

## MODUL VIII-PHP-XML-JSON

### 8.1 Tujuan

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat:

1. Memahami fungsi dan struktur dokumen berjenis XML.
2. Memahami fungsi dan struktur dokumen berjenis JSON.
3. Mampu membuat file berjenis PHP yang dapat mengakses dokumen baik yang jenis XML maupun berjenis JSON.

### 8.2 Pengantar XML

XML Kependekan dari eXtensible Markup Language, dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998 teknologi yang digunakan XML sebenarnya bukan teknologi baru, tapi turunan dari SGML yang dikembangkan pada awal 80-an dan telah banyak digunakan pada dokumentasi teknis proyek-proyek berskala besar.

Ketika HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian penting pada SGML dan dengan berpedoman pada pengembangan HTML menghasilkan markup language yang tidak kalah hebatnya dengan XML. Seperti halnya HTML, XML juga menggunakan elemen yang ditandai dengan Tag pembuka dan penutup ( diawali dengan '<' dan diakhiri dengan '>'). Tag penutup ( diawali dengan '</' dan diakhiri dengan '>' ).

XML adalah bahasa markup yang berisikan kode-kode berupa parameter tertentu disesuaikan dengan aturan penulisannya untuk memformat dokumen text dengan tag sendiri agar bisa dapat dimengerti. XML menyediakan format untuk mendeskripsikan data terstruktur.

### 8.3 Keuntungan Menggunakan XML

XML untuk saat ini bukan merupakan pengganti HTML. Masing-masing dikembangkan untuk tujuan yang berbeda. Kalau HTML digunakan untuk menampilkan informasi dan berfokus pada bagaimana informasi terlihat, XML mendeskripsikan susunan informasi yang terstruktur dan berfokus pada informasi itu sendiri. XML terutama dibutuhkan untuk menyusun dan menyajikan informasi dengan format yang tidak mengandung format standar layaknya heading, paragraph, table dan lain sebagainya. Berikut ini adalah keuntungan-keuntungan dalam penggunaan XML antara lain:

1. Ekstensibilitas : dapat ditukar /digabung dengan dokumen XML lain.
2. Memungkinkan pemrograman yang lebih baik : maka dibuat suatu software pengolah XML.
3. Memisahkan data dan presentasi. Yang akan direpresentasikan dalam XML dan XSLT.
4. Pencarian data cepat karena XML merupakan data dalam format yang terstruktur.
5. XML berfungsi sekali untuk pertukaran data.

## 8.4 EXTENSIBLE STYLESHEET LANGUAGE:TRANSFORMATION (XSLT)

XSLT adalah kependekan dari eXtensible StyleSheet Language:Transformation, adalah bagian dari XSL yang dikembangkan sebelumnya. XSL adalah Stylesheet yang khusus dikembangkan sebagai komplemen XML, untuk merubah informasi pada XML ke dalam bentuk lain agar bisa ditampilkan di layar.

Pada dasarnya proses ini di bagi menjadi dua bagian proses yakni pertama Transformasi Struktural yang meliputi pengumpulan, pengelompokan dan pengurutan data maupun penyusunan ulang, penambahan dan penghapusan tag dan atribut, dan yang kedua adalah proses merubah format menjadi pixel di layar. Proses yang pertama itulah yang kemudian disebut XSLT, sedangkan yang kedua biasa disebut XSLFO (eXtensible StyleSheet Language:Formatting Object).

## 8.5 XSLT PROCESSOR

XSLT Processor atau yang biasa disebut Parser adalah software bantu yang tugasnya menerapkan perintah-perintah dalam XSLT pada dokumen sumber XML, dan menghasilkan dokumen keluaran baik berupa HTML, Text file ataupun XML.

## 8.6 STRUKTUR XML

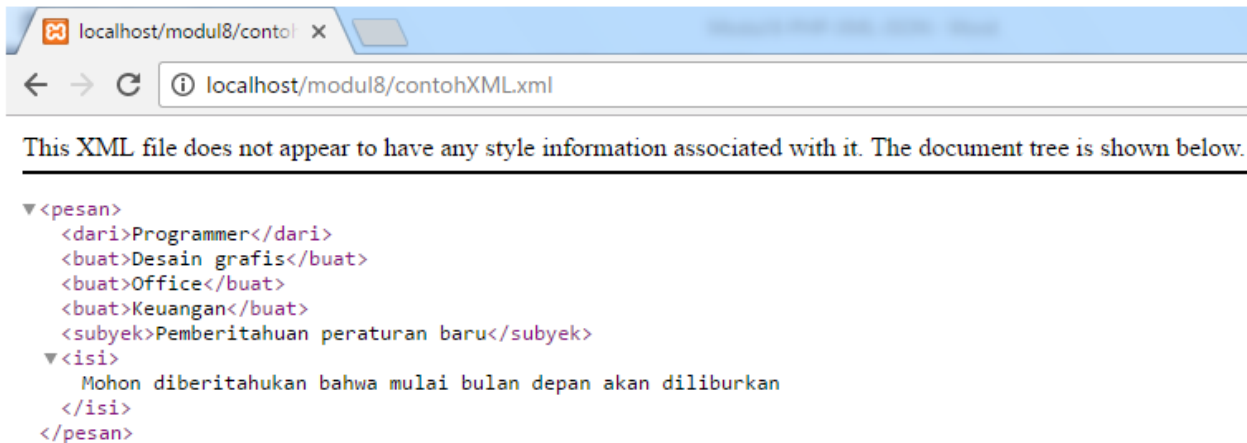
Seperti halnya dengan HTML, XML juga menggunakan elemen yang sama dengan HTML diawali dengan tag pembuka (diawali dengan '<' dan diakhiri dengan '>'), tag penutup (diawali dengan '</' dan diakhiri '>') dan atribut elemen (parameter yang dinyatakan dalam tag pembuka misal <form name="isidata">). Hanya bedanya, HTML mendefinisikan dari awal tag dan atribut yang dipakai didalamnya.

Berikut contoh dokumen XML:

### contohXML.xml

```
1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <pesan>
3   <dari>Programmer</dari>
4   <buat>Desain grafis</buat>
5   <buat>Office</buat>
6   <buat>Keuangan</buat>
7   <subyek>Pemberitahuan peraturan baru</subyek>
8   <isi>Mohon diberitahukan bahwa mulai bulan depan akan diliburkan</isi>
9 </pesan>
10
```

Buka file XML di atas melalui browser:



## 8.7 APLIKASI PHP AJAX & XML

Pada bagian ini kita akan membuat contoh aplikasi AJAX dan XML, dimana pada aplikasi web AJAX akan meminta respon berdasarkan dari File XML yang telah kita buat, tujuan dari pengembangan web adalah untuk menciptakan sebuah web yang interaktif dan dinamis.

### MEMBUAT FILE INDEX.PHP

---

```
<html>
<head>
<title>Aplikasi PHP AJAX XML</title>
<script type="text/javascript" src="createObject.js">
</script>
<script language="javascript">
function requestContent(){
oRequest = createObject();
var url = "contohData.xml";
oRequest.open("GET",url true);
oRequest.onreadystatechange = function() {
document.getElementById("divContent").innerHTML="<div align='center'>Menunggu
Respon server</div>";
if (oRequest.readyState==4) {
document.getElementById("divContent").innerHTML = "";
var xmlData = oRequest.responseXML;
var photoList = xmlData.getElementsByTagName("photo");
for (var i=0; i < photoList.length; i++){
var currentPhoto = photoList[i];
var newThumb = document.createElement("div");
newThumb.setAttribute("className", "thumbnail");
```

```

newThumb.setAttribute("class", "thumbnail");
var newNama = document.createElement("div");
newNama.setAttribute("className", "nama");
newNama.setAttribute("class", "nama");
var nama_ = document.createTextNode(currentPhoto.
getElementsByTagName("nama")[0].firstChild.data);
newNama.appendChild(nama_);
var newFile = document.createElement("div");
newFile.setAttribute("className", "photo");
newFile.setAttribute("class", "photo");
var file_ = "<img
src='"+currentPhoto.getElementsByTagName("file")[0].firstChild.data+"'>";
newFile.innerHTML= file_;
var newKeterangan = document.createElement("div");
newKeterangan.setAttribute("className", "keterangan");
newKeterangan.setAttribute("class", "keterangan");
var keterangan_ =document.createTextNode(
currentPhoto.getElementsByTagName("keterangan")[0].firstChild.data);;
newKeterangan.appendChild(keterangan_);
newThumb.appendChild(newNama);
newThumb.appendChild(newFile);
newThumb.appendChild(newKeterangan);
document.getElementById("divContent").
appendChild(newThumb);
}
}
oRequest.send(null);
}
}
</script>

```

```

</head>
<body>
<table width="748" cellpadding="0" cellspacing="0"
align="center" style="border:1px solid:#cccccc">
<tr>
<td colspan="2" style="border-bottom:1px solid #cccccc;">
</td>
</tr>
<tr>
<td width="200" height="291" valign="top" bgcolor="#F1F4F6"
style="border-bottom:1px solid #cccccc;">
<table width="100%" border="0" cellpadding="0"
cellspacing="0" bgcolor="#F1F4F6">
<tr>
<td valign="top" bgcolor="#F1F4F6">

```

```

<div>
<marquee direction=up scrollamount="3" height="187" width="170">
<p>XML (Extensible Markup Language) adalah bahasa Markup
untuk keperluan umum yang disarankan W3C untuk membuat dokumen markup keperluan
khusus.
Keperluan utama XML adalah untuk pertukaran data antar sistem yang beranekaragam.
Perlu diingat, AJAX akan berkomunikasi dengan server tanpa harus
merefresh Semua Halaman, Konsep ini berbeda dengan aplikasi
web sistem tradisional. Ajax adalah aplikasi web yang lebih
baik. Aplikasi web menambah keuntungan dibanding aplikasi
dekstop. dengan demikian, untuk mempelajari ajax, harus
pahami konsep standar.</p>
</marquee>
</div>
</td>
</tr>
</table>
</td>
<td valign="top" width="546" bgcolor="#F1F4F6"
style="border-bottom:1px solid #cccccc;border-left:1px
solid #cccccc">
<table width="100%" border="0" cellpadding="0"
cellpadding="0" style="border-bottom:1px solid #cccccc;">
<tr>
<td><span class="style2">Homepage</span></span></td>
</tr></table>
<table width="100%">
<tr>
<td>
<div>
<p><strong>RESPON AJAX DAN XML </strong></p>
<p>Berikut ini adalah Halaman web yang akan melakukan
permintaan request ke server untuk meminta menampilkan gambar
pada File XML. lalu gambar yang diminta akan tampil melalui
respon dari server berdasarkan permintaan.<br><br>
<a href="#" onclick="javascript:requestContent();">Click
disini untuk menampilkan gambar </a></p>
</tr>
<td>
<div id="divContent"></div>
</td>
</tr>
</div>
</td>
</tr>
</table>

```

```
</td>
</tr>
<tr>
</tr>
</table>
</body>
</html>
```

## MEMBUAT FILE XML (contohData.xml)

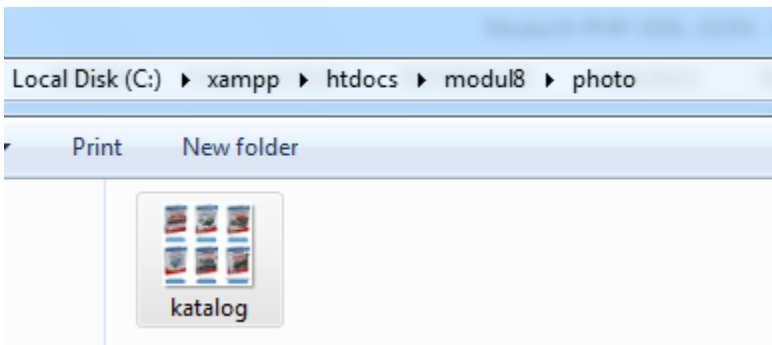
---

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xml id="photoBook">
  <photoList>
    <photo>
      <id>1</id>
      <nama>Katalog Buku</nama>
      <file>photo/katalog.jpg</file>
      <keterangan>Keterangan Buku disini</keterangan>
    </photo>
  </photoList>
</xml>
```

## MENYIAPKAN GAMBAR

---

Buat folder photo dan simpan satu file gambar, simpan dengan nama katalog.jpg



## MEMBUAT XMLHTTPREQUEST OBJECT

---

Pada tahap selanjutnya Anda akan mengembangkan aplikasi web dengan teknik AJAX beberapa hal yang harus diperhatikan :

1. Javascript untuk membuat object XMLHttpRequest yang akan digunakan untuk berkomunikasi dengan server.
2. DOM (Document Object Model), hasil proses yang diterima akan ditampilkan dengan

memanipulasi object DOM yang telah Anda buat sebelumnya untuk menampilkan data hasil respon yang diberikan server.

3. XML (eXtensible Markup Language) format data yang dikembalikan oleh server, data XML ini akan dibaca dan ditampilkan untuk mengupdate content pada halaman web.

Pada langkah berikut ini Anda akan membuat file JavaScript untuk membuat Object XMLHttpRequest pada halaman Web untuk mendeteksi Web browser yang digunakan.

createObject.js

```
function createObject(){
var oAJAX = false;
try{
oAJAX      =      new      ActiveXObject("Msxml2.XMLHTTP");
}catch(e){
try{
oAJAX =new ActiveXObject("Microsoft.XMLHTTP");
}catch(e2)  {
oAJAX      = false;
}
}
if      (!oAJAX      &&      typeof XMLHttpRequest      !=      'undefined')  {
oAJAX      = new XMLHttpRequest();
}
if      (!oAJAX){
alert("Error saat membuat XMLHttpRequest!");
}
return oAJAX;
}
```

Buka web browser untuk melihat hasilnya dengan mengetikkan URL:

<http://localhost/modul8/>

[Homepage](#)

Berikut ini adalah Halaman web yang akan melakukan permintaan request ke server untuk meminta menampilkan gambar pada File XML. lalu gambar yang diminta akan tampil melalui respon dari server berdasarkan permintaan.

[Click disini untuk menampilkan gambar](#)

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.

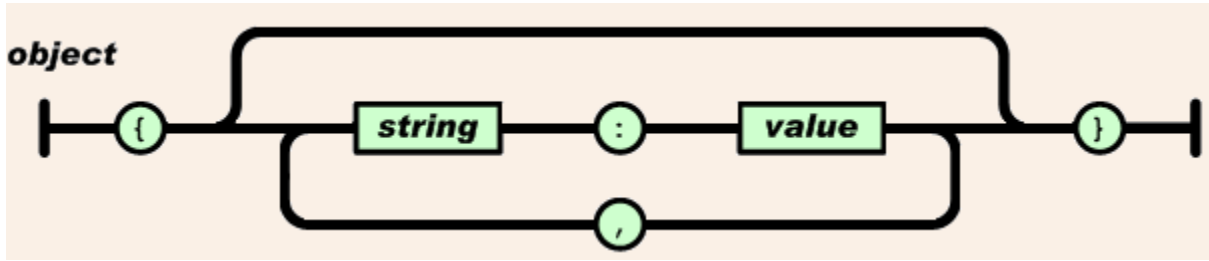
Beberapa orang lebih suka JSON, karena paling mudah untuk mem-parse-nya.

- 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).

JSON menggunakan bentuk sebagai berikut:

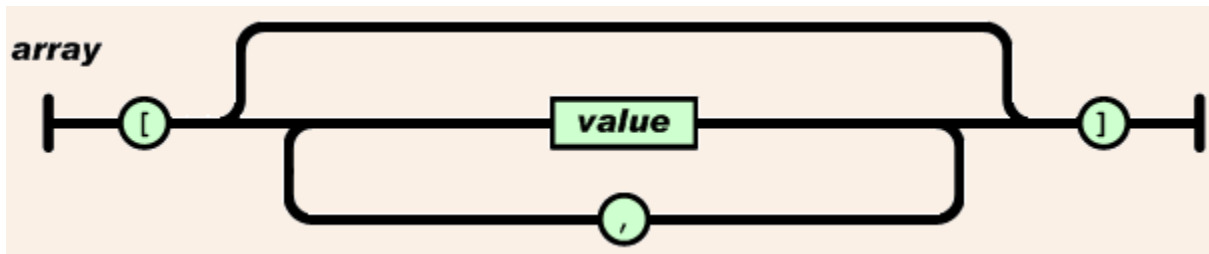


Object adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



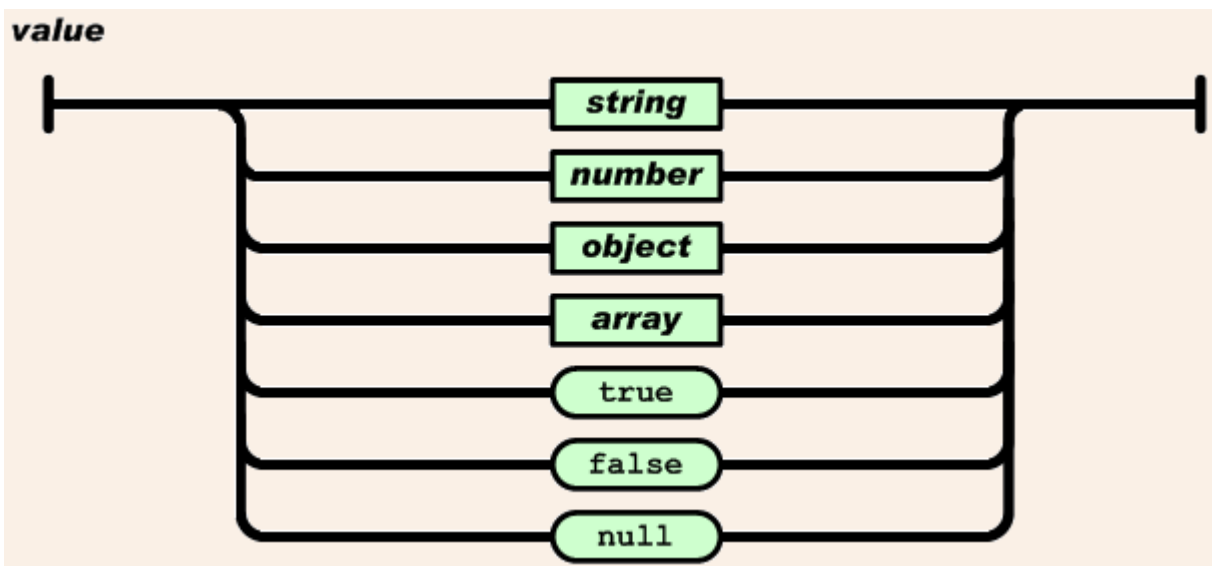
*Sumber : [www.json.org](http://www.json.org)*

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



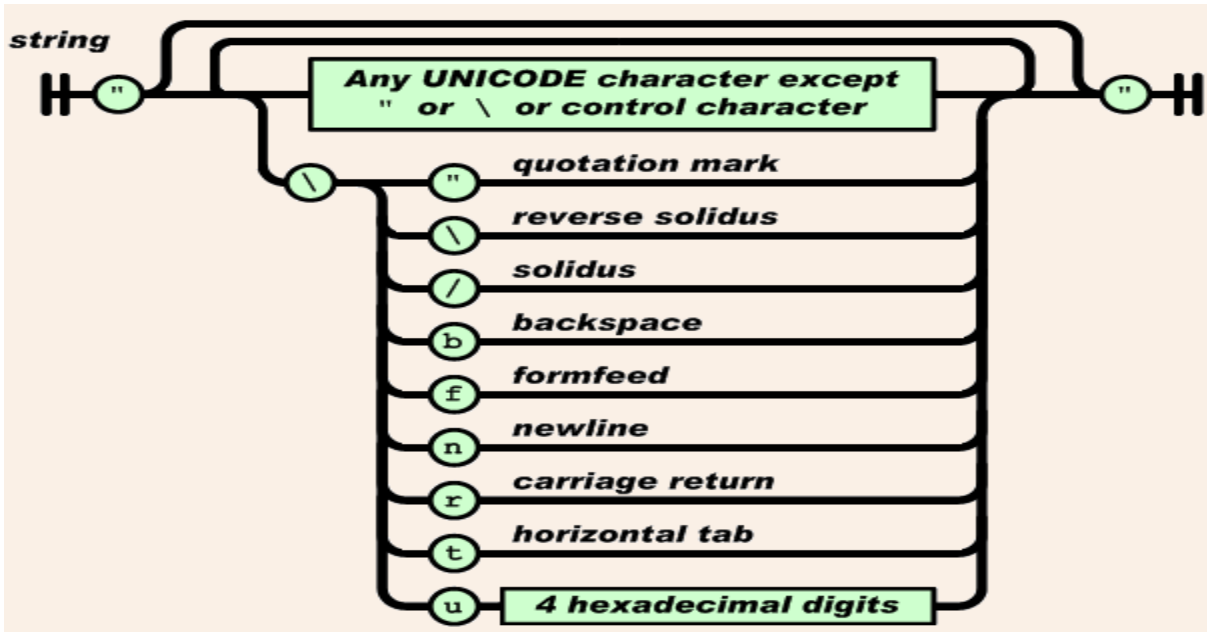
*Sumber : [www.json.org](http://www.json.org)*

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.



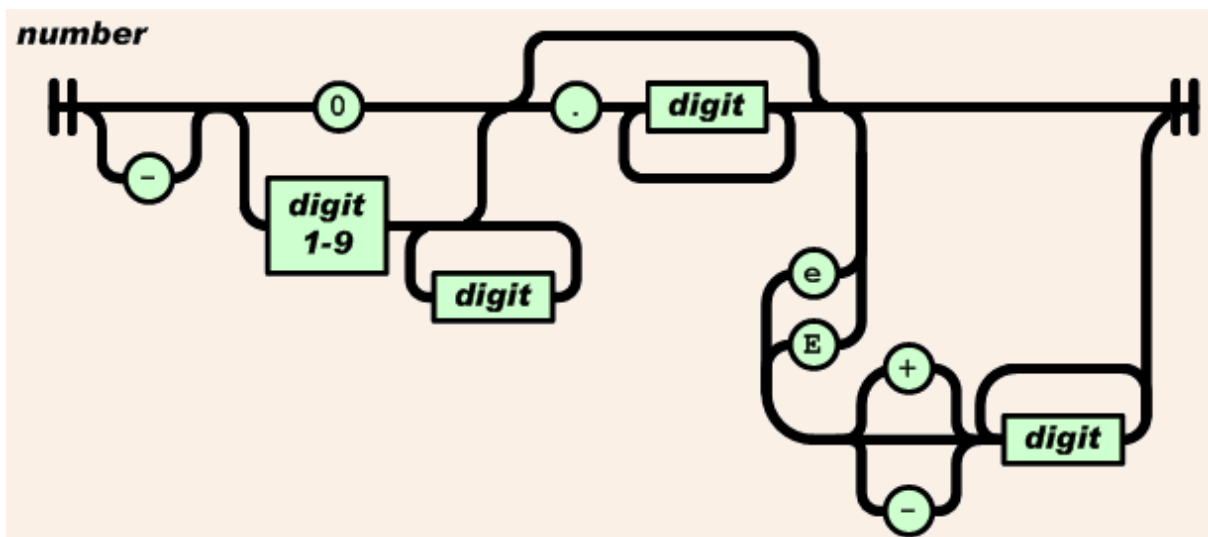
Sumber : [www.json.org](http://www.json.org)

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.



Sumber : [www.json.org](http://www.json.org)

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Sumber : [www.json.org](http://www.json.org)

## Contoh JSON

Sekarang mari kita lihat contoh-contoh bentuk JSON yang diambil dari json.org. Berikut bentuk XML, yang telah familiar dengan kita.

```
<menu id="file" value="File">
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()" />
    <menuitem value="Open" onclick="OpenDoc()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>
```

Jika kita ubah ke bentuk JSON adalah :

```
{ "menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      { "value": "New", "onclick": "CreateNewDoc()" },
      { "value": "Open", "onclick": "OpenDoc()" },
      { "value": "Close", "onclick": "CloseDoc()" }
    ]
  }
} }
```

Bentuk XML :

```

<widget>
  <debug>on</debug>
  <window title="Sample Konfabulator Widget">
    <name>main_window</name>
    <width>500</width>
    <height>500</height>
  </window>
  <image src="Images/Sun.png" name="sun1">
    <hOffset>250</hOffset>
    <vOffset>250</vOffset>
    <alignment>center</alignment>
  </image>
  <text data="Click Here" size="36" style="bold">
    <name>text1</name>
    <hOffset>250</hOffset>
    <vOffset>100</vOffset>
    <alignment>center</alignment>
    <onMouseUp>
      sun1.opacity = (sun1.opacity / 100) * 90;
    </onMouseUp>
  </text>
</widget>

```

Bentuk JSON :

```

{"widget": {
  "debug": "on",
  "window": {
    "title": "Sample Konfabulator Widget",
    "name": "main_window",
    "width": 500,
    "height": 500
  },
  "image": {
    "src": "Images/Sun.png",
    "name": "sun1",
    "hOffset": 250,
    "vOffset": 250,
    "alignment": "center"
  },
  "text": {
    "data": "Click Here",
    "size": 36,
    "style": "bold",
    "name": "text1",
    "hOffset": 250,
    "vOffset": 100,
    "alignment": "center",
    "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;"
  }
}}

```

XML	JSON
< xx yy='nn'>< /xx>	{ "xx": { "yy": "nn" } }

### Membuat JSON di Sisi Server

Sekarang mari kita lihat bagaimana cara membuat bentuk bentuk JSON. Dari sisi server kita akan membuat JSON menggunakan PHP, pada PHP versi 5.2 telah terdapat fungsi untuk membuat JSON, yaitu fungsi `json_encode()`.

Syntax-nya adalah sebagai berikut

```
string json_encode ( mixed $value [, int $options= 0 ] )
```

\$value, adalah nilai yang akan di encode menjadi string JSON

\$option, bisa berupa **JSON\_HEX\_QUOT**, **JSON\_HEX\_TAG**,  
**JSON\_HEX\_AMP**, **JSON\_HEX\_APOS**, **JSON\_FORCE\_OBJECT**,  
defaultnya adalah 0.

### Contoh :

```
<?php
$arr = array ('a'=>1, 'b'=>2, 'c'=>3, 'd'=>4, 'e'=>5);

echo json_encode($arr);
?>
```

Kode di atas akan menghasilkan :

```
{ "a":1, "b":2, "c":3, "d":4, "e":5 }
```

Selain menggunakan `json_encode()`, kita juga bisa membuat JSON secara manual dengan menggunakan PHP.

Misal, kita memiliki tabel di MySQL dengan struktrur sebagai berikut :

```
MariaDB [(none)]> use dbJSON;
```

```
Database changed
```

```
MariaDB [dbJSON]> 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)

-> )  
-> ;

Isi dari tabel tersebut adalah sebagai berikut:

message_id	chat_id	user_id	user_name	message	post_time
1	1	1	Nurmi	Halo apa kabar	4/28/2010 17:30
2	1	2	Desrizal	Kabar Baik	4/28/2010 17:30
3	1	1	Nurmi	Lagi di mana	4/28/2010 17:30
4	1	2	Desrizal	Belitung	4/28/2010 17:30

```
{ "messages":  
  { "pesan": [  
    { "id": "1",  
      "user": "Nurmi",  
      "text": "Halo apa kabar",  
      "time": "4/28/2010 17:30"  
    }, { "id": "2",  
      "user": "Desrizal",  
      "text": "Kabar Baik",  
      "time": "4/28/2010 17:30"  
    }, { "id": "3",  
      "user": "Nurmi", "text":  
      "Lagi di mana",  
      "time": "4/28/2010 17:30"  
    }, { "id": "4",  
      "user": "Desrizal",  
      "text": "Belitung",  
      "time": "4/28/2010 17:30"  
    } ]  
  }  
}
```

Dan kita ingin membuat JSON dari tabel di atas dengan bentuk seperti ini:

Berikut kode PHP untuk membuat JSON tersebut.

Membuat JSON dengan PHP secara manual (phpjson.php)

```

1 <?php
2     $link = mysqli_connect("localhost", "root", "", "dbJSON");
3     $json = '{"messages": {';
4     $query = mysqli_query($link, "select * from message");
5     $json .= '"pesan":[';
6     while($x = mysqli_fetch_array($query, MYSQLI_ASSOC)){
7         $json .= '{';
8         $json .= '"id": "' . $x['message_id'] . '",
9         "user": "' . htmlspecialchars($x['user_name']) . '",
10        "text": "' . htmlspecialchars($x['message']) . '",
11        "time": "' . $x['post_time'] . '"
12        },';
13    }
14
15    //hilangkan koma (,) di akhir
16    $json = substr($json, 0, strlen($json)-1);
17
18    //lengkapi penutup format JSON
19    $json .= ']';
20    $json .= '}}';
21
22    echo $json;
23 ?>

```