

MODUL VIII-PHP-XML-JSON (*tambahan*)

8.1 PHP to XML

Pembuatan file XML dapat dilakukan dengan berbagai cara, salah satunya dengan menggunakan fungsi dari PHP yaitu **SimpleXMLElement**. Fungsi tersebut dapat digunakan untuk mengubah string pada PHP ke dalam format bahasa markup yaitu XML. Berikut ini adalah contoh code PHP yang disimpan dalam file **xml.php**:

```
<?php
$xml = new SimpleXMLElement('<tracks/>');

for ($i = 1; $i <= 8; ++$i) {
    $data = $xml->addChild('data');
    $data->addChild('path', 'song'.$i.'.mp3');
    $data->addChild('title', 'Track '.$i.' - Track Title');
}

Header('Content-type: text/xml');
print($xml->asXML());
file_put_contents('data.xml',$xml->saveXML()); //export to "data.xml"
?>
```

Gambar di bawah ini merupakan hasil jika menjalankan file **xml.php** pada localhost:



Di dalam file **xml.php** tersebut terdapat beberapa baris fungsi yang berfungsi untuk meng-export code yang telah dibuat sebelumnya menjadi file XML. Berikut baris program tersebut:

```
Header('Content-type: text/xml');  
print($xml->asXML());  
file_put_contents('data.xml',$xml->saveXML()); //export to "data.xml"
```

Jika fungsi export tersebut berhasil, file **data.xml** akan secara otomatis muncul pada directory yang sama dengan file **xml.php**:



Berikut contoh jika **data.xml** dijalankan pada browser:



8.2 XML to PHP

Secara fungsional, XML dapat digunakan untuk menyimpan data. Penyimpanan data yang dimaksud adalah data-data yang dimiliki suatu website dapat disimpan lalu dibagikan menggunakan XML, dimana hal tersebut disebut Web API/Web Services. Layanan Web API/Web Services yang diberikan oleh suatu website pun tetap harus ditampilkan secara representatif jika ingin menggunakannya. Maka dari itu terdapat beberapa fungsi dari PHP

yang dapat melakukan parsing file XML ke dalam bentuk yang lebih representatif, salah satu fungsinya yaitu **simplexml_load_file** yang dapat digunakan untuk melakukan load data XML.

Masih menggunakan contoh sebelumnya, berikut contoh code PHP yang digunakan untuk mem-parsing XML ke dalam bentuk tabel:

```
<html>
  <head></head>
  <body>
    <center>
      <h1>XML Parser - PHP</h1>
      <table border="2">
        <tr>
          <td>No.</td>
          <td>Track Path</td>
          <td>Track Title</td>
        </tr>
        <?php
          $tracks = simplexml_load_file('data.xml');
          $sno = 1;
          foreach ($tracks as $row){
            $path=$row->path;
            $title=$row->title;
            echo "<tr>
              <td>". $sno++."</td>
              <td>$path</td>
              <td>$title</td>
            </tr>";
          }
        <?>
      </table>
    </center>
  </body>
</html>
```

Pada code di atas terdapat baris code `$row->path`, dimana *path* yang dimaksud merupakan nama atribut pada XML yaitu `<path>`, begitu juga untuk atribut *title*. Baris code di atas apabila dicompile pada browser akan menampilkan data ke dalam bentuk tabel seperti di bawah ini:

```
<?xml version="1.0" encoding="UTF-8" ?>
<tracks>
  <data>
    <path>song1.mp3</path>
    <title>Track 1 - Track Title</title>
  </data>
  <data>
    <path>song2.mp3</path>
    <title>Track 2 - Track Title</title>
  </data>
  <data>
    <path>song3.mp3</path>
    <title>Track 3 - Track Title</title>
  </data>
  <data>
    <path>song4.mp3</path>
    <title>Track 4 - Track Title</title>
  </data>
  <data>
    <path>song5.mp3</path>
    <title>Track 5 - Track Title</title>
  </data>
  <data>
    <path>song6.mp3</path>
    <title>Track 6 - Track Title</title>
  </data>
  <data>
    <path>song7.mp3</path>
    <title>Track 7 - Track Title</title>
  </data>
  <data>
    <path>song8.mp3</path>
    <title>Track 8 - Track Title</title>
  </data>
</tracks>
```

XML Parser - PHP

No.	Track Path	Track Title
1	song1.mp3	Track 1 - Track Title
2	song2.mp3	Track 2 - Track Title
3	song3.mp3	Track 3 - Track Title
4	song4.mp3	Track 4 - Track Title
5	song5.mp3	Track 5 - Track Title
6	song6.mp3	Track 6 - Track Title
7	song7.mp3	Track 7 - Track Title
8	song8.mp3	Track 8 - Track Title

8.3 PHP to JSON

Sama seperti XML, pembuatan file JSON dapat dilakukan berbagai cara, salah satunya dengan menggunakan sebuah fungsi dari PHP yaitu **json_encode**. Fungsi tersebut digunakan untuk mengubah string pada PHP ke dalam format bahasa markup yaitu JSON. Berikut ini adalah contoh code PHP yang disimpan dalam file **json.php**:

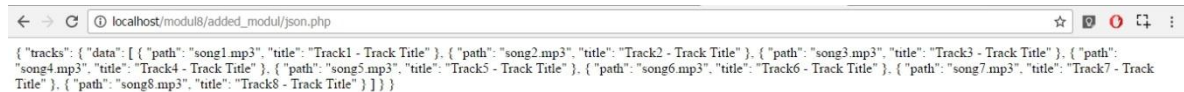
```
<?php
$tracks = array();
$data = array();
$contents = array();

for ($i = 1; $i <= 8; ++$i) {
    $contents[] = array(
        'path'=> 'song'.$i.'.mp3',
        'title'=> 'Track'.$i.' - Track Title'
    );
}

$data['data'] = $contents;
$tracks['tracks'] = $data;

$json_data = json_encode($tracks, JSON_PRETTY_PRINT);
echo $json_data;
file_put_contents('data.json', $json_data); //export to "data.json"
?>
```

Gambar di bawah ini merupakan hasil jika menjalankan file **json.php** pada localhost:



```
{ "tracks": { "data": [ { "path": "song1.mp3", "title": "Track1 - Track Title" }, { "path": "song2.mp3", "title": "Track2 - Track Title" }, { "path": "song3.mp3", "title": "Track3 - Track Title" }, { "path": "song4.mp3", "title": "Track4 - Track Title" }, { "path": "song5.mp3", "title": "Track5 - Track Title" }, { "path": "song6.mp3", "title": "Track6 - Track Title" }, { "path": "song7.mp3", "title": "Track7 - Track Title" }, { "path": "song8.mp3", "title": "Track8 - Track Title" } ] } }
```

Di dalam file **json.php** tersebut terdapat beberapa baris fungsi yang berfungsi untuk meng-export code yang telah dibuat sebelumnya menjadi file XML. Berikut baris program tersebut:

```
$json_data = json_encode($tracks, JSON_PRETTY_PRINT);
echo $json_data;
file_put_contents('data.json', $json_data); //export to "data.json"
```

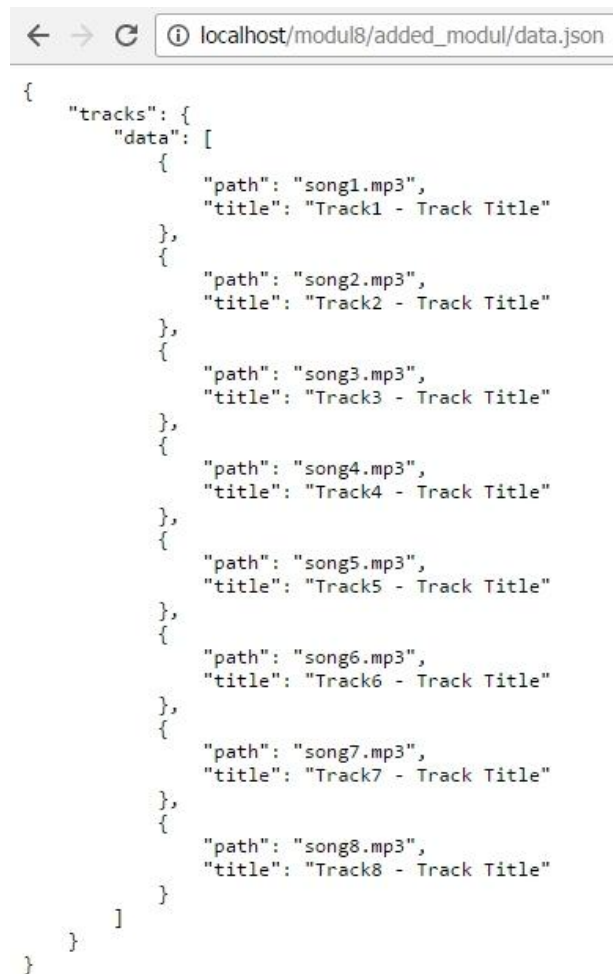
Jika fungsi export tersebut berhasil, file **data.json** akan secara otomatis muncul pada directory yang sama dengan file **json.php**:

 data.json

3/18/2017 10:50 AM

JSON File

Berikut contoh jika **data.json** dijalankan pada browser:



```
{
  "tracks": {
    "data": [
      {
        "path": "song1.mp3",
        "title": "Track1 - Track Title"
      },
      {
        "path": "song2.mp3",
        "title": "Track2 - Track Title"
      },
      {
        "path": "song3.mp3",
        "title": "Track3 - Track Title"
      },
      {
        "path": "song4.mp3",
        "title": "Track4 - Track Title"
      },
      {
        "path": "song5.mp3",
        "title": "Track5 - Track Title"
      },
      {
        "path": "song6.mp3",
        "title": "Track6 - Track Title"
      },
      {
        "path": "song7.mp3",
        "title": "Track7 - Track Title"
      },
      {
        "path": "song8.mp3",
        "title": "Track8 - Track Title"
      }
    ]
  }
}
```

8.4 JSON to PHP

Secara fungsional, JSON dapat digunakan untuk menyimpan data. Penyimpanan data yang dimaksud adalah data-data yang dimiliki suatu website dapat disimpan lalu dibagikan menggunakan JSON, dimana hal tersebut disebut Web API/Web Services. Layanan Web API/Web Services yang diberikan oleh suatu website pun tetap harus ditampilkan secara representatif jika ingin menggunakannya. Maka dari itu terdapat beberapa fungsi dari PHP yang dapat melakukan parsing file JSON ke dalam bentuk yang lebih representatif, jika pada saat mengubah PHP ke dalam format JSON menggunakan **json_encode** maka jika ingin mengubah format JSON ke dalam bentuk PHP yang lebih representatif gunakan **json_decode**.

Masih menggunakan contoh sebelumnya, berikut contoh code PHP yang digunakan untuk mem-parsing JSON ke dalam bentuk tabel:


```

<html>
<head></head>
<body>
<center>
<h1>JSON Parser - PHP</h1>
<table border="2">
<tr>
<td>No.</td>
<td>Track Path</td>
<td>Track Title</td>
</tr>
<?php
$json_url = 'data.json';
$json = file_get_contents($json_url);
$links = json_decode($json);
$no = 1;
foreach($links->tracks->data as $row) {
$path=$row->path;
$title=$row->title;
echo "<tr>
<td>". $no++."</td>
<td>$path</td>
<td>$title</td>
</tr>";
}
?>
</table>
</center>
</body>
</html>

```

Pada code di atas terdapat baris code `$links->tracks->data`, dimana *tracks* yang dimaksud merupakan main parent dari data yang terdapat pada JSON. Sedangkan *data* pada baris tersebut adalah child dari main parent yang akan diambil datanya. Baris code di atas apabila dicompile pada browser akan menampilkan data ke dalam bentuk tabel seperti di bawah ini:

```

{
  "tracks": {
    "data": [
      {
        "path": "song1.mp3",
        "title": "Track1 - Track Title"
      },
      {
        "path": "song2.mp3",
        "title": "Track2 - Track Title"
      },
      {
        "path": "song3.mp3",
        "title": "Track3 - Track Title"
      },
      {
        "path": "song4.mp3",
        "title": "Track4 - Track Title"
      },
      {
        "path": "song5.mp3",
        "title": "Track5 - Track Title"
      },
      {
        "path": "song6.mp3",
        "title": "Track6 - Track Title"
      },
      {
        "path": "song7.mp3",
        "title": "Track7 - Track Title"
      },
      {
        "path": "song8.mp3",
        "title": "Track8 - Track Title"
      }
    ]
  }
}

```

JSON Parser - PHP

No.	Track Path	Track Title
1	song1.mp3	Track1 - Track Title
2	song2.mp3	Track2 - Track Title
3	song3.mp3	Track3 - Track Title
4	song4.mp3	Track4 - Track Title
5	song5.mp3	Track5 - Track Title
6	song6.mp3	Track6 - Track Title
7	song7.mp3	Track7 - Track Title
8	song8.mp3	Track8 - Track Title