

## **Format File**

### **Teori Dasar**

Format file yang umumnya digunakan di internet adalah:

- JSON
- XML
- RDF
- Spreadsheet
- File Dipisahkan Koma (CSV)
- Dokumen Teks (Word, ODF, OOXML, atau PDF)
- Plain Text (.txt)
- Gambar terpindai (TIFF dan JPEG-2000)

### **Format hak milik**

Beberapa sistem terdedikasi, dll. mempunyai format data mereka sendiri yang dapat menyimpan dan mengekspor data yang masuk di dalamnya. Terkadang cukup mengekspos data dengan format demikian - terutama jika ke depannya berencana untuk penggunaan kembali sistem yang sama sebagaimana sebelumnya. Di mana selanjutnya informasi dengan format hak milik ini dapat ditemukan harus selalu dapat ditunjukkan, misalnya dengan memberikan tautan ke situs penyedia. Umumnya dianjurkan untuk menampilkan data dalam format non-hak milik di mana hal itu lebih layak.

### **HTML**

Biasanya, akan sangat tepat untuk menggunakan tabel dalam dokumen HTML untuk menyimpan data, serta menjadi hal yang penting pula untuk menampilkan field data yang beragam dan masing-masing diberikan ID dapat membuatnya mudah untuk menemukan dan memanipulasi data. Yahoo telah mengembangkan sebuah perangkat (<http://developer.yahoo.com/yql/>) yang dapat mengekstrak informasi terstruktur dari sebuah situs web, dan bahkan perangkat seperti itu dapat melakukan lebih banyak lagi dengan data jika ditandai dengan seksama.

### **Format File Terbuka**

Sekalipun bila informasi disediakan secara elektronik, format yang bisa dibaca mesin, dan dalam rinciannya, mungkin terdapat beberapa isu berkaitan dengan format file-nya.

Keuntungan dari format file terbuka adalah mereka memperkenalkan para pengembang untuk membuat multi paket perangkat lunak dan layanan yang menggunakan format ini. Hal ini meminimalisir kendala dalam menggunakan ulang informasi yang dimilikinya.

Menggunakan file berformat hak milik yang spesifikasinya tidak tersedia untuk umum dapat menciptakan ketergantungan pada perangkat lunak pihak ketiga atau pemegang lisensi format file tersebut. Dalam skenario terburuk, ini dapat berarti informasi hanya dapat dibaca dengan menggunakan paket perangkat lunak tertentu, yang bisa saja mahal, atau mungkin sudah usang.

### **Contoh: Data lalu-lintas UK**

Andrew Nicolson adalah seorang pengembang perangkat lunak yang terlibat dalam sebuah (sangat sukses) kampanye menentang pembangunan jalan bebas hambatan Westbury Eastern di Inggris. Andrew telah tertarik dengan mengakses dan menggunakan data lalu lintas jalan yang digunakannya untuk memperkuat usulannya. Ia berupaya mendapatkan sejumlah data relevan melalui kebebasan dalam mendapatkan informasi, tetapi pemerintah lokal

menyediakan datanya dalam format hak milik yang hanya bisa dibaca menggunakan perangkat lunak yang dibuat oleh Saturn, sebuah perusahaan yang khusus bergerak dalam pemodelan dan prediksi lalu lintas. Masalahnya tidak ada ketentuan yang dapat menyediakan versi “hanya baca” dari perangkat lunak tersebut, sehingga kelompok Andrew tidak punya pilihan selain membeli lisensi perangkat lunak tersebut dengan harga £500 (€600) itupun dengan potongan untuk versi edukasi. Paket utama perangkat lunak tersebut hingga April 2010 harga yang ditawarkan Saturn berkisar antara £13,000 (hingga lebih dari €15,000), sebuah harga yang tidak akan terjangkau oleh kebanyakan warga masyarakat biasa.

Meskipun tidak ada akses ke informasi, hukum memberi hak untuk mengakses informasi dalam format terbuka, setiap inisiatif data terbuka pemerintah perlu dimulai dengan didampingi dokumen-dokumen kebijakan yang menyatakan informasi resmi harus tersedia dalam format file terbuka. Pengaturan standar emas telah menjadi agenda kerja Obama melalui Ketentuan Terbuka Pemerintah yang diterbitkan pada Desember 2009 yang mengatakan:

*Untuk tujuan praktis dan tunduk pada pembatasan yang valid, lembaga-lembaga harus mempublikasikan informasi secara daring dalam format terbuka yang dapat diambil, diunduh, diindeks, dan dicari oleh aplikasi web yang umum digunakan dalam pencarian. Format terbuka adalah salah satu platform yang independen, dapat dibaca mesin, dan dibuat tersedia untuk umum tanpa pembatasan yang akan menghambat penggunaan kembali informasi tersebut.*

### **Bagaimana saya bisa menggunakan format yang diberikan?**

Ketika pihak otoritas harus menunjukkan data baru, anda harus memilih format yang dapat memberikan keseimbangan terbaik antara biaya dan kesesuaian dengan tujuan. Untuk setiap formatnya ada beberapa hal yang harus anda sadari, dan bagian ini bertujuan untuk menjelaskannya.

Bagian ini hanya berfokus pada bagaimana memotong permukaan menjadi cara pengaturan terbaik sehingga mesin dapat mengaksesnya secara langsung. Nasihat dan bimbingan tentang bagaimana situs dan solusi web harus didesain agar dapat ditemukan dari tempat lain.

### **Layanan web**

Untuk data yang berubah secara berkala, dan ketika untuk mendapatkannya telah dibatasi jumlahnya, menjadi sangat relevan untuk mengekspos data melalui layanan web. Terdapat beberapa cara untuk membuat layanan web, tetapi beberapa di antaranya yang paling sering digunakan adalah SOAP dan REST. Umumnya, SOAP melampaui REST, namun layanan REST lebih mudah untuk dikembangkan dan digunakan, sehingga ia menjadi standar yang banyak digunakan.

### **Basisdata**

Seperti layanan web, basisdata menyediakan akses langsung ke data secara dinamis. Basisdata mempunyai kelebihan yaitu di mana para penggunaanya bisa menyimpan secara bersama hanya dengan mengekstraksi apa yang mereka butuhkan.

Terdapat beberapa masalah keamanan mengenai fasilitas ekstraksi basisdata dengan kendali jarak jauh dan akses basisdata hanya akan berguna bila strukturnya dan kepercayaannya dari tabel-tabel individualnya dan field-field telah didokumentasikan dengan baik. Seringkali, adalah hal yang relatif sederhana dan murah untuk membuat layanan web yang mengekspos data dari basisdata, yang dapat menjadi cara mudah untuk mengatasi masalah keamanan.

## **Apa masalah seputar format file?**

Format File harus dipilih untuk memastikan pembagian, akses jangka panjang, dan pelestarian data. Pilih standar dan format terbuka yang mudah digunakan kembali. Jika menggunakan format yang berbeda selama fase pengumpulan dan analisis penelitian, pastikan untuk menyertakan informasi dalam dokumentasi tentang fitur yang mungkin hilang saat file dimigrasikan ke format preservasinya, serta perangkat lunak khusus apa pun yang akan diperlukan untuk melihat atau bekerja dengan data.

## **Praktik terbaik untuk pemilihan format file meliputi:**

- bukan hak milik
- tidak terenkripsi
- tidak terkompresi
- terbuka, standar terdokumentasi
- biasanya digunakan oleh komunitas riset
- menggunakan pengkodean karakter umum – ASCII, Unicode, UTF-8

Ingatlah untuk menyimpan data mentah asli yang belum diedit dalam format aslinya sebagai data sumber. Jangan mengubah atau mengeditnya. Dokumentasikan alat, instrumen, atau perangkat lunak yang digunakan dalam pembuatannya. Buat salinannya sebelum analisis atau manipulasi data apa pun.

## **Format Data Digital yang Disarankan:**

- Teks, Dokumentasi, Skrip: XML, PDF/A, HTML, Teks Biasa.
- Gambar Diam: TIFF, JPEG 2000, PNG, JPEG/JFIF, DNG (negatif digital), BMP, GIF.
- Geospasial: Shapefile (SHP, DBF, SHX), GeoTIFF, NetCDF.
- Format raster: TIFF, JPEG2000, PNG, JPEG/JFIF, DNG, BMP, GIF.
- Format vektor: grafik vektor yang dapat diskalakan, AutoCAD Drawing Interchange Format, Encapsulated Postscripts, Shape files.
- kartografi: Data terlengkap, GeoTIFF, GeoPDF, GeoJPEG2000, Shapefile.
- Audio: GELOMBANG, AIFF, MP3, MXF, FLAC.
- Video: MOV, MPEG-4, AVI, MXF.
- Basis data: XML, CSV, TAB.

## **Praktikum**

Penggunaan umum JSON adalah untuk membaca data dari server web, dan menampilkan data di halaman web. Pada praktikum ini akan mempelajari cara bertukar data JSON antara klien dan server PHP.

PHP memiliki beberapa fungsi bawaan untuk menangani JSON:

## **Mengonversi Array atau Objek PHP ke String JSON**

Fungsi `json_encode()` digunakan untuk mengencode nilai (array atau objek) ke format JSON.

Sintaks: `json_encode (value, options, depth)`

Value: wajib. Menentukan nilai yang akan dikodekan

## **Konversi Objek dalam PHP menjadi JSON**

Objek dalam PHP dapat dikonversi menjadi JSON dengan menggunakan fungsi PHP `json_encode()`:

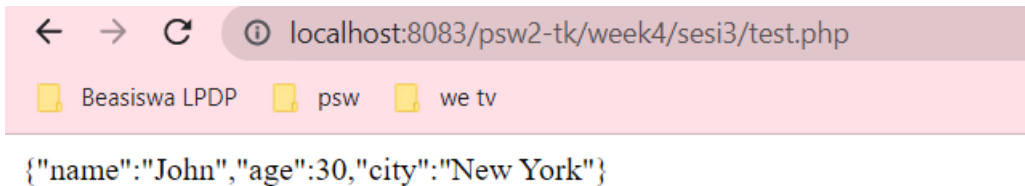
Contoh:

```

1  <?php
2  $myObj = new stdClass();
3  $myObj->name = "John";
4  $myObj->age = 30;
5  $myObj->city = "New York";
6
7  $myJSON = json_encode($myObj);
8
9  echo $myJSON;
10 ?>

```

Output:



localhost:8083/psw2-tk/week4/sesi3/test.php

Beasiswa LPDP psw we tv

{"name":"John","age":30,"city":"New York"}

### Konversi Array dalam PHP menjadi JSON

Array dalam PHP juga akan dikonversi menjadi JSON saat menggunakan fungsi PHP `json_encode()`:

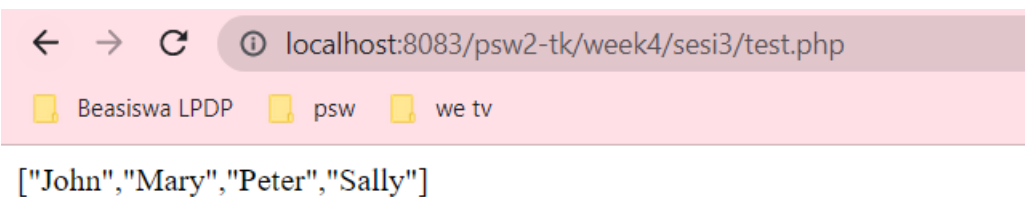
Contoh1:

```

1  <?php
2  $myArr = array("John", "Mary", "Peter", "Sally");
3
4  $myJSON = json_encode($myArr);
5
6  echo $myJSON;
7  ?>

```

Output1:



localhost:8083/psw2-tk/week4/sesi3/test.php

Beasiswa LPDP psw we tv

["John","Mary","Peter","Sally"]

Contoh2:

```

3.convert_php.php > ...
4  kelas    :
5  -->
6  <?php
7  // Array
8  $someArray = [
9      ["name" => "Jonathan Suh", "gender" => "male"],
10     ["name" => "William Philbin", "gender" => "male"],
11     ["name" => "Allison McKinnery", "gender" => "female"]
12 ];
13 // Convert Array to JSON String
14 $someJSON = json_encode($someArray);
15 echo $someJSON;
16 ?>

```

Contoh3:

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <?php
6  $age = array("Peter"=>35, "Ben"=>37, "Joe"=>43);
7
8  echo json_encode($age);
9  ?>
10
11 </body>
12 </html>

```

Contoh4:

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <?php
6  $cars = array("Volvo", "BMW", "Toyota");
7
8  echo json_encode($cars);
9  ?>
10
11 </body>
12 </html>

```

Contoh5:

```

1  <?php
2  $arr = array('a' => 1, 'b' => 2, 'c' => 3, 'd' => 4, 'e' => 5);
3  $x = json_encode($arr)."\n";
4  echo $x;
5  var_dump($x);
6  ?>
7

```

## Mengonversi String JSON ke Array atau Objek PHP

Fungsi `json_decode()` digunakan untuk mendekode atau mengubah objek JSON menjadi objek PHP.

Sintaks: `json_decode (string, assoc, depth, options)`

String: Wajib. Menentukan nilai yang akan didekodekan

Assoc: Opsional. Menentukan nilai Boolean. Saat diberikan nilai `true`, objek yang dikembalikan akan diubah menjadi array asosiatif. Saat diberikan nilai `false`, ini mengembalikan objek. `false` adalah default.

Contoh1:

```
1 json_string.php / ...
2 kelas :
3 -->
4
5
6
7 <?php
8 // JSON string
9 $someJSON = ' [{"name": "Jonathan Suh", "gender": "male"}, {"name": "William
10 Philbin", "gender": "male"}, {"name": "Allison McKinnery", "gender": "female"} ]
11 ';
12
13 // Mengonversi string JSON ke Array
14 $someArray = json_decode($someJSON, true);
15
16 // Dump all data of the Array
17 print_r($someArray);
18 echo '<br>';
19
20 // Access Array data
21 echo $someArray[0]["name"];
22 echo '<br>';
23
24 // Convert JSON string to Object
25 $someObject = json_decode($someJSON);
26
27 // Dump all data of the Object
28 print_r($someObject);
29 echo '<br>';
30
31 // Access Object data
32 echo $someObject[0]->name;
33 ?>
```

Contoh2: Simpan data JSON dalam variabel PHP, dan kemudian decode ke dalam objek PHP

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5
6
7 <?php
8 $jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';
9
10 var_dump(json_decode($jsonobj));
11 ?>
12
13 </body>
14 </html>
```

Contoh3: Simpan data JSON dalam variabel PHP, lalu dekodekan ke dalam array asosiatif PHP

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <?php
6  $jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';
7
8  var_dump(json_decode($jsonobj, true));
9  ?>
10
11 </body>
12 </html>

```

Contoh4: Cara mengakses nilai dari objek PHP

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <?php
6  $jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';
7
8  $obj = json_decode($jsonobj);
9
10 echo $obj->Peter;
11 echo $obj->Ben;
12 echo $obj->Joe;
13 ?>
14
15 </body>
16 </html>

```

Contoh4: Cara mengakses nilai dari array associative PHP

```

1
2  <!DOCTYPE html>
3  <html>
4  <body>
5
6  <?php
7  $jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';
8
9  $arr = json_decode($jsonobj, true);
10
11 echo $arr["Peter"];
12 echo $arr["Ben"];
13 echo $arr["Joe"];
14 ?>
15
16 </body>
17 </html>

```

**Contoh tambahan:**

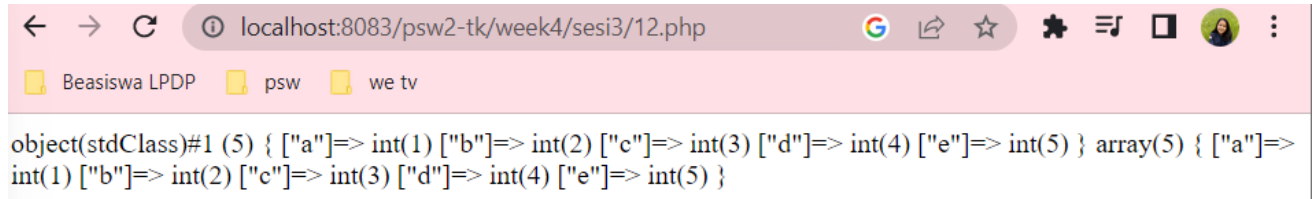
Bagaimana mengonversi string JSON ke array atau objek PHP?

```

1  <?php
2  //JSON string
3  $json = '{"a":1,"b":2,"c":3,"d":4,"e":5}';
4
5  //JSON Object
6  var_dump(json_decode($json));
7
8  //JSON Array
9  var_dump(json_decode($json, true));
10 ?>

```

Contoh diatas akan menghasilkan output sebagai berikut:



```

object(stdClass)#1 (5) { ["a"]=> int(1) ["b"]=> int(2) ["c"]=> int(3) ["d"]=> int(4) ["e"]=> int(5) } array(5) { ["a"]=> int(1) ["b"]=> int(2) ["c"]=> int(3) ["d"]=> int(4) ["e"]=> int(5) }

```

## Mengonversi String JSON ke Array atau Objek PHP

```

1  <?php
2  // JSON string
3  $soJSON = '[
4      {"name":"Suh","gender":"female"},
5      {"name":"WillPhilbin","gender":"male"},
6      {"name":"McKinnery","gender":"female"}
7  ]';
8
9  // Convert JSON string to Array
10 $soArray = json_decode($soJSON, true);
11
12 // Dump all data of the Array
13 print_r($soArray);
14
15 // Access Array data
16 echo $soArray[0]["name"];
17
18 // Convert JSON string to Object
19 $soObject = json_decode($soJSON);
20
21 // Dump all data of the Object
22 print_r($soObject);
23
24 // Access Object data
25 echo $soObject[0]->name;
26 ?>

```

## JSON Arrays

```

1  <?php
2  //Hello world
3  $json = '["Hello", "world"]';
4  $array = json_decode($json);
5
6  echo($array[0]); //Hello
7  ?>
8

```

## JSON Objects

Objek JSON dikonversi menjadi objek PHP secara default



```

1 <?php
2 $jsonObject = '{"Name":"John","Surname":"Doe"}';
3 $obj = json_decode($jsonObject);
4 echo($obj->Surname); // Doe
5 ?>

```

Fungsi `json\_decode()` memberikan argumen kedua boolean opsional bahwa jika dilewatkan sebagai `true` mengubah objek menjadi array asosiatif PHP. Dengan menggunakan contoh di atas, dan mendekodekannya seperti itu, elemen dapat diakses menggunakan tombol masing-masing.

```

1 <?php
2 $jsonObject = '{"Name":"John","Surname":"Doe"}';
3 $assocArray = json_decode($jsonObject, true);
4 // Accessing elements using standard array syntax
5 echo($assocArray['Name']); // John
6 ?>

```

### Menangani Objek Multidimensi

`json\_decode()` dapat mendekode string JSON secara rekursif. Argumen opsional ketiga memberi tahu fungsi kedalaman decoding yang harus terjadi. Pertimbangkan string JSON yang mewakili array objek yang diindeks.

```

<?php
$json = '[ {"id":"1", "name":"foo" }, {"id":"2", "name":"bar" } ]';
$baz = json_decode($json);
echo($baz[0]->name); // foo
?>

```

Jika benar akan diteruskan sebagai argumen kedua ke `json\_decode()` di atas, satu akan berakhir dengan array asosiatif yang diindeks secara numerik.

```

<?php
$json = '[ {"id":"1", "name":"foo" }, {"id":"2", "name":"bar" } ]';
$baz = json_decode($json,true);
echo($baz[1]['name']); //bar
?>

```

### Mengonversi String JSON ke Objek JavaScript

JavaScript memiliki metode `JSON.parse()` bawaan yang mem-parsing string JSON dan mengembalikan objek.

Contoh1:

```

5.php > script
4 kelas :
5 -->
6
7 <script>
8 // Convert JSON String to JavaScript Object
9 var JSONString = '[{"name":"Jonathan Suh",
  "gender":"male"}, {"name":"William Philbin",
  "gender":"male"}, {"name":"Allison McKinnery",
  "gender":"female"}]';
10
11 var JSONObject = JSON.parse(JSONString);
12
13 // Dump all data of the Object in the console
14 console.log(JSONObject);
15 // Access Object data
16 alert(JSONObject[0]["name"]);
17 </script>

```

Contoh2:

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h2>Creating an Object from a JSON String</h2>
6
7 <p id="demo"></p>
8
9 <script>
10 const txt = '{"name":"John", "age":30, "city":"New York"}'
11 const obj = JSON.parse(txt);
12 document.getElementById("demo").innerHTML = obj.name + ", " + obj.age;
13 </script>
14
15 </body>
16 </html>

```

## Array sebagai JSON

Saat menggunakan `JSON.parse()` pada JSON yang berasal dari array, metode ini akan mengembalikan array JavaScript, bukan objek JavaScript.

Contoh3:

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h2>Parsing a JSON Array.</h2>
6 <p>Data written as an JSON array will be parsed into a JavaScript array.</p>
7 <p id="demo"></p>
8
9 <script>
10 const text = '[ "Ford", "BMW", "Audi", "Fiat" ]';
11 const myArr = JSON.parse(text);
12 document.getElementById("demo").innerHTML = myArr[0];
13 </script>
14
15 </body>
16 </html>

```

## Mengonversi Objek JavaScript ke String JSON

JavaScript memiliki metode `JSON.stringify` untuk mengonversi objek javascript menjadi string JSON.

Contoh1:

```

1 <script>
2 var JSONObject = [
3   {
4     "name": "Jonathan Suh",
5     "gender": "male"
6   },
7   {
8     "name": "William Philbin",
9     "gender": "male"
10  },
11  {
12    "name": "Allison McKinnery",
13    "gender": "female"
14  }
15 ];
16 var JSONString = JSON.stringify(JSONObject);
17 alert(JSONString);
18 </script>

```

Contoh2:

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h2>Create a JSON string from a JavaScript object.</h2>
6  <p id="demo"></p>
7
8  <script>
9  const obj = {name: "John", age: 30, city: "New York"};
10 const myJSON = JSON.stringify(obj);
11 document.getElementById("demo").innerHTML = myJSON;
12 </script>
13
14 </body>
15 </html>
```

### Tugas:

1. Apa yang dimaksud:
  - JSON
  - XML
  - RDF
  - Spreadsheet
  - File Dipisahkan Koma (CSV)
  - Dokumen Teks (Word, ODF, OOXML, atau PDF)
  - Plain Text (.txt)
  - Gambar terpindai (TIFF dan JPEG-2000)
2. Buatlah masing-masing 1 contoh dari tiap format diatas yang diolah/*handle* dengan Bahasa pemrograman PHP.