"}, {"id":"4","nama":"Roni","alamat":"Jlan. Mekar sari indah "}]]

// seperti yang sudah saya contohkan di atas untuk membaca format JSON di PHP
anda harus merubahnya kembali ke Array dengan menggunakan json_decode
echo 'Ini hasil dari Array Object : ';
$json = json_decode($json);
print_r($json);
echo '<hr />';

// meloop data Array Object
foreach($json as $data){
    echo 'ID : '.$data->id;
    echo 'NAMA : '.$data->nama;
    echo 'ALAMAT : '.$data->alamat;
    echo '<hr />'; }
// jika anda buka di browser maka akan menghasilkan
// ID : 1
// NAMA : Rohmat
// ALAMAT : Jln. Pantura ciasem tengah
// ..... dan seterusnya
?>
</body>
</html>

```

OK sampai disini dulu pembahasan mengenai JSON di PHP, selamat mencoba dan semoga bisa bermanfaat

Cara Menggunakan JSON Di Javascript

<https://cahbiyen.blogspot.com/2015/09/menggunakan-json-di-javascript.html>

Dalam artikel sebelumnya yang berjudul **Pengertian JSON** sudah saya jelaskan perbedaan JSON Object dan JSON Array beserta cara menggunakan format JSON, dalam tutorial kali ini saya akan menjelaskan cara menggunakan JSON di Javascript, perhatikan kode di bawah, kode JSON di bawah jika anda cetak akan menghasilkan **value** dari **key** nama

```
123456789101112131415<!DOCTYPE html>
<html>
<head>
<title>BELAJAR JSON</title>
</head>
<body>
<script type="text/javascript">
  var json = {'id':1, 'nama':'Mochamad Rohmat'};
  document.write(json.nama);

  // hasilnya adalah Mochamad Rohmat
</script>
</body>
</html>
```

jika script di atas di jalankan maka akan menghasilkan **Mochamad Rohmat** Sekarang bedakan dengan contoh berikut ini, hasil 1 akan mencetak undefined karena formatnya adalah JSON Array bukan JSON Object sedangkan hasil 2 akan mencetak **Mochamad Rohmat** karena posisi JSON Object ada di Array 0.

```
1234567891011121314151617<!DOCTYPE html>
<html>
<head>
<title>BELAJAR JSON</title>
</head>
<body>
<script type="text/javascript">
  var json = [{ 'id':1, 'nama':'Mochamad Rohmat' }];
  document.write('Hasil 1 : ' + json.nama);
  document.write('Hasil 2 : ' + json[0].nama);

  // hasil 1 akan mencetak undefined
  // hasil 2 akan mencetak Mochamad Rohmat
</script>
</body>
</html>
```

Sekarang bagaimana jika hasil dari JSON di loop dengan for, silahkan lihat kode berikut

```
123456789101112131415161718192021<!DOCTYPE html>
<html>
<head>
<title>BELAJAR JSON</title>
</head>
<body>
<script type="text/javascript">
  var json = [{ 'nama': 'Mochamad Rohmat' }, { 'nama': 'Heny Winangsih' }, { 'nama': 'Maman
Drajat' } ];

  for (var i = 0; i < Object.keys(json).length; i++) {
    document.write((i + 1) + '. ' + json[i].nama);
  }

  // akan menghasilkan
  // 1. Mochamad Rohmat
  // 2. Heny Winangsih
  // 3. Maman Drajat
</script>
</body>
</html>
```

OK. sampai disini dulu tutorial kali ini, selamat mencoba dan semoga bisa bermanfaat.

Memproses Data JSON Pada Android

<http://bayu.freelancer.web.id/2012/02/24/memproses-data-json-pada-android/>

JSON sepertinya sudah menjadi format data standar dalam komunikasi data antara API di internet. Semakin banyak website yang menyediakan API, menggunakan JSON sebagai format data yang dikirimkan. Untuk itu, rasanya cocok tepat kalau sekarang kita belajar cara mengolah data JSON di Android.

Sebelum kita memulai ada baiknya anda mengerti terlebih dahulu, apa itu JSON.

Perhatikan contoh JSON berikut:

```
{  "pelatih": {
    "nama": "adi saka",
    "kota": "Langit",
    "murid": [
      { "nama": "sila gading",
        "umur": 6
      },
      { "nama": "putri tunggal",
        "umur": 8
      },
      { "nama": "anka satu",
        "umur": 7
      }
    ]
  }
}
```

Dari kode di atas, bisa dilihat bahwa ada `pelatih` yang memiliki properti `nama`, `kota`, dan `murid`. Data `pelatih` ini berupa object. Dari mana kita bisa tahu? hal ini diindikasikan pada penggunaan karakter `{}`. Lalu untuk `murid`, berisikan array object, yang ditunjukkan pada penggunaan karakter `[]`. Untuk mengetahui lebih lanjut, silahkan pelajari diagram dan notasi-notasi yang ada pada website resmi [JSON](#). Sekarang kita masuk ke kode android. Kita akan menggunakan kode JSON di atas sebagai bahan. Yang pertama, bagaimana cara mengambil informasi `pelatih`, seperti `nama` dan `kota`?. Di android, proses manipulasi JSON ada pada paket `org.json.*`. Untuk membaca data pelatih, kita buat terlebih dahulu representasi object-nya:

Membuat representasi object dari JSON

```
JSONObject object = new JSONObject(jsonString);
```

Membuat representasi object dari elemen `pelatih`

```
JSONObject objData = object.getJSONObject("pelatih");
```

Membaca properti dari elemen `pelatih`

```
Log.i("Log", "Nama: " + objData.getString("nama"));
Log.i("Log", "Kota: " + objData.getString("kota"));
```

Cukup sederhana bukan?

Berikutnya, cara membaca elemen array JSON. Pada kasus kita, membaca elemen `murid` dari `pelatih`. Caranya juga cukup sederhana. Berikut step2nya:

Membuat representasi object dari elemen `murid`. Dalam hal ini, representasi JSONArray

```
JSONArray dataMurid = objData.getJSONArray("murid");
```

Lakukan looping data array `murid`

```
for(int i = 0; i < dataMurid.length(); i++){
    //di sini nantinya kita akan memproses elemen object masing2 murid
}
```

Membuat representasi object masing2 elemen anak dari `murid` dan membaca propertinya

```
JSONObject objMurid = dataMurid.getJSONObject(i);
Log.i("Log", "    Nama: " + objMurid.getString("nama"));
Log.i("Log", "    Umur: " + objMurid.getString("umur"));
```

Perhatikan, cara membaca elemen anak dari `murid`, mirip dengan cara membaca `pelatih`. Hal ini, karena `pelatih` dan `murid` keduanya merupakan object.

Seperti ditunjukkan pada notasi mereka yang berupa `{}`

Sangat sederhana kan? nah, ini kode lengkapnya:


```
String jsonString = "{\"pelatih\":{\"nama\":\"adi saka\",\"kota\":\"Langit\",\"murid\": [{\"nama\":\"sila gading\",\"umur\":6}, {\"nama\":\"putri tunggal\",\"umur\":8}, {\"nama\":\"anka satu\",\"umur\":7}]}}";
try {
    JSONObject object = new JSONObject(jsonString);
    JSONObject objData = object.getJSONObject("pelatih");
    Log.i("Log", "Nama: " + objData.getString("nama"));
    Log.i("Log", "Kota: " + objData.getString("kota"));
    Log.i("Log", "Murid: ");
    JSONArray dataMurid = objData.getJSONArray("murid");
    for(int i = 0; i < dataMurid.length(); i++){
        JSONObject objMurid = dataMurid.getJSONObject(i);
        Log.i("Log", "    Nama: " + objMurid.getString("nama"));
        Log.i("Log", "    Umur: " + objMurid.getString("umur"));
    }

} catch (JSONException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

Mengenal Json Beserta Kelebihan Dan Manfaatnya

[kangtanto](#) | February 13, 2016 | [Technology](#), [Web Desain](#) | [No Comments](#)

<http://kangtanto.com/technology/mengenal-json-beserta-kelebihan-dan-manfaatnya>

Apa itu JSON? apa yang bisa JSON lakukan? Pada tulisan kali ini akan membahas tentang pengenalan JSON. Selain itu juga akan membahas berbagai manfaat dan kegunaan JSON. Apabila anda pernah membuat web atau mengembangkan aplikasi berbasis web, tentunya anda pernah mendengar JSON atau setidaknya pernah membacanya secara sekilas. Mari kita mengenal JSON.



MENGENAL JSON, APA ITU JSON?

JSON merupakan open standard untuk pertukaran data yang berupa teks. Dengan menggunakan syntax yang sederhana, anda bisa menyimpan apapun, mulai dari data berupa number, string, array dan object dalam bentuk teks. Sekali anda membuat string JSON, maka cukup mudah untuk mengirimkannya ke aplikasi lain, karena ia hanya berupa teks.

JSON merupakan kepanjangan dari JavaScript Object Notation. Sesuai dengan namanya, maka ia berdasarkan bagaimana anda mendefinisikan obyek dan array.

KELEBIHAN JSON

1. Beberapa kelebihan dan manfaat dari JSON antara lain:
2. Mudah dibaca atau ditulis oleh komputer atau manusia
3. Hampir semua bahasa pemrograman menyediakan library atau function yang memudahkan untuk membaca atau membuat struktur JSON
4. JSON mudah sekali dipetakan pada struktur data yang digunakan oleh sebagian besar bahasa pemrograman terkait data berupa number, string, boolean, null, array dan associative array.

MANFAAT JSON

1. JSON pada umumnya sering digunakan pada aplikasi web untuk mengirim data dari server ke browser. Sebagai contoh kasus misalnya seperti ini:
2. Seorang user melakukan klik pada gambar thumbnail sebuah produk yang dijual disebuah toko online
3. Selanjutnya script Javascript dijalankan pada browser, melakukan Ajax request ke script PHP yang dijalankan pada server, melemparkan ID dari produk yang dipilih
4. Kemudian script PHP mengambil data nama produk, deskripsi, harga dan info-info lainnya dari database. Kemudian data dirubah dalam bentuk JSON dan dikirimkan kembali ke browser
5. Javascript yang jalan di browser kemudian membaca format JSON dan menampilkan detail informasi pada user
6. Saat proses itu terjadi, browser pada user tidak perlu reload atau berganti halaman. Semuanya terjadi di background.

Setelah kita mengenal JSON, maka tulisan selanjutnya akan kita bahas bagaimana cara membuat JSON dan memahami strukturnya.

Pengenalan JSON

object

```
{ }  
{ members }
```

members

```
pair  
pair , members
```

pair

```
string : value
```

array

```
[ ]  
[ elements ]
```

elements

```
value  
value , elements
```

value

```
string  
number  
object  
array  
true  
false  
null
```

```

string      ""
            " chars "
chars
char
char chars
char
    any-Unicode-character-
    except-"-or-\-or-
    control-character
    \"
    \\
    \/
    \b
    \f
    \n
    \r
    \t
    \u four-hex-digits
number
    int
    int frac
    int exp
    int frac exp
int
    digit
    digit1-9 digits
    - digit
    - digit1-9 digits
frac
    . digits
exp
    e digits
digits
    digit
    digit digits
e
    e
    e+
    e-
    E
    E+
    E-

```

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat

(generate) oleh komputer. Format ini dibuat berdasarkan bagian dari **Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999**. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

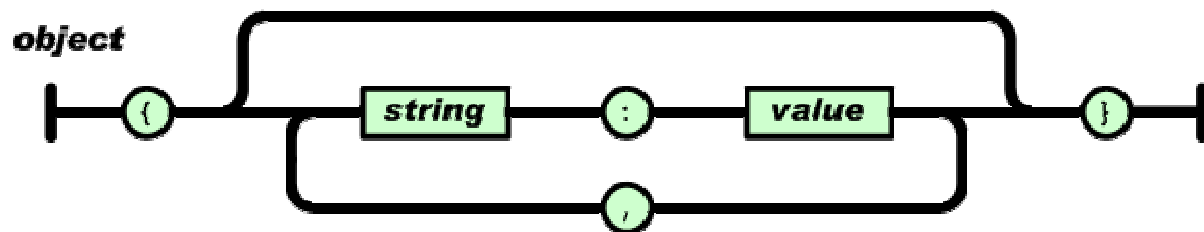
JSON terbuat dari dua struktur:

- 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*.
- Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

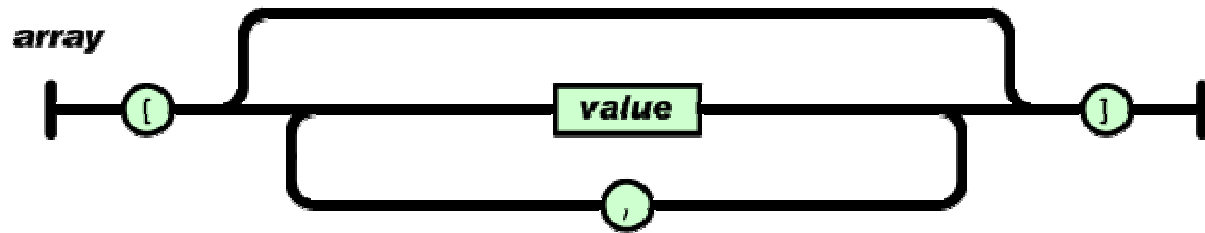
Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan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

JSON menggunakan bentuk sebagai berikut:

Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).

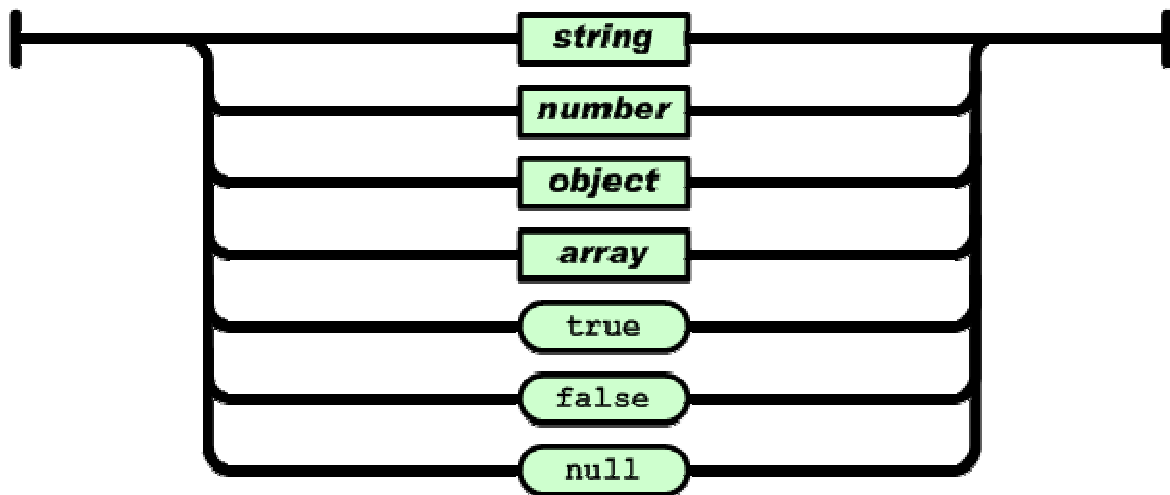


Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).

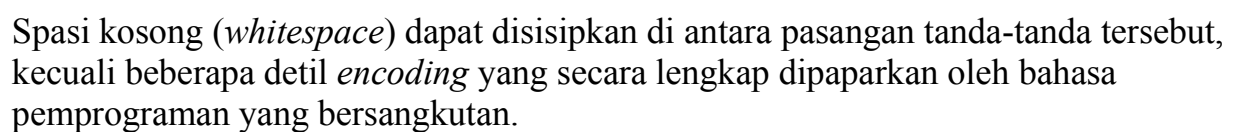
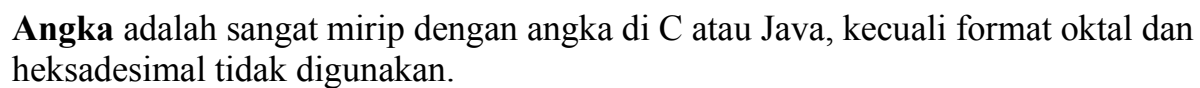


Nilai (*value*) dapat berupa sebuah **string** dalam tanda kutip ganda, atau *angka*, atau `true` atau `false` atau `null`, atau sebuah *objek* atau sebuah *larik*. Struktur-struktur tersebut dapat disusun bertingkat.

value



String adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.



-
- 8th:
 - [json>](#).
 - ABAP:
 - [EPO Connector](#).
 - ActionScript:
 - [ActionScript3](#).
 - Ada:
 - [GNATCOLL.JSON](#).
 - AdvPL:
 - [JSON-ADVPL](#).
 - ASP:
 - [JSON for ASP](#).
 - [JSON ASP utility class](#).
 - AWK:
 - [JSON.awk](#).
 - [rhawk](#).
 - Bash:
 - [Jshon](#).
 - [JSON.sh](#).
 - BlitzMax:
 - [bmx-rjson](#).
 - C:
 - [JSON_checker](#).
 - [YAJL](#).
 - [LibU](#).
 - [json-c](#).
 - [json-parser](#).
 - [jsonsl](#).
 - [WJElement](#).
 - [M's JSON parser](#).
 - [cJSON](#).
 - [Jansson](#).
 - [jsmn](#).
 - [parson](#).
 - [ujson4c](#).
 - [nxjson](#).
 - [frozen](#).
 - [microjson](#).
 - [mjson](#).
 - C++:
 - [JSONKit](#).
 - [jsonme--](#).
 - [ThorsSerializer](#).
 - [JsonBox](#).
 - [jvar](#).

- rapidjson.
- JSON for Modern C++.
- ArduinoJson.
- minijson.
- jsoncons.
- QJson.
- jsoncpp.
- CAJUN.
- libjson.
- nosjob.
- JSON++.
- JSON library for IoT.
- qmjson.
- JSON Support in Qt.
- JsonWax for Qt.
- C#:
 - fastJSON.
 - JSON_checker.
 - Jayrock.
 - Json.NET - LINQ to JSON.
 - JSON for .NET.
 - JSONSharp.
 - fluent-json.
 - Manatee Json.
 - FastJsonParser.
 - LightJson.
 - liersch.json.
- Ciao:
 - Ciao JSON encoder and decoder.
- Clojure:
 - data.json.
- Cobol:
 - XML Thunder.
 - Redvers COBOL JSON Interface.
- ColdFusion:
 - SerializeJSON.
 - toJSON.
- D:
 - Libdjson.
- Dart:
 - json library.
- Delphi:
 - Delphi Web Utils.
 - JSON Delphi Library.
- E:
 - JSON in TermL.

- Fantom:
 - [Json](#).
- FileMaker:
 - [JSON](#).
- Fortran:
 - [json-fortran](#).
 - [YAJL-Fort](#).
- Go:
 - [package json](#).
- Groovy:
 - [groovy-io](#).
- Haskell:
 - [RJson package](#).
 - [json package](#).
- Java:
 - [JSON-java](#).
 - [JSONUtil](#).
 - [jsonp](#).
 - [Json-lib](#).
 - [Stringtree](#).
 - [SOJO](#).
 - [json-taglib](#).
 - [Flexjson](#).
 - [JON tools](#).
 - [Argo](#).
 - [jsonij](#).
 - [fastjson](#).
 - [mjson](#).
 - [jjson](#).
 - [json-simple](#).
 - [json-io](#).
 - [JsonMarshaller](#).
 - [google-gson](#).
 - [Json-smart](#).
 - [FOSS Nova JSON](#).
 - [Corn CONVERTER](#).
 - [Apache johnzon](#).
 - [Genson](#).
 - [JSONUtil](#).
 - [cookjson](#).
- JavaScript:
 - [JSON](#).
 - [json2.js](#).
 - [clarinet](#).
 - [Oboe.js](#).
- LabVIEW:

- [flatten](#).
- Lisp:
 - [Common Lisp JSON](#).
 - [Emacs Lisp](#).
- LiveCode:
 - [mergJSON](#).
- LotusScript:
 - [JSON LS](#).
- Lua:
 - [JSON Modules](#).
- M:
 - [DataBallet](#).
- Matlab:
 - [JSONlab](#).
 - [20565](#).
 - [23393](#).
- Net.Data:
 - [netdata-json](#).
- Nim:
 - [Module json](#).
- Objective C:
 - [NSJSONSerialization](#).
 - [json-framework](#).
 - [JSONKit](#).
 - [yajl-objc](#).
 - [TouchJSON](#).
- OCaml:
 - [jsonm](#).
- PascalScript:
 - [JsonParser](#).
- Perl:
 - [CPAN](#).
 - [perl-JSON-SL](#).
- Photoshop:
 - [JSON Photoshop Scripting](#).
- PHP:
 - [PHP 5.2](#).
- PicoLisp:
 - [picolisp-json](#).
- Pike:
 - [Public.Parser.JSON](#).
 - [Public.Parser.JSON2](#).
- PL/SQL:
 - [pljson](#).
- PureBasic:
 - [JSON](#).

- Puredata:
 - [PuRestJson](#).
- Python:
 - [The Python Standard Library](#).
 - [simplejson](#).
 - [pyson](#).
 - [Yajl-Py](#).
 - [ultrajson](#).
 - [metamagic.json](#).
- R:
 - [rjson](#).
 - [jsonlite](#).
- Racket:
 - [json-parsing](#).
- Rebol:
 - [json.r](#).
- RPG:
 - [JSON Utilities](#).
- Rust:
 - [Serde JSON](#).
 - [json-rust](#).
- Ruby:
 - [yajl-ruby](#).
 - [json-stream](#).
- Scheme:
 - [MZScheme](#).
 - [PLT Scheme](#).
- Squeak:
 - [Squeak](#).
- Symbian:
 - [s60-json-library](#).
- Tcl:
 - [JSON](#).
- Visual Basic:
 - [VB-JSON](#).
 - [PW.JSON](#).
 - [.NET-JSON-Transformer](#).
- Visual FoxPro:
 - [fwJSON](#).
 - [JSON](#).
 - [vfpjson](#).

Mengenal Format JSON

<https://www.codepolitan.com/mengenal-format-json-59e8152dd0e51>



Pendahuluan

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 dan 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.

Sintaks dan Struktur

Sebuah objek JSON adalah format data key-value yang biasanya di render di dalam kurung kurawal. Saat kita bekerja dengan JSON, kita akan sering melihat objek JSON disimpan di dalam sebuah file `.json`, tapi mereka juga dapat disimpan sebagai objek JSON atau string di dalam sebuah program.

Sebuah objek JSON terlihat seperti berikut ini:

```
{
  "first_name": "Sammy",
  "last_name": "Shark",
  "location": "Ocean",
  "online": true,
  "followers": 987
}
```

Meskipun contoh di atas sangat singkat dan JSON dapat memiliki isi yang sangat banyak, contoh di atas secara umum menggambarkan dua kurung kurawal `{ }` di awal dan di akhir dengan pasangan key-value diantara kedua tanda kurung. Sebagian besar data yang dipakai di JSON dienkapsulasi di dalam sebuah objek JSON.

Pasangan key-value memiliki tanda titik dua diantara mereka `"key" : "value"`. Setiap key-value dipisahkan oleh sebuah koma, sehingga ditengah isi sebuah JSON terlihat seperti in: `"key" :`

"value", "key" : "value", "key": "value". Pada contoh di atas, nilai pertama pasangan key-value kita adalah "first_name" : "Sammy".

Key JSON berada di sebelah kiri tanda titik dua. Mereka perlu dibungkus oleh tanda petik dua seperti ini: "key", dan dapat berupa string apapun yang valid. Di dalam setiap objek, key haruslah unik. Key ini dapat memiliki spasi seperti di "first name", namun menambahkannya akan membuat kita lebih repot saat akan mengaksesnya di proses ngoding sehingga disarankan untuk menggunakan underscore seperti pada "first_name".

Value JSON ada di sebelah kanan tanda titik dua. Ada enam tipe data dasar yang bisa dipakai untuk mengisinya yaitu:

- strings
- numbers
- objects
- arrays
- Booleans (true atau false)
- null

Secara lebih luas, value juga dapat berisi tipe data yang lebih kompleks misalnya JSON object atau JSON array yang akan kita bahas di bagian berikutnya.

Setiap tipe data yang dimasukkan sebagai value ke dalam JSON akan mengingat sintaksnya, jadi string akan diberikan tanpa petik, namun tidak dengan angka.

Meskipun di dalam file .json kita sering melihat format ini ditulis dalam beberapa baris, JSON juga dapat ditulis disatu baris saja.

```
{ "first_name" : "Sammy", "last_name": "Shark", "online" : true, }
```

Menulis format JSON dalam bentuk beberapa baris akan membantunya lebih mudah dibaca terutama saat sudah memiliki banyak data. Karena JSON mengabaikan spasi antara elemennya, kita bisa memberikan spasi antara key-value sehingga menjadi lebih mudah di baca:

```
{  
  "first_name" : "Sammy",  
  "last_name" : "Shark",  
  "online" : true  
}
```

Penting juga untuk diingat, meskipun terlihat sama sebuah objek JSON tidak memiliki format yang sama dengan objek JavaScript. Jadi, meskipun kita bisa menambah fungsi kedalam objek JavaScript, kita tidak bisa menggunakannya sebagai value di JSON. Kelebihan utama JSON adalah kemudahan dan kesiapannya untuk ditransfer antar bahasa pemrograman. Objek JavaScript hanya bisa digunakan di dalam bahasa pemrograman JavaScript.

Sejauh ini kita sudah melihat format JSON dalam bentuk yang sederhana, namun JSON Juga dapat menjadi lebih kompleks yang terdiri dari objek dan array bersarang. Kita akan membahas JSON yang lebih kompleks di bagian berikutnya.

Bekerja dengan Data Kompleks di JSON

JSON dapat menyimpan objek bersarang maupun array bersarang. Seperti data lain, objek atau array ini dapat disimpan ke dalam sebuah key

Object Bersarang

Di dalam file `users.json` berikut, untuk empat pengguna ("`sammy`", "`jesse`", "`drew`", "`jamie`") ada sebuah objek JSON didalam setiap value key keempatnya yang memiliki key-nya sendiri ("`username`" dan "`location`") milik setiap user.

`users.json`

```
{
  "sammy" : {
    "username" : "SammyShark",
    "location" : "Indian Ocean",
    "online" : true,
    "followers" : 987
  },
  "jesse" : {
    "username" : "JesseOctopus",
    "location" : "Pacific Ocean",
    "online" : false,
    "followers" : 432
  },
  "drew" : {
    "username" : "DrewSquid",
    "location" : "Atlantic Ocean",
    "online" : false,
    "followers" : 321
  },
  "jamie" : {
    "username" : "JamieMantisShrimp",
    "location" : "Pacific Ocean",
    "online" : true,
    "followers" : 654
  }
}
```

Pada contoh di atas, tanda kurung kurawal digunakan untuk membuat objek JSON bersarang di mana setiap username dan location masing-masing menjadi milik keempat user. Seperti data lainnya, koma dipakai untuk memisahkan antar elemen.

Array Bersarang

Data dapat dimasukkan ke dalam format JSON menggunakan array JavaScript sebagai sebuah value. JavaScript menggunakan kurung siku `[]` di awal dan akhir sebuah array. Array adalah sebuah koleksi terurut dan memiliki tipe data yang berbeda.

Kita dapat menggunakan array saat bekerja dengan banyak data yang dapat dikelompokkan, seperti profil sosial media yang dimiliki oleh seorang user.

Setelah kita membahas tentang array bersarang, profil user untuk Sammy dapat terlihat sebagai berikut:

user_profile.json

```
{
  "first_name" : "Sammy",
  "last_name" : "Shark",
  "location" : "Ocean",
  "websites" : [
    {
      "description" : "work",
      "URL" : "https://www.digitalocean.com/"
    },
    {
      "description" : "tutorials",
      "URL" : "https://www.digitalocean.com/community/tutorials"
    }
  ],
  "social_media" : [
    {
      "description" : "twitter",
      "link" : "https://twitter.com/digitalocean"
    },
    {
      "description" : "facebook",
      "link" : "https://www.facebook.com/DigitalOceanCloudHosting"
    },
    {
      "description" : "github",
      "link" : "https://github.com/digitalocean"
    }
  ]
}
```

Key `"websites"` dan `"social_media"` dapat menggunakan array untuk menyimpan data yang dimiliki oleh Sammy berupa 2 link website dan 3 profil sosial media. Kita tahu bahwa data tersebut array karena ada kurung sikunya.

Menggunakan data bersarang di dalam format JSON memungkinkan kita untuk bekerja pada data yang lebih kompleks.

Perbandingan dengan XML

XML, atau **eXtensible Markup Language**, adalah sebuah cara untuk menyimpan data yang dapat dibaca baik oleh manusia maupun mesin. Format XML tersedia secara luas bagi banyak bahasa pemrograman.

XML sangat mirip dengan JSON tapi membutuhkan lebih banyak teks sehingga lebih panjang isinya dan lebih lama untuk membaca dan menulisnya. XML harus dibaca dengan sebuah XML parser, namun JSON dapat dibaca menggunakan fungsi standar. Juga seperti JSON, XML tidak bisa menggunakan array.

Kita akan melihat contoh data yang disimpan ke dalam format XML dan JSON.

users.xml

```
<users>
  <user>
    <username>SammyShark</username> <location>Indian Ocean</location>
  </user>
  <user>
    <username>JesseOctopus</username> <location>Pacific Ocean</location>
  </user>
  <user>
    <username>DrewSquir</username> <location>Atlantic Ocean</location>
  </user>
  <user>
    <username>JamieMantisShrimp</username> <location>Pacific Ocean</location>
  </user>
</users>
```

users.json

```
{"users": [
  {"username": "SammyShark", "location": "Indian Ocean"},
  {"username": "JesseOctopus", "location": "Pacific Ocean"},
  {"username": "DrewSquid", "location": "Atlantic Ocean"},
  {"username": "JamieMantisShrimp", "location": "Pacific Ocean"}
]}
```

Kita dapat lihat bahwa JSON jauh lebih ringkas dan tidak memerlukan tag penutup seperti halnya XML. Sebagai tambahan, XML tidak menggunakan sebuah array seperti yang tadi kita pelajari.

Jika pembaca sudah mengenal HTML, pembaca akan melihat kemiripan dari penggunaan tag-nya. Meski JSON lebih singkat dan lebih mudah dari XML serta dapat dipakai diberbagai situasi

termasuk aplikasi AJAX, kita perlu memahami tipe proyek yang sedang dikerjakan sebelum memutuskan struktur data apa yang akan di pakai.

Sumber-sumber Lain

JSON merupakan format natural untuk JavaScript dan memiliki banyak implementasi yang dapat langsung dipakai diberbagai bahasa pemrograman populer. Pembaca dapat melihat bahasa yang didukung pada situs "[Introducing JSON](#)", dan [jQuery library](#) juga memiliki kemampuan untuk membaca format ini.

Seringkali, kita tidak perlu menulis JSON utuh namun mengambilnya dari suatu sumber data atau mendapatnya dari mengubah jenis data lain ke JSON. Kita dapat mengubah CSV atau data yang terpisah dengan tab ke dalam JSON menggunakan aplikasi open source [Mr. Data Converter](#). Kita juga dapat mengubah XML ke JSON atau sebaliknya dengan aplikasi berlisensi Creative Commons ini [utilities-online.info site](#).

Saat mengubah tipe data lain ke JSON, atau membuat tipe data sendiri (dengan objek bersarang), anda dapat memvalidasi JSON dengan [JSONLint](#) dan dapat menguji file JSON dalam konteks web development dengan [JSFiddle](#).

Penutup

JSON adalah format yang ringan dan mempermudah kita dapat membagi, menyimpan dan bekerja dengan data. Sebagai sebuah format, JSON telah mendapatkan dukungan yang makin meningkat dalam bentuk API, misalnya di [Twitter API](#).

Karena kita mungkin akan jarang membuat file [.json](#) sendiri namun mengunduhnya dari sumber lain, pikirkan bagaimana menggunakan JSON dengan baik diprogram daripada memikirkan strukturnya.

Diterjemahkan dari [An Introduction to JSON](#) di bawah [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#). Sumber gambar: [speakingjs](#)

Konsep Dasar JSON dan penggunaannya

<https://terusbelajar.wordpress.com/2010/09/03/konsep-dasar-json-dan-penggunaannya/>

Apa itu JSON ? Bagaimana menggunakannya dalam code-code kita ? Mungkin teman pernah mendengar istilah JSON ini saat membaca tutorial-tutorial mengenai implementasi dari AJAX. Ya memang saat ini JSON sangat diperlukan sekali guna implementasi AJAX. Saya sendiri pertama kali dengar istilah ini sekitar 2 tahun lalu saat dosen saya mengajarkan AJAX dengan PHP. Tapi gak mudeng :)...

Sekarang sudah mengerti dikit, jadi saya coba share agar nanti tidak lupa dan dapat digunakan untuk teman-teman yang sedang belajar JSON juga.

Sama seperti [XML](#), [JSON](#) berfungsi untuk merepresentasikan data. Ingat hanya data saja focus utamanya. Yang tentunya antara XML dan JSON memiliki kelebihan dan kekurangannya serta cara merepresentasikan data juga berbeda syntaxnya. Teman bisa perjelas hal itu [di artikel ini](#).

JSON sendiri singkatan dari JavaScript Object Notation. Ada hubungannya dengan Javascript ya..

Lalu bagaimana menggunakannya ? Berikut adalah contoh codenya.

```
<html>
<head>
<title>JSON testing - adiputra</title>
<script type='text/javascript'>
    var JSON_adi = {nama:'adiputra'};
    document.write (JSON_adi.nama);
</script>
</head>
<body>
</body>
</html>
```

JSON_adi adalah nama variable pada javascript yang menunjukan bahwa variable tersebut berbentuk Object JSON karena terdapat kurung kurawal `{}`. Di dalam kurung kurawal terdapat atribut `* nama *` dengan value `* adiputra *`. Ingat `* nama *` adalah atribut, sedangkan `* adiputra *` adalah value dari atribut `* nama *` dan keduanya dipisahkan dengan titik dua `:`. Lalu untuk memanggil valuenya kita dapat panggil dengan code

```
1    document.write (JSON_adi.nama);
```

Begitu kurang lebihnya, jadi variable JSON_adi merepresentasikan data dimana data tersebut memiliki atribut `* nama *` dengan value `* adiputra *`. Datanya kalau integer bagaimana ? lalu bisa data array tidak ?

Berikut contoh tambahan code lagi

```
<html>
<head>
<title>JSON testing - adiputra</title>
<script type='text/javascript'>
    var JSON_adi = {
        nama:'adiputra',
        umur:21,
        status_nikah:false,
        hobi:['coding','main game','ngetik']
    };

    var nikah;

    if(JSON_adi.status_nikah){
        nikah = "sudah nikah";
```

```

    }else{
        nikah = "belum nikah";
    }

    document.write (JSON_adi.nama +" berumur "+JSON_adi.umur+" suka "+JSON_adi.hobi[0]+ " dan
    "+nikah);

</script>
</head>
<body>

</body>
</html>

```

Pada code diatas, kita dapat membuat data tipe integer, Boolean dan juga array. Cara pemanggilannya pun sederhana bukan...Didalam array kita juga dapat membuat data object lagi berikut contohnya :

```

<html>
<head>
<title>JSON testing - adiputra</title>
<script type='text/javascript'>
    var JSON_adi = {
        nama:'adiputra',
        umur:21,
        status_nikah:false,
        hobi:['coding','main game','ngetik'],
        teman:[
            {nama:'anas', universitas:'UGM'},
            {nama:'wahyu', universitas:'UNIKOM'},
            {nama:'adesetio', universitas:'UGM'},
            {nama:'nano', universitas:'UNIKOM'}
        ]
    };

    var nikah;

    if(JSON_adi.status_nikah){
        nikah = "sudah nikah";
    }else{
        nikah = "belum nikah";
    }

    document.write(JSON_adi.nama +" berumur "+JSON_adi.umur+" suka "+JSON_adi.hobi[0]+ " dan
    "+nikah+ "<br />");
    document.write("List teman : <br />");
    document.write("<ol>")
    for(var i=0; i<JSON_adi.teman.length; i++){

```

```

        document.write("<li>" + JSON_adi.teman[i].nama + " kuliah di " + JSON_adi.teman[i].universitas +
"</li>");
    }
    document.write("</ol>")

</script>
</head>
<body>

</body>
</html>

```

Dari code diatas, moga teman mengerti ya. Silahkan komentar / kritik bila kurang mengerti.

Lalu bagaimana implementasi ke PHP-nya ? Berikut ini saya coba membuat code untuk menjelaskannya :

```

$array_adi = Array(
    "nama" => "adiputra",
    "nim" => "10107633",
    "jurusan" => "TI",
    "kelas" => "IF-13"
);
echo "Array dalam PHP : ";
print_r($array_adi);
echo "<br />";
echo "-----<br />";

echo "Array yang sudah convert ke JSON : ";
$json_adi = json_encode($array_adi);
print_r($json_adi);

echo "<br /><br />"
?>

<html>
<head>
    <script language='javascript'>
        function ubah_json(){
            var text_json = document.getElementById("json_id").value;

            var json_obj = eval('(' + text_json + ')');

            alert(json_obj.nama+ " kelas " +json_obj.kelas+ " jurusan " +json_obj.jurusan+ " NIM
"+json_obj.nim);
        }
    </script>
</head>

```

```

<body>
<form action="tes_json.php" method="POST">
  <input type='hidden' name='json_text' id='json_id' value='<?php echo $json_adi; ?>' />
  <input type='submit' name='tbl' value="Convert ke Array lagi" />
  <input type='button' name='button1' value='Ubah ke Object JSON' onclick='ubah_json()' />
</form>
</body>
</html>

```

```

<?php
if($_POST[tbl]=="Convert ke Array lagi"){
  // ubah bentuk string menjadi bentuk array
  echo "<b>Sebelum Slash dihapus : </b>".$_POST[json_text]."<br /><br />";

  $hilang_slash = stripslashes($_POST[json_text]);
  echo "<b>Sesudah Slash dihapus : </b>".$hilang_slash."<br /><br />";

  $array_back = json_decode($hilang_slash);
  echo "<b>Hasil convert menjadi array kembali : </b>";
  print_r($array_back);
}
?>

```

Awalnya saya membuat array seperti ini.

```

$array_adi = Array(
  "nama" => "adiputra",
  "nim" => "10107633",
  "jurusan" => "TI",
  "kelas" => "IF-13"
);

```

Lalu setelah itu saya ubah menjadi bentuk sementara JSON sehingga nanti bisa dimanipulasi di javascript. Untuk mengubahnya PHP tinggal menggunakan code `json_encode()`. Sebenarnya hasil dari `json_encode()` berbentuk string.

Disini, saya coba dahulu melakukan convert bentuk string yang baru saja di encode dengan code `json_encode()` menjadi sebuah array kembali dengan perintah `json_decode()`.

Perintah `stripslashes()` berfungsi untuk menghapus tanda slash `*/` karena bentuk dari array yang sudah diconvert menjadi String .

Hasilnya bisa dilihat pada gambar dibawah ini.

```

Array dalam PHP : Array ( [nama] => adiputra [nim] => 10107633 [jurusan] => TI [kelas] => IF-13 )
-----
Array yang sudah convert ke JSON : ("nama":"adiputra","nim":"10107633","jurusan":"TI","kelas":"IF-13")

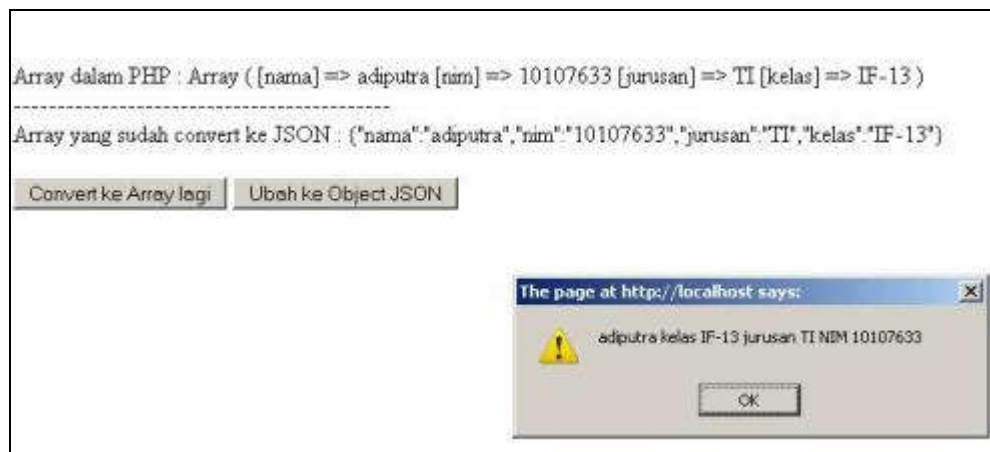
Convert ke Array lagi  Ubah ke Object JSON

Sebelum Slash dihapus : (\'nama\'\'adiputra\'\'\'nim\'\'10107633\'\'\'jurusan\'\'TI\'\'\'kelas\'\'IF-13\'')
Sesudah Slash dihapus : ("nama":"adiputra","nim":"10107633","jurusan":"TI","kelas":"IF-13")

Hasil convert menjadi array kembali : stdClass Object ( [nama] => adiputra [nim] => 10107633 [jurusan] => TI [kelas] => IF-13 )

```

Lihat, kita telah berhasil mengubah kembali menjadi array bukan...
 Sekarang kita ubah menjadi object dalam Javascript. Klik saja tombol Ubah ke Object JSON, lalu lihat alert yang muncul ? taadaaa....kita sudah membuatnya dalam bentuk object javascript



Kok bisa ? lihat codenya ya..Yang paling penting adalah code

```
1 var json_obj = eval('(' + text_json + ');');
```

dimana text_json didapat dari id element text input yang valuenya dari encode_json() pada coding PHP.

Sampai sini, moga ada sedikit pemahaman mengenai apa itu JSON dan hubungannya dengan PHP. Nanti saya coba buat artikel bagaimana menggunakan json untuk aplikasi Ajax.

Mohon koreksi juga kalau ada yang salah ya...

Dasar AJAX dengan jQuery,JSON dan PHP

<https://terusbelajar.wordpress.com/2010/09/07/dasar-ajax-dengan-jqueryjson-dan-php/>

Setelah kemarin sempet *posting dasar JSON*, kini saya coba buat tutor sederhana bagaimana kita mengakses data pada database menggunakan jQuery, JSON, PHP dengan kata lain AJAX. Konsep AJAX sebelumnya

mesti kita pahami minimal mengerti kerja dari teknologi web satu ini yang kemungkinan masih terus berkembang. Bagi yang belum memahami, teman bisa liat gambar menariknya pada [artikel ini](#). Saya sendiri kalau koding di AJAX manual belum pernah karena ribet namun secara konsep ngerti dan paham,,tinggal latihan aja kali ya, tapi inget saran senior lewat chatting kiri-kanan kalau bahasa pemrograman itu hanya tool yang memudahkan kita ngedevelop program. Penguasaan salah satu tool itu mesti dimiliki,,cari tool yang mudah dan sesuai dengan program yg akan kita develop. Ya sudah,,saya belajar AJAX di jQuery saja...ups,,malah curhat..)

Baik sekarang kita mulai,

Awalnya saya membuat data yang berisi informasi teman dalam tipe file .json. Berikut adalah codenya :

```
[
  {"nama":"adiputra","hobby":"main game"},
  {"nama":"nano","hobby":"jalan-jalan"},
  {"nama":"ade","hobby":"baca buku"},
  {"nama":"husni","hobby":"denger radio"}
]
```

Saya simpan dengan nama **data.json**. syntax apa tuh ? klu belum paham dengan JSON,bisa liat diartikel saya sebelumnya.

Lalu saya buat file dengan nama **json_jquery.html** yang berisi code untuk nge-load file data.json. berikut adalah codenya :

```
<html>
<head>
<title>json + jquery</title>
<script type='text/javascript' src='../..../js/jquery-1.4.2.min.js'></script>
<script type='text/javascript'>
$(function(){
  $("#form_json").submit(function(){

    $.ajax({
      url:"data.json",
      dataType:"json",
      success:function(data){
        teman = "<ol>";
        $.each(data, function(i,n){
          teman = teman + "<li>" + n["nama"] + " suka : " + n["hobby"] + "</li>";
        });
        teman = teman + "</ol>";
        $(".result_json").append(teman);
      }
    });

    return false;
  });
});
```

```

</script>
</head>
<body>
Untuk menampilkan data json, klik tombol disamping...
<form id='form_json'>
  <input type='submit' class='tbl_ok' value='Lihat' />
</form>

<div class='result_json'>

</div>
</body>
</html>

```

Oke,,simpan dan kita pelan-pelan mulai pahami code sederhana itu :

```
$("#form_json").submit(function() {
```

Adalah code yang mendefinisikan bahwa bila form dengan id form_json itu di submit (event submit) maka return kan false agar tidak ngeload sehingga biar si AJAX yang bekerja nge-load data. Ini bahasa sederhana saya kalau mendefinisikan AJAX...)

lihat, return bernilai false kan..

Lalu setelah si form di submit sebelum return-nya false, kita mendefinisikan code AJAX

```

$.ajax({
    url:"data.json",
    dataType:"json",
    success:function(data) {
        teman = "<ol>";
        $.each(data, function(i,n){
            teman = teman + "<li>" + n["nama"] + " suka : " +
n["hobby"] + "</li>";
        });
        teman = teman + "</ol>";
        $(".result_json").append(teman);
    }
});

```

Secara sederhana apabila kita mendefinisikan AJAX pada jQuery seperti ini :

```

$.ajax({
    type: 'GET',
    url: 'getDetails.php',
    data: { id: 142 },
    success: function(data) { // code fungsi callback };
});

```

Sebenarnya tidak hanya empat parameter, banyak, namun yg sering digunakan seperti itu dan sesuaikan dengan program yang ingin kita buat.

- Parameter type : mendefinisikan method HTTP request dari AJAX. Ada 2 tipe GET dan POST dan keduanya memiliki kelebihan masing-masing dan lagi-lagi kita mesti sesuaikan dengan program. Kalau tidak didefinisikan, defaultnya adalah GET.
- Parameter URL : alamat dari data yang akan kita kirim. Kalau saya menyebutnya file query ke dbnya.

- Parameter DATA : adalah ya data yang akan dikirim ke db, misalnya diatas disebutkan bahwa id=142. Berarti pada file getDetails.php ada perintah mendapatkan variable id (\$id=\$_GET[id]) .
- Parameter SUCCESS : nah ini yang penting juga,,jadi apabila si AJAX telah selesai ngambil data di server, lakukan respons dan aksi. Biasa disebut fungsi callback. Kalau pada sampe AJAX sederhana saya hanya menampilkan data dalam bentuk tag (order list).

Parameter yang lainnya bisa dibaca di alamat ini : <http://api.jquery.com/jquery.ajax/>

Untuk Demo sampe AJAX pertama ini bisa di liat disini : [[demo](#)]

Sample yang pertama hanya ngeload data dari file data.json. Lalu bagaiman ngeload data dari database ?

Berikut ini sample keduanya,,awalnya saya buat databasenya terlebih dahulu :

```
CREATE TABLE `teman` (
  `id_teman` INT( 11 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `nama` VARCHAR( 50 ) NOT NULL ,
  `hobby` TEXT NOT NULL
) ENGINE = MYISAM ;
```

```
INSERT INTO
  `blog_tes`.`teman` (
    `id_teman` ,
    `nama` ,
    `hobby`
  )
VALUES
  (NULL , 'adiputra', 'main game'),
  (NULL , 'nano', 'jalan-jalan'),
  (NULL , 'ade', 'baca buku'),
  (NULL , 'husni', 'denger radio');
```

Setelah itu saya buat file **index.php** yang sama persis dengan sample pertama file **json_jquery.html** :

```
<html>
<head>
<title>json + jquery</title>
<script type='text/javascript' src='../..../js/jquery-1.4.2.min.js'></script>
<script type='text/javascript'>
$(document).ready(function(){
  $("#loading").hide();
});

$(function(){
  $("#form_json").submit(function(){

    $("#loading").show();
    $("#loading").fadeIn(500).html("<img src='loading.gif' /> Loading data...");
$.ajax({
  url:"proses.php",
  dataType:"json",
  success:function(data){
    teman = "<ol>";
```

```

        $.each(data, function(i,n){
            teman = teman + "<li>" + n["nama"] + " suka : " + n["hobby"] + "</li>";
        });
        teman = teman + "</ol>";
        $(".result_json").append(teman);

        $("#loading").hide();
    }
});

return false;
});
</script>
</head>
<body>
    Untuk menampilkan data json dari database, klik tombol disamping...
    <form id='form_json'>
        <input type='submit' class='tbl_ok' value='Lihat' />
    </form>

    <div id='loading'></div>
    <div class='result_json'>
    </div>
</body>
</html>

```

Lalu saya buat **proses.php** dimana file proses.php ini akan menjadi value dari parameter URL. Berikut codenya :

```

<?php
    # koneksi ke db
    $h = "localhost";
    $u = "root";
    $p = "";
    $db = "blog_tes";

    mysql_connect($h,$u,$p) or die('error : '.mysql_error());

    mysql_select_db($db) or die('error : '.mysql_error());
    # akhir dari code koneksi ke db

    # query ke db
    $sql = "SELECT * FROM teman";
    # jalankan query
    $req_db = mysql_query($sql);

    # ambil hasil query, simpan dalam bentuk array.
    while ($data = mysql_fetch_array($req_db)) {
        $array_data[] = array(

```

```

        "nama"=>$data['nama'],
        "hobby"=>$data['hobby']
    );
}
# encode menjadi bentuk data JSON
echo(json_encode($array_data));

```

?>

Isi file proses.php hanya settingan DB dan query ke DB, lalu liat datanya diparse ke bentuk JSON dengan perintah json_encode().

Penjelasan Struktur Data JSON

<https://www.candra.web.id/penjelasan-struktur-data-json/>

Published by [Candra Adi Putra](#) on 12 June , 2014



JSON adalah kependekan dari Javascript Object Notation. JSON dipakai untuk dua hal; yang pertama untuk format transfer data antar device/OS/bahasa pemrograman yang berbeda dan sebagai media penyimpanan data seperti MongoDB. JSON dalam artikel ini lebih menekankan pada transfer data. Sebenarnya selain JSON, format transfer data antar sistem bisa dilakukan XML, plaintext ataupun HTML, namun 3 format tersebut punya banyak kekurangan, terutama dari sisi ukuran. Saingan terdekat dari JSON adalah XML, dalam layanan Web API atau webservice, sebagian layanan menggunakan JSON, sebagian yang lain menggunakan format XML. Dua format ini bisa saling menggantikan.

Kelebihan JSON

Walaupun XML sudah sejak lama dipakai dalam format transfer data, namun beberapa tahun terakhir, JSON menggeser posisi XML. Kelebihan JSON dari XML adalah sbb:

Ukuran lebih kecil dibanding XML, efeknya transfer data lebih cepat dan lebih hemat resource, terutama bandwidth

JSON adalah format data bawaan di Javascript, artinya jika data dari server di kirim ke client , dan client menggunakan javascript, maka tidak perlu library tambahan untuk memprosesnya

Dibanding XML, format JSON lebih sederhana.

library JSON ada di setiap bahasa pemrograman sehingga memudahkan programmer yang berbeda bahasa pemrograman. (lihat di :

Perbandingan format JSON dan XML

berikut ini adalah contoh data JSON tentang event kegiatan

```

{"list_event":
[

```

```

{"eventID":"1",
"judul":"Pengajian Akbar menyambut ramadhan",
"tanggal":"2014-05-19",
"jam":"20:00",
"lokasi":"Masjid Kampus UGM"
},
{"eventID":"2",
"judul":"Bedah buku kesesatan syiah",
"tanggal":"2014-05-21",
"jam":"09:00",
"lokasi":"Masjid Kampus UII "
}
]
}

```

list_event adalah Array JSON. Array terdiri dari beberapa object. Dalam kasus ini, kita punya 2 object event. Masing masing object event ini mempunyai 5 field.

Representasi Data dalam format XML

Format JSON diatas bisa direpresentasikan dengan XML. Berikut ini adalah format XML untuk data yang sama

```

<list_event>
  <event>
    <eventID>1</eventID>
    <judul>Pengajian Akbar menyambut ramadhan</judul>
    <tanggal>2014-05-19</tanggal>
    <jam>20:00</jam>
    <lokasi>Masjid Kampus UGM</lokasi>
  </event>
  <event>
    <eventID>2</eventID>
    <judul>Bedah buku kesesatan syiah</judul>
    <tanggal>2014-05-21</tanggal>
    <jam>09:00</jam>
    <lokasi>Masjid Kampus UII </lokasi>
  </event>
</list_event>

```

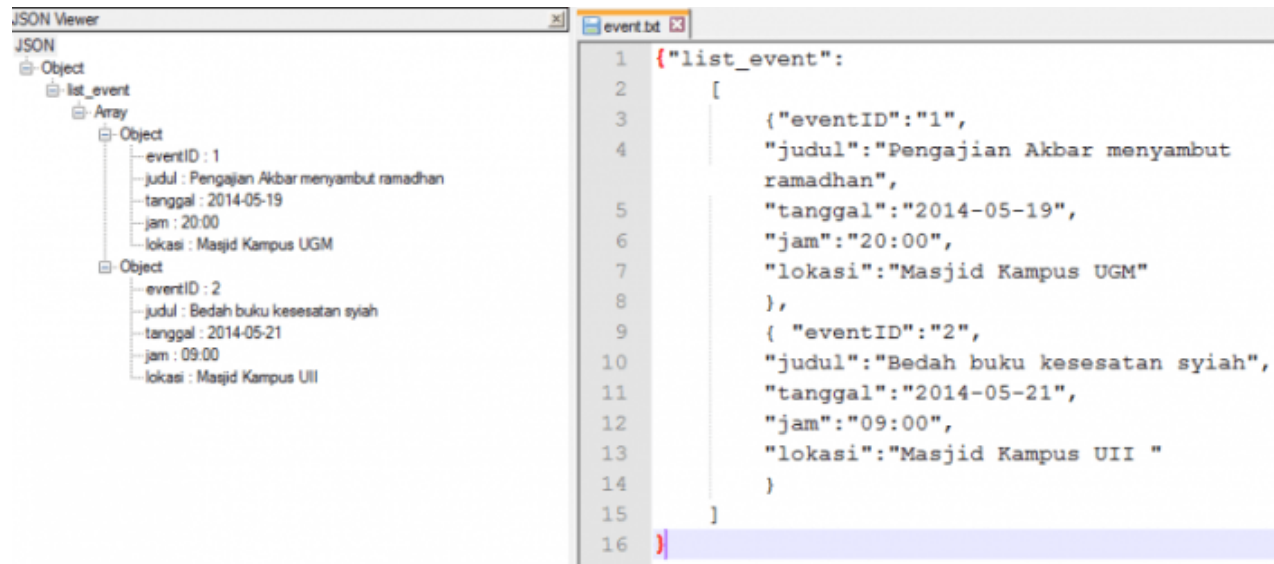
Tampak perbedaan yang paling mendasar adalah ukuran JSON lebih kecil, perhatikan jumlah text dan tag penutup di XML yang membuat ukuran XML menjadi besar. Dari dua format diatas juga terlihat format XML lebih kompleks.

Cara mudah menganalisa format JSON

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.

Untuk memakai plugin ini, install, lalu blok kode JSONnya lalu klik menu plugin->JSON viewer->Show JSON Viewer



JSON viewer di Notepad++

<http://codeborneo.com/belajar-menguraikan-json/>

Membuat format JSON secara manual sangat jarang dilakukan dalam pembuatan program ataupun layanan webservice. Sekenario yang umum adalah data didapat dari database lalu sebelum di kirim ke client, data di format sebagai JSON. Dalam artikel selanjutnya kita akan membahas cara membuat format JSON dari database MySQL dengan Bahasa pemrograman PHP.

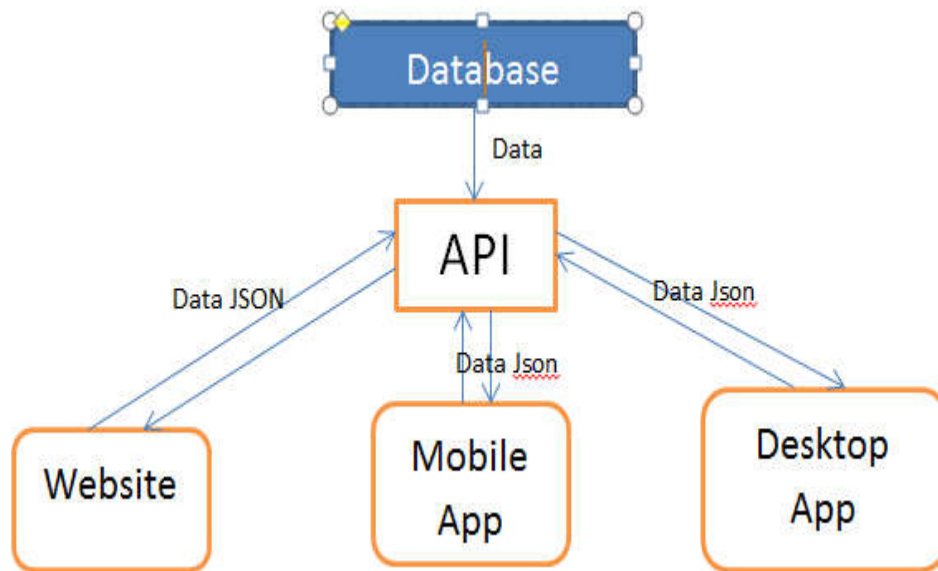
Belajar menguraikan JSON

Hi para sobat borneo, hari ini kita akan belajar menguraikan JSON (Javascript object notation).

Kenapa kita harus mengerti materi ini?

Itu karena JSON merupakan salah satu format pertukaran data yang mudah di mengerti manusia atau manusiabile (halah apa coba hehe) dan juga sangat sering digunakan dalam bertukar data pada banyak **aplikasi android**.

Pada tutorial-tutorial selanjutnya pun kita akan banyak menggunakan format json ini sebagai format pertukaran data antara database kita di server dengan aplikasi android di client. Kira-kira seperti inilah contoh gambarannya.



Nah oleh karena itu kita harus benar-benar paham dulu nih tentang JSON sebelum kita merambah ke tutorial-tutorial berikutnya. Pada tutorial-tutorial selanjutnya kita akan banyak membangun aplikasi-aplikasi kompleks yang memanfaatkan arsitektur seperti pada gambar diatas.

Contoh JSON

Nah dibawah ini merupakan contoh dari data berformat JSON. Contoh berikut sangatlah simpel karena tujuan kita adalah untuk belajar. Pada aplikasi nyata biasanya format data json akan jauh lebih kompleks dan banyak. Data json dibawah ini dapat diakses melalui link

berikut <http://codeborneo.com/febapp/v1/eveng/0> dan untuk mengaksesnya sobat borneo dapat menggunakan browser biasa dengan memasukan url tersebut atau menggunakan bantuan tools seperti [Advance Rest Client](#) ataupun [POSTMAN](#)

```
{
  "success": true,
  "barang": [
    {
      "no": 1,
      "id": "1",
      "nama": "Lemari",
      "harga": 500000,
      "stok": 0,
      "gambar": "poster/grand-launching-gerakan-ui-mengajar-51.png"
    },
    {
      "no": 2,
      "id": "2",
      "nama": "AC 100 PK",
      "harga": 600000,

```

```

        "stok": 39,
        "gambar": "poster/Donor_darah.jpg"
    },
    {
        "no": 3,
        "id": "3",
        "nama": "Kursi lipat serbaguna",
        "harga": 80000,
        "stok": 80,
        "gambar": "poster/Career-Expo-by-PNJ-Fair-Balairung-UI-Depok-24-25-April-2015.jpg"
    },
    {
        "no": 4,
        "id": "9",
        "nama": "stik ps",
        "harga": 80000,
        "stok": 80,
        "gambar": "poster/Career-Expo-by-PNJ-Fair-Balairung-UI-Depok-24-25-April-2015.jpg"
    },
    {
        "no": 5,
        "id": "10",
        "nama": "hardisk",
        "harga": 80000,
        "stok": 80,
        "gambar": "poster/Career-Expo-by-PNJ-Fair-Balairung-UI-Depok-24-25-April-2015.jpg"
    },
    {
        "no": 6,
        "id": "11",
        "nama": "ampli",
        "harga": 500000,
        "stok": 0,
        "gambar": "poster/grand-launching-gerakan-ui-mengajar-51.png"
    },
    {
        "no": 7,
        "id": "12",
        "nama": "Laptop asus a455lf",
        "harga": 600000,
        "stok": 39,
        "gambar": "poster/Donor_darah.jpg"
    }
],
}

```

Perbedaan antara [dan {

Jika para sobat borneo perhatikan lebih rinci pada contoh data json diatas maka kita dapat melihat tanda [dan tanda { apa maksud dari kedua tanda ini? Oke kita akan bahas satu persatu.

tanda {

Tanda { artinya adalah JsonObject, jadi kalau data json kita dimulai dengan tanda ini { maka itu berarti data json kita dimulai dengan JsonObject dan kita dapat mengaksesnya dengan method getJsonObject. Dalam contoh data json diatas kita dapat melihat bahwa data json diatas dimulai dengan tanda {. Itu berarti diawal kita menggunakan method getJsonObject untuk mengaksesnya. Lalu terdapat 2 objek utama pada contoh data json diatas yaitu objek success dan objek barang.

tanda [

Untuk tanda [artinya adalah JsonArray, jadi kalau data json kita dimulai dengan tanda ini [maka itu berarti data json kita dimulai dengan JsonArray dan kita dapat mengaksesnya dengan method getJSONArray. Dalam contoh data json diatas kita dapat melihat bahwa data json barang merupakan sebuah jsonArray yang didalamnya terdapat array 0 sampai dengan 6 sehingga ada 7 data. Lalu masing-masing array tersebut memiliki objek no, id, nama, harga, stok, dan gambar.

Key dan value

Lalu untuk objek sendiri memiliki key dan valuenya masing-masing. Apa itu key dan value? Key sendiri merupakan kunci yang mana jika kita ingin mengakses sebuah data json kita bisa menggunakan key tersebut untuk mengidentifikasi data mana yang mau kita gunakan. Lalu value merupakan isi dari key tersebut atau bisa dibilang datanya.

Jadi sebagai contoh pada contoh data json diatas kita mempunyai data berikut "no": 1, "id": "1", "nama": "Lemari", "harga": 500000, "stok": 0, "gambar": "poster/grand-launching-gerakan-ui-mengajar-51.png" lalu misalkan kita ingin menggunakan data poster/grand-launching-gerakan-ui-mengajar-51.png, maka kita dapat mengaksesnya dengan memanggil keynya yaitu gambar.

Sekarang kita sudah memahami dasar dari penguraian json. Paling tidak sobat borneo harus memahami terlebih dahulu materi diatas untuk melanjutkan ke tutorial-tutorial berikutnya karena pada tutorial-tutorial berikutnya kita akan menggunakan JSON untuk bertukar data pada setiap aplikasi yang akan kita buat.