

## Penyajian Dokumen XML dengan Teknik Pengikatan Data

Aji Supriyanto

Fakultas Teknologi Informasi Universitas Stikubank Semarang

e-mail : ajisup@gmail.com

**ABSTRAK** : Fleksibilitas XML mengakibatkan pemakaian dokumen XML semakin banyak diminati para pengguna aplikasi berbasis web, hal ini dikarenakan fasilitas yang didukung lebih lengkap dibanding dengan HTML. Sebuah dokumen XML yang valid harus memenuhi syarat *well-formed* dan dalam bentuk DTD, syarat ini sebagai landasan untuk mengembangkan penyajian dokumen dalam berbagai bentuk. Teknik yang digunakan dalam penyajian dokumen XML dapat berbagai macam, hal ini sangat tergantung pada kebutuhan penyajian. Teknik pengikatan merupakan teknik yang mengaitkan sebuah dokumen XML ke sebuah halaman HTML, selanjutnya mengaitkannya ke elemen HTML standar pada elemen XML individual. Elemen HTML selanjutnya secara otomatis menampilkan isi elemen XML dimana mereka terikat. Teknik tersebut akan memberikan kemudahan dalam menyajikan dokumen XML dalam tampilan browsingnya, dengan format yang telah ditentukan sesuai dengan keinginnya.

**Kata kunci** : XML, HTML, pengikatan data, *well-formed*, DTD.

### PENDAHULUAN

XML (*eXtensible Markup Language*) merupakan sebuah himpunan bagian (*subset*) dari *Standard Generalized Markup Language* (SGML), yang bertujuan agar SGML secara generik dapat melayani, menerima, dan memproses dalam Web dengan cara seperti yang dimungkinkan *HyperText Markup Language* (HTML) saat ini. XML didesain untuk kemudahan implementasi dan interoperabilitas dengan SGML maupun HTML. XML adalah bahasa *markup* yang dirancang untuk penyampaian informasi melalui *World Wide Web* (WWW) atau sering disebut web saja. (W3C, 2000).

XML banyak mendukung aplikasi dibanding HTML yang mendukung dirinya sendiri dengan hanya satu aplikasi yaitu *web browsing*. XML secara khusus tidak mengatakan aplikasi lain apa yang akan didukung. Terlepas dari *tool* dan keahlian SGML yang sudah ada yang dapat siap untuk diadaptasi ke XML, ada banyak kekuatan yang melekat didalam SGML yang tidak diteruskan ke dalam spesifikasi HTML. Didalam HTML, misalnya tidak dapat

menyimpan dari sekumpulan tag yang telah ditentukan. Kemampuan XML untuk melibatkan semua tag sehingga mengharuskan tujuan tertentu atas nama "*Extensible*" merupakan warisan langsung dari SGML.

Setiap dokumen XML memiliki dua struktur utama, yaitu struktur logika dan struktur fisik. Secara fisik, dokumen terdiri dari satuan-satuan yang disebut *entitas*. Entitas adalah string khusus dari karakter-karakter yang memiliki maksud untuk penyimpanan. Suatu entitas bisa terdapat pada entitas lain sehingga entitas ini juga termasuk ke dalam dokumen.

Secara logika, dokumen tersusun dari beberapa deklarasi, elemen, komentar, acuan karakter, dan instruksi pemrosesan (*Processing Instruction-PI*), yang kesemuanya disebutkan dalam dokumen oleh suatu *markup* secara eksplisit. Struktur logika dan struktur fisik harus tersarang dengan benar.

Terdapat 3 bagian dokumen XML yaitu *prolog*, *root element* atau disebut *Document element*, dan *epilog*. Namun sebuah dokumen XML utamanya memiliki dua bagian yaitu *prolog* dan elemen dokumen atau elemen *root*.

(Marchal, 2000). Hanya elemen *root* yang diperlukan dalam dokumen yang tersusun baik (*well formed*), sedangkan untuk dokumen valid harus memasukkan prolog. Elemen epilog berada pada akhir dokumen yang keberadaanya hampir tidak pernah diperlukan.

### Dokumen XML yang *Well-Formed* dan yang valid

XML merupakan bahasa *markup* yang penulisannya harus memenuhi struktur yang benar yaitu memenuhi unsur validitas dan tersusun dengan baik (*well formed*). XML dikatakan *valid* jika dapat menjelaskan dirinya secara lengkap tentang segala sesuatu yang berkaitan dengan struktur dan isi dokumen, terdapat dalam dokumen dasar itu sendiri atau dalam file penolong tertentu yang dirujuk dalam dokumen dasar tersebut. Sedangkan dokumen yang *well formed* berisi hal-hal yang paling sederhana yaitu tentang dokumen XML yang dapat diproses oleh perangkat XML (prosesor XML).

Hubungan istilah valid dan *well formed* adalah semua dokumen yang valid juga disusun dengan baik. Namun sebaliknya dokumen yang disusun dengan baik belum tentu valid. Sehingga Dokumen XML Valid adalah dokumen yang memiliki struktur dan isi elemennya secara formal dinyatakan dalam DTD (*Document Type Definition*). Dokumen tersusun baik (*well formed*) adalah dokumen yang jika tidak memerlukan pemakaian DTD dalam memahami struktur dan isi elemennya, tetapi masih mengikuti prinsip-prinsip umum XML, seperti tag *nested* (tersarang) secara tepat. DTD secara formal menyatakan struktur dan isi elemen (tag, hubungan antar tag-tag yang berbeda, dan seterusnya) dari dokumen XML valid yang diberikan.

### MODEL PENGIKATAN

Pengikatan data merupakan bentuk pengaitan sebuah dokumen XML ke sebuah halaman HTML, selanjutnya mengaitkannya ke elemen HTML standar seperti SPAN atau TABLE ke elemen XML individual. Elemen HTML selanjutnya secara otomatis menampilkan isi elemen XML dimana mereka terikat.

### Pengikatan dengan Mengaitkan Dokumen XML ke halaman HTML

Dalam menampilkan sebuah dokumen XML ke halaman HTML, harus mengaitkan dokumen tersebut ke halaman HTML. Langkah ini biasanya dilaksanakan dengan memasukkan sebuah elemen HTML yang dinamai XML dalam halaman HTML. Untuk memasukkan dokumen XML ke halaman HTML dapat dilakukan dengan dua cara, pertama yaitu memasukkan seluruh teks dokumen XML antara tag-awal dan tag-akhir XML, dan kedua yaitu elemen HTML XML dibiarkan kosong dan hanya memasukkan URL dari dokumen XML. Contoh elemen berikut ada didalam elemen HTML yang menghubungkan dokumen XML Buku.xml ke halaman tersebut:

```
<XML ID="dsoBuku" SRC="Buku.xml"></XML>
```

- a. Bentuk pertama, memasukkan seluruh teks dokumen XML antara tag-awal dan tag-akhir XML. Contohnya:

```
<HTML>
<HEAD>
  <TITLE>Daftar Buku
  Perpustakaan</TITLE>
</HEAD>
<BODY>
  <XML ID="dsoBuku">
    <?xml version="1.0"?>
  <BUKU>
    <JUDUL>Pengantar Teknologi
    Informasi</JUDUL>
    <PENGARANG> Aji
    Supriyanto</PENGARANG>
    <PENERBIT>Salemba Infotek</PENERBIT>
    <HALAMAN>499</HALAMAN>
    <HARGA>Rp. 60.000</HARGA>
  </BUKU>
</XML>
<!-- Elemen HTML lainnya -->
</BODY>
</HTML>
```

- b. Bentuk kedua, elemen HTML XML dibiarkan kosong, dan hanya memasukkan URL dari dokumen XML. Contoh file HTML-nya:

```
<HTML>
<HEAD>
  <TITLE>Daftar Buku
  Perpustakaan</TITLE>
</HEAD>
```

```
<BODY>
  <XML ID="dsoBuku"
  SRC="Buku.xml"></XML>
  <!-- Elemen HTML lainnya -->
</BODY>
</HTML>
```

Elemen bernama XML diatas bukan sebuah elemen XML, melainkan sebuah elemen HTML yang berisi elemen XML. Sehingga jika sintaks XML elemen kosong, <XML ID="dsoBuku.xml"/>, akan menjadi ilegal. Contoh file XML yang diikatkan pada file HTML. File Buku.xml :

```
<?xml version="1.0"?>
<!-- Nama File :Buku.xml -->

<BUKU>
  <JUDUL>Pengantar Teknologi
  Informasi</JUDUL>
  <PENGARANG> Aji
  Supriyanto</PENGARANG>
  <PENERBIT>Salemba
  Infotek</PENERBIT>
  <HALAMAN>499</HALAMAN>
  <HARGA>Rp. 60.000</HARGA>
</BUKU>
```

Bentuk kedua lebih tunduk pada filosofi dasar XML dalam menjaga data itu sendiri (dokumen XML) terpisah dari informasi pemformatan dan pemrosesan (*style sheet*). Cara kedua ini lebih mudah dalam mengelola dokumen XML, khususnya dokumen tunggal yang ditampilkan pada sejumlah halaman HTML yang berbeda. Dan jika diberikan atribut SRC URL, maka bentuknya menjadi :

```
<XML ID="dsoBuku"
SRC=http://www.my_domain.com/dokumen
/Buku.xml>
</XML>
```

dan jika alamat lokasi file Buku.xml sama dengan halaman HTML-nya maka dapat dideklarasikan lebih singkat, yaitu :

```
<XML ID="dsoBuku" SRC=Buku.xml>
</XML>
```

Namun penggunaan URL relatif umum digunakan dan aman.

### Penyimpanan data XML

Pengenal yang diberikan ke atribut ID menyatakan DSO (*Data Source Object*). DSO merupakan sebuah objek program yang menyimpan atau menyimpan sementara (*caching*) data XML dan memberikan akses ke data ini. DSO menyimpan data XML sebagai sebuah *recordset*, yaitu sebuah kumpulan record dan field. Contohnya jika memasukkan dokumen Pustaka.xml dalam halaman dokumen, maka setiap elemen BUKU adalah record, dan setiap elemen anaknya (JUDUL, PENGARANG, PENERBIT, HALAMAN, HARGA) adalah sebagai field.

Jika mengikat sebuah elemen HTML ke sebuah elemen XML, DSO otomatis memberikan nilai pada elemen XML dan menangani semua detailnya. DSO juga mengijinkan langsung untuk mengakses dan memanipulasi recordset yang tersimpan melalui sekumpulan metode, properti, dan even.

Metode adalah fungsi yang bisa dipanggil dari suatu halaman untuk mengakses atau mengubah suatu recordset, misalnya bisa menggunakan metode untuk berpindah diantara record-record. Properti adalah seting fitur yang ada dan bisa dibaca, dan terkadang dapat diubah dari halaman tersebut, misalnya bisa membaca sebuah properti yang menyatakan telah mencapai record terakhir. Even adalah kejadian misalnya perubahan record yang dapat ditangani dari sebuah fungsi script yang dimasukkan dalam halaman tersebut.

### Pengikatan dengan Mengaitkan elemen HTML ke elemen XML

Untuk mengaitkan atau mengikatkan elemen HTML ke elemen XML dapat dilakukan dengan dua cara yaitu :

- Pengikatan data tabel, yaitu mengaitkan sebuah elemen tabel HTML ke data XML sehingga tabel otomatis menampilkan seluruh kumpulan record yang dimiliki dokumen XML.
- Pengikatan data record-tunggal, yaitu mengikat atau mengaitkan elemen HTML nontabel (misalnya SPAN) ke elemen XML sehingga hanya satu record pada satu waktu yang ditampilkan.

### Pengikatan Data Tabel

Model pengikatan tabel HTML ke data XML dapat dilakukan dengan beberapa cara seperti model pengaitan langsung tabel (recordset sederhana), model *paging*, model recordset tersarang (hirarki).

Model Recordset Langsung (sederhana), yaitu dokumen XML yang diorganisasikan sebagai :

- Elemen root berisi serangkaian elemen record, semuanya bertipe sama
- Setiap elemen record berisi kumpulan elemen *field* yang sama
- Setiap elemen field berisi data karakter saja.

Contoh dokumen XML Pustaka.xml, elemen root-nya adalah DATABUKU, yang berisi empat elemen record BUKU, dan masing-masing record memiliki kumpulan elemen field yang berisi data karakter yaitu JUDUL, PENGARANG, PENERBIT, HALAMAN, dan HARGA. Jika tabel dikaitkan ke dokumen XML dengan elemen record ditampilkan pada baris tabel tersendiri, dan masing-masing elemen field anak ditampilkan pada kolom tersendiri (nama file TabelPustaka.htm) seperti dibawah.

Jika file TabelPustaka.htm tersebut setelah dikaitkan dengan file Pustaka.xml dengan perintah :

```
<XML ID="dsoPustaka"
SRC="Pustaka.xml"></XML>
```

dengan mengaitkan seluruh dokumen XML dengan memberikan atribut DATASRC pada elemen ID dsoPustaka (disebut juga elemen pulau data) sehingga harus didahului dengan karakter *cross*(#), sebagai berikut :

```
<TABLE DATASRC="#dsoPustaka"
BORDER="1" CELLPADDING="5">
```

```
<!-- Nama file: TabelPustaka.htm -->
<HTML>
<HEAD>
<TITLE>Buku Pustaka</TITLE>
</HEAD>
<BODY>
<XML ID="dsoPustaka"
SRC="Pustaka.xml"></XML>
<H2 ALIGN="center">DATA BUKU
PERPUSTAKAAN</H2>
<TABLE DATASRC="#dsoPustaka"
BORDER="1" CELLPADDING="5">
<THEAD>
<TH>Judul Buku</TH>
<TH>Nama Pengarang</TH>
<TH>Penerbit</TH>
<TH>Halaman</TH>
<TH>Harga</TH>
</THEAD>
<TR ALIGN="center">
<TD><SPAN DATAFLD="JUDUL"
STYLE="font-style:italic"></SPAN></TD>
<TD><SPAN
DATAFLD="PENGARANG"></SPAN></TD>
<TD><SPAN
DATAFLD="PENERBIT"></SPAN></TD>
<TD><SPAN
```

sehingga hasil eksekusinya nampak seperti dibawah ini :



Gambar 1. Hasil eksekusi model recordset langsung

Model *Paging*, yaitu jika dokumen XML disajikan dengan berisi sejumlah record secara berkelompok atau sebagian dalam satu waktu



jumlah variabel kejadian (nol atau lebih) pada sebuah record tersarang. Contoh:  
 Model Recordset tersarang (hirariki).File :  
 Pustakahirarki.xml

```

<?xml version="1.0"?>
<!-- Nama File: Pustakahirarki.xml -->
<DATABUKU>
  <KELOMPOK>
    <NAMA>Teknologi Komputer </NAMA>
    <BUKU>
      <JUDUL> Pengantar Teknologi
      Informasi</JUDUL>
      <PENGARANG>Aji
      Supriyanto</PENGARANG>
      <PENERBIT>Salemba
      Infotek</PENERBIT>
      <HALAMAN>499</HALAMAN>
      <HARGA>Rp 60.000 </HARGA>
    </BUKU>
    <BUKU>
      <JUDUL> Merakit, Mengupgrade, dan
      Memelihara PC</JUDUL>
      <PENGARANG>Aji
      Supriyanto</PENGARANG>
      <PENERBIT>Graha Ilmu</PENERBIT>
      <HALAMAN>326</HALAMAN>
      <HARGA>Rp 40.000 </HARGA>
    </BUKU>
    <BUKU>
      <JUDUL> XML Security</JUDUL>
      <PENGARANG>Blake
      Dournaee</PENGARANG>
      <PENERBIT>McGraw-Hill</PENERBIT>
      <HALAMAN>412</HALAMAN>
      <HARGA>Rp 230.000 </HARGA>
    </BUKU>
  </KELOMPOK>

```

```

<KELOMPOK>
  <NAMA>Bahasa dan Sastra</NAMA>
  <BUKU>
    <JUDUL> Kamus Inggris-
    Indonesia</JUDUL>
    <PENGARANG>John M. Echols dan
    Hassan Shandily</PENGARANG>
    <PENERBIT>Gramedia</PENERBIT>
    <HALAMAN>660</HALAMAN>
    <HARGA>Rp 98.000 </HARGA>
  </BUKU>
  <BUKU>
    <JUDUL> Belajar Sastra Jawi
    Kuno</JUDUL>
    <PENGARANG>Hartono
    Sumantri</PENGARANG>
    <PENERBIT>Balai
    Pustaka</PENERBIT>
    <HALAMAN>445</HALAMAN>
    <HARGA>Rp 68.000 </HARGA>
  </BUKU>
</KELOMPOK>
<KELOMPOK>
  <NAMA>Ekonomi</NAMA>
  <BUKU>
    <JUDUL> Manajemen
    Strategi</JUDUL>
    <PENGARANG>Budiono</PENGARANG>
    <PENERBIT>Gramedia</PENERBIT>
    <HALAMAN>556</HALAMAN>
    <HARGA>Rp 76.000 </HARGA>
  </BUKU>
  <BUKU>
    <JUDUL> Manajemen Akuntansi
    Keuangan</JUDUL>
    <PENGARANG>Sri
    Mulyani</PENGARANG>
    <PENERBIT>Salemba</PENERBIT>
    <HALAMAN>419</HALAMAN>
    <HARGA>Rp 66.000 </HARGA>
  </BUKU>
</KELOMPOK>
</DATABUKU>

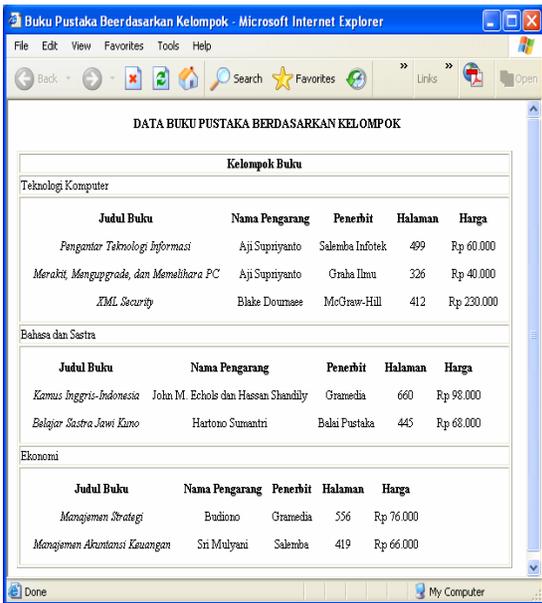
```

Model Recordset tersarang (hirariki).File :  
 Pustakahirarki.htm

```

<!-- Nama file: Pustakahirarki.htm -->
<HTML>
<HEAD>
  <TITLE>Buku Pustaka Beerdasarkan
  Kelompok</TITLE>
</HEAD>
<BODY>
  <XML ID="dsoPustaka"
  SRC="Pustakahirarki.xml"></XML>
  <H3 ALIGN="center">DATA BUKU PUSTAKA
  BERDASARKAN KELOMPOK </H3>
  <TABLE DATASRC="#dsoPustaka"
  BORDER="1">
    <THEAD>
      <TH>Kelompok Buku</TH>
    </THEAD>
    <TR>
      <TD><SPAN
  DATAFLD="NAMA "></SPAN></TD>
    </TR>
    <TR>
      <TD>
        <TABLE DATASRC="#dsoPustaka"
        DATAFLD="BUKU" BORDER="0"
        CELLSPACING="10">
          <THEAD>
            <TH>Judul Buku</TH>
            <TH>Nama Pengarang</TH>
            <TH>Penerbit</TH>
            <TH>Halaman</TH>
            <TH>Harga</TH>
          </THEAD>
          <TR ALIGN="center">
            <TD><SPAN DATAFLD="J UDUL"
            STYLE="font-style:italic"></SPAN></TD>
            <TD><SPAN
            DATAFLD="PENGARANG"></SPAN></TD>
            <TD><SPAN
            DATAFLD="PENERBIT"></SPAN></TD>
            <TD><SPAN
            DATAFLD="HALAMAN"></SPAN></TD>
            <TD><SPAN
            DATAFLD="HARGA"></SPAN></TD>
          </TR>
        </TABLE>
      </TD>
    </TR>
  </TABLE>
</BODY>
</HTML>

```



Gambar 3. Hasil Eksekusi Model RecordSet Tersarang

**Model Pengikatan Data Record-Tunggal**

Model Pengikatan Data Record-Tunggal, yaitu mengikatkan data XML untuk disajikan menjadi per-data record (record tunggal). Bentuknya menyerupai paging namun tidak dalam bentuk tabel. Contoh untuk file xml gunakan file: Pustaka.xml, sedangkan file HTML-nya didefinisikan serbagai berikut. Model Pengikatan Data Record-Tunggal. File : PustakaTunggal.htm

```

<!-- Nama file: PustakaTunggal.htm -->
<HTML>
<HEAD>
  <TITLE>Buku Pustaka</TITLE>
</HEAD>
<BODY>
  <XML ID="dsoPustaka" SRC="Pustaka.xml"></XML>
  <H2 ALIGN="center">DATA BUKU PERPUSTAKAAN</H2>
  <SPAN STYLE="font-style:italic">Judul Buku:</SPAN>
  <SPAN DATASRC="#dsoPustaka" DATAFLD="JUDUL"
  STYLE="font-weight:bold"></SPAN>
  <BR>
  <SPAN STYLE="font-style:italic">Nama Pengarang :</SPAN>
  <SPAN DATASRC="#dsoPustaka" DATAFLD="PENGERANG"
  STYLE="font-weight:bold"></SPAN>
  <BR>
  <SPAN STYLE="font-style:italic">Penerbit :</SPAN>
  <SPAN DATASRC="#dsoPustaka" DATAFLD="PENERBIT"
  STYLE="font-weight:bold"></SPAN>
  <BR>
  <SPAN STYLE="font-style:italic">Halaman :</SPAN>
  <SPAN DATASRC="#dsoPustaka" DATAFLD="HALAMAN"
  STYLE="font-weight:bold"></SPAN>
  <BR>
  <SPAN STYLE="font-style:italic">Harga Buku :</SPAN>
  <SPAN DATASRC="#dsoPustaka" DATAFLD="HARGA"
  STYLE="font-weight:bold"></SPAN>
  <HR>
  <BUTTON ONCLICK="dsoPustaka.recordset.moveFirst()">
    &lt; Awal
  </BUTTON>
  <BUTTON ONCLICK="dsoPustaka.recordset.movePrevious();
    if (dsoPustaka.recordset.BOF)
      dsoPustaka.recordset.moveNext()">
    &lt; Sebelum
  </BUTTON>
  <BUTTON ONCLICK="dsoPustaka.recordset.moveNext();
    if (dsoPustaka.recordset.EOF)
      dsoPustaka.recordset.movePrevious()">
    Sesudah &gt;
  </BUTTON>
  <BUTTON ONCLICK="dsoPustaka.recordset.moveLast()">
    Akhir &gt;|
  </BUTTON>
</BODY>
</HTML>

```

Sehingga hasil eksekusinya adalah :



Gambar 4. Hasil Eksekusi Model Pengikatan Data RecordSet Tunggal

### Model Script dengan DSO

Pada bagian sebelumnya telah dijelaskan bahwa DSO (*Data Source Objects*) merupakan objek pemrograman yang dapat menyimpan, atau menyimpan sementara (*caching*) data XML dalam bentuk recordset dan memberikan sebuah akses ke data XML tersebut. Berikut ini disajikan script untuk pencarian sebuah recordset.

Model Script DSO. File CariPustaka.htm



## **DAFTAR PUSTAKA**

1. Benz B., Duran J.R., 2003, “*XML Programming Bible*”, Wiley Publishing, Inc.
2. Eric T. Ray, 2001 , “*Learning XML*”.
3. Michail J. Young, 2001, “*Step By Step XML*”, Elex Media – Microsoft.
4. Reed D., 2004, “*Web Programming*”, Spring,<http://www.creighton.edu/~davereed/>
5. <http://msdn.microsoft.com/xml/>
6. <http://www.w3c.org/TR/>
7. <http://www.w3schools.com>