**JSON: Pengertian, Fungsi dan Cara Menggunakannya**

JSON adalah turunan JavaScript yang digunakan dalam transfer dan penyimpanan data. Kekinian, bahasa ini sering dimanfaatkan dalam pembuatan aplikasi web.

## **Apa Itu JSON?**

Seperti yang tadi disebutkan, JavaScript object notation atau JSON adalah format yang digunakan untuk menyimpan dan mentransfer data.

Berbeda dengan XML (extensive markup language) dan format lainnya yang memiliki fungsi serupa, JSON memiliki struktur data yang sederhana dan mudah dipahami. Itulah mengapa JSON sering digunakan pada [API](https://www.niagahoster.co.id/blog/api-adalah/).

JSON sendiri terdiri dari dua struktur, yaitu:

* Kumpulan value yang saling berpasangan. Dalam JSON, contohnya adalah object.
* Daftar value yang berurutan, seperti array.

JSON juga tidak hanya dapat digunakan dengan [JavaScript](https://www.niagahoster.co.id/blog/belajar-javascript/). Berikut ini adalah beberapa bahasa pemrograman lain yang mendukung penggunaan format tersebut:

* PHP
* Python
* Ruby
* C++
* Perl

## **Fungsi JSON**

Fungsi JSON kurang lebih sama dengan XML. Sebelum JSON diciptakan, format bahasa markup ini sering digunakan dengan JavaScript dalam AJAX (asynchronous JavaScript and XML) untuk penyimpanan dan perpindahan data.

JSON memiliki fungsi yang sama. Bedanya, ia memiliki beberapa keunggulan yang membuatnya lebih sering digunakan daripada XML, termasuk:

* File yang lebih ringan
* Struktur kode yang lebih sederhana
* Kode yang mudah dipahami oleh manusia

### **Perbandingan JSON dan XML**

Seberapa sederhanakah kode JSON jika dibandingkan dengan kode XML? Untuk mengetahuinya, Anda dapat memperhatikan kedua contoh di bawah ini. Contoh pertama adalah kode XML.

<users>

<user>

<username>Anton</username> <lokasi>Bandung</lokasi>

</user>

<user>

<username>Budi</username> <lokasi>Semarang</lokasi>

</user>

<user>

<username>Nana</username> <lokasi>Surabaya</lokasi>

</user>

<user>

<username>Jamal</username> <lokasi>Tangerang</lokasi>

</user>

</users>

Berikut ini adalah contoh kode JSON.

{"users": [

{"username" : "Anton", "lokasi" : "Bandung"},

{"username" : "Budi", "lokasi" : "Semarang"},

{"username" : "Nana", "lokasi" : "Surabaya"},

{"username" : "Jamal", "lokasi" : "Tangerang"}

] }

Dapat Anda lihat bahwa kode JSON lebih ringkas karena tidak memerlukan tag pembuka dan penutup. Dengan demikian, kodenya juga lebih mudah dipahami. Di samping itu, kelebihan ini juga memberikan dua manfaat lainnya, yaitu:

* Proses loading data yang lebih ringan karena ukuran file yang kecil.
* Penulisan kode yang lebih cepat dengan kode yang sederhana.

## **Cara Menggunakan JSON**

### **Syntax JSON**

Pada contoh di bagian sebelumnya memang menunjukkan bahwa JSON memiliki struktur kode yang sederhana. Namun, mari lihat contoh yang lebih simpel lagi agar Anda tahu apa saja yang ada dalam syntax-nya:

{"city":"New York", "country":"United States "}

JSON selalu dibuka dan ditutup dengan tanda {} atau kurung kurawal. Syntax-nya terdiri dari dua elemen, yaitu  **key** dan **value**. Keduanya dipisahkan oleh titik dua agar jelas.

Contoh **key** adalah “city”, sedangkan “New York” adalah contoh **value**. Keduanya selalu dibuka dan ditutup dengan tanda kutip ganda. Perlu Anda ingat bahwa kedua tidak ada spasi di antara kedua elemen tersebut.

Apabila ada lebih dari satu pasang key dan value, Anda perlu memisahkannya dengan tanda koma yang diikuti spasi. Ini dapat Anda lihat pada contoh di atas.

Nah, meskipun **key** harus berupa string, Anda dapat menggunakan beberapa jenis data sebagai **value**-nya. Apa saja itu? Mari kita bahas di bagian selanjutnya.

### **Jenis-Jenis Value JSON**

Setidaknya ada enam jenis data yang dapat digunakan sebagai value JSON, yaitu:

* String
* Object
* Array
* Boolean
* Number
* Null

#### **String**

String adalah data yang terdiri dari karakter unicode, seperti “Anton” pada contoh di bawah ini.

“nama”:”Anton”

#### **Object**

Object adalah sepasang key dan value. Seperti pada contoh syntax tadi, object dibuka dan ditutup dengan kurung kurawal. Jika ada lebih dari satu object, masing-masing dipisahkan oleh koma dan spasi.

“karyawan”: {“nama”:”Anton”, “asal”:”Bandung”}

Pada contoh di atas, semua yang ada di dalam kurung kurawal adalah object yang merupakan value dari “karyawan”.

#### **Array**

Array adalah kumpulan object. Kumpulan ini dibuka dan ditutup dengan kurung siku [].

"karyawan":[

{"nama":"Anton", "asal":"Bandung"},

{"nama":"Nana", "asal":"Surabaya"},

{"nama":"Jamal", "asal":"Tangerang"}

]

#### **Boolean**

Boolean adalah jenis data yang hanya berisikan pernyataan benar atau salah. Anda hanya tinggal memasukkan true atau false, seperti pada contoh berikut ini:

“karyawan tetap”:”false”

#### **Number**

Seperti namanya, jenis data ini berupa angka saja. Dengan catatan, angkanya harus merupakan integer atau angka bulat. Artinya, 21,8 atau √2 tidak dapat dijadikan sebagai value . Contoh penggunaannya seperti di bawah ini:

“usia”:“29”

#### **Null**

Jika sebuah key tidak memiliki value, Anda dapat mengetikkan null. Contohnya seperti pada kode berikut:

“golonganDarah”:”null”

## **Membuat Nested Objects dan Nested Arrays**

Anda dapat membuat nested object dan nested array dalam kode JSON. Untuk memahami penggunaannya, amati kedua contoh yang dibahas pada bagian ini.

Pertama, mari perhatikan contoh nested object berikut:

“karyawan”: {

“nama”:“Anton”,

“asal”:“Bandung”.

“hobi”: {

“hobi1”:“berenang”,

“hobi2”:“melukis”,

“hobi3”:“jogging”

}

}

Pada contoh tersebut, “hobi1”, “hobi2”, dan “hobi3” merupakan value dari key “hobi”. Perhatikan bahwa ketiga value tersebut ada di dalam kurung kurawal. Dengan demikian, “hobi” beserta ketiga value-nya merupakan object di dalam object.

Penggunaan nested array tidak jauh berbeda. Perhatikan contoh di bawah ini:

“karyawan”: {

“nama”:“Anton”,

“asal”:“Bandung”.

“hobi”: [ “berenang”, “melukis”, “jogging” ]

}

Pada contoh di atas, “berenang”, “melukis”, dan “jogging adalah array yang merupakan value dari “hobi”.