



XML

Minggu 11

Sumber : Materi Pengantar XML, Moh Junaedi,
www.ilmukomputer.com



XML

- Apakah XML
- XML vs HTML
- XML Struktur
- DTD
- XLST



Apakah XML ?

- **eXtensible Markup Language** adalah meta-language yang menerangkan tentang isi dari suatu document (self describing data)

Java = Portable Program

XML = Portable Data

- XML tidak menerangkan *tag-set* atau grammar dari XML itu sendiri
- Menggunakan DTD (Document Type Definition) untuk menerangkan data
- XML bukan pengganti HTML



XML vs HTML ?

Kenapa tidak HTML ? , karena HTML :

- Tidak memberikan “informasi” tentang isi dari sebuah halaman page
- Tampilan tidak bisa di parsing, dan tidak bisa digunakan kembali
- Terbatas hanya untuk melakukan format tampilan pada web browser



XML vs HTML

- XML juga menggunakan elemen yang ditandai dengan tag pembuka (diawali '<' dan diakhiri dengan '>'), tag penutup (diawali '</' diakhiri dengan '>')
- XML mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri.
- XML terutama digunakan untuk menyusun dan menyajikan informasi dengan format yang tidak mengandung format standar contoh : paragraph, table dan lain sebagainya



XML vs HTML

- Contoh XML

<pesan>

<dari>MIS Manager</dari>

<buat>HRD Manager</buat>

<buat>Bagian rekrut</buat>

<buat>Computer Suport team</buat>

<subyek>Permohonan Tenaga kerja baru</subyek>

<isi>Mohon diberikan tenaga kerja baru untuk mengisi lowongan di Departemen MIS</isi>

</pesan>

```
- <pesan>
  <dari>MIS Manager</dari>
  <buat>HRD Manager</buat>
  <buat>Bagian rekrut</buat>
  <buat>Computer Suport team</buat>
  <subyek>Permohonan Tenaga kerja baru</subyek>
  <isi>Mohon diberikan tenaga kerja baru untuk mengisi lowongan di
    Departemen MIS</isi>
</pesan>
```



Aplikasi XML

- Konfigurasi File :
 - Arsitektur J2EE
- Media untuk pertukaran data
- B2B transaksi
 - Electronics Bussines Order (ebXML)
 - Financial Exchange (IFX)
 - Messaging Exchange (SOAP)



Bagian-Bagian Dokumen XML

Sebuah dokumen XML terdiri dari bagian-bagian yang disebut dengan node. Node-node itu adalah:

- **Root Node**, yaitu yang melingkupi keseluruhan dokumen. Dalam satu dokumen XML hanya ada satu root node
- **Element Node**, yaitu bagian dari dokumen XML yang ditandai dengan tag pembuka dan tag penutup, atau bisa juga sebuah tag tunggal dengan elemen kosong, seperti `<anggota nama =“budi”/>`
- **Attribut Node**, termasuk nama dan nilai atribut yang ditulis pada tag awal sebuah elemen atau pada tag tunggal
- **Text Node**, adalah text yang merupakan isi dari sebuah elemen, ditulis diantara tag pembuka dan tag penutup
- **Comment Node**, baris yang tidak dieksekusi
- **Processing Instruction Node**, perintah pengolahan dalam dokumen XML. Node ini diawali dengan karakter `<?` dan diakhiri dengan `?>`
- **NameSpace Node**, node ini mewakili deklarasi namespace



Contoh Dokumen XML

Processing Instruction

Standard Header

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<?xml-stylesheet  
  type="text/xsl"  
  href="belajar.xsl"  
?>
```

```
<!--Dokumen ini menjelaskan tentang isi buku-->
```

```
<Buku
```

```
  Judul="Manajemen Data dan Informasi dengan  
  XML/XSLT"
```

```
  Penulis="Moh Junaedi">
```

```
  <pengantar/>
```

```
  <Bab No="1">Berkenalan dengan XML/XSLT</Bab>
```

```
  <Bab No="2">Lebih dekat dengan XML</Bab>
```

```
  <Bab No="3">Elemen-elemen XSLT</Bab>
```

```
  .
```

```
  .
```

```
  .
```

```
</Buku>
```

Attribute

Element kosong

Rootnode

Comment



XML Element

XML Elements

- Menggunakan building block yang sama seperti HTML : Element, Attribute dan Value
- Sebuah element mengandung tag pembuka dan tag penutup :
Contoh : `<animal>Lion</animal>`
- `<animal class="mammals">Lion</animal>`,
maka : `animal` adalah *element*, `class` adalah *attribute* dan `mammals` adalah nilai dari attribute



Aturan XML

- Dokumen XML harus “well-formed” yang artinya harus memenuhi aturan2 sebagai berikut :
 - Harus mempunyai sebuah root element yang berisikan element2 lainnya.
 - Semua elemen harus diakhiri dengan tag penutup,
Contoh :
`<picturefile=“test.jpg”/>`
`<name>Lion</name>`
 - Element tidak boleh overlapping



Aturan XML

- XML bersifat case sensitive
- Untuk nilai harus berada dalam kutip satu (‘) atau kutip dua (“)
- Simbol2 special harus dideklarasikan pada DTD



Aturan XML

Pendeklarasian file sebagai sebuah file XML :

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

Penulisan komentar pada file xml :

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



DTD

- DTD berfungsi untuk menjelaskan spesifikasi dan aturan terhadap element2 dan attributes yang harus di miliki oleh sebuah xml dokument.
- Kumpulan dari aturan2 tersebut disebut dengan SCHEMA
- SCHEMA tidak wajib dimiliki, tetapi di perlukan untuk memastikan konsistensi dokumen



Unsur –Unsur DTD

- Element, satu blok data yang diawali dengan tag pembuka dan tag penutup
- Attribute, informasi pendukung element yang disertakan pada tag pembuka
- Entity, karakter pengganti untuk sekumpulan informasi yang didefinisikan



DTD

Mendefinisikan elemen pada DTD :

- `<!ELEMENT animal (lion)>` : element animal hanya boleh memiliki satu elemen lion.
- `<!ELEMENT picture EMPTY>` : element picture tidak mempunyai elemen lain.
- `<!ELEMENT animal ANY>` : element animal dapat berisikan element lainnya.

Mendefinisikan element yang berisikan text

```
<!ELEMENT name (#PCDATA)>
```




DTD

Mendefinisikan element yang mempunyai beberapa element

```
<!ELEMENT animal(name, weight)>
```

Berdasarkan aturan diatas maka element harus mempunyai element
nama dan weight :

```
<animal>
```

```
<name>lion</name>
```

```
<weight>350 pounds</weight>
```

```
</animal>
```



DTD

Mendefinisikan element dengan beberapa pilihan element

```
<!ELEMENT animal ((name, weight) | (picture))>
```

Berdasarkan aturan diatas maka element animal harus mempunyai elemen name dan weight atau hanya punya element picture.

Pendefinisian Unit pada element :

```
<!ELEMENT animal (name+, weight?, picture, subspecies*)
```

Artinya :

- name harus tampil setidaknya satu kali
- weight dapat tampil sekali atau tidak tampil sama sekali
- picture hanya harus tampil sekali
- subspecies dapat tampil beberapa kali pun atau tidak tampil sama sekali



Attribute

- Terkadang penggunaan attribute dapat lebih berarti dari pada memecah element menjadi sub-sub element.
- Kedua element ini berarti sama :
 - `<population animal="lion">80</population>`
 - `<population><animal>lion</animal><quantity>80</quantity></population>`
- Attribute harus dideklarasikan juga pada DTD untuk dapat di pergunakan
 - `<!ELEMENT population (#PCDATA)>`
`<!ATTLIST population year CDATA #IMPLIED>`



Attribute

- Mendefinisikan Attribute wajib (Required Attribute) :

```
<!ELEMENT population (#PCDATA)>
```

```
<!ATTLIST population year (2000 | 2001) #REQUIRED>
```

Berarti : element populasi harus mempunyai attribute year yang bernilai 2000 atau 2001

```
<population year="2000">80</population>
```

- Mendefinisikan Attribute default (Default Attribute) :

```
<!ELEMENT population (#PCDATA)>
```

```
<!ATTLIST population year CDATA "2000">
```



Attribute

- Mendefinisikan Attribute default (Default Attribute) :

```
<!ELEMENT population (#PCDATA)>
```

```
<!ATTLIST population year CDATA #FIXED "2000">
```

Invalid : <population year="2001">80</population>

Valid : <population year="2000">80</population>

Valid : <population>80</population>

- Mendefinisikan Unique Attribute :

```
<!ELEMENT animal (name)>
```

```
<!ATTLIST animal code ID #REQUIRED>
```



Deklarasi DTD

Internal :

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

```
<!DOCTYPE animal [ ]>
```

```
<animal> </animal>
```

- animal is the root element

- The DTD berada dalam kurung “[]” brackets pada “DOCTYPE”

Eksternal :

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

```
<!DOCTYPE animal SYSTEM "http://www.animals.com/xml/animal.dtd">
```



Entity

- **Entity XML digunakan untuk menggantikan kalimat yang panjang atau satu blok elemen yang sering digunakan dengan sebuah pengenal singkat. Misalnya ingin menggantikan kalimat “Manajemen Data dan Informasi dengan XML/XSL” dengan entity &judul;**
- **Sekali entity didefinisikan didalam DTD, maka dapat digunakan dimana saja pada seluruh dokumen XML**

```
<?xml version="1.0" encoding="iso-8859-1">
<!DOCTYPE organisasi [
<!ENTITY judul "Manajemen data dan informasi dengan XML/XSLT"> ]>
<resensi>
<buku judul="&judul;">
<ulasan>Buku yang berjudul &judul; ini ditulis oleh Moh Junaedi..</ulasan>
</buku>
</resensi>
```



XSLT

- **XSLT adalah eXtensible StyleSheet Language:Transformation, merupakan 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 dilayar, dicetak di kertas atau didengarkan ditelinga.**
- **Dibagi menjadi 2 bagian proses:**
 1. **Transformasi Struktural yang meliputi pengumpulan, pengelompokkan dan pengurutan data maupun penyusunan ulang, penambahan dan penghapusan tag dan atribut (disebut dengan XSLT)**
 2. **Proses merubah format menjadi menjadi pixel dilayar, nohtah tinta di kertas atau nada di speaker (disebut XSLFO)**
- **Hasil keluaran XSLT bisa berupa HTML, Text file atau XML dengan format baru.**
- **CSS dapat juga digunakan untuk menampilkan dokumen XML agar lebih menarik dilihat di browser, tetapi CSS tidak mampu melakukan tugas-tugas yang rumit seperti memformat angka desimal, menjumlah, menghitung rata-rata, menampilkan gambar dan lain-lain**



XSLT Processor

- XSLT Processor atau 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.
- XSLT Processor yang dapat digunakan bila menggunakan Internet Explorer versi 5.5 kebawah adalah MSXML atau MSXML2



Contoh Sederhana

Untuk memahami bagaimana XML dan XSLT bekerja, perhatikan contoh berikut ini:

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<berita>Saya sedang belajar XML</berita>
```

- Simpanlah dokumen diatas dengan nama belajar.xml
- XSLT yang diperlukan adalah sebagai berikut

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

```
<xsl:template match ="/>
```

```
<html>
```

```
<head>
```

```
<title>XML Saya yang Pertama</title>
```

```
</head>
```

```
<body>
```

```
<p><xsl:value-of select="berita"/></p>
```

```
</body>
```

```
</html>
```

```
</xsl:template>
```

```
</xsl:stylesheet>
```

- Simpanlah dokumen XSLT ini dengan nama belajar.xsl



Hasil

- Bila menggunakan MSXML3 sebagai parser, maka harus ditambahkan satu baris heading standard sehingga dokumen XML akan menjadi seperti berikut:

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<?xml-stylesheet type="text/xsl" href="belajar.xsl"
```

```
<berita>Saya sedang belajar XML</berita>
```

Maka akan tampil di browser adalah sebagai berikut :

