

Arsitektur Aplikasi Perangkat Enterprise #7

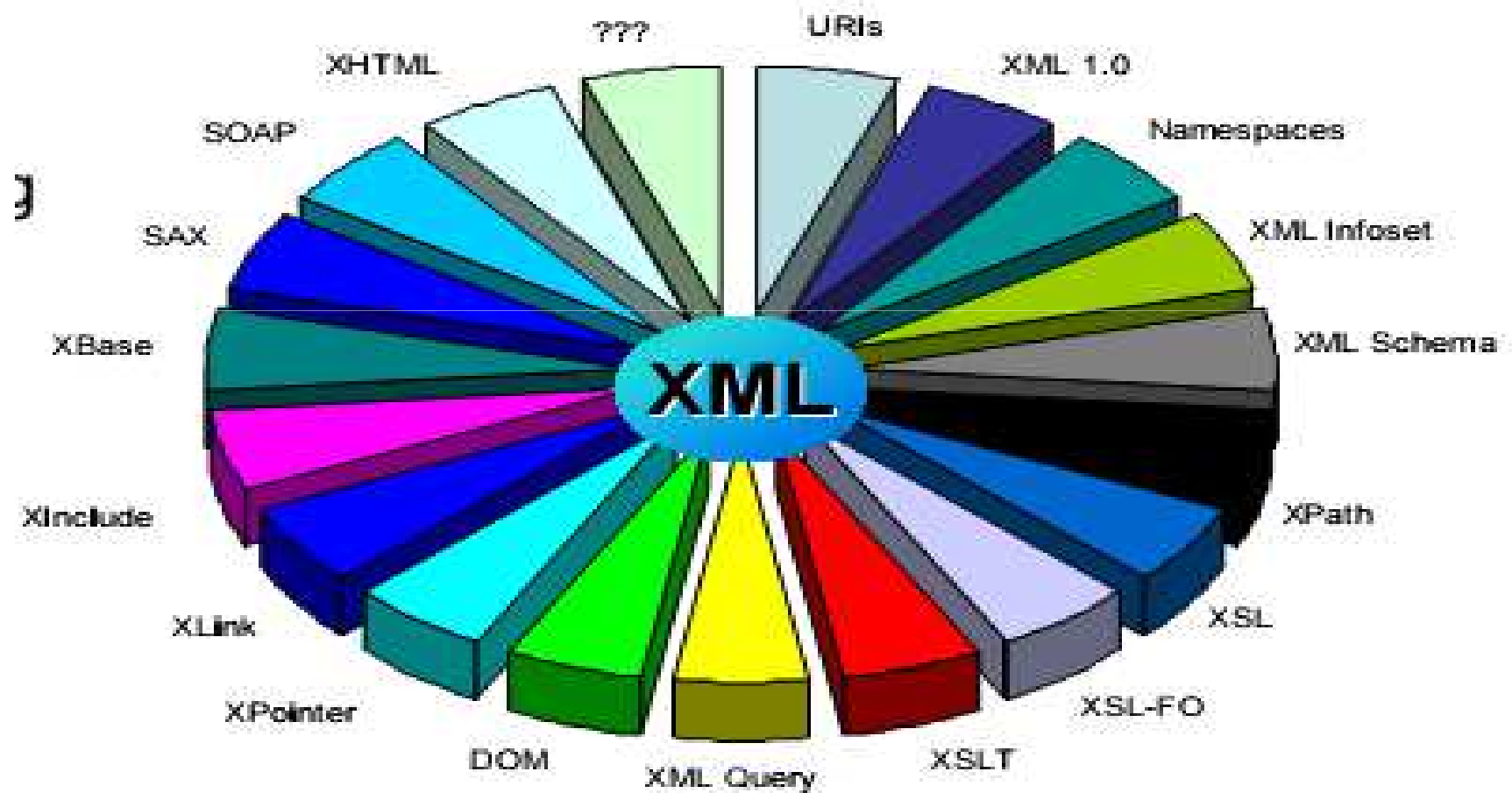


Antonius Rachmat C, S.Kom, M.Cs

XML (eXtensible Markup Language)

- ❑ XML kependekan dari eXtensible Markup Language, dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada 10 Februari 1998.
 - Kadang disebut **eXtensible Meta Language**
- ❑ Teknologi XML adalah teknologi keturunan dari SGML (Standard Generalized Markup Language, ISO 8879) yang dikembangkan pada tahun 1980-an.
- ❑ XML adalah suatu bahasa *Markup*, yaitu bahasa yang berisikan kode-kode berupa tanda-tanda tertentu dengan aturan tertentu untuk memformat dokumen teks dengan tag sendiri agar dapat dimengerti.
 - Contoh yang mirip : bahasa HTML, Postscript, RTF, Word star
- ❑ XML adalah bahasa untuk mengidentifikasi dan menandai data terstruktur.

XML Relations



XML (2) - Keuntungan

- Keunggulan dan keuntungan dari XML adalah:
 - Ekstensibilitas dan reusable
 - Memungkinkan pemrograman yang lebih baik:
 - Dapat memperlihatkan hubungan antar data.
 - XML dapat diprogram dan sudah banyak software pengolah XML.
 - Memisahkan data dan presentasi.
 - Data disimpan dalam XML sedangkan untuk presentasi (tampilan) dibuat dengan menggunakan XSLT.
 - Pencarian data cepat karena XML merupakan data dalam format yang terstruktur.
 - Hal ini dapat dilakukan dengan menggunakan XPath dan XQuery berdasarkan suatu elemen tertentu.
 - Bersifat plain text dan platform independent.
 - Terstruktur dan standard

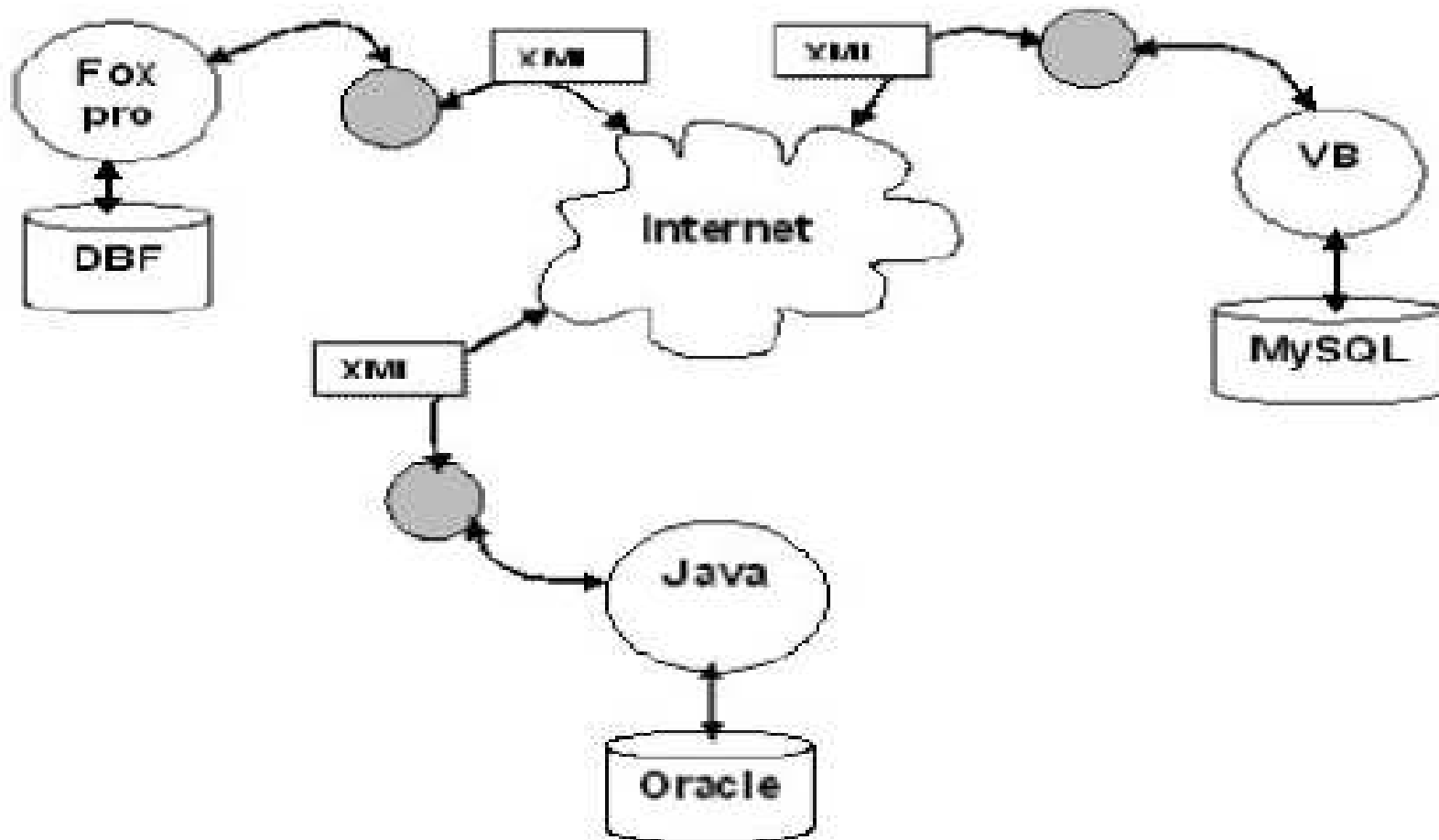
```
<bookstore>
- <book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>
- <book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
- <book category="WEB">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>
- <book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>
```

XML doc

Xquery Example

- ❑ **for** \$x in doc("books.xml")/bookstore/book
where \$x/price>30
return \$x/title
- ❑ Result:
<title lang="en">XQuery Kick Start</title>
<title lang="en">Learning XML</title>
- ❑ **for** \$x in doc("books.xml")/bookstore/book
return if (\$x/@category="CHILDREN")
then <child>{data(\$x/title)}</child>
else <adult>{data(\$x/title)}</adult>
- ❑ Result:
<adult>Everyday Italian</adult>
<child>Harry Potter</child>
<adult>Learning XML</adult>
<adult>XQuery Kick Start</adult>

XML (3) – Pertukaran data



Perbedaan antara XML dan HTML

- HTML merupakan tag untuk memformat **tampilan** (menitik-beratkan pada unsur presentasi)
 - Standar dan baku
- XML merupakan tag untuk mengontrol data dan tidak memperhatikan tampilan (menitik-beratkan pada **hirarki** data dan **penyimpanan** data)
 - Dapat membuat tag-tag sendiri sesuai dengan data yang ingin ditampilkan.

Document- vs. Data-Centric XML

- There are two broad application areas of XML:
- Document-centric XML
 - Used in publishing systems as a mechanism for representing documents such as technical manuals, legal documents, and product catalogues
- Data-centric XML
 - Used to mark up highly structured information such as the textual representation of relational data from databases, financial information, and programming language data structures
- Web services are about data-centric uses of XML

<HTML>

<H1>Memorandum</H1>

<P>From: The Big Cheese</P>

<P>To: One and all</P>

<P>Date: 11/1</P>

<P>Please be advised that effective
as of the date above, our casual
dress policy has been revised.
It's back to ties and jackets</P>

</HTML>

HTML

XML

<memo>

<from>The Big Cheese</from>

<to>One and all</to>

<Date month="11" day="1"/>

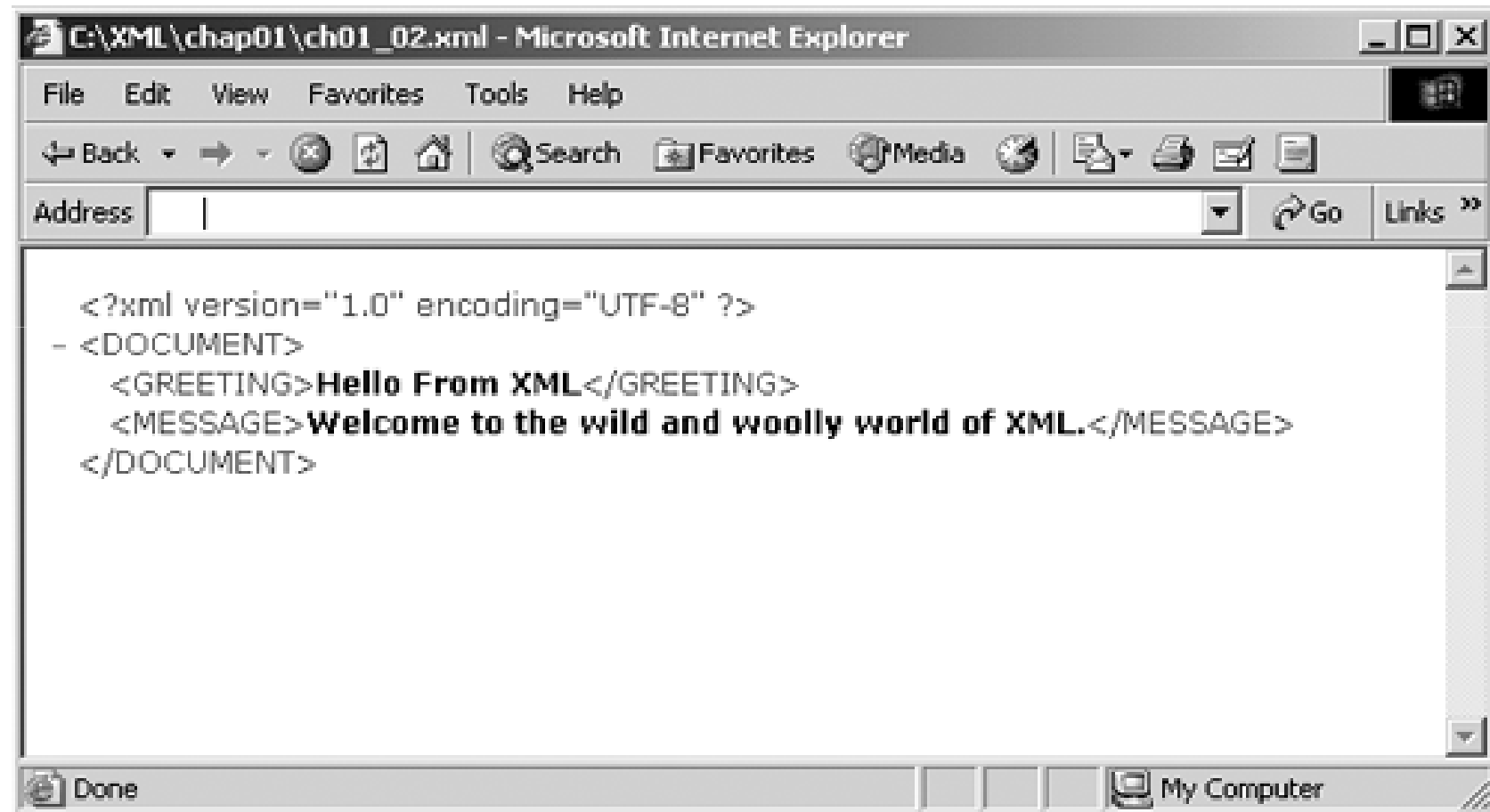
<bad_news>Please be advised that
effective as of the date above,
our popular casual dress policy
has been revoked. It's back to
ties and jackets</bad_news>

</memo>

Contoh XML

```
□ <?xml version="1.0" encoding="UTF-8"?>
  <DOCUMENT>
    <GREETING>
      Hello From XML
    </GREETING>
    <MESSAGE>
      Welcome to the wild and woolly world
      of XML.
    </MESSAGE>
  </DOCUMENT>
```

Tampilan XML di browser

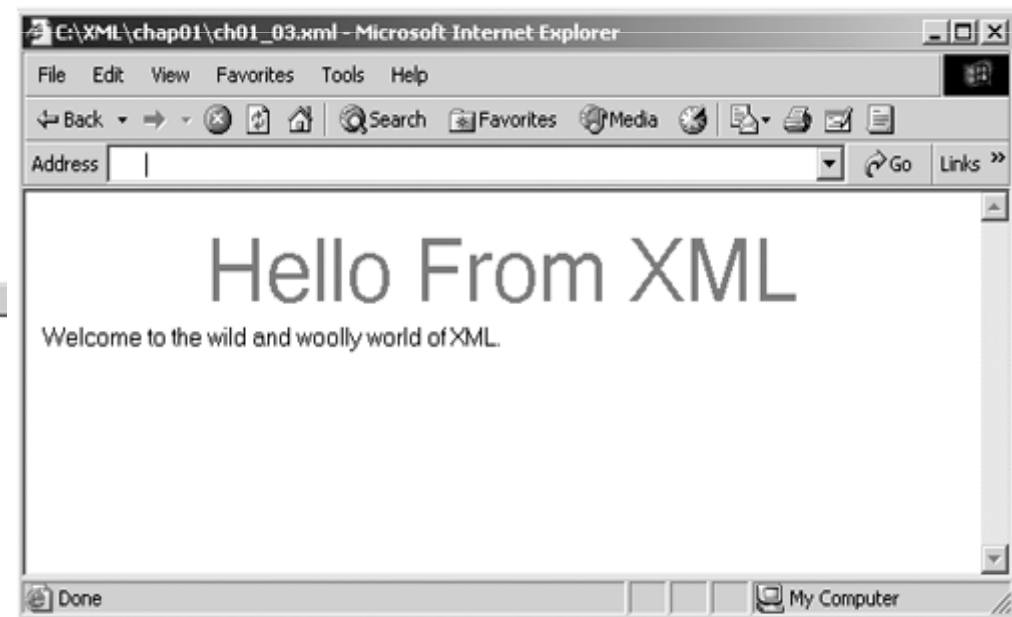


Jelek? Tambah saja dengan CSS

- ❑

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="coba.css"?>
<DOCUMENT>
  <GREETING>
    Hello From XML
  </GREETING>
  <MESSAGE>
    Welcome to the wild and woolly world of XML.
  </MESSAGE>
</DOCUMENT>
```
- ❑ File coba.css:
GREETING {display: block; font-size: 36pt; color: #FF0000; text-align: center}
MESSAGE (display: block; font-size: 18pt; color: #000000}

Ok, sekarang di browser (dengan CSS)



Tool & XML programming ?

- Notepad?
- Word?
- Altova XMLSpy?



- Program: Delphi, VB, .NET, Java, PHP, Javascript (JSON)!
- http://en.wikipedia.org/wiki/List_of_XML_markup_languages

Aturan XML

- Dokumen XML digunakan untuk **mendeskripsikan** dokumen itu sendiri dengan sintaks yang sederhana.

- Contoh:

```
<?xml version="1.0" encoding="ISO-8859-1" ? standalone="yes" >
```

```
<note>
```

```
  <to>Tove</to>
```

```
  <from>Jani</from>
```

```
  <heading>Reminder</heading>
```

```
  <body>Don't forget me this weekend!</body>
```

```
</note>
```


Aturan XML

- Tag `<xml>` : mendefinisikan versi `xml`, *encoding* yang digunakan dan status `xml`.
- Tag `<note>` adalah elemen *root* dalam dokumen.
- Tag `<to>`, `<from>`, `<heading>`, dan `<body>` adalah *child elements* dari elemen `<note>`.

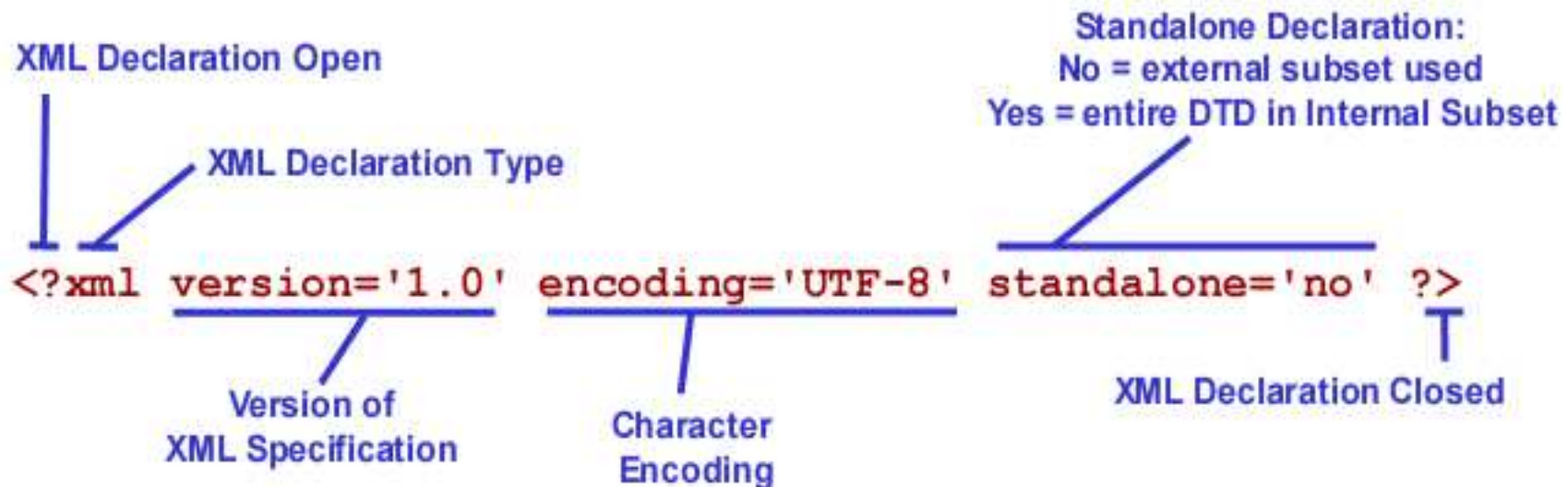
Aturan XML

- ❑ Semua tag xml harus memiliki tag penutup.
- ❑ Pada HTML beberapa elemen tidak harus berpasangan
 - <p>paragraph pertama**
 - <p>paragraph kedua**
- ❑ XML harus menulis tag penutup untuk setiap tag pembuka
 - <p>paragraph pertama</p>**
 - <p>paragraph kedua</p>**
- ❑ Tag tunggal hanya diperbolehkan untuk elemen kosong.
 - <anggota nama="anton" />**
 -
**
 - <paragraph heading="awal" />**

Aturan XML

XML Declaration:

- Setiap membuat dokumen XML diawali dengan heading standard XML. Formatnya adalah sebagai berikut:



Aturan XML

- Tag XML adalah *case sensitive* (membedakan antara huruf besar dan kecil).
- Contoh berikut adalah contoh yang salah:

<buku>

<judul>AASE</Judul>

<penGarang>Antonius</pengarang>

<BUKU>

Aturan XML

- Semua tag XML harus tersarang dengan benar dan tidak terbalik-balik.

- Contoh:

<i>This text is bold and italic - wrong</i>

<i>This text is bold and italic - right</i>

Aturan XML

- Semua dokumen XML harus memiliki elemen *root*. Elemen *root* adalah elemen teratas/tertinggi yang harus memiliki tag penutup.
- Contoh:
<orangtua>
 <anak>
 <namaanak>.....</namaanak>
 </anak>
</orangtua>
- Dalam contoh di atas elemen **<orangtua>** adalah elemen *root*.

Aturan XML

- Nilai **atribut** harus diapit dalam tanda petik ganda (*double quote*) atau petik tunggal (*single quote*).

- Contoh:

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

```
<note date='12/11/2002'>
```

```
  <to>Tove</to>
```

```
</note>
```

Aturan XML

- ❑ Dalam XML karakter spasi akan dibaca sebagai karakter spasi.
- ❑ Dalam HTML jika ada lebih dari satu karakter spasi, karakter spasi tersebut dihilangkan sehingga menjadi 1 spasi saja, Contoh:
- ❑ "Hello <spasi> <spasi> my name is Tove" akan menjadi "Hello my name is Tove".
- ❑ Dalam XML karakter spasi tetap dibaca sebagai karakter spasi.

Aturan XML

- ❑ Komentar adalah kalimat/baris yang tidak dieksekusi oleh compiler, browser atau parser.
- ❑ Penulisan komentar di XML menggunakan tanda `<!-- komentar -->`
- ❑ Dalam XML, karakter `'<'` dan `'>'` adalah illegal:

**`<syarat>jika jumlah < 1000
maka</syarat>`**

Legal:

**`<syarat>jika jumlah < 1000
maka</syarat>`**

Entity with References

Entity References	Character
<	<
>	>
&	&
"	"
'	'

Representasi Data

- Elemen XML adalah blok utama penyusun XML yang memiliki relasi antar elemen-elemen lainnya dan diapit oleh tag-tag "<" dan ">".
- Misal ada data:

My First XML

Introduction to XML

- What is HTML
- What is XML

XML Syntax

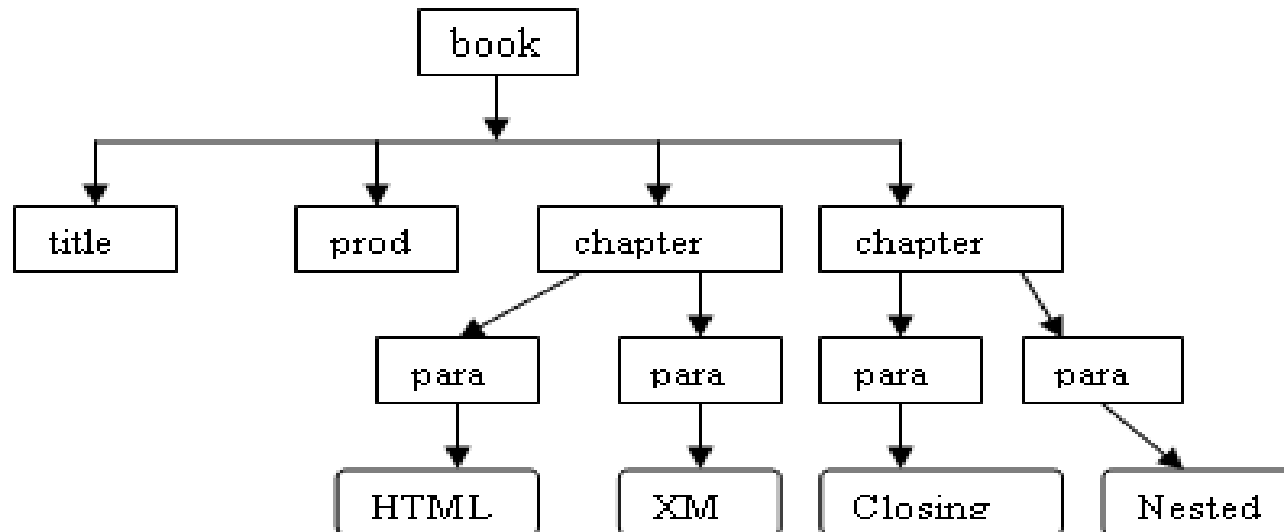
- Elements must have a closing tag
- Elements must be properly nested

Hasil XML

```
<book>  
  <title>My First XML</title>  
  <prod id="33-657" media="paper"></prod>  
  <chapter>Introduction to XML  
    <para>What is HTML</para>  
    <para>What is XML</para>  
  </chapter>  
  <chapter>XML Syntax  
    <para>Elements must have a closing tag</para>  
    <para>Elements must be properly nested</para>  
  </chapter>  
</book>
```

Keterangan XML

- Elemen **<book>** adalah *root element* **dan** *parent element*. Elemen **<title>**, **<prod>**, dan **<chapter>** adalah *child elements* dari **<book>**.
- Elemen **<title>**, **<prod>**, dan **<chapter>** adalah kembar (atau memiliki relasi yang disebut *sister elements*) sebab mereka memiliki *parent* yang sama.



XML Documents

- XML Documents may contain:
 - Elements
 - Attributes
 - Comments
 - Entities
 - Processing instructions
 - Character data (CDATA)
 - XML text and whitespace
 - DTD

Node dalam XML

Sebuah dokumen XML terdiri dari bagian bagian yang disebut dengan **node**.

- ❑ **Root node**: yaitu node 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 elemen kosong seperti `<anggota nama="anton" />`.
- ❑ **Attribute node**: berupa elemen tambahan dari elemen utama.
- ❑ **Text node**: adalah text yang merupakan isi dari sebuah elemen, ditulis diantara tag pembuka dan tag penutup.
- ❑ **Comment node**: adalah baris yang tidak dieksekusi oleh parser.

Node dalam XML

- **Processing Instruction node**: adalah perintah pengolahan dalam dokumen XML. Node ini ditandai dengan karakter <? Dan diakhiri dengan ?>.
 - Tapi perlu diingat bahwa header standard XML <?xml version="1.0" encoding="iso-8859-1"?> bukanlah processing instruction node.
 - **Header standard bukanlah bagian dari hirarki pohon dokumen XML.**
- **Namespace Node**: node ini mewakili deklarasi namespace.

Namespace XML

- Bila ada dokumen yang mendeskripsikan tentang kebutuhan material pembuatan gardu jaga dari “bambu”:
- **<bambu>**
 - <jenis>Jawa</jenis>**
 - <panjang>2</panjang>**
- **</bambu>**
- Sedangkan jika terdapat dokumen yang mendeskripsikan “bambu” sebagai merk produk:
- **<bambu>**
 - <jumlah>246</jumlah>**
 - <hargasatuan>200</hargasatuan>**
- **</bambu>**

Namespace XML

- Untuk mengatasi hal ini, Namespace menyediakan metode dengan menggunakan prefiks yang berbeda untuk dokumen pertama:
- **<a:bambu>**
 - **<a:jenis>Jawa</a:jenis>**
 - **<a:panjang>2</a:panjang>**
- **</a:bambu>**
- Sedangkan untuk dokumen kedua menjadi seperti berikut:
- **<b:bambu>**
 - **<b:jumlah>246</b:jumlah>**
 - **<b:hargasatuan>200</b:hargasatuan>**
- **<b:/bambu>**

- Aturan penggunaan namespace adalah sebagai berikut:
<a:bambu xmlns:a="http://www.somewhere.com/gardu">
-
- W3C namespace specification menyatakan bahwa namespace haruslah merupakan Uniform Resource Identifier (URI).

Another example of XML

`<xh:p xmlns:xh='http://www.w3.org/1999/xhtml'>` **MathML**

So the result can be expressed as `<!-- (a+b)2 -->`

`<mml:apply xmlns:mml='http://www.w3.org/TR/REC-MathML'>`

`<mml:power/>`

`<mml:apply>`

`<mml:plus/>`

`<mml:ci>a</mml:ci>`

`<mml:ci>b</mml:ci>`

`</mml:apply>`

`<mml:cn>2</mml:cn>`

`</mml:apply>`

`</xh:p>`

WML

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml" >
<wml>
  <card id="main" title="First Card">
    <p mode="wrap">This is a sample WML page.</p>
  </card>
</wml>
```

Example: XHTML

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head><title>Hello world!</title></head>
  <body>
    <h1>This is a heading</h1>
    This is some text.
  </body>
</html>
```

Example: CML

```
<molecule id="METHANOL">  
  <atomArray>  
    <stringArray builtin="id">a1 a2 a3 a4 a5 a6</stringArray>  
    <stringArray builtin="elementType">C O H H H H</stringArray>  
    <floatArray builtin="x3" units="pm">  
      -0.748 0.558 ...  
    </floatArray>  
    <floatArray builtin="y3" units="pm">  
      -0.015 0.420 ...  
    </floatArray>  
    <floatArray builtin="z3" units="pm">  
      0.024 -0.278 ...  
    </floatArray>  
  </atomArray>  
</molecule>
```

Example: ebXML

```
<MultiPartyCollaboration name="DropShip">
  <BusinessPartnerRole name="Customer">
    <Performs initiatingRole='//binaryCollaboration[@name="Firm Order"]/
      InitiatingRole[@name="buyer"]' />
  </BusinessPartnerRole>
  <BusinessPartnerRole name="Retailer">
    <Performs respondingRole='//binaryCollaboration[@name="Firm Order"]/
      RespondingRole[@name="seller"]' />
    <Performs initiatingRole='//binaryCollaboration[...]/
      InitiatingRole[@name="buyer"]' />
  </BusinessPartnerRole>
  <BusinessPartnerRole name="DropShip Vendor">
    ...
  </BusinessPartnerRole>
</MultiPartyCollaboration>
```

Elemen XML

- ❑ Merupakan inti dari XML
- ❑ Menggunakan tag pembuka dan penutup
- ❑ Bisa juga empty tag
- ❑ Bersifat hirarkis
- ❑ Nama elemen hanya mengandung huruf, nama, dan karakter lain.
- ❑ Nama elemen tidak boleh diawali dengan angka maupun karakter *punctuation*.
- ❑ Nama elemen tidak boleh diawali dengan karakter "xml" atau "XML" atau "Xml".
- ❑ Nama elemen tidak boleh mengandung spasi.

Elemen XML

```
<book>
  <title>My First XML</title>
  <prod id="33-657" media="paper"></prod>
  <chapter>Introduction to XML
    <para>What is HTML</para>
    <para>What is XML</para>
  </chapter>
  <chapter>XML Syntax
    <para>Elements must have a closing tag</para>
    <para>Elements must be properly nested</para>
  </chapter>
</book>
```

- ❑ *element content*: dalam contoh di atas, <book> memiliki *element content*, karena mengandung elemen lain.
- ❑ *mixed content*: dalam contoh diatas, <chapter> memiliki *mixed content* karena berisi teks dan elemen lain.
- ❑ *simple content*: dalam contoh diatas, <para> memiliki *simple content* atau *text content* sebab mengandung hanya teks saja.
- ❑ *empty content*: dalam <prod> memiliki *empty content* karena tidak membawa satu informasi pun.
- ❑ Sebuah elemen juga dapat memiliki atribut
 - Hanya elemen **<prod>** yang memiliki atribut. Atribut "id" memiliki nilai "33-657". Atribut bernama "media" memiliki nilai "paper".

Atribut XML

- ❑ Elemen XML dapat memiliki atribut seperti pada tag HTML
- ❑ Atribut biasanya menyediakan informasi ekstra dari elemen (metadata)
- ❑ Atribut adalah bagian dari elemen
- ❑ Selalu berupa string, diapit oleh double quote
- ❑ Bersifat unlimited, tapi tidak boleh dobel
- ❑ Letaknya boleh tidak terurut
- ❑ Contoh: **File type** tidak relevan dengan data, tetapi penting untuk software yang ingin memanipulasi elemen tersebut.

<file type="gif">computer.gif</file>

Atribut VS Elemen

- Atribut sulit mengandung banyak nilai tetapi elemen bisa.
 - Contoh yang aneh: `<anak nama="a,b,c"></anak>`
- Atribut tidak mudah diubah untuk pengembangan masa yang akan datang, tapi elemen bisa
- Atribut tidak dapat mendeskripsikan strukturnya tetapi elemen bisa.
- Atribut lebih sulit di-*parsing* oleh program penerjemah XML.
- Nilai atribut tidak mudah untuk ditangani oleh DTD yang digunakan untuk mendefinisikan validasi dan legalitas elemen dokumen XML.
- Tapi atribut baik untuk:
 - Jika kita butuh suatu nilai yang strict (misal hanya bisa a,b,c saja)
 - Jika kita butuh suatu nilai default
 - Jika kita butuh suatu satuan spesifik
 - `<weight>10 kg</weight>`
 - `<weight unit="kg">10</weight>`

Contoh-contoh lain

- Attribute values require quotes

```
<billnum type="senate">123</billnum>
```

- The case of elements names must match

```
<billnum type="senate">123</Billnum>
```

- Non-empty elements need end tags

```
<billnum type="senate">123</billnum>
```

- Empty elements require termination

```
<billnum type="senate" reference="5"/>
```

CDATA

- ❑ Semua teks dalam dokumen XML:
 - PCDATA (*Parsed Character Data*) di-parse oleh XML Parser.
 - CDATA (*Character Data*) dilewati oleh XML Parser.
- ❑ Bentuk umum: `<![CDATA[...]]>`
- ❑ CDATA tidak boleh berisi string `"]]>`
- ❑ CDATA tidak boleh bersarang
- ❑ Tidak boleh ada spasi atau enter di dalam string `"]]>`.

Contoh CDATA:

```
<script><![CDATA[  
function matchwo(a,b)  
{  
    if (a < b && a < 0) then  
    {return 1;}  
    else  
    {return 0;}  
}  
]]>/script>
```

Tipe XML

- ❑ **Well Formed XML:** jika mengikuti spesifikasi / aturan XML namun tidak memiliki DTD / XML Schema
- ❑ **Valid XML:** jika mengikuti spesifikasi / aturan XML dan memiliki & sesuai dengan DTD / XML Schema.
- ❑ **Invalid XML:** jika tidak mengikuti aturan XML & tidak memiliki DTD.

Processing Instruction: DTD

- ❑ DTD = Document Type Definition
- ❑ DTD bertujuan untuk mendefinisikan sebuah blok yang valid dari sebuah dokumen XML dan agar setiap dokumen XML memiliki deskripsi yang sesuai dengan formatnya.
- ❑ DTD berisi aturan-aturan yang digunakan untuk memvalidasi sebuah dokumen XML
- ❑ DTD dapat ditulis bersama dengan dokumen XML (internal) ataupun terpisah menjadi file DTD sendiri (eksternal).

DTD

- Jika DTD ditulis dalam file XML (internal) maka harus mengikuti sintaks:
 - **<!DOCTYPE root-element [element-declarations]>.**
- Sedangkan untuk eksternal DTD sintaksnya adalah:
 - **<!DOCTYPE root-element SYSTEM "filename">** dan **<!DOCTYPE root-element PUBLIC FPI "URL">.**
- root – element adalah nama elemen root dari dokumen XML
- Kata kunci SYSTEM berarti DTD tersebut merancu pada DTD local berupa file,
- Kata kunci PUBLIC mengacu pada DTD berupa file pada URL tertentu.

Contoh DTD

Contoh DOCTYPE internal:

```
<?xml version="1.0" ?>
<!DOCTYPE notes [
<!ELEMENT notes (note) +>
<!ELEMENT note (to,from,heading,body*) >
<!ELEMENT to (#PCDATA) >
<!ELEMENT from (#PCDATA) >
<!ELEMENT heading (#PCDATA) >
<!ELEMENT body (#PCDATA) > ] >
<notes>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
</notes>
```

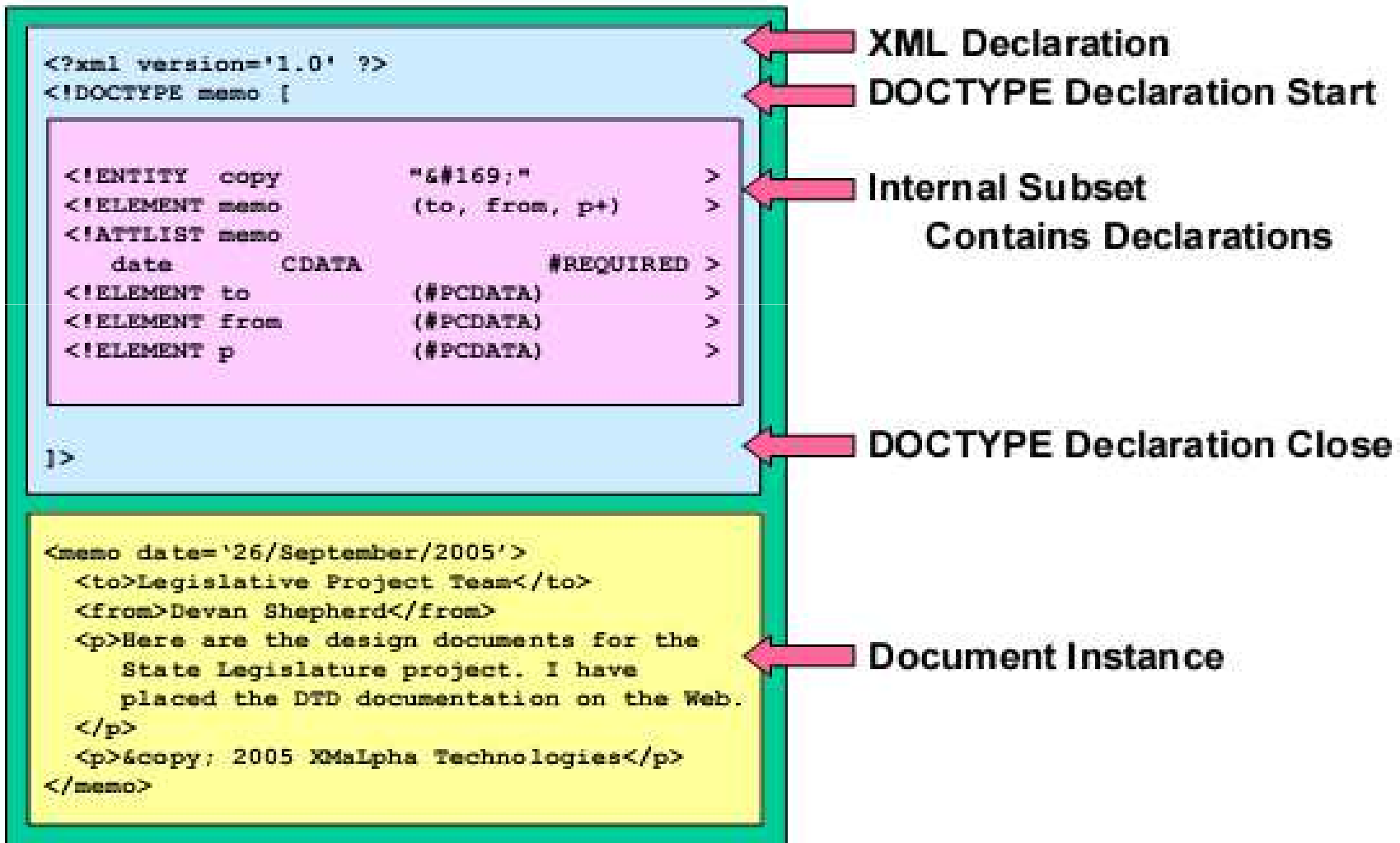
Contoh DOCTYPE SYSTEM:

```
<?xml version="1.0" ?>
<!DOCTYPE notes SYSTEM "note.dtd">
<notes><note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note></notes>
```

note.dtd

```
<!ELEMENT notes (note) +>
<!ELEMENT note (to,from,heading,body*) >
<!ELEMENT to (#PCDATA) >
<!ELEMENT from (#PCDATA) >
<!ELEMENT heading (#PCDATA) >
<!ELEMENT body (#PCDATA) >
```


More Internal DTD



More System DTD

```
<?xml version='1.0' standalone='no'?>
<!DOCTYPE memo SYSTEM 'memo.dtd' >

<memo date='26/September/2005'>
  <to>Legislative Project Team</to>
  <from>Devan Shepherd</from>
  <p>Here are the design documents for the
    State Legislature project. I have
    placed the DTD documentation on the Web.
  </p>
  <p>&copy; 2005 XMaAlpha Technologies</p>
</memo>
```

**XML Declaration,
DOCTYPE Declaration**
Referencing external subset
Followed by DOCTYPE Declaration Close

Document Instance

memo.dtd

External Subset
Contains All Declarations

```
<!ENTITY copy      "&#169;"          >
<!ENTITY % hilite  'b | i | u'      >
<!ELEMENT memo     (to, from, p+)   >
<!ATTLIST memo
  date      CDATA      #REQUIRED   >
<!ELEMENT to      (#PCDATA)        >
<!ELEMENT from    (#PCDATA)        >
<!ELEMENT p       (#PCDATA | %hilite;)* >
```

CONTOH Public DTD

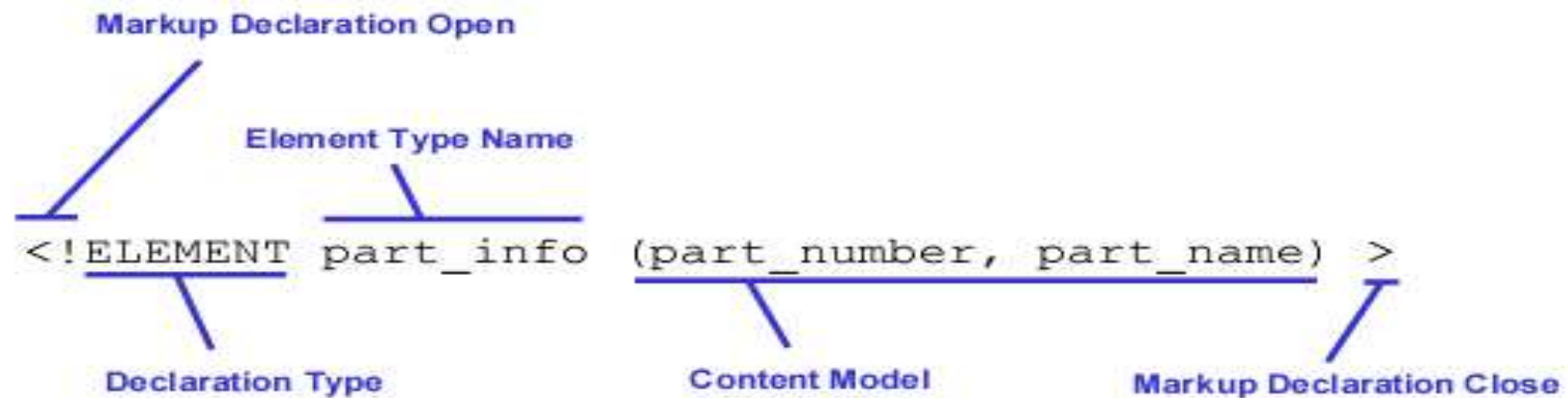
Contoh DTD PUBLIC:

```
<?xml version="1.0" ?>
<!DOCTYPE notes PUBLIC "-//anton/Antonie's Note/ID"
"http://lecturer.ukdw.ac.id/anton/note.dtd">
<notes><note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note></notes>
```

- ❑ **Format FPI (Format Public Identifier)**
- ❑ **"-//anton/Antonie's Note/ID"**
 - Minus (-) : private DTD
 - Plus (+) : non standard DTD yang sudah disepakati
 - Pengguna yang bertanggung jawab terhadap DTD (anton)
 - Deskripsi DTD (Antonie's Note)
 - Bahasa yang digunakan (ID-indonesian)

DTD Element

- Elemen dalam DTD dapat dideklarasikan dengan sintaks:
 - **<!ELEMENT element-name category>** atau
 - **<!ELEMENT element-name (element-content)>**.



DTD Element

- ANY : sembarang data yang well formed XML (kombinasi dari data yang dapat diparsing). Ditulis dengan sintaks:
 - **<!ELEMENT element-name ANY>**
 - Contoh: `<!ELEMENT para0 (#PCDATA | para0_number | subb | supp | br | dol | bl | graphic | fd | f)* >`
- Empty Element : tidak berisi data, seperti: `
`, yang ditulis sesuai sintaks:
 - **<!ELEMENT element-name EMPTY>**
- PCDATA : Elemen yang hanya berisi data karakter ditulis sesuai sintaks:
 - `<!ELEMENT element-name (#PCDATA)>`
 - Contoh: `<!ELEMENT bill_number (#PCDATA)>`
- Child Element : berisi elemen anak yang valid, yang ditulis dengan sintaks:
 - `<!ELEMENT element-name (child-element-name)>` atau
 - `<!ELEMENT element-name (child-element-name,child-element-name,.....)>`
 - Contoh: `<!ELEMENT resolution_structure (bill_info, bill_draft) >`

DTD Element

□ Kardinalitas Element:

- [none] : default/required (only 1 elemen)
- ? : 0 .. 1 (optional)
- * : 0,1 .. N (zero or more)
- + : 1 .. N (one or more)

□ Daftar Operator Element:

- , (koma) : deret urutan
- | : pilihan (satu atau beberapa)
- () : pengelompokkan

Contoh-2

- ❑ **MCM Elements** contain other elements AND data

```
<para1>This paragraph was modified in a Legislative  
Editor with change tracking turned on <add user="Devan  
Shepherd" time="1089610561"> this text was added to the  
paragraph and will be underlined by the  
stylesheet</add>this is a mixed model example</para1>
```

- ❑ Must declare data content before elements in content model
- ❑ Must use "or" separator
- ❑ Must have Optional/Repeatable indicator (*)
- ❑ `<!ELEMENT para1 (#PCDATA | add)* >`

Contoh-2

- "resolution_structure" contains one required "bill_info", followed immediately by one required "bill_draft":
 - <!ELEMENT resolution_structure (bill_info, bill_draft)>
- "bill_shell" contains either one "resolution" or one "session_law":
 - <!ELEMENT bill_shell (resolution | session_law) >
- "Memo" must have one "date", followed by one or more "name", followed by one or more "para", followed by one optional "enclosure":
 - <!ELEMENT memo (date, name+, para+, enclosure?) >
- "bill_section_number" can have any number of "bno_ref" elements or Parsed Character Data in any order:
 - <!ELEMENT bill_sect_number (#PCDATA | bno_ref)*>
- "bno_ref" can only contain Parsed Character Data:
 - <!ELEMENT billno_ref (#PCDATA) >

Soal-soal

```
<!ELEMENT stat_sect_insert (stat_title_info, part_info,  
                             chapter_info, stat_sect_text) >
```

```
<!ELEMENT date ((month, day, year) | (day, month, year)) >
```

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

```
<!ELEMENT tvlisting (show, time, date, (actor, role?)*,  
                     description, length?, (mparating |  
                     tvrating | comments)+ ) >
```


```
<!ELEMENT br EMPTY >
```

Contoh Lain DTD Element

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE Person [
  <!ELEMENT Person ( (Mr|Ms|Miss)?, FirstName,
    MiddleName*, LastName, (Jr|Sr)? )>
  <!ELEMENT FirstName (#PCDATA)>
  <!ELEMENT MiddleName (#PCDATA)>
  <!ELEMENT LastName (#PCDATA)>
  <!ELEMENT Mr EMPTY>
  <!ELEMENT Ms EMPTY>
  ...
  <!ELEMENT Sr EMPTY>
]>
<Person>
  <Mr/>
  <FirstName>Lawrence</FirstName>
  <LastName>Brown</LastName>
</Person>
```

```
<!ELEMENT XXX (AAA+ , BBB+)>
<!ELEMENT AAA (BBB | CCC )
<!ELEMENT BBB (#PCDATA | CCC )*>
<!ELEMENT CCC (#PCDATA)>
```

```
<?xml version="1.0"?>
<!DOCTYPE tutorial SYSTEM "tutorial.dtd">
<XXX>
  <AAA>
    Element: <CCC/>
  </AAA>
  <BBB>
    Element: <CCC/>
  </BBB>
</XXX>
```



Atribut DTD : type

- ❑ Untuk mendefinisikan atribut dalam XML
- ❑ Bentuk umum:
<!ATTLIST elementname attributename attributetype attributevalue>
- ❑ Attribute Type:
 - CDATA : data yang tidak di parsing, yaitu merupakan teks biasa
 - Enumeration: (element1 | element2 | element3)
 - ID : nilai adalah id yang unik untuk elemen tersebut
 - IDREF dan IDREFS : nilai adalah id atau list id (dipisah dengan spasi) dari elemen yang ada
 - NMTOKEN : berupa angka, karakter, koma, titik koma, dan titik tanpa whitespace
 - NMTOKENS : berupa NMTOKEN dipisahkan dgn whitespace
 - ENTITY dan ENTITIES : nilai adalah entity atau list entity dipisahkan dengan spasi
 - NOTATION (value1|value2) : nilai adalah nama notasi

Atribut DTD Element : Value

Nilai Atribut	Arti	Contoh DTD	Contoh XML
Value	Nilai default	<code><!ATTLIST square width CDATA "0"></code>	<code><square width="0"></code>
#REQUIRED	nilai atribut harus diikuti dalam elemen	<code><!ATTLIST person number CDATA #REQUIRED></code>	<code><person number="5677" /></code>
#IMPLIED	nilai atribut tidak harus diikuti dalam elemen	<code><!ATTLIST contact fax CDATA #IMPLIED></code>	<code><contact /></code>
#FIXED	nilai atribut adalah tetap.	<code><!ATTLIST sender company CDATA #FIXED "Microsoft"></code>	<code><sender company="Microsoft" /></code>

- Tipe ID dapat berisi hanya karakter yang diijinkan oleh NMTOKEN dan harus dimulai dengan huruf. Tidak ada tipe elemen yang memiliki lebih dari satu ID. Nilai atribut ID harus unik antara semua atribut ID.

```

<!ELEMENT XXX (AAA+ , BBB+ , CCC+) >
<!ELEMENT AAA (#PCDATA) >
<!ELEMENT BBB (#PCDATA) >
<!ELEMENT CCC (#PCDATA) >
<!ATTLIST AAA id ID #REQUIRED>
<!ATTLIST BBB code ID #IMPLIED
list NMTOKEN #IMPLIED>
<!ATTLIST CCC
X ID #REQUIRED
Y NMTOKEN #IMPLIED>

```

More DTD Atribut : Value

```
<!ELEMENT XXX (AAA+, BBB+)>
<!ELEMENT AAA (#PCDATA)>
<!ELEMENT BBB (#PCDATA)>
<!ATTLIST AAA true ( yes | no ) #REQUIRED>
<!ATTLIST BBB
  month (1|2|3|4|5|6|7|8|9|10|11|12) #IMPLIED>
```

```
<!ELEMENT XXX (AAA+, BBB+)>
<!ELEMENT AAA (#PCDATA)>
<!ELEMENT BBB (#PCDATA)>
<!ATTLIST AAA true ( yes | no ) "yes">
<!ATTLIST BBB month NMTOKEN "1">
```

```
<!ATTLIST image id          ID          #REQUIRED
                  caption   CDATA       #IMPLIED
                  rights     (private | public) 'public'
                  format     NOTATION    (jpeg | gif) 'jpeg' >
```

```
<image id='G1234' format='gif'>
```

Default Value Type

Default Value

DTD Comments

```
<!-- Devan, XMaAlpha Technologies: 25 September, 2005  
bill_sect_info comprises an information block with  
references to components used to create a bill. Since the  
model permits multiple (one or more) bill sections, these  
referants are identifying for each individual section. -->  
<!ELEMENT bill_sect_info (bill_sect_number, bill_sect_name,  
bill_sect_text)>
```

DTD Entity

- Dengan menggunakan entity XML, kita bisa menggantikan kalimat yang panjang atau satu blok elemen yang sering kita gunakan dengan sebuah pengenal singkat.
- Misalnya kita ingin menggantikan kalimat “Manajemen Data dan Informasi dengan XML/XSL” dengan entity **&judul**.

Contoh Entity

```
<?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>
```


Hasil



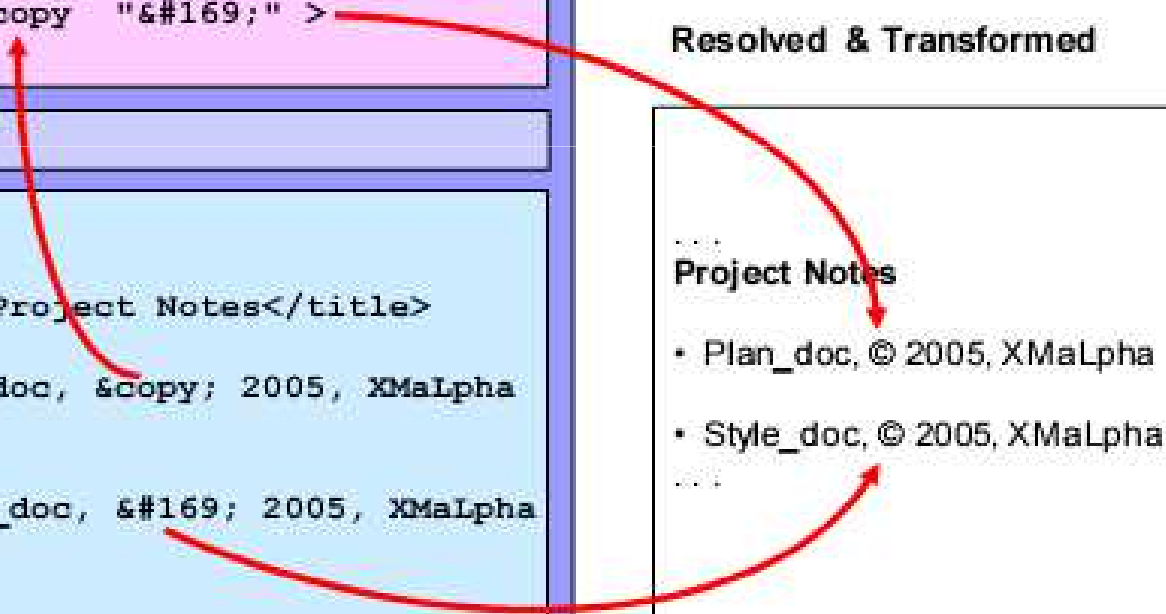
DTD Entity

Source

```
<!DOCTYPE memo [  
  <!ENTITY copy "&#169;" >  
>  
...  
<list>  
  <title>Project Notes</title>  
  <item>  
    Plan_doc, &copy; 2005, XMaLpha  
  </item>  
  <item>  
    Style_doc, &#169; 2005, XMaLpha  
  </item>  
</list>  
...</pre>
```

Resolved & Transformed

```
...  
Project Notes  
• Plan_doc, © 2005, XMaLpha  
• Style_doc, © 2005, XMaLpha  
...
```



NEXT

- XML Schema, Xpath,
- XSL dan XSLT