

Pengenalan XML

Oleh :

Janner Simarmata
sijanner@yahoo.com
<http://simarmata.cogia.net>

*Dipublikasikan dan didedikasikan
untuk perkembangan pendidikan di Indonesia melalui*

MateriKuliah.Com

Lisensi Pemakaian Artikel:

*Seluruh artikel di **MateriKuliah.Com** dapat digunakan, dimodifikasi dan disebarakan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut Penulis. Hak Atas Kekayaan Intelektual setiap artikel di **MateriKuliah.Com** adalah milik Penulis masing-masing, dan mereka bersedia membagikan karya mereka semata-mata untuk perkembangan pendidikan di Indonesia. **MateriKuliah.Com** sangat berterima kasih untuk setiap artikel yang sudah Penulis kirimkan.*

Pengantar

XML dirancang untuk menguraikan data dan untuk memusatkan pada data.

HTML dirancang untuk menampilkan data dan untuk memusatkan bagaimana melihat data.

Apa yang harus anda perlu ketahui

Sebelum kita melanjutkan pembahasan ini, anda harus mempunyai beberapa pemahaman dasar yang berikut:

- WWW, HTML dan dasar membangun Halaman web
- Bahasa Script Web seperti Javascript atau VBScript

Apa itu XML?

- XML adalah singkatan dari **EX**tensible **M**arkup **L**anguage
- XML adalah suatu bahasa **markup language** seperti halnya HTML
- XML dirancang untuk **menguraikan data**
- Tag-tag XML tidak didefinisikan. Anda harus mendefinisikan tagmu sendiri
- XML menggunakan sebuah **Document Type Definition** (DTD) atau sebuah **skema XML** untuk menguraikan data
- XML dengan sebuah DTD atau skema XML dirancang menjadi self-descriptive

Language

- Mempunyai sebuah tatabahasa
- Mempunyai sebuah kosa kata
- Dapat diuraikan oleh mesin

Markup Language

- Bukanlah sebuah bahasa pemrograman
- Tidak di-compile

Extensible

- Dapat menambahkan kata-kata pada bahasa

Perbedaan utama antara XML dan HTML

XML dirancang untuk membawa data.

- XML bukanlah suatu pengganti untuk HTML.
- XML dan HTML telah dirancang dengan tujuan yang berbeda:
- XML dirancang untuk menguraikan data dan untuk memusatkan pada data.
- HTML dirancang untuk menampilkan data dan untuk memusatkan bagaimana melihat data.
- HTML adalah tentang bagaimana menampilkan informasi, sedangkan XML adalah tentang bagaimana menggambarkan informasi.

XML adalah sebuah komplemen HTML

XML bukanlah sebuah pengganti untuk HTML.

Adalah penting untuk memahami XML, bahwa XML itu bukanlah sebuah pengganti untuk HTML. Pada masa yang akan datang pengembangan Web sudah hampir bisa dipastikan, XML akan digunakan untuk menguraikan data, sedangkan HTML akan digunakan untuk format dan menampilkan data yang sama.

Uraian XML: XML adalah sebuah **cross-platform, perangkat lunak dan perangkat keras untuk menyebarkan informasi.**

XML pada masa depan Pengembangan Web

XML akan berada dimana-mana.

XML menjadi sangat mengagumkan untuk melihat seberapa cepat standard XML telah dikembangkan dan seberapa cepat jumlah penjual perangkat lunak yang besar sudah mengadopsi standard XML ini.

Kita benar-benar percaya bahwa XML akan menjadi sama pentingnya bagi masa depan Web seperti halnya HTML telah menjadi pondasi bagi Web, dan XML akan menjadi alat yang paling umum untuk semua manipulasi data dan transmisi data.

Bagaimana XML Digunakan?

Adalah penting untuk memahami XML itu dirancang untuk menyimpan, membawa, dan menukar data. XML tidaklah dirancang untuk menampilkan data.

XML dapat Memisahkan Data dari HTML

Dengan XML, data anda disimpan diluar HTML anda.

Ketika HTML digunakan untuk menampilkan data, data disimpan didalam HTML anda. Dengan XML, data dapat disimpan terpisah dalam file XML. Dengan cara ini anda dapat berkonsentrasi pada penggunaan HTML untuk data layout dan tampilan.

XML digunakan untuk Pertukaran Data

Dengan XML, data dapat ditukar antara sistem yang tidak sesuai.

Pada dunia nyata, sistem komputer dan database berisi data didalam format yang tidak sesuai. Salah satu dari tantangan yang paling memakan waktu bagi pengembang harus menukar data antara sistem melalui Internet.

Merubah data ke XML dapat mengurangi kompleksitas ini dan menciptakan data yang dapat dibaca oleh banyak jenis aplikasi yang berbeda .

XML dan B2B (Business To Business)

Dengan XML, informasi keuangan dapat ditukar melalui Internet.

Harapan untuk melihat banyak tentang XML dan B2B (Business To Business) dimasa mendatang sudah hampir dekat.

XML akan menjadi bahasa yang utama untuk menukarkan informasi keuangan antara bisnis melalui Internet.

XML dapat digunakan untuk Berbagi Data

Dengan XML, file plain text dapat digunakan untuk berbagi data.

Sejak data XML disimpan dalam format plain text, XML menyediakan sebuah perangkat lunak - dan hardware-independent cara untuk berbagi data.

Ini membuatnya banyak lebih mudah untuk menciptakan data yang berbeda aplikasi.

Plain text juga membuatnya lebih mudah untuk memperluas atau mengupgrade sebuah sistem ke sistem operasi baru, server, aplikasi, dan browsers baru.

XML dapat digunakan untuk Menyimpan Data

Dengan XML, file plain text dapat digunakan untuk menyimpan data.

XML dapat juga digunakan untuk menyimpan data didalam file atau database. Aplikasi dapat ditulis untuk menyimpan dan mengambil kembali informasi dari penyimpanan, dan aplikasi umum dapat digunakan untuk menampilkan data.

Aturan sintaks XML adalah yang sangat tegas dan sangat sederhana. Aturan adalah sangat mudah untuk dipelajari, dan sangat mudah untuk menggunakannya.

Oleh karena itu, menciptakan perangkat lunak yang dapat membaca dan memanipulasi XML adalah sangat mudah untuk dilakukan.

Contoh sebuah dokumen XML

Dokumen XML menggunakan sebuah self-describing dan sintaks yang sederhana.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Ojak</to>
<from>Jay</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Baris yang pertama dalam dokumen - deklarasi XML - menggambarkan versi XML dan karakter pengkodean yang digunakan dalam dokumen. Dalam hal ini dokumen menyesuaikan diri pada 1.0 spesifikasi XML dan menggunakan ISO-8859-1 (*Latin-1/West European*) kumpulan karakter.

Baris yang berikutnya menguraikan elemen akar dokumen. ("dokumen ini adalah sebuah catatan"):

```
<note>
```

Baris 4 berikutnya menguraikan 4 elemen-elemen anak akar (to, from, heading, and body):

```
<to>Ojak</to>
<from>Jay</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
```

Dan baris yang terakhir menggambarkan akhir elemen akar:

```
</note>
```

Semua elemen XML harus mempunyai sebuah tag penutup.

Dengan XML, itu tidak sah jika tidak ada tag penutup.

Didalam HTML beberapa elemen-elemen tidak mempunyai sebuah tag penutup. Kode yang berikut adalah sah pada kode HTML:

```
<p>This is a paragraph
<p>This is another paragraph
```

Didalam XML semua elemen harus mempunyai sebuah tag penutup, seperti ini:

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

Tag-tag XML adalah case sensitive

Tidak seperti HTML, tag XML adalah case sensitive.

Dengan XML, tag <Surat> adalah berbeda dari tag <surat>.

Karena:

- tag yang pertama <Surat> (S adalah huruf BESAR)
- tag yang kedua <surat> (s adalah huruf kecil)

Tag pembuka dan penutup harus ditulis sama, seperti contoh dibawah ini:

```
<Message>Ini adalah salah</message>
```

```
<message>Ini adalah benar</message>
```

Semua elemen XML harus bersarang dengan baik

Didalam HTML beberapa elemen bisa tidak sesuai didalam tag satu sama lain seperti ini:

```
<b><i> Teks Ini bold dan italic</b></i>
```

Didalam XML semua elemen harus bersarang dengan baik didalam satu sama lain seperti ini:

```
<b><i> Teks Ini bold dan italic </i></b>
```

Semua dokumen XML harus mempunyai sebuah elemen akar

Semua dokumen XML harus berisi tag tunggal dipasang untuk menggambarkan sebuah elemen akar.

Semua elemen lain harus didalam elemen akar ini.

Semua elemen-elemen dapat mempunyai sub elemen-elemen (elemen-elemen anak). Sub elemen-elemen harus dengan tepat bersarang didalam elemen orangtua:

```
<akar>
  <anak>
    <subanak> . . . . . </subanak>
  </anak>
</akar>
```

Nilai-nilai Atribut harus selalu diberi tanda kutip

Dengan XML, itu adalah tidak sah jika tidak membuat tanda kutip disekitar nilai-nilai atribut.

Elemen XML dapat mempunyai atribut pasangan nama seperti halnya HTML. Didalam XML atribut nilai harus selalu diberi tanda kutip. Lihat contoh dibawah ini. Nomor satu adalah salah, yang kedua adalah benar:

Nomor satu

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note date=12/11/2002>
<to>Ojak</to>
<from>Jay</from>
</note>
```

Nomor dua

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note date="12/11/2002">
<to>Ojak</to>
<from>Jay</from>
</note>
```

Kesalahan pada dokumen yang pertama adalah bahwa tanggal menunjukkan elemen catatan yang tidak diberi tanda kutip.

Ini adalah benar: date="12/11/2002".

Ini adalah salah: date=12/11/2002.

Komentar Dalam XML

Sintaks untuk menulis komentar didalam XML adalah serupa dengan HTML.

```
<!-- Ini adalah sebuah komentar-->
```

Elemen –elemen XML

Elemen-elemen XML dapat diperluas dan mereka mempunyai hubungan.

Elemen-elemen XML mempunyai aturan penamaan yang sederhana.

Elemen-elemen XML Dapat Diperluas

Dokumen XML dapat diperluas untuk membawa banyak informasi.

Lihat contoh XML yang berikut:

```
<note>
<to>Ojak</to>
<from>Jay</from>
<body>Don't forget me this weekend!</body>
</note>
```

Mari kita bayangkan, menciptakan sebuah aplikasi yang diberi tanda kutip <to>, <from>, dan <body> elemen-elemen dari dokumen XML untuk menghasilkan keluaran ini:

MESSAGE

To: Ojak

From: Jay

Don't forget me this weekend!

Bayangkan pembuat dokumen XML menambahkan beberapa informasi tambahan untuk itu:

```
<note>
<date>2002-08-01</date>
<to>Ojak</to>
<from>Jay</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
```

</note>

Elemen-elemen XML mempunyai Hubungan

Elemen-elemen terkait seperti orang tua dan anak.

Untuk memahami istilah XML, anda harus mengetahui bagaimana hubungan antara elemen-elemen XML, dan bagaimana isi elemen-elemen itu diuraikan.

Bayangkan, ini adalah sebuah uraian suatu buku:

XML Pertamaku

Pengantar XML

Apa itu HTML

Apa itu XML

Sintaks XML

Element harus punya tag penutup

Element harus bersarang

Bayangkan, dokumen XML ini menguraikan buku itu:

```
<book>
<title> XML Pertamaku
</title>
<prod id="33-657" media="paper"></prod>
<chapter>Introduction to XML
<para> Apa itu HTML</para>
<para> Apa itu XML</para>
</chapter>

<chapter> Sintaks XML
<para> Element harus punya tag penutup </para>
<para> Element harus bersarang </para>
</chapter>

</book>
```

Buku adalah elemen akar. Title, prod, dan chapter adalah elemen-elemen anak buku.

Buku adalah elemen orangtua title, prod, dan chapter. Title, prod, dan chapter adalah saudara kandung (atau elemen saudara) sebab mereka mempunyai orangtua yang sama.

Elemen-elemen mempunyai Isi

Elemen-elemen dapat mempunyai isi tipe yang berbeda.

Sebuah **elemen XML** adalah segalanya dari (*termasuk*) elemen tag awal pada elemen tag akhir.

Sebuah elemen dapat mempunyai isi **elemen**, isi yang **dicampur**, isi **sederhana**, atau isi **kosong**. Sebuah elemen dapat juga mempunyai atribut.

Pada contoh di atas, buku mempunyai **isi elemen**, sebab berisi elemen-elemen lain. Bab telah **mencampur isi** sebab berisi teks kedua-duanya dan elemen-elemen lain. Para mempunyai **isi sederhana** (*atau isi teks*) sebab hanya berisi teks. Prod mempunyai **isi kosong**, sebab tidak membawa informasi apapun.

Pada contoh diatas hanya elemen prod mempunyai **atribut**. **Atribut** dinamakan ID mempunyai **nilai** " 33-657". **Atribut** dinamakan media mempunyai **nilai**"kertas".

Penamaan Elemen

Elemen-elemen XML harus mengikuti aturan penamaan ini:

- Nama-nama dapat berisi surat, angka-angka, dan karakter lain
- Nama-nama harus tidak dimulai dengan sebuah nomor atau pemberian karakter tanda baca
- Nama-nama harus tidak mulai dengan xml (atau XML atau Xml..)
- Nama-nama tidak bisa berisi spasi

Nama apapun dapat digunakan, tidak ada kata-kata yang khusus, tetapi gagasan untuk membuat nama yang deskriptif. Menuliskannya dengan sebuah garis bawah.

Contoh: <nama_depan> , <nama_belakang>.

Nama elemen dapat sepanjang yang anda suka, tapi jangan melebihi-lebihkan. Nama harus sederhana dan pendek, seperti ini: <book_title> bukan seperti ini: <the_title_of_the_book>.

Dokumen XML sering mempunyai sebuah kesesuaian dengan database, dimana field ada yang sesuai dengan elemen didalam dokumen XML. Sebuah praktek yang baik adalah untuk menggunakan peraturan penamaan tentang database untuk elemen didalam dokumen XML.

Atribut XML

Elemen-elemen XML dapat mempunyai atribut tag awal, seperti halnya HTML.

Atribut digunakan untuk menyediakan informasi tambahan tentang elemen-elemen.

Elemen-elemen XML dapat mempunyai atribut.

Dari HTML anda akan ingat ini: . atribut SRC menyediakan informasi tambahan tentang elemen IMG.

Didalam HTML (*dan didalam XML*) atribut menyediakan informasi tambahan tentang elemen-elemen:

```

<a href="contoh.php">
```

Atribut sering menyediakan informasi yang bukan bagian dari data. Pada contoh dibawah ini, jenis file tidak relevan terhadap data, tetapi penting untuk perangkat lunak yang ingin memanipulasi elemen itu:

```
<file type="gif">informasi.gif</file>
```


Gaya Tanda kutip, " Wanita" atau 'Wanita'?

Nilai-Nilai Atribut harus selalu terlampir pada tanda kutip, tetapi baik tunggal maupun tanda kutip ganda dapat digunakan. Untuk jenis kelamin seseorang, tag orang dapat ditulis seperti ini:

```
<jeniskelamin="wanita">
```

Atau seperti ini:

```
<jeniskelamin ='wanita'>
```

Penggunaan elemen-elemen Versus Atribut

Data dapat disimpan dalam elemen-elemen anak atau didalam atribut.

Perhatikan contoh ini:

Contoh1

```
<orang>
<jeniskelamin="wanita">
  <namadepan>Anna</namadepan>
  <namabelakang>Smith</namabelakang>
</orang>
```

Contoh2

```
<orang>
  <jeniskelamin>wanita</jeniskelamin>
  <namadepan>Anna</namadepan>
  <namabelakang>Smith</namabelakang>
</orang>
```

Pada contoh jenis kelamin yang pertama adalah sebuah atribut. Pada yang terakhir, jenis kelamin adalah seorang elemen anak. Contoh kedua-duanya menyediakan informasi yang sama.

Tidak ada aturan tentang kapan untuk menggunakan atribut, dan kapan untuk menggunakan elemen-elemen anak.

Cara Favoritku

Aku suka menyimpan data didalam elemen-elemen anak.

Yang berikut tiga dokumen XML berisi informasi yang sama pada contoh sebelumnya: Sebuah atribut tanggal digunakan pada contoh yang pertama:

```
<note date="12/11/2002">
<to>Ojak</to>
<from>Jay</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
```

```
</note>
```

Sebuah elemen tanggal digunakan pada contoh yang kedua:

```
<note>
<date>12/11/2002</date>
<to>Ojak</to>
<from>Jay</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Sebuah elemen tanggal diperluas dengan menggunakan cara yang ketiga: (INI ADALAH CARA FAVORIT KU):

```
<note>
<date>
  <day>12</day>
  <month>11</month>
  <year>2002</year>
</date>
<to>Ojak</to>
<from>Jay</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Hindari Penggunaan atribut?

Perluakah anda menghindari penggunaan atribut?

Sebagian dari permasalahan dengan penggunaan atribut adalah:

- atribut tidak bisa berisi berbagai nilai-nilai (*elemen-elemen anak*)
- atribut tidak dapat diperluas (*untuk perubahan masa yang akan datang*)
- atribut tidak bisa menguraikan struktur (*elemen-elemen anak*)
- atribut menjadi lebih sukar untuk memanipulasi dengan kode program

Jika anda menggunakan atribut sebagai kontainer untuk data, anda mengakhiri dengan dokumen yang sukar untuk membaca dan memelihara. Coba untuk menggunakan **elemen-elemen** untuk menguraikan data. Penggunaan atribut hanya untuk menyediakan informasi yang tidak relevan terhadap data.

```
<note day="12" month="11" year="2002"
to="Ojak" from="Jay" heading="Reminder"
body="Don't forget me this weekend!">
</note>
```

Sebuah Perkecualian Untuk Aturan Atribut

Aturan selalu mempunyai perkecualian.

Aturan tentang atribut mempunyai satu perkecualian:

Kadang-kadang menugaskan acuan ID ke elemen-elemen. Acuan ID ini dapat digunakan untuk mengakses elemen-elemen XML pada cara yang hampir sama seperti NAMA atau atribut ID dalam HTML.

Contoh:

```
<messages>
  <note id="p501">
    <to>Ojak</to>
    <from>Jay</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>

  <note id="p502">
    <to>Jay</to>
    <from>Ojak</from>
    <heading>Re: Reminder</heading>
    <body>I will not!</body>
  </note>
</messages>
```

ID didalam contoh ini adalah sekedar counter, atau sebuah pengenal yang unik, untuk mengidentifikasi catatan yang berbeda didalam file XML, dan bukan bagian dari data catatan.

Aku sedang mencoba untuk mengatakan disini bahwa metadata (*data tentang data*) harus disimpan sebagai atribut, dan data itu sendiri harus disimpan sebagai elemen-elemen.

Editor XML

Editor XML	URL	Tipe
Amaya	http://www.w3.org/amaya/	Windows, UNIX
BEdit	http://www.bbedit.com/	Macintosh
EditML Pro	http://www.editml.com	Windows
Emile	http://www.in-progress.com/emile/	Macintosh
Merlot	http://www.merlotxml.org	Java
Visual XML	http://www.pierlou.com/visxml/index.html	Java
XML Pro	http://www.vervet.com/xmlpro.html	Java
XML Spy	http://www.xmlspy.com/	Windows
XML Writer	http://www.xmlwrite.net/	Windows

Aplikasi XML

Aplikasi XML	Deskripsi	URL
CDF	(Channel Definition Format) Mengijinkan sebuah penerbit web untuk menawarkan secara otomatis penyampaian informasi ke PC atau peralatan informasi lain	http://www.w3.org/TR/NOTE-CDFsubmit.html

CML	(Chemical Markup Language) Kode yang digunakan untuk informasi molekul dan kimia	http://www.xml-cml.org/
MathML	(Mathematical Markup Language) Digunakan untuk mengevaluasi persamaan matematika	http://www.w3.org/Math/
MusicML	(Musical Markup Language) Digunakan untuk menampilkan notasi music	http://www.tcf.nl/3.0/musicml/index.html
OFX	(Open Financial Exchange) Pertukaran yang digunakan untuk data keuangan antara lembaga keuangan, bisnis, dan konsumen melalui internet	http://www.otx.net/
SMIL	(Synchronized Multimedia Integration Language) Digunakan untuk mengedit presentasi audiovisual interaktif yang menyertakan audio, video, teks, dan tipe media lain	http://www.w3.org/AudioVideo/
VoiceXML	(Voice Markup Language) Digunakan untuk menciptakan bentuk dialog audio, audio digitalisasi, dan mengenal suara	http://www.voicexml.org/

Referensi:

Introduction to XML

<http://www.ibiblio.org/xml/>

New Perspectives on XML, Comprehensive

[http://www.course.com/downloads/additional/0188-1asf2.exe">](http://www.course.com/downloads/additional/0188-1asf2.exe)

XML Tutorial

<http://www.w3schools.com/xml/default.asp>

BIOGRAFI PENULIS



Janner Simarmata. Lahir di Aek Nabara, 07 Januari 1976. Tamat dari STM GKPS Pematang Siantar tahun 1995. Menyelesaikan program S1 pada jurusan Teknik Informatika di STMIK BANDUNG pada tahun 2000. Pernah mengajar di beberapa Perguruan Tinggi Swasta seperti: STMIK Mikroskil, STMIK Multimedia Prima, Unika Santo Thomas Sumatera Utara. Pada tahun 2004 melanjutkan studi pada program S2 (M.Kom) pada jurusan Ilmu Komputer Universitas Gadjja Mada sampai sekarang.

Informasi lebih lanjut tentang penulis:

KEYWORD: *Janner Simarmata*

Email: sijanner@yahoo.com