

# REPRESENTASI DATA SEMI TERSTRUKTUR MENGGUNAKAN *EXTENSIBLE MARKUP LANGUAGE (XML)*

Titin Pramiyati

## Abstrak

Perkembangan teknologi informasi yang mengarah pada *World Wide Web* sebagai sebuah sumber informasi yang luas menyebabkan timbulnya kebutuhan akan sebuah model data yang tidak terkungkung pada sebuah skema tertentu dan dapat merepresentasikan data semi terstruktur. Untuk pemenuhan kebutuhan tersebut pemilihan terhadap model data XML sebagai representasi dari data semi terstruktur yang banyak digunakan pada sumberdata web sehingga dapat memfasilitasi *interoperability* pada sistem basisdata tersebar.

Kata kunci : data semi terstruktur, representasi data, XML

## A. Pendahuluan

Perkembangan teknologi informasi yang mengarah pada *World Wide Web* sebagai sebuah sumberdata informasi yang luas, telah menjadi tujuan pencarian informasi bagi masyarakat informasi. Sumberdata informasi pada web sebagian besar merupakan kumpulan dokumen-dokumen tidak terstruktur yang terkoneksi satu dengan yang lain.

Sumberdata web memiliki ketentuan karakteristik yang harus dipenuhi, yaitu, data web tidak dapat dibatasi oleh sebuah tipe skema tertentu, maksudnya adalah representasi data dalam web tidak mengikuti skema tertentu dalam menempatkan datanya. Karakteristik lain yang terdapat pada data web adalah struktur yang tidak beraturan dengan penyarangan (*nested*) data yang cukup mendalam (*deep*), data web merupakan data yang tersebar dan terkoneksi dengan luas. Data dengan karakteristik seperti ini sesuai dengan format data yang disebut sebagai data semi terstruktur (*semistructured data*). Dengan demikian dapat juga dikatakan bahwa data web tidak menggunakan skema dan format tertentu dalam representasi datanya.

Keberadaan data semi terstruktur didasarkan pada beberapa alasan, yaitu; Pertama, telah terdapat sumberdata seperti web yang dapat diolah sebagai sebuah basisdata (*database*) yang tidak dibatasi oleh sebuah skema basisdata tertentu. Kedua, adanya keinginan untuk memiliki sebuah format yang sangat fleksibel dalam pertukaran data antar basisdata. Ketiga, berkaitan dengan keberadaan data terstruktur yang telah ada, akan sangat membantu jika data terstruktur tersebut dapat dianggap sebagai data semi terstruktur untuk kebutuhan *browsing*. Pada data semi terstruktur, informasi biasanya diasosiasikan dengan sebuah skema yang ditempatkan didalam data yang dikenal dengan sebutan "*self-describing*".

Makin meningkatnya penggunaan data semi terstruktur di berbagai bidang, yang menginginkan terpublikasinya informasi secara luas maka pilihan di gunakannya *extensible markup language (XML)* pada representasi data tersebut. Pemilihan XML sebagai model data semi terstruktur

disebabkan karena XML memberikan suatu model data yang bersifat *schema independent* dimana model data ini tidak mengikuti skema tertentu. Selain itu XML juga memfasilitasi kemudahan dalam pertukaran data pada web sehingga *interoperability* pada sistem basisdata tersebar tetap dapat terjaga dengan baik.

XML adalah salah satu bahasa yang berfungsi sebagai *tools* dalam memfasilitasi pertukaran data antar sistem, dokumen XML memiliki format yang dapat mendeskripsikan dokumen itu sendiri (*self-described*). XML juga merupakan representasi bagi data semi terstruktur (*semistructured data*), hal ini dikarenakan representasi data pada XML merupakan graph berlabel (*labeled graph*) yang memiliki *root*, *parent node* dan *child node*. Sehingga dengan kemampuan XML, maka representasi data semi terstruktur dapat dibangun menggunakan XML yang dituangkan kedalam bentuk dokumen XML.

Validasi XML dapat dilakukan dengan menggunakan *Data Type Definition* (DTD) dan skema, hal ini dikarenakan DTD digunakan untuk mendefinisikan struktur dan isi dari XML dokumen.

Skema XML merupakan skema yang dapat dibentuk sesuai dengan kebutuhan dokumen yang akan dibangun, tidak terdapat bentuk skema tertentu dalam pembangunan skema XML. Dengan skema dan DTD, kebenaran data yang disimpan dalam dokumen XML akan diperoleh. Skema dapat melengkapi kekurangan yang terdapat dalam DTD, skema dapat mendefinisikan tipe data yang kompleks, skema dapat menampilkan bentuk hirarki berdasarkan pada *relationship*, dan skema bersifat *reusable* dan *extensible*

Penggunaan XML dalam representasi data telah banyak digunakan, pada bidang ilmu ekologi telah digunakan pada *The Knowledge Network for Biocomplexity Project* yang mengakomodasikan representasi data dengan format dan skema data yang beragam pada bidang ekologi. XML juga digunakan sebagai media simpanan data pada *A Hybrid Search in the Online Knowledge Center*, yang memungkinkan pencarian isi dokumen pada sebuah metadata.

Dengan meluasnya penggunaan model data XML pada web, pada tulisan ini akan diperlihatkan bagaimana XML merepresentasikan data. Sebagai bahan bahasan tulisan ini, akan digunakan data tentang wisudawan. Data wisudawan ini terdiri dari data mahasiswa (Nrp, Nama, TempatLhr, TanggalLhr, Agama, Jenis kelamin, alamat), data kelulusan (Tahun Lulus, IPK Lulus), data program studi (Nama Prodi) dan tanggal yudisium.

Perangkat lunak yang digunakan pada pembentukan model data XML ini adalah XML Spy.

## A. Pembahasan

Umumnya pembentukan simpanan data yang berbasis XML, dilakukan dalam 2 (dua) tahap yaitu; pertama membentuk DTD, dan kedua adalah membuat skema dari XML dokumen yang digunakan sebagai simpanan data. Akan tetapi dengan menggunakan perangkat lunak XML Spy proses pembuatan dapat dilakukan dengan membentuk skema XML terlebih dahulu.

Terdapat 2 (dua) kebutuhan dalam membentuk XML yaitu; *well-formed* dan *validate-formed*. *Well-formalization* merupakan kebutuhan dokumen XML agar dapat mengikuti beberapa sintaks, seperti, adanya satu elemen yang tepat yang berisi semua elemen lain, atau

elemen dapat berupa elemen yang bersarang (*nest*) akan tetapi tidak tumpang tindih (*overlap*), dan sebagainya. Kebutuhan atas *validation-formed*, dikarenakan XML berisi elemen dan atribut yang telah ditentukan menurut tipe data dan hubungan (*relationship*) antar elemen. *Document Type Definition* (DTD) dan skema XML adalah dua bentuk mekanisme validasi utama yang digunakan pada XML. Sebelum XML instance dapat divalidasi dan digunakan, XML instance harus diuraikan oleh XML parsers. Sebuah parser akan membuat isi data yang terdapat pada struktur data XML tersedia untuk digunakan oleh aplikasi lain.

Komponen dasar dari dokumen XML adalah elemen, yaitu satu bagian teks yang dibatasi dalam sebuah tag yang berpasangan, seperti `<Mahasiswa>` dan `</Mahasiswa>`. Isi dari elemen dapat berupa *raw text*, elemen lain atau kombinasi dari keduanya. Elemen dapat berisi atribut yang merupakan isi yang terkandung pada elemen.

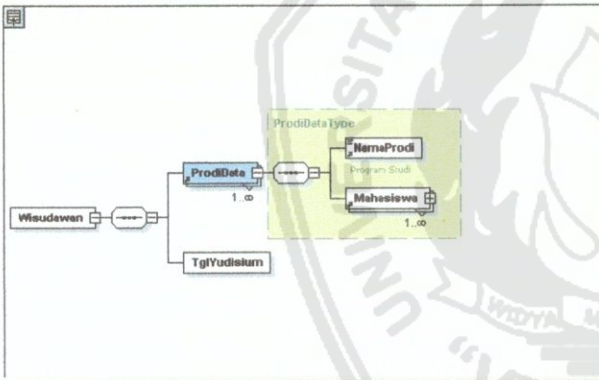
Pembuatan skema XML ini dituangkan dalam bentuk XML file, didalam XML file ini yang harus ditentukan adalah *node-node* yang berfungsi baik sebagai *parent node* atau *child node*. Bagian yang terdapat dalam skema XML yang bertugas untuk mendefinisikan node dari XML adalah *ElementType declaration*

Sesuai dengan bahan yang digunakan pada tulisan ini, maka *root* dari dokumen yang akan dibuat dapat dinamakan Wisudawan yang memiliki elemen ProdiData dan TglYudisium. Elemen ProdiData merupakan elemen yang memiliki anggota elemen lainnya, yaitu NamaProdi dan Mahasiswa sehingga elemen ProdiData merupakan elemen *parent node*. Pada elemen Mahasiswa terdapat beberapa elemen yang menjadi *child node*.

Kebutuhan akan elemen yang nantinya digunakan dalam dokumen XML Wisudawan harus didefinisikan pada skema, berikut adalah kerangka penulisan penentuan elemen pada skema XML Wisudawan dan tampilan dalam bentuk struktur pohon.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.2 U
(http://www.xmlspy.com) by Ba-k
(ZonaWarez.com) -->
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema
" elementFormDefault="qualified"
attributeFormDefault="unqualified">
<xs:element name="Mahasiswa">
<xs:complexType>
<xs:sequence>
<xs:element name="ThMasuk"/>
<xs:element
name="ThLulusSMU"/>
<xs:element name="Nrp"/>
<xs:element ref="Nama"/>
<xs:element
name="TempatLhr"/>
<xs:element
name="TanggalLhr"/>
<xs:element name="Agama"/>
<xs:element
name="JnsKelamin"/>
<xs:element name="Alamat"/>
<xs:element name="ThLulus"/>
<xs:element name="IPKLulus"/>
</xs:sequence>
</xs:complexType>
<xs:element name="Wisudawan">
<xs:complexType>
<xs:sequence>
<xs:element ref="ProdiData"
maxOccurs="unbounded"/>
<xs:element
name="TglYudisium"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ProdiData"
type="ProdiDataType"/>
<xs:complexType name="ProdiDataType">
<xs:sequence>
<xs:element ref="NamaProdi"/>
<xs:element ref="Mahasiswa"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

Penulisan kerangka diatas jika dipadankan dengan gambar dapat dijelaskan sebagai berikut : pada skema terdapat elemen Mahasiswa yang didefinisikan sebagai elemen *complextipe*. Elemen dengan tipe *complextipe*, merupakan elemen yang memiliki sejumlah elemen lain sebagai *child node*, dan pembentukan child node dilakukan dengan membentuk node-node tersebut sebagai sebuah *sequence*. Perbedaan yang terdapat pada elemen Mahasiswa dengan elemen ProdiData adalah bahwa elemen ProdiData merupakan elemen yang memiliki tipe ProdiDataType yang didefinisikan sebagai tipe baru yang bersifat *complextipe*, dimana elemen yang menjadi *child node* pada *elementype* ini adalah elemen NamaProdi dan elemen Mahasiswa.



Gambar 1. Struktur pohon dari skema XML (wisudawan)

Penentuan jumlah maksimal record yang terdapat pada dokumen XML dilakukan dengan cara menentukan apakah elemen *complextipe* tidak terbatas (*unbounded*), jika elemen tersebut merupakan elemen dengan jumlah record yang tidak terbatas maka akan diperlihatkan pada struktur pohon tanda 1..8 pada node.

Perlu diketahui berbeda dengan data terstruktur yang dengan jelas membedakan struktur data sebagai baris dan kolom, dimana

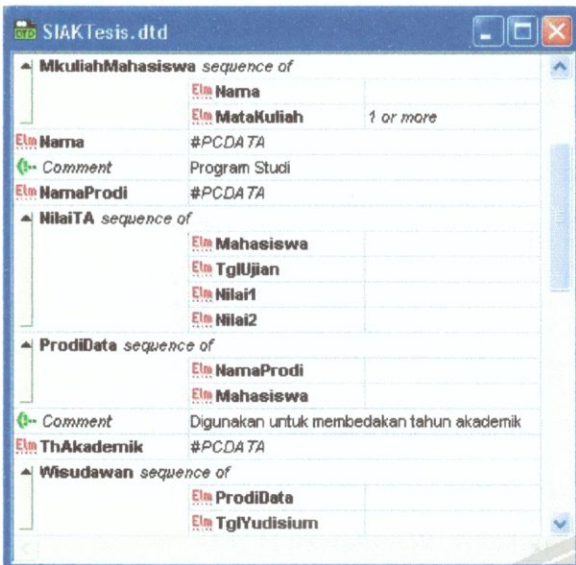
baris menyatakan *record* sedangkan kolom menyatakan atribut. Maka pada data semi terstruktur khususnya XML, baik *record* maupun atribut, semua dikenal sebagai elemen.

Setelah skema dokumen XML terbentuk dengan menggunakan utilitas yang terdapat pada XML Spy, dibentuk DTD dokumen tersebut, isi dari DTD adalah definisi elemen yang digunakan pada dokumen XML. Berikut adalah penulisan DTD dari Wisudawan.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.2
U (http://www.xmlspy.com) by Ba-k
(ZonaWarez.com) -->
<!--DTD generated by XML Spy v4.2 U
(http://www.xmlspy.com)-->
<!ELEMENT Mahasiswa (ThMasuk,
ThLulusSMU, Nrp, Nama, TempatLhr,
TanggalLhr, Agama, JnsKelamin, Alamat,
ThLulus, IPKLulus)>
<!ELEMENT Nama (#PCDATA)>
<!--Program Studi-->
<!ELEMENT NamaProdi (#PCDATA)>
<!ELEMENT ProdiData (NamaProdi,
Mahasiswa)>
<!--Digunakan untuk membedakan tahun
akademik-->
<!ELEMENT Wisudawan (ProdiData,
TglYudisium)>
<!ELEMENT Nrp ANY>
<!ELEMENT TempatLhr ANY>
<!ELEMENT TanggalLhr ANY>
<!ELEMENT Agama ANY>
<!ELEMENT JnsKelamin ANY>
<!ELEMENT Alamat ANY>
<!ELEMENT ThLulus ANY>
<!ELEMENT IPKLulus ANY>
```

.....

Bentuk *grid* dari DTD ditampilkan seperti terlihat pada gambar dibawah ini.



Gambar 2. DTD XML

Pada gambar terlihat bahwa setiap elemen baik yang bersifat *complextype* ataupun tidak didefinisikan dengan lengkap, misalnya pada elemen NamaProdi, elemen ini bukan elemen *complextype* maka pada DTD didefinisikan memiliki tipe elemen PCDATA. Maksud dari PCDATA disini adalah sebuah notasi yang digunakan untuk menandakan sebuah elemen merupakan variabel bukan suatu *character string*. Bentuk data lain yang dapat digunakan adalah CDATA atau *character data*, bentuk data ini merupakan suatu *unparsed data*. Jika suatu elemen dideklarasikan sebagai data dengan tipe CDATA, maka akan ditemukan *markup character* ( contohnya: <, >, &, dll) didalam string yang tidak akan diproses, tetapi akan diperlakukan sebagai sebuah teks sederhana.

Untuk elemen yang beranggotakan elemen lain yang merupakan sequence dideklarasikan seperti :

```
<ELEMENT Mahasiswa (ThMasuk, ThLulusSMU, Nrp, Nama, TempatLhr, TanggalLhr, Agama, JnsKelamin, Alamat, ThLulus, IPKLulus)>
```

Pada elemen Nrp, TempatLhr dan yang lain dideklarasikan sebagai elemen dengan tipe ANY, tipe ini dapat berisi apapun termasuk PCDATA, elemen atau kombinasi

dari elemen dan PCDATA, dapat juga tidak berisi elemen atau kosong.

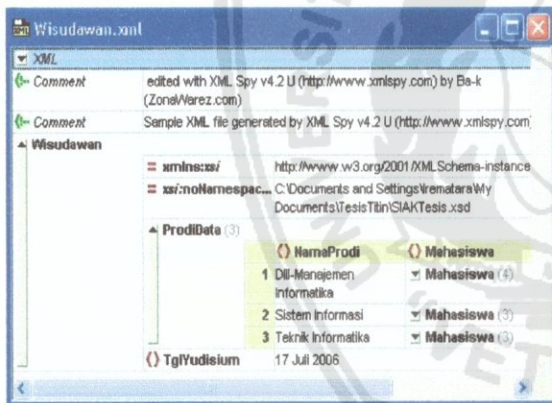
Ketika DTD dan skema XML tersedia, maka dokumen XML dapat dibuat, untuk mendapatkan dokumen XML yang *valid* atau *nonvalid* digunakan *parser*, yang bertugas untuk membaca DTD yang dibangun dan menentukan apakah dokumen sesuai dengan DTD atau tidak, jika sesuai maka dokumen valid (*validation-formed*), tetapi jika tidak maka dokumen salah, meskipun secara sintak benar (*well-formed*).

Jika DTD dan skema dari dokumen XML telah dibentuk, maka instance dari dokumen XML dapat dibentuk dengan membuat XML *stylesheet* atau lebih dikenal sebagai XSL. Pembuatan XSL dapat digunakan untuk menghasilkan *instance* dari dokumen XML, pada instance inilah data wisudawan ditempatkan. Berikut ini XSL dari dokumen XML Wisudawan,

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.2 U
(http://www.xmlspy.com) by Ba-k (ZonaWarez.com) -->
<!-- Sample XML file generated by XML Spy v4.2 U
(http://www.xmlspy.com)-->
<?xml-stylesheet type="text/xsl"
href="C:\Documents and Settings\Irematara\My
Documents\TesisTitin\Wisuda.xsl"?>
<Wisudawan
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Documents
and Settings\Irematara\My
Documents\TesisTitin\SIAKTesis.xsd">
  <ProdiData>
    <NamaProdi>DIII-Manajemen
Informatika</NamaProdi>
    <Mahasiswa>
      <ThMasuk>2001</ThMasuk>
      <ThLulusSMU>2001</ThLulusSMU>
      <Nrp>201502001</Nrp>
      <Nama>Mutia Delia Nadifah</Nama>
      <TempatLhr>Jakarta</TempatLhr>
      <TanggalLhr>09 September
1987</TanggalLhr>
      <Agama>Islam</Agama>
      <JnsKelamin>Perempuan</JnsKelamin>
      <Alamat>Jl. Surabaya III No.31</Alamat>
      <ThLulus>2006</ThLulus>
```

```
<IPKulus>3.50</IPKulus>
</Mahasiswa>
</ProdiData>
<TglYudisium>17 Juli 2006</TglYudisium>
</Wisudawan>
```

Untuk menambahkan data mahasiswa pada masing-masing program studi dapat dilakukan dengan membentuk tag `<Mahasiswa></Mahasiswa>` yang baru. Pada instance ini konten dari masing-masing elemen dapat ditempatkan diantara pasangan tag elemen dengan mengikuti semantik yang telah ditentukan pada DTD Wisudawan. Jika semantik data yang ditempatkan pada instance tidak sesuai maka parser akan melakukan validasi terhadap konten tersebut. Berikut adalah tampilan instance dalam bentuk grafik,



Gambar 3. Instance Wisudawan.XML

Jika XSL untuk dokumen Wisudawan telah terbentuk, maka akan dihasilkan output dari XSL tersebut dalam bentuk format HTML, berikut ini adalah output dari XSL Wisudawan,

Tanggal Yudisium				
08 April 2005				
Nama Program Studi				
DIII- Manajemen Informatika				
N r p	N a m a	N a m a O r a n g T u a	T h n M a s u k	T h n L u l u s S M U
201502001	Muthia Della Nadhifah	Jayanta	2001	2001
201502003	Diva Aicilla Fitri	Endang Susryana	2001	2000
201502005	Rusky Anis	Agustono	2001	2001
S1-Teknik Informatika				
N r p	N a m a	N a m a O r a n g T u a	T h n M a s u k	T h n L u l u s S M U
201511001	Andhika Wibisono	Hari Purmana	2001	2001
201511010	Bayu Ardiansyah	M. Yusuf	2001	2000
200511016	Ade Yumi Fajarsari	Kusriani Puri	2000	2000
S1-Sistem Informasi				
N r p	N a m a	N a m a O r a n g T u a	T h n M a s u k	T h n L u l u s S M U
201512040	Deasy Amelia	JG Sudarsono	2001	2000
200512020	Maya Safitri	Sudiono	2000	2000
200512005	M. Misdianto	Misdianto	2001	2001

Gambar 4. Wisudawan HTML

Dengan terbentuknya dokumen dalam format HTML seperti terlihat diatas, maka data tersebut dapat langsung di *upload* pada sebuah situs.

### C. Simpulan

XML dapat digunakan untuk merepresentasikan data semi terstruktur dengan baik, hal ini dapat dilihat dengan digunakannya skema yang tidak terlalu kaku pada dokumen XML dan penggunaan DTD untuk menentukan semantik dari data yang akan dikandung oleh dokumen tersebut. XML juga memberi kemudahan dalam penempatan data ke dalam *instance* dokumen yang telah dibuat. Dengan menggunakan instance dokumen XML akan dihasilkan informasi yang dibutuhkan, hal ini dapat dengan mudah dibentuk melalui pembangunan XSL. Keluaran yang dihasilkan dari XSL dapat berupa informasi dengan format HTML.

### DAFTAR PUSTAKA

- Abiteoul S, Buneman P, Suciu D, *Data on the Web*, Morgan Kauffman, 2000
- Buneman P, *Semistructured Data*, Department of Computer and Information Science University of Pennsylvania, 1997
- Connolly T, Begg C, Strachan A, *Database Systems*, Addison Wesley, 2003
- McBrien, P., Poulouvassilis, *A Semantic Approach to Integrating XML and Structured Data Sources*, In 13<sup>th</sup> CaiSE Conference, 2001
- NISO Press, *Understanding Metadata*, NISO Press, 2004
- Özsu MT, Valduriez P, *Principles Distributed Database Systems, Second Edition*, Prentice Hall, 1999
- Turban E, Rainer RK, Potter RE, *Introduction To Information Technology, Second Edition*, John Wiley & Sons, Inc., 2003
- Sams, *Teach Yourself XML in 24 Hours*, Sams Publishing, 2000

