Pemrograman Basis Data Berbasis Web

Pertemuan Ke-8

(XML)

Sub Pokok Bahasan

- XML?
- Keuntungan XML
- Perbedaan XML dan HTML
- Contoh sederhana dokumen XML
- Struktur *Tree*
- *Tag* XML
- Aturan dalam Sintaks XML
- Tampilan XML dalam *Web Browser*
- Memformat XML
- Aplikasi XML

XML?

- eXtensible Markup Language (XML) adalah sebuah tipe bahasa baru yang dikembangkan untuk *web* yang berbeda dengan tipe bahasa *scripting* atau pemrograman lain yang terdapat sebelumnya
- XML tidak dikonsentrasi untuk pemrosesan dan penampilan data tetapi terutama dimaksudkan untuk memberitahu komputer apa arti sesungguhnya dari data yang disertakan
- XML bukanlah suatu cara untuk merancang halaman *web* dan XML tidak akan mengubah cara seseorang membangun *website*. Ini telah membuat banyak orang percaya bahwa XML tidak berguna, sepertinya tidak mendapatkan cara yang akan memberikan keuntungan kepadanya

Keuntungan XML

- Dengan menggunakan XML, komputer akan dapat memahami dengan pasti apa yang sesungguhnya ada di halaman tersebut. Tanpa XML, suatu *search engine* atau komputer lain tidak akan mengetahui yang sebenarnya ada di dalam halaman *web* karena semuanya dianggap sebagai kumpulan huruf dan bilangan, dengan format HTML yang menyertainya. Komputer bahkan tidak dapat memberitahukan apakah yang ada di halaman itu sebuah *heading*, teks atau sekedar iklan. Jika *search engine* mengetahui dengan pasti apa yang ada di suatu halaman, *engine* tersebut akan dapat menyediakan hasil pasti kepada seseorang yang melakukan pencarian informasi, tidak dengan pencocokan tidak akurat dan halaman yang setengah relevan.
- Karena XML digunakan untuk mendefinisikan apa arti dari data dan bukan bagaimana data ditampilkan maka XML dengan mudah menggunakan data yang sama pada beberapa platform berbeda

Perbedaan XML dan HTML

- XML digunakan dengan cara yang menyerupai HTML. Akan tetapi terdapat perbedaan besar antara keduanya, yaitu :
  - HTML digunakan untuk mendeskripsikan bagaimana data diformat
  - XML digunakan untuk mendeskripsikan apa arti sebenarnya dari data

Contoh Sederhana Dokumen XML

```
<?xml version="1.0"?>
<academic>
<student>
<id>12345</id>
<name>Almira Wijaya</name>
</student>
</academic>
```

Keterangan:

- Baris pertama adalah baris Deklarasi XML untuk mendefinisikan XML *version* (1.0)
- Baris kedua merupakan elemen root (<academic>) dari dokumen
- Baris tiga merupakan elemen child dari root <academic>
- Baris empat dan lima adalah elemen subchild dari child <student>

Struktur *Tree*

- Dokumen XML harus memiliki elemen *root*. Elemen ini merupakan "*parent*" dari seluruh elemen yang ada di dalam dokumen
- Elemen-elemen di dalam suatu dokumen XML membentuk sebuah dokumen *tree*

```
<root>
  <child>
        <subchild>.....</subchild>
  </child>
</root>
```

Tag XML

- XML terlihat dan distruktur sangat menyerupai HTML. Keduanya menggunakan *tag-tag* untuk membatasi data yang dirujuk (dijadikan acuan). Keduanya dapat menggunakan *tag* bersarang (*nested tag*) dan juga dapat mempunyai atribut-atribut yang ditambahkan ke *tag*-nya
- Di dalam XML tidak dibatasi dengan penggunaan *tag* normal, *tag pre-defined* seperti “font” dan “br”, bahkan *tag-tag* di dalam dokumen XML **harus dibuat sendiri** oleh si pemrogram. *Tag-tag* tersebut dapat dinamai sesuai keinginan dan dapat digunakan untuk menyajikan sesuatu yang diinginkan. Ini adalah fitur yang tidak dapat ditemukan pada bahasa *scripting* web lain
- *Tag* yang digunakan dalam XML yang sangat menyerupai kontruksi HTML, juga terlihat seperti *tag* HTML. *Tag* dibentuk oleh suatu kata atau sejumlah kata yang dibatasi di dalam tanda <> dan </>

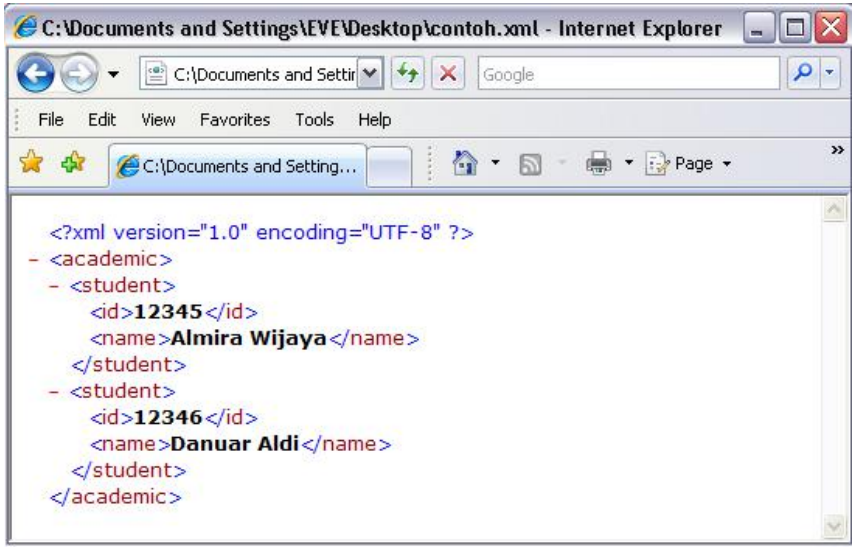
Aturan dalam Sintaks XML

- **Seluruh elemen XML harus memiliki *tag* penutup**
  Contoh:
  <student> </student>
- ***Tag* XML adalah *case-sensitive***
  Contoh:
  <Name>Almira Wijaya</name> adalah salah
  Elemen XML harus disarangkan dengan tepat
  Contoh:
  ```
  <student>
  <id>12345</id>
  <name>Almira Wijaya</name>
  </student>
  ```
- **Dokumen XML harus memiliki sebuah elemen *Root***
- **Nilai atribut XML harus diberi tanda *quote* (“ “)**
  Contoh:
  ```
  <student sex=“female”>
  <id>12345</id>
  <name>Almira Wijaya</name>
  </student>
  ```

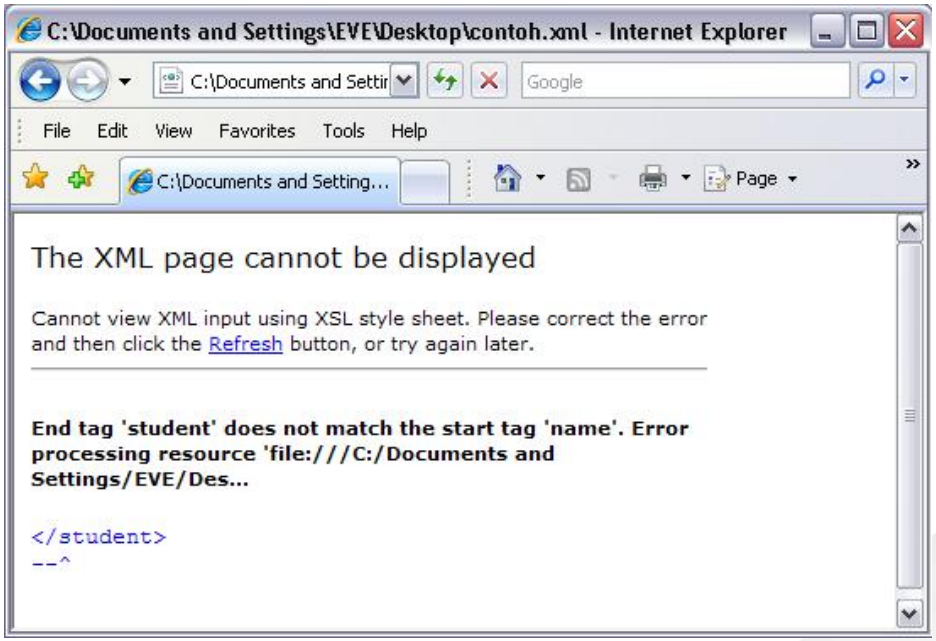
Tampilan XML dalam *Web Browser*

Script XML:

```
<?xml version="1.0"?>
<academic>
<student>
<id>12345</id>
<name>Almira Wijaya</name>
</student>
<student>
<id>12346</id>
<name>Danuar Aldi</name>
</student>
</academic>
```



- Jika terdapat suatu kesalahan di dalam dokumen XML, maka web browser akan menyediakan suatu pesan bantuan yang memberitahukan di mana error tersebut terjadi dan menampilkan potongan kode yang salah.



Memformat XML

- Terdapat 2 cara untuk memformat data XML untuk menampilkannya pada *web browser*, yaitu dengan menggunakan:
  1. CSS (*Cascading Style Sheets*)
  2. XSL (*eXstensible Stylesheet Language*)

Memformat XML dengan CSS

- Selain dapat digunakan untuk memformat dokumen HTML, CSS juga dapat digunakan untuk memformat dokumen XML. CSS dapat melakukan redefinisi tag HTML, memungkinkan tag tersebut disajikan dalam cara berbeda. Dengan cara yang sama, CSS dapat digunakan untuk mendefinisikan bagaimana tag XML ditampikan.

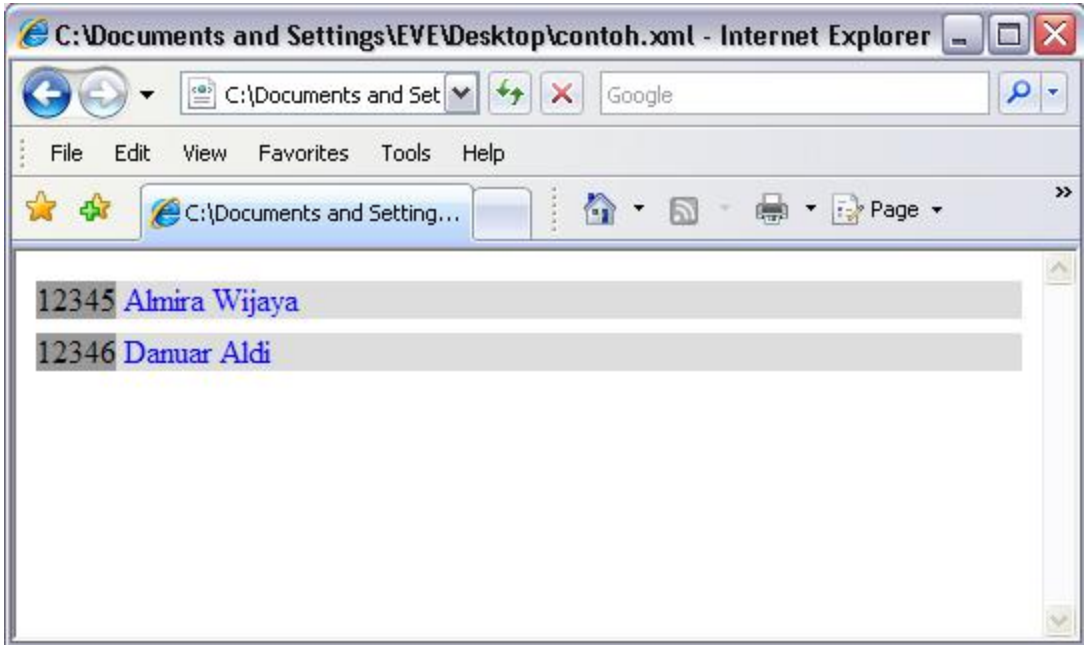
- Script contoh.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css"
    href="estyle.css"?>
<academic>
<student>
<id>12345</id>
<name>Almira Wijaya</name>
</student>
<student>
<id>12346</id>
<name>Danuar Aldi</name>
</student>
</academic>
```

Script estyle.css:

```
academic
{
background-color: #ffffff;
width: 100%;
}
student
{
display: block;
background-color: #DDDDDD;
margin-bottom: 5pt;
}
id
{
background-color: #999999;
margin-bottom: 12pt;
}
name
{
color: #0000FF;
font-size: 12pt;
}
```

Tampilan dokumen di dalam web browser setelah diformat dengan menggunakan CSS:



Memformat XML dengan XSL

- XSL merupakan singkatan dari *eXstensible Stylesheet Language*, dan merupakan suatu bahasa yang dikembangkan untuk memformat dokumen XML.
- Untuk menformat kode maka harus dibuat terlebih dahulu suatu stylesheet XLS. Meskipun XLS adalah sebuah bahasa yang lengkap dan untuk menyelesaikan masalah di atas cukup memanfaatkan beberapa fiturnya saja.

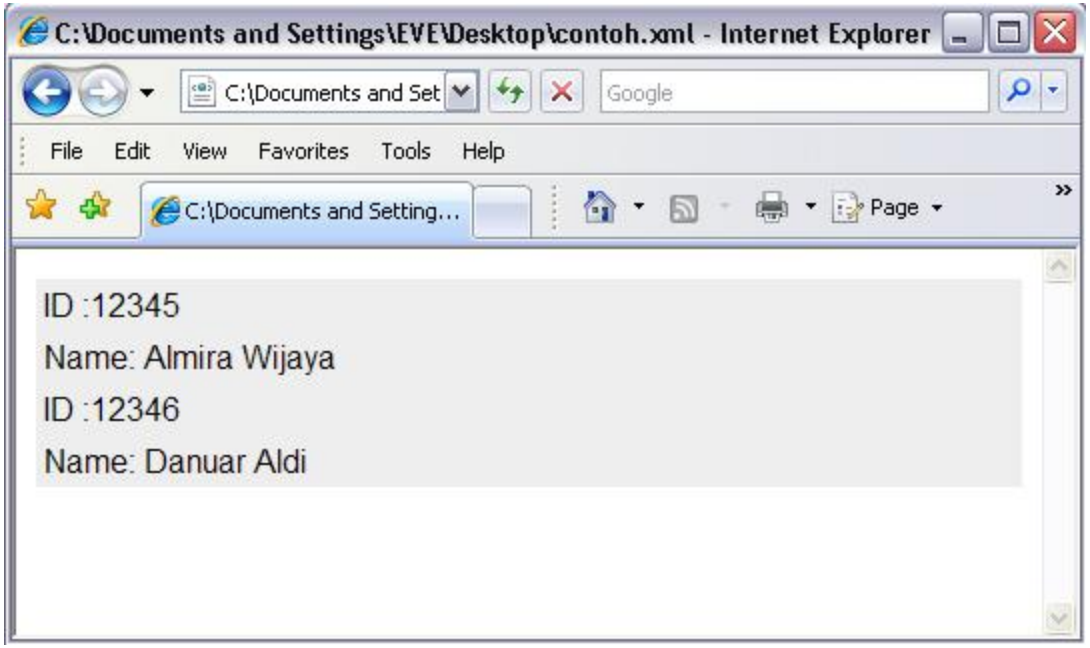
Script estyle.xsl

- Script contoh.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
    href="estyle.xsl"?>
<academic>
<student>
<id>12345</id>
<name>Almira Wijaya</name>
</student>
<student>
<id>12346</id>
<name>Danuar Aldi</name>
</student>
</academic>
```

```
<?xml version="1.0"?>
<HTML xmlns:xsl="http://www.w3.org/TR/WD-
  xsl">
<BODY STYLE="font-family:Arial, helvetica,
  sans-serif; font-size:12pt;
  background-color:#FFFFFF">
<xsl:for-each select="academic/student">
    <DIV STYLE="background-
      color:#EEEEEE; padding:4px">
    <SPAN
      STYLE="color:black">ID  :<xsl:value-of
      select="id"/></SPAN>
    </DIV>
    <DIV STYLE="background-
      color:#EEEEEE; padding:4px">
    <SPAN STYLE="color:black">Name:
      <xsl:value-of select="name"/></SPAN>
    </DIV>
</xsl:for-each>
</BODY>
</HTML>
```

Tampilan dokumen di dalam web browser setelah diformat dengan menggunakan XSL:



- *Scripting* yang ada di dalam file **estyle.xsl** menggunakan tag-tag HTML DIV dan SPAN dan dikombinasikan dengan sedikit kode XSL. Tag DIV dan SPAN memberikan area halaman yang berisi penformatan. Dokumen XSL sebenarnya hanya sebuah halaman HTML dengan sedikit kode XLS di tambahkan ke dalamnya
- Header standar untuk dokumen XSL:

  <?xml version="1.0"?>

  <HTML xmlns:xsl="http://www.w3.org/TR/WD-xsl">
- *Looping*:

  <xsl:for-each select="academic/student">

  Kode di atas bekerja seperti sebuah "*loop for*" di dalam bahasa pemrograman. Kode tersebut memberitahu *browser* untuk melakukan perulangan sepanjang (terhadap) semua item <student> di dalam *tag* <academic>
- Pencarian nilai elemen:

  ID: <xsl:value-of select="id"/>

  Dengan menggunakan XSL (karena XSL sebenarnya hanya dokumen HTML dengan kode tambahan di dalamnya), dalam *browser* dapat ditampilkan tulisan "**ID :**" sebelum nilai dari *tag* ditampilkan. Bagian kedua dari baris ini memberitahu *browser* untuk mengeluarkan nilai dari *tag* <id> dalam posisi *tag* XLS

Aplikasi XML

- XML telah banyak digunakan sebagai bagian dari deskripsi data dalam aplikasi, banyak digunakan oleh vendor besar seperti Microsoft dan Sun Micro System. Platform dot Net dan Java adalah dua software besar yang memberikan dukungan sangat baik kepada XML. Bahasa pemrograman seperti PHP dan JSP juga mendukung XML Bahkan aplikasi seperti Microsoft Word dan Open Office dapat menghasilkan halaman HTML dan nantinya akan disebut sebagai open document.
- XMLNews adalah sebuah sistem yang memungkinkan *news stories* disimpan sebagai XML. Dengan menggunakan tag-tag seperti <headline>, <byline>, <location> dan <story> halaman web dan sistem dapat dikembangkan - akan mengambil data XML dan akan menampilkannya sebagai suatu halaman web yang terformat dengan benar. Kenyataannya, cerita yang sama dapat ditampilkan pada suatu WAP phone, news website, headlines news ticker, news e-mail, SMS message atau di dalam suatu potongan software, semua dari file source yang sama. Sehingga pada aplikasinya, sebuah cerita dapat ditulis sekali oleh seorang jurnalis tetapi didistribusikan ke seluruh dunia dalam berbagai format. Informasi lebih lanjut dapat dilihat di XMLNews.org.

Summary

- XML (eXtensible Markup Language) dimaksudkan untuk memberitahu komputer apa arti sesungguhnya dari data yang disertakan bukan untuk pemrosesan dan penampilan data
- XML sangat membantu dalam penggunaan *search engine* dan juga dapat diaplikasikan pada platform yang berbeda-beda
- Perbedaan antara XML dan HTML adalah bahwa XML digunakan untuk mendeskripsikan apa arti sebenarnya dari data sedangkan HTML digunakan untuk mendeskripsikan bagaimana data diformat
- Dokumen XML harus memiliki elemen *root* yang merupakan "*parent*" dari seluruh elemen yang ada di dalam dokumen. Sedangkan elemen-elemen di dalam suatu dokumen XML membentuk sebuah dokumen *tree*
- *Tag-tag* yang digunakan di dalam XML tidak perlu berdasarkan *tag-tag predefined* seperti halnya yang digunakan di dalam HTML. Akan tetapi tag-tag di dalam XML dapat diciptakan sendiri oleh si pemrogramn sesuai dengan keinginan dan kebutuhan
- Terdapat lima aturan dalam sintaks XML, yaitu: seluruh elemen XML harus memiliki *tag* penutup, *tag* XML adalah *case-sensitive,* elemen XML harus disarangkan dengan tepat, dokumen XML harus memiliki sebuah elemen *root, n*ilai atribut XML harus diberi tanda *quote* (" ")

Daftar Pustaka

- Chris Bates [2006]. **Web Programming: Building Internet Applications**, Third Edition, John Wiley & Sons Ltd, England.
- Husni [2007]. **Pemrograman Database Berbasis Web**, Graha Ilmu, Yogyakarta.
- http://www.w3schools.com/xml