

# Panduan Lengkap Memahami JSON

Agus Prawoto Hadi, 27-01-2018, <http://jagowebdev.com/memahami-json/>

Pada kesempatan kali ini kita akan membahas tentang JSON. Meskipun diinternet sudah banyak dibahas, namun kali ini kita akan mengupasnya lebih dalam lagi sehingga sobat benar benar dapat memahami JSON dengan baik.

## I. Memahami JSON – Apa Itu JSON ?

JSON merupakan kependekan dari JavaScript Object Notation.

JSON sendiri dapat diartikan sebagai **teks** dengan format tertentu yang digunakan untuk memfasilitasi pertukaran data antar **bahasa pemrograman**. Apa maksudnya?

1. **Teks**. Format standar untuk pertukaran data adalah teks, mengapa teks? karena dengan teks, semua bahasa pemrograman dapat menerimanya.

2. **Antar bahasa pemrograman**: ya antar bahasa pemrograman (misal PHP ke Javascript). Setiap bentuk format pertukaran data, pasti ditujukan untuk berbagai bahasa pemrograman, meskipun juga dapat untuk satu bahasa pemrograman.

Jika pertukaran hanya dilakukan dalam satu bahasa pemrograman, maka banyak alternatif yang lebih baik dan lebih mudah diantaranya menggunakan fitur bawaan dari bahasa pemrograman tersebut. Misal menggunakan fungsi serialize pada PHP.

## Bentuk Format JSON

Penulisan JSON mengacu pada standar ECMA Script 262 Edisi ke 3 (1999). Adapun contoh format penulisannya adalah sebagai berikut:

```
{
    "judul": "Panduan Lengkap Memahami JSON",
    "penulis": "Agus Prawoto Hadi",
    "date": "2018-01-25"
}
```

Jika diperhatikan, bentuk diatas sama persis dengan bentuk objek Javascript. Ya, karena JSON ini terinspirasi dengan objek literal pada Javascript, namanya saja JavaScript Object Notation

Karena FULL mengadopsi bentuk Javascript, maka segala sesuatu yang digunakan di JSON ada di Javascript, misal bentuk objek, array, string, dll.

Dengan demikian, jika sobat sudah terbiasa menggunakan Javascript maka tidak ada sesuatu yang baru disini, dan dengan mudah dapat mempelajari JSON.

Meski bentuknya mirip atau bahkan sama dengan Javascript, JSON tetap merupakan *language independent* artinya tidak terikat ke bahasa pemrograman manapun dan punya standar penulisan sendiri.

## Standar resmi....

JSON telah masuk kedalam standar ECMA Internasional, yaitu ECMA-404 Standar. Segala hal tentang JSON ada disana, mulai dari format penulisan, tipe data yang dapat digunakan, karakter yang diperbolehkan, dll.

ECMAScript sendiri merupakan standar scripting language, Standar ini digunakan juga oleh Javascript, sehingga dapat dikatakan bahwa Javascript dan JSON mengacu pada standar yang sama.

Dengan standar yang sama maka syntax penulisannya juga sama, inilah mengapa JSON dan Javascript Object memiliki format yang sama.

### Lalu, **apa pentingnya standar?**

Standar resmi ini penting, karena agar menjadi perhatian bagi setiap bahasa pemrograman agar bahasa mereka mensupport format ini, seperti misal php mulai versi 5.2.0 menyediakan fungsi untuk mengelola JSON seperti fungsi `json_encode`.

Selain itu, dengan masuk ke standar internasional, maka format tersebut dapat dipastikan dapat bertahan lama, tidak muncul terus hilang.

## Ekstensi dan Content Type

Ketika masuk ke standar Internasional, maka ditetapkan bahwa ekstensi JSON ini adalah `.json` dan mime typenya adalah `application/json`

Fungsi penetapan ekstensi ini salah satunya untuk memudahkan memilih default program untuk membuka file JSON, sedangkan content type ini salah satunya digunakan untuk HTTP header sehingga akan memudahkan browser mengenali data apa yang sedang di kirim, selain itu akan memudahkan kita mengenali file JSON (misal ketika melakukan scanning file dengan bahasa pemrograman tertentu).

Namun demikian, jika kita hanya fokus pada content, maka tidak masalah ekstensi apa yang kita gunakan, selama bentuk format datanya sesuai standar yang ditetapkan maka data JSON tersebut akan tetap valid.

## II. Struktur dan Tipe Data JSON

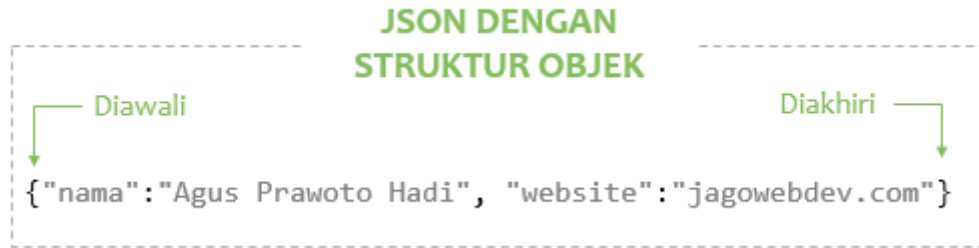
### 1. Struktur JSON

Jika dilihat dari strukturnya, JSON dibagi menjadi dua yaitu:

1. Bentuk pasangan `key:value` yang pada bahasa pemrograman Javascript disebut objek, atau dalam bahasa pemrograman lain seperti PHP disebut associative array.  
Contoh:

```
{"nama":"Agus Prawoto Hadi", "website":"jagowebdev.com"}
```

Selanjutnya, dalam JSON bentuk diatas **disebut objek** (Object). Perhatikan bahwa pada bentuk diatas, karakter pembuka dan penutup adalah kurung kurawa.

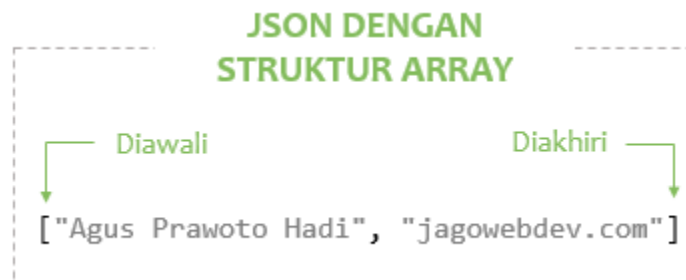


*JSON Dengan Struktur Objek*

2. Bentuk sederetan value saja (meskipun value tersebut juga dapat berbentuk object), yang dalam bahasa pemrograman Javascript dan PHP disebut array, contoh:

```
[ "Agus Prawoto Hadi", "jagowebdev.com" ]
```

Selanjutnya, dalam JSON bentuk diatas **disebut array**. Perhatikan bahwa pada bentuk diatas, karakter pembuka dan penutup yang digunakan adalah kurung siku.

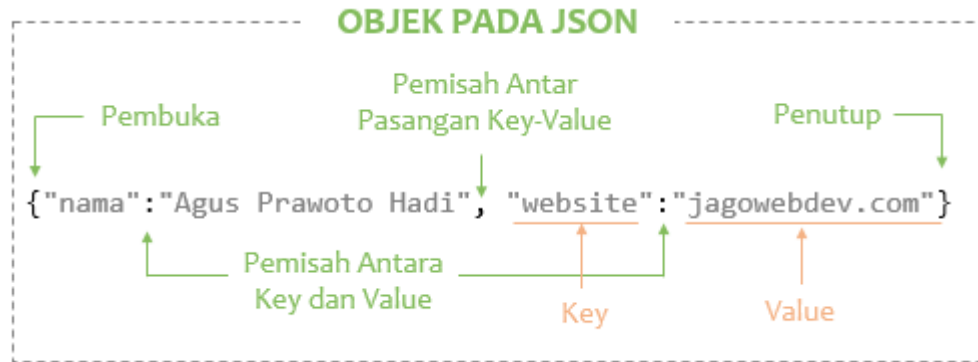


*JSON Dengan Struktur Array*

## 2. Tipe Data JSON

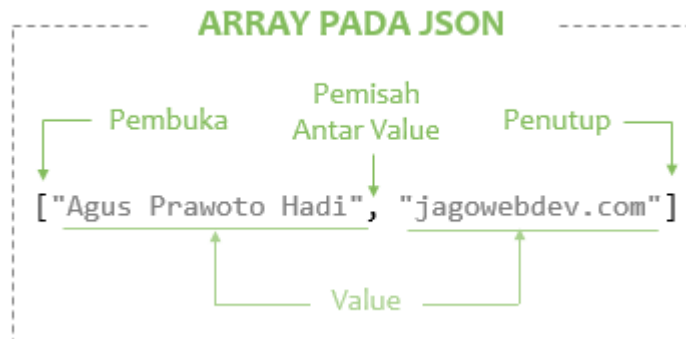
Maksud tipe data disini adalah jenis data yang dapat digunakan di dalam JSON, adapun tipe data yang dapat digunakan adalah sebagai berikut:

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 {}



*Objek Pada JSON*

2. **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 []



*Array Pada JSON*

3. **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
4. **Boelan.** Boelan berupa teks true dan false tanpa tanda kutip.
5. **Null.** Null berupa teks null dan ditulis langsung tanpa tanda kutip.
6. **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.

Contoh JSON dengan berbagai tipe data:

```
{
  "title": "Panduan Lengkap Memahami JSON",
  "author": {
    "id" : 1,
    "username" : "agusph",
    "nama": "Agus Prawoto Hadi"
  },
  "date": {
    "first_published" : "2018-01-10",
    "revision" : ["2018-01-31", "2018-01-15"]
  },
  "allow_comment" : true,
  "comment" : [
    {
      "id": 1,
      "name" : "Fulan",
      "date": "2018-02-01",
      "comment" : "..."
    },
    {
      "id": 2,
      "name" : "Abdullah",
      "date": "2018-02-05",
      "comment" : "..."
    }
  ],
  "pingback" : null
}
```

Pada contoh diatas, JSON berbentuk Objek, disamping itu di dalam JSON tersebut terdapat data berbentuk objek sehingga disebut nested object (object di dalam objek) dan array, untuk lebih jelasnya perhatikan gambar berikut:

## JSON DENGAN STRUKTUR OBJEK

```
{
  "title": "Panduan Lengkap Memahami JSON",
  "author": {
    "id": 1,
    "username": "agusph",
    "nama": "Agus Prawoto Hadi"
  },
  "date": {
    "first_published": "2018-01-10",
    "revision": ["2018-01-31", "2018-01-15"]
  },
  "allow_comment": true,
  "comment": [
    {
      "id": 1,
      "name": "Fulan",
      "date": "2018-02-01",
      "comment": "..."
    },
    {
      "id": 2,
      "name": "Abdullah",
      "date": "2018-02-05",
      "comment": "..."
    }
  ],
  "pingback": null
}
```

Diagram illustrating JSON structure with annotations:

- objek**: Points to the `"author"` object.
- boolean**: Points to the `allow_comment` value (`true`).
- array**: Points to the `revision` array (`["2018-01-31", "2018-01-15"]`).
- number**: Points to the `id` value (`2`) in the second comment object.
- null**: Points to the `pingback` value (`null`).

*Memahami JSON – Contoh Tipe Data JSON*

Selanjutnya, berikut ini contoh JSON dengan struktur array

```
[
  {
    "title": "Panduan Lengkap Memahami JSON",
    "author": {
      "id": 1,
      "username": "agusph",
      "nama": "Agus Prawoto Hadi"
    }
  },
  {
    "title": "Memahami JSON Pada PHP",
    "author": {
      "id": 1,
      "username": "agusph",
      "nama": "Agus Prawoto Hadi"
    }
  }
]
```

Pada contoh diatas juga terdapat beberapa nested object.

## II.1 Tipe Data STRING

Seperti disebutkan sebelumnya, salah satu tipe data yang sering digunakan pada JSON adalah string. Tipe data ini agak ribet ketika diterapkan pada JSON untuk itu perlu pembahasan tersendiri.

Harus menggunakan dua tanda kutip

Pada bagian sebelumnya telah disebutkan bahwa string harus diapit menggunakan dua tanda kutip, jika tetap memaksakan menggunakan satu tanda kutip maka JSON tersebut tidak dapat di parsing dengan baik oleh parser, sebagai contoh:

```
<?php
$json = '{"Judul' : 'Panduan Lengkap Memahami JSON'}';
$parse = json_decode($json, true);
print_r($parse); // Hasil: halaman kosong
```

Untuk itu selalu gunakan dua tanda kutip.

### Karakter Khusus

**Penting diperhatikan juga bahwa** jika di dalam string terdapat karakter khusus, maka agar karakter tersebut dapat terbaca perlu di-*escape* menggunakan back slash ( \ ). Adapun karakter khusus tersebut adalah:

Karakter	Escaped	Nama Karakter
“	\”	Tanda kutip dua
/	\/	Solidus (slash)
\	\\	Reverse solidus (Backslash)
r	\r	Cariage return
n	\n	Line feed
t	\t	Tabulation (tab)
b	\b	Backspace
f	\f	Form feed



Dalam praktik, karakter khusus yang sering digunakan adalah tanda kutip dua, slash, dan backslash, contohnya adalah sebagai berikut:

```
{
  "file": "logo_json.jpg"
  , "path" : "\\public\\img\\"
  , "permalink" : "http:\\\\jagowebdev.com\\public\\img\\logo_json.png"
  , "judul" : "Official logo JSON"
  , "deskripsi" : "Official logo JSON. JSON merupakan kependekan dari \\\"Javascript Object Notation\\\""
}
```

### III. Implementasi JSON

Dalam praktik, format data JSON digunakan untuk berbagai keperluan, diantaranya untuk load halaman dengan sistem AJAX dan untuk file konfigurasi.

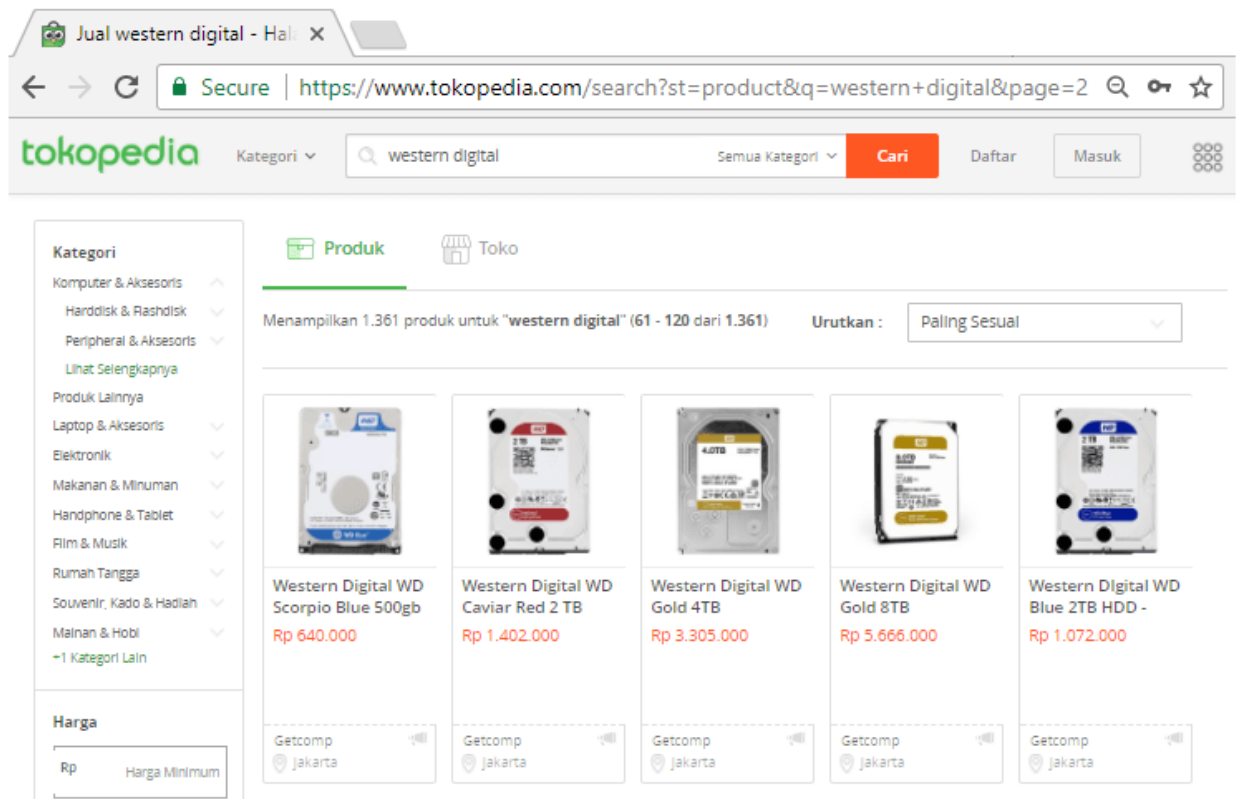
#### 1. AJAX

JSON ini lebih banyak digunakan dalam teknologi AJAX dimana untuk merubah content tidak perlu merefresh halaman, hal ini lebih disebabkan karena ukuran teks data yang kecil dan sudah tersedia built in parser yang ada di Javascript.

Sebagai contoh untuk pagination. Ketika meload halaman berikutnya, aplikasi cukup mengambil data-data teksnya saja, tanpa perlu meload satu halaman penuh dokumen HTML, misal ketika kita buka url ini:

<https://www.tokopedia.com/search?st=product&q=western+digital&page=1>

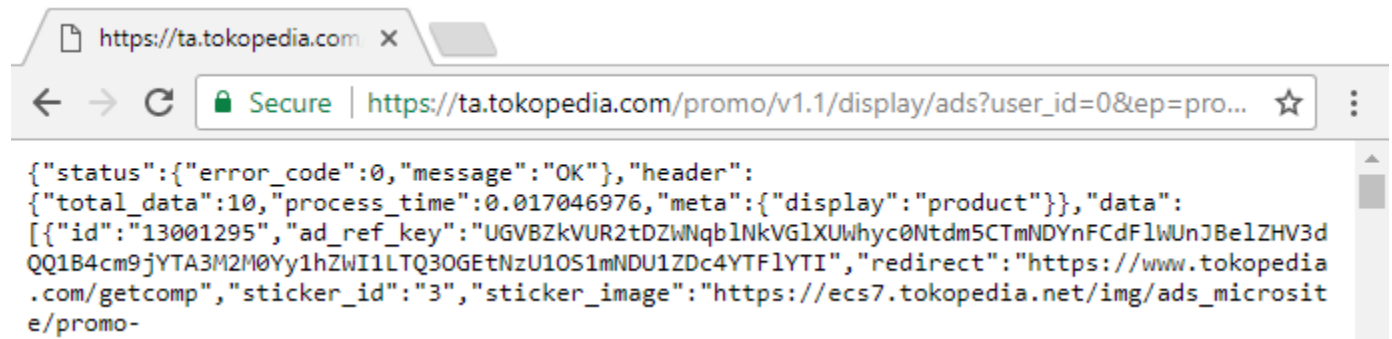
kemudian klik tombol page 2 sehingga url berubah menjadi <https://www.tokopedia.com/search?st=product&q=western+digital&page=2> tampilan yang kita peroleh adalah sebagai berikut:



*Tampilan Tokopedia Untuk Pencarian Western Digital – 2018-01-20*

Jika diperhatikan, ketika berpindah ke page 2, halaman yang ada tidak ter-refresh, hanya item produk saja yang berganti.

Mengapa demikian? ketika mengklik tombol page 2, di belakang layar, aplikasi mengambil banyak sekali data JSON yang salah satunya daftar produk. Adapun bentuk JSON nya adalah sebagai berikut:



*Penggalan JSON dari Tokopedia.com*

## Source JSON

Selanjutnya data JSON tersebut diolah sehingga yang tampil di halaman web adalah gambar produk beserta deskripsinya.

Demikian juga ketika kita ingin membagi data kita ke pihak lain, misal, dengan PHP saya buat data JSON untuk menampilkan data 5 artikel terakhir yang ada di jagowebdev.com.

Adapun data JSON yang tampil adalah sebagai berikut (untuk satu artikel):

```
[
  {
    "post_title": "Memahami Tipe Data Pada PHP",
    "excerpt": "Tutorial ini membahas segala yang perlu anda ketahui tentang tipe data pada PHP. Disamping teori, juga di berikan contoh penerapannya dilapangan",
    "category": "PHP Tutorial",
    "post_date": "2017-12-02 05:11:03",
    "post_url": "http://jagowebdev.com/tipe-data-pada-php/",
    "author": {
      "link": "http://jagowebdev.com/author/agusph/",
      "name": "Agus Prawoto Hadi"
    },
    "feature_image": {
      "alt": "Tipe Data Pada PHP",
      "url": "http://jagowebdev.com/wp-content/uploads/2017/12/Tipe-Data-Pada-PHP.png"
    }
  }
]
```

Untuk data JSON lengkap bisa dilihat disini: [Jagowebdev Last Article](#)

Dari kedua contoh diatas tidak ada yang berekstensi `.json` tetapi format datanya tetap standar JSON dan hal tersebut sah sah saja.

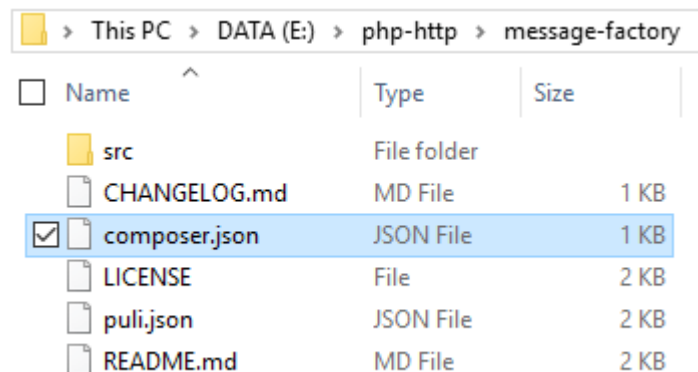
Note: Lebih jauh tentang implementasi JSON pada PHP akan kita bahas pada artikel tersendiri.

## 2. File Konfigurasi

Contoh lain penggunaan JSON adalah untuk file konfigurasi.

Pada bentuk ini, biasanya file di share ke pengguna lain dengan ekstensi `.json` sehingga dapat langsung diketahui bahwa data yang ada di dalam file tersebut berformat JSON.

Sebagai contoh, berikut ini file `.json` yang ketika diperoleh ketika menginstall package menggunakan composer:



This PC > DATA (E:) > php-http > message-factory		
<input type="checkbox"/> Name	Type	Size
src	File folder	
CHANGELOG.md	MD File	1 KB
<input checked="" type="checkbox"/> composer.json	JSON File	1 KB
LICENSE	File	2 KB
puli.json	JSON File	2 KB
README.md	MD File	2 KB

*Contoh File Composer.json*

Isi file tersebut adalah sebagai berikut:

```
{
  "name": "php-http/message-factory",
  "description": "Factory interfaces for PSR-7 HTTP Message",
  "license": "MIT",
  "keywords": ["http", "factory", "message", "stream", "uri"],
  "homepage": "http://php-http.org",
  "authors": [
    {
      "name": "Márk Sági-Kazár",
      "email": "mark.sagikazar@gmail.com"
    }
  ]
}
```

```

],
"require": {
    "php": ">=5.4",
    "psr/http-message": "^1.0"
},
"autoload": {
    "psr-4": {
        "Http\\Message\\": "src/"
    }
},
"extra": {
    "branch-alias": {
        "dev-master": "1.0-dev"
    }
}
}

```

atau misal kita membuat file `.json` sendiri untuk konfigurasi website kita, file tersebut kita beri nama `config.json` dengan data sebagai berikut:

```

{
    "database" : {
        "host" : "localhost",
        "user" : "root",
        "pass" : "Pas123"
    },
    "theme" : {
        "color_scheme" : "blue",
        "font" : {
            "family" : "open sans",
            "size" : "14px",
            "color" : "#000000"
        }
    }
}

```

## IV. JSON Viewer dan JSON Formatter

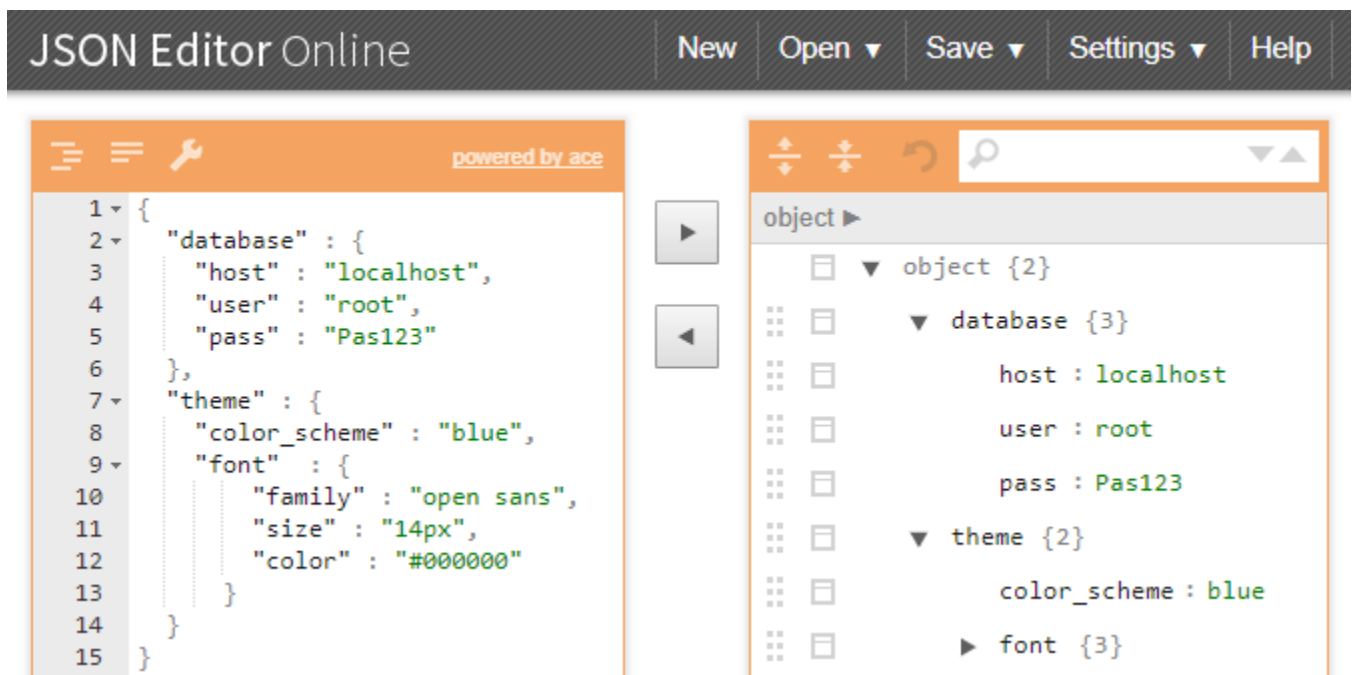
### 1. JSON Viewer

Pada JSON dengan content sederhana, kita dapat dengan mudah mengidentifikasi bentuk susunan data pada JSON tersebut, namun jika bentuknya kompleks dan berjenjang, kita akan kesulitan membacanya, untuk itu kita perlu bantuan tools seperti JSON Viewer

### JSON berupa string

Untuk JSON berbentuk string, kita dapat menggunakan tools dengan JSON Viewer yang ada di <http://jsoneditoronline.org/> atau <https://codebeautify.org/jsonviewer>

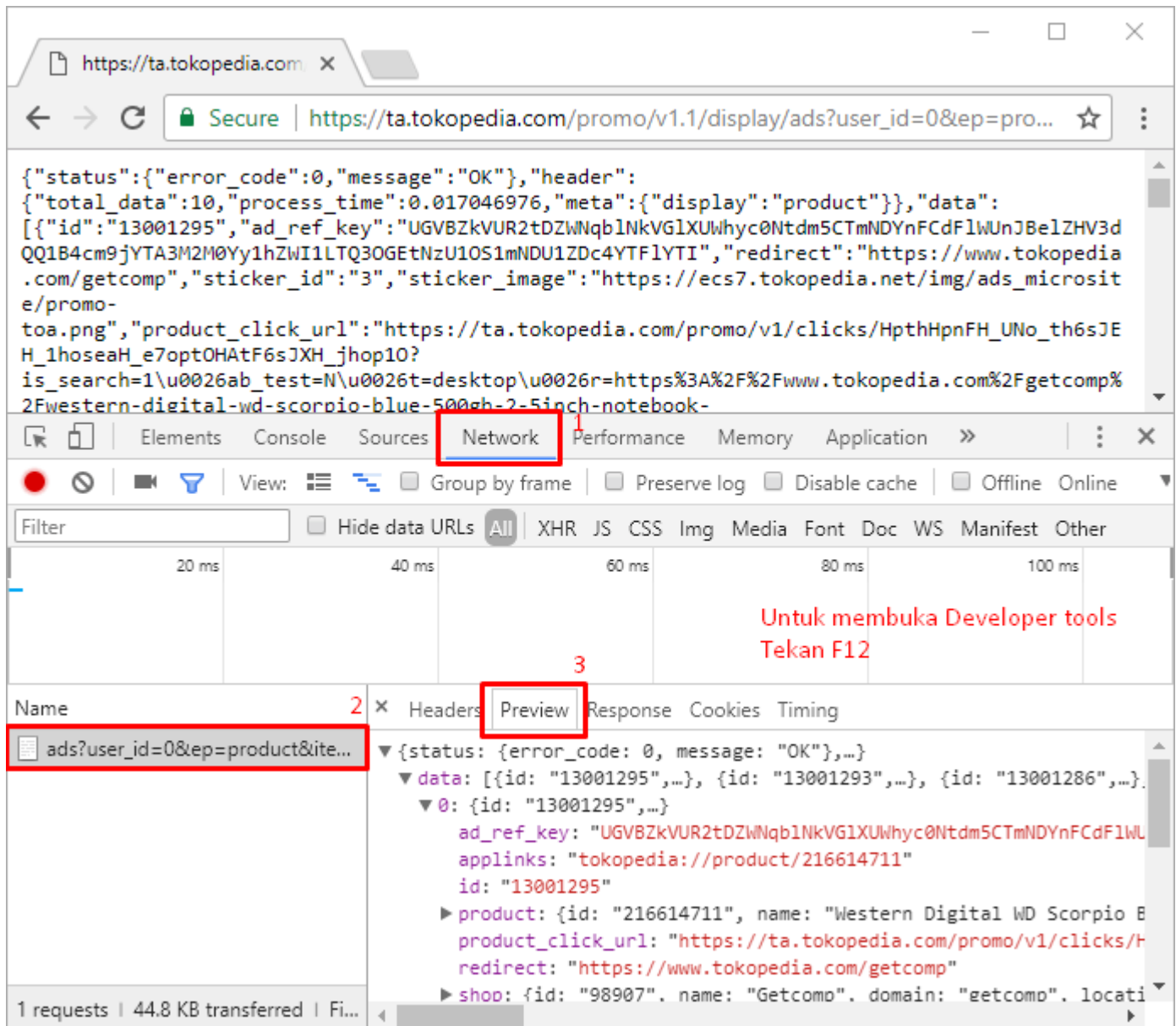
Pada website tersebut, kita tinggal copy paste string JSON yang ada dan JSON Tree nya dapat langsung terlihat, contoh:



JSON Viewer – Jsoneditoronline.org

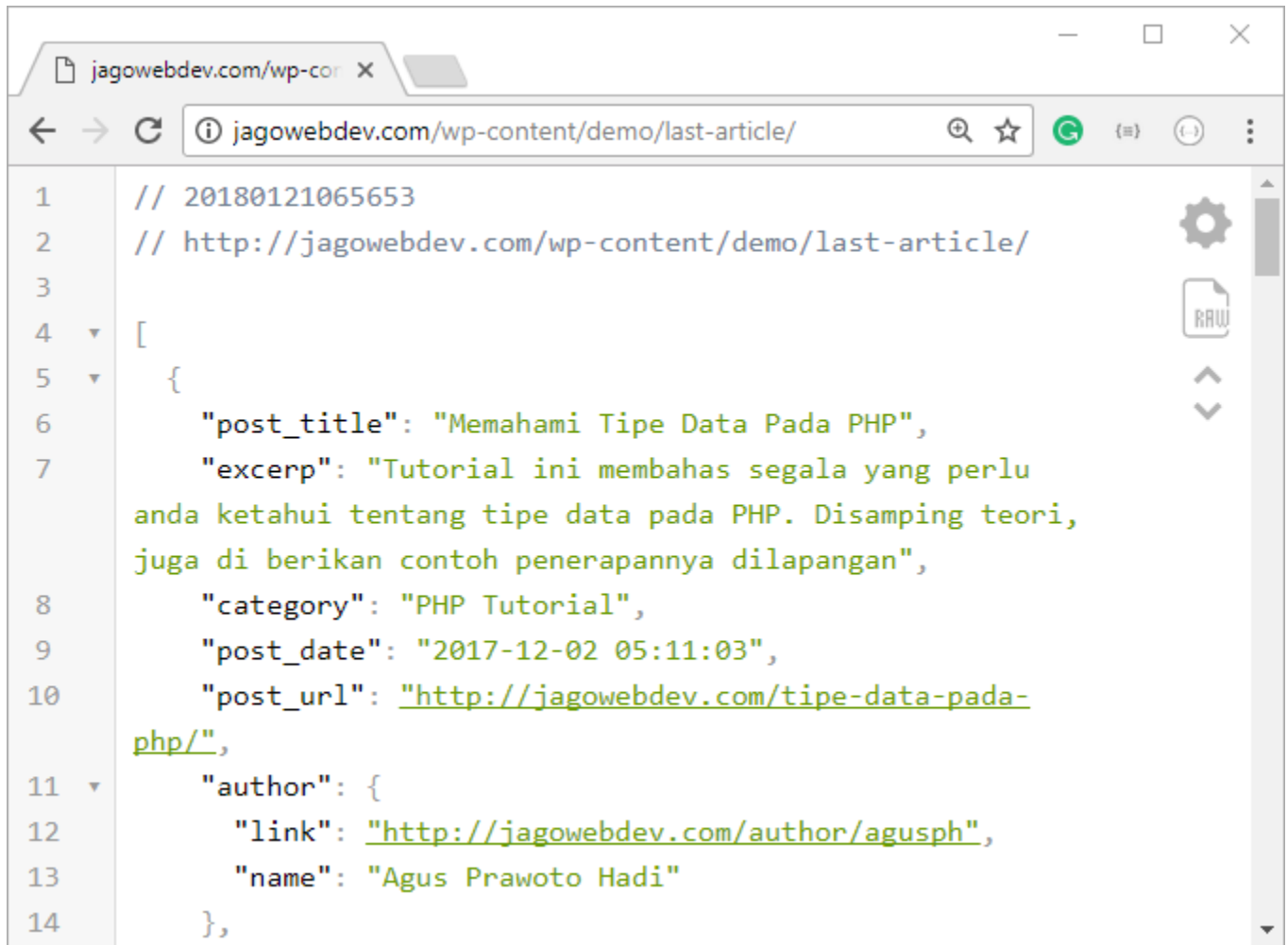
## JSON Via URL

Jika data JSON tersebut dapat di load dengan browser, misal seperti pada Tokopedia tadi: [Source JSON](#), maka kita dapat melihat JSON Tree nya melalui developer tools pada browser, contoh:



JSON Tree Pada Browser Google Chrome

atau bisa menggunakan add-on browser seperti [JSON Viewer](#) pada browser google chrome. Dengan add on ini, ketika kita membuka halaman yang isinya FULL data JSON, maka data tersebut otomatis akan diubah menjadi bentuk tree, seperti contoh berikut:



*JSON Tree Dengan Add On Browser Google Chrome*

## 2. JSON Formatter

Untuk keperluan tertentu kita ingin memformat data JSON menjadi bentuk yang lebih mudah dibaca (ada cukup indentasi dan spasi). Untuk keperluan tersebut kita perlu tools seperti JSON Formatter.



Tools online yang dapat digunakan

adalah <https://codebeautify.org/jsonviewer>, <https://jsonlint.com/>,

atau <https://jsonformatter.org/> sedangkan untuk offline, bisa menggunakan browser extension, seperti [JSON Viewer](#) pada Google Chrome

## V. Apakah Saya Harus Menggunakan JSON?

JSON hanya merupakan tools untuk memudahkan pertukaran data, jika sobat ingin menggunakan format lain tidak masalah, seperti misal: [Protocol Buffer](#) atau [Message Pack](#) atau memiliki format yang dibuat sendiri yang digunakan bersama team anda.

Yang penting adalah format tersebut sudah teruji dan didukung oleh bahasa pemrograman yang sobat gunakan, artinya bahasa pemrograman tersebut menyediakan fungsi (baik bawaan maupun extension) yang dapat digunakan untuk (1) building: membuat data sesuai format dimaksud maupun (2) parsing: meng ekstraknya kembali menjadi data yang dapat digunakan oleh bahasa pemrograman tersebut.

## VI. Tambahan: Objek JSON vs Objek Javascript

Mungkin sobat ada yang bertanya tanya apa bedanya objek pada JSON dengan objek pada Javascript karena jika diperhatikan bentuk dan strukturnya sama persis.

Yup betul keduanya sama persis, tetapi... tipe keduanya berbeda, sekali lagi JSON ini adalah teks biasa, sehingga objek pada JSON juga merupakan string (teks) dan sekali lagi karena JSON mengadopsi ECMAScript – Javascript maka format yang digunakan juga akan sama. Perhatikan contoh berikut:

```
<html>
<script type="text/javascript">
// Objek
var artikel = {"judul": "Panduan Lengkap Memahami JSON",
"permalink": "http://jagowebdev.com/panduan-lengkap-memahami-json", "penulis": "Agus Prawoto Hadi", "date": "2018-01-31"};
document.write(artikel.judul); // Hasil: Panduan Lengkap Memahami JSON
```

```
// JSON
var artikel = '{"judul": "Panduan Lengkap Memahami JSON",
"permalink": "http://jagowebdev.com/panduan-lengkap-memahami-json", "penulis": "Agus
Prawoto Hadi", "date": "2018-01-31"}';
document.write(artikel.judul); // Hasil: Undefined
</script>
</html>
```

Dari contoh diatas terlihat karena JSON berbentuk string, maka key nya tidak bisa langsung diakses, sehingga menghasilkan output `Undefined`, begitu juga dengan array:

```
<html>
<script type="text/javascript">
// Array (tipe data)
var penulis = ["Agus Prawoto Hadi", "jagowebdev.com"];
document.write(penulis[0]); // Hasil: Agus Prawoto Hadi

// Array JSON (string)
var penulis = '["Agus Prawoto Hadi", "jagowebdev.com"]';
document.write(penulis[0]); // Hasil: [
</script>
</html>
```

## VII. Tambahan: JSON vs XML?

Meskipun pembahasan kita kali tentang JSON, ada baiknya kita sedikit membahas tentang XML, kenapa? Karena XML sangat populer dan lebih dulu digunakan untuk format pertukaran data dibanding JSON.

Adapun perbedaan keduanya adalah sebagai berikut:

### 1. Ukuran JSON Lebih Kecil dibanding XML

Dilihat dari ukuran file nya, JSON lebih kecil hingga 30% dibanding XML, hal ini dikarenakan untuk menyediakan informasi sama, jumlah karakter yang digunakan XML lebih banyak dibanding JSON

Karena keduanya berupa teks, maka ukuran file (dalam byte) dihitung berdasarkan banyaknya karakter yang digunakan, misal: bandingkan kedua bentuk berikut ini:

JSON (95 byte)

```
{"judul": "Panduan Lengkap Memahami JSON", "penulis": "Agus Prawoto Hadi",  
"date": "2018-01-25"}
```

XML (168 byte)

```
<?xml version="1.0" encode="utf-8">  
<article>  
  <title>Panduan Lengkap Memahami JSON</title>  
  <author>Agus Prawoto Hadi</author>  
  <date>2018-01-31</date>  
</article>
```

Dari perbandingan diatas, terlihat jelas bahwa xml menggunakan lebih banyak karakter sehingga mengakibatkan ukuran juga menjadi lebih besar.

Perbedaan ukuran inilah yang menjadi salah satu alasan kenapa banyak yang memilih JSON, terutama ketika menggunakan protokol http (internet), karena semakin kecil data yang di transfer, semakin cepat load datanya dan semakin kecil bandwidth yang digunakan.

Meskipun pada XML penulisan value dapat ditempatkan pada atribut element sehingga dapat memperkecil ukuran file, misal seperti ini:

XML (130 byte)

```
<?xml version="1.0" encode="utf-8">  
<article title="Panduan Lengkap Memahami JSON" author="Agus Prawoto Hadi" date="2018-01-31"/>
```

namun hal tersebut tidak umum digunakan dan memiliki banyak keterbatasan.

## 2. Dukungan Bahasa Pemrograman

**Parsing** (mengakses element JSON dan XML)

Untuk parsing data JSON maupun XML sudah didukung oleh berbagai bahasa pemrograman seperti Javascript dan PHP. Untuk Javascript, XML diperlakukan seperti element HTML sehingga model parsingnya sama seperti parsing element HTML (menggunakan DOM)

Sedangkan untuk JSON, Javascript dapat langsung mengubahnya ke tipe objek (terdapat fungsi bawaan), sehingga dapat langsung diperlakukan seperti objek Javascript pada umumnya.

### **Building** (membuat JSON dan XML)

Pada Javascript, membuat dokumen JSON sangat mudah karena sudah terdapat built in function yaitu `JSON.Stringify` atau dengan bantuan library JQuery.

Sedangkan untuk membuat dokumen XML agak ribet karena sama seperti membuat dokumen HTML sehingga perlu menjalankan berbagai fungsi yang berkaitan dengan DOM.

Dengan demikian, dapat disimpulkan bahwa pada Javascript, JSON lebih mudah digunakan dibanding XML.

## Penutup

Saat ini JSON menjadi format standar untuk pertukaran data dan lebih banyak digunakan untuk pertukaran data via internet (protokol HTTP) karena load time yang lebih cepat dan bandwidth yang diperlukan lebih kecil.

Dengan **struktur yang simpel**, format JSON lebih mudah digunakan dibanding format yang lain, terutama jika data yang di pertukarkan jumlahnya besar.

Kita akan lebih banyak merasakan manfaat JSON ketika menggunakannya bersama Javascript, seperti singkatannya (**Javascript** Object Notation).

Demikian pembahasan mengenai “Memahami JSON” semoga dapat bermanfaat.