

# **XML vs JSON**

**by: Ahmad Syauqi Ahsan**

# What is XML ?

- XML adalah singkatan dari eXtensible Markup Language yang dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada 10 Februari 1998.
- XML merupakan turunan dari SGML (Standart Generalized Markup Language), seperti juga HTML.
- XML tidak mempunyai mekanisme untuk menentukan aturan dalam menampilkan data ke user.
- XML dapat digunakan untuk menyimpan data terstruktur kedalam text file.
  - *easy to read*
  - *unambiguous*
  - *extensible*
  - *platform-independent*

# XML vs HTML (Persamaan)

- Persamaan XML dan HTML
  - *Keduanya menggunakan elemen yang diletakkan dalam tag.*
    - `<body>PENS Surabaya</body>`
    - `<linebreak></linebreak` atau `<linebreak />`
  - *Keduanya dapat menggunakan atribut dalam tag.*
    - `<font face="Verdana" color="red">Text</font>`
  - *Keduanya mempunyai entities (&lt;, &gt;, &amp;, &quot;, &apos;).*

# XML vs HTML (Perbedaan)

## XML

- Tag dapat dikembangkan (extensible)
- Berorientasi pada isi
- Ada infrastructure untuk menstandarkan data
- Memungkinkan adanya beberapa tampilan

## HTML

- Mempunyai tag-tag yang tetap/fixed.
- Berorientasi pada tampilan
- Tidak mempunyai kemampuan untuk melakukan validasi data
- Tampilan tetap

# Aturan dasar XML

- XML adalah case sensitive
- Setiap start tags harus mempunyai end tags
- Setiap dokumen harus mempunyai element root
- Nilai dari suatu atribut harus berada dalam tanda kutip
- Ada beberapa karakter yang tidak dapat digunakan (reserved for parsing).
- Deklarasi XML harus menjadi statemen pertama.

# Contoh dokumen XML

```
<?xml version="1.0"?>
<ramalanCuaca>
  <kota nama="Surabaya">
    <tanggal>30/05/2012</tanggal>
    <suhu skala="C">40</suhu>
    <pagi>Cerah</pagi>
    <siang>Panas @amp Berawan</siang>
    <malam cuaca="Gelap" />
  </kota>
  <kota nama="Malang">
    <tanggal>30/05/2012</tanggal>
    <suhu skala="C">32</suhu>
    <pagi>Berawan</pagi>
    <siang>Panas @amp Berawan</siang>
    <malam cuaca="Gelap" />
  </kota>
</ramalanCuaca>
```

# Struktur Global

- Dokumen XML dapat dimulai dengan satu atau lebih directives (Processing Instruction):

```
<?xml version="1.0"?>
```

- Setelah directive, **harus** ada **satu** tag yang disebut sebagai **elemen root**.

Semua data dalam XML harus berada didalam elemen root ini.

```
<ramalanCuaca>
```

```
    . . . . .  
    . . . . .
```

```
</ramalanCuaca>
```

# Blok-blok pembentuk XML

Disamping directives, dokumen XML dapat dibentuk dari:

- Elements:  
`tanggal` di `<tanggal>30/05/2012</tanggal>`
- Tags: berpasangan:  
`<suhu skala="C">40</suhu>`
- Attributes:  
`<suhu skala="C">40</suhu>`
- Entities  
`<siang>Panas @amp; Berawan</siang>`
- Character data  
`<pagi>Cerah</pagi>`



# XML declaration

- XML declaration dapat terlihat seperti berikut:  
`<?xml version="1.0" encoding="UTF-8" standalone="yes"?>`
- XML declaration tidak dibutuhkan oleh browser, tetapi dibutuhkan oleh hampir semua XML processor.
- Jika digunakan, XML declaration harus berada paling awal. (Bahkan tidak boleh didahului oleh karakter spasi)
- Perlu dicatat tanda kurung yang digunakan adalah `<?>` dan `?>`.
- `version="1.0"` adalah required (versi terakhir saat ini)
- `encoding` dapat berupa `"UTF-8"` (ASCII) atau `"UTF-16"` (Unicode) atau yang lain. Dapat juga dihilangkan.
- `standalone` dapat digunakan untuk memberi tahu apakah ada DTD yg terpisah atau tidak.

# Elemen dan Atribut

- Dalam banyak kasus, elemen dan atribut dapat saling dipertukarkan.

- Contoh: menyimpan data dalam elemen

```
<nama>
```

```
  <awal>Jono</awal>
```

```
  <akhir>Andromeda</akhir>
```

```
</nama>
```

- Bisa juga dituliskan sebagai atribut:

```
<nama awal="Jono" akhir="Andromeda"></nama>
```

- Didalam program, penggunaan elemen akan lebih mudah dibandingkan atribut.
- Atribut sering digunakan untuk menyimpan metadata, seperti unique IDs.

# Entities

- Ada lima karakter spesial yang harus ditulis sebagai entities:
  - &amp;*; for *&* (almost always necessary)
  - &lt;*; for *<* (almost always necessary)
  - &gt;*; for *>* (not usually necessary)
  - &quot;*; for *"* (necessary inside double quotes)
  - &apos;*; for *'* (necessary inside single quotes)
- Hanya lima entities diatas yang sudah terdefinisi (predefined) didalam XML.

# Processing Instruction (PI)

- PI dapat berada di bagian manapun dalam dokumen XML, tetapi biasanya berada di awal.
- PI adalah perintah bagi program untuk memproses dokumen XML dengan cara tertentu.
- Dokumen XML biasanya diproses oleh lebih dari satu program.
- Program-program yang tidak dapat mengenali PI dapat mengabaikannya.
- Format umum dari PI adalah `<?target_instruction?>`
- Contoh:  
`<?xml-stylesheet type="text/css" href="mySheet.css"?>`

# Komentar

- Contoh komentar:  
<!-- Ini adalah komentar baik di HTML maupun XML -->
- Komentar dapat diletakkan dimanapun didalam dokumen XML
- Komentar berguna untuk:
  - *Menjelaskan struktur dari dokumen XML*
  - *Membuat agar beberapa bagian dari XML tidak diproses (biasanya selama proses development dan testing).*
- Komentar bukan merupakan elemen, jadi tidak memerlukan end tag.
- Komentar tidak ditampilkan dalam browser, tetapi dapat dilihat dari source codenya.

# Validasi dokumen XML

- Pada prinsipnya, pengaturan isi dari dokumen XML adalah bebas.
- Tetapi, jika dibutuhkan, kita bisa membuat aturan-aturan.
- Aturan-aturan tersebut dapat didefinisikan didalam DTD (Document Type Definition).
- Dengan DTD, kita bisa membatasi tag-tag apa saja yang diperbolehkan dan tempat dimana tag-tag tersebut dapat diletakkan.
- DTD dapat disertakan didalam dokumen XML, tetapi biasanya merupakan file yang terpisah.
- Alternatif lain dari penggunaan DTD adalah XML schemas atau Relax NG.

# Menampilkan XML

- Dokumen XML didesain untuk diproses oleh program komputer, tidak untuk dibaca langsung oleh manusia.
- Tapi bagaimanapun juga, saat ini hampir semua browser dapat menampilkan dokumen XML.
  - *Tampilan antara satu browser dengan yang lain tidak selalu sama.*
  - *Bisa jadi browser tidak menampilkan isi dari dokumen XML jika terdapat error didalamnya.*
- Yang perlu diingat:
  - *HTML didesain untuk **ditampilkan**.*
  - *XML didesain untuk **digunakan**.*

# Standar tambahan dokumen XML

Anda dapat mendefinisikan XML tag sets, tetapi ada beberapa standar yang telah tersedia:

- XHTML: HTML didalam XML
- XAML: eXtensible Application Markup Language
  - *Bahasa deklaratif untuk Microsoft Silverlight dan DirectX graphics programming pada Windows Vista keatas.*
- SOAP: Simple Object Access Protocol
  - *Mendefinisikan bentuk-bentuk dari pesan-pesan dan RPC(Remote Procedure Calls)*
- MathML: Mathematical Markup Language
  - *Mendeskrripsikan notasi matematika dalam mengintegrasikan formula matematika kedalam HTML dan tipe dokumen yang lain.*
- SVG: Scalable Vector Graphics
  - *Mendeskrripsikan vektor grafik dua dimensi.*



# What is JSON?

- JSON singkatan dari JavaScript Object Notation
- Merupakan format untuk pertukaran data yang berbasiskan teks, ringan, platform-independent, serta mudah di-parse.
- JSON merupakan subset dari JavaScript
- JSON merupakan alternative dari XML untuk pertukaran data
- Hampir semua Web Browser modern sudah mendukung format JSON

# JSON dibangun dengan 2 struktur

- Kumpulan pasangan nama dan nilai (name & value pairs)
  - Pada bahasa pemrograman lain, hal ini dapat berbentuk sebuah object, record, struct, dictionary, hash table, keyed list, atau associative array
  - Misal: sebuah objek dengan 3 property bernama "a", "b", dan "c" dapat ditulis dengan format JSON seperti berikut:  

```
{ "a": 1, "b": 5, "c": 3 }
```
- Daftar nilai yang berurutan (ordered list of values)
  - Pada Bahasa pemrograman lain, hal ini dapat berbentuk array, vector, list, atau sequence.
  - Misal: sebuah array yang terdiri dari 3 integer dan 1 nilai string dapat ditulis dengan format JSON seperti berikut:  

```
[ 1, 2, 3, "nilai ke-4" ]
```

# Contoh JSON

```
{  
  "fullname": "Jono Gumilang",  
  "phones": [  
    {"type": "work", "value": "+62315914321"},  
    {"type": "mobile", "value": "+628565914321"}  
  ],  
  "addresses": [  
    {"type": "work",  
      "value": "Jalan Raya ITS no 1"},  
    {"type": "home",  
      "value": "Jalan Kertajaya Indah no 1"}  
  ]  
}
```

# JSON vs XML

- Keduanya memiliki simplicity, openness, dan interoperability yang sama.
- Keduanya mempunyai format data yang mudah untuk didistribusikan ke berbagai macam user
- JSON tidak bisa di-extend, sedangkan XML bisa di-extend untuk menyimpan data gambar, video, dll.
- JSON mempunyai ukuran yang lebih kecil dibandingkan dengan XML untuk menyimpan data yang sama
- JSON lebih mudah digunakan pada JavaScript.
- Di aplikasi yang menggunakan AJAX, proses parsing data JSON lebih cepat dibandingkan dengan XML
- Saat ini, sebagian besar web-services hanya menyediakan data dalam bentuk XML



**THANK YOU**