XML (eXtensible Markup Language) -from few books and W3School

Why XML?

- Web app bergerak ke Web Service
- Web Service adalah web application dengan kemampuan brkomunikasi dengan apllikasi yg lain
- Untuk integrasi atau bekerja bersama
- Memanfaatkan data atau prosedur/ function dari sistem yg berbeda
- Mashup di layer machine

- Backbone tech Web Serv adalah XML
- Data dipaket dalam bentuk XML dan dikomunikasikan dgn open protokol (ex: SOAP)
- WSDL (Web Service Desc Lang) → mendefinisikan lang yg digunakan dalam SOAP.
- Juga berbentuk XML
- PHP can handle it :-)

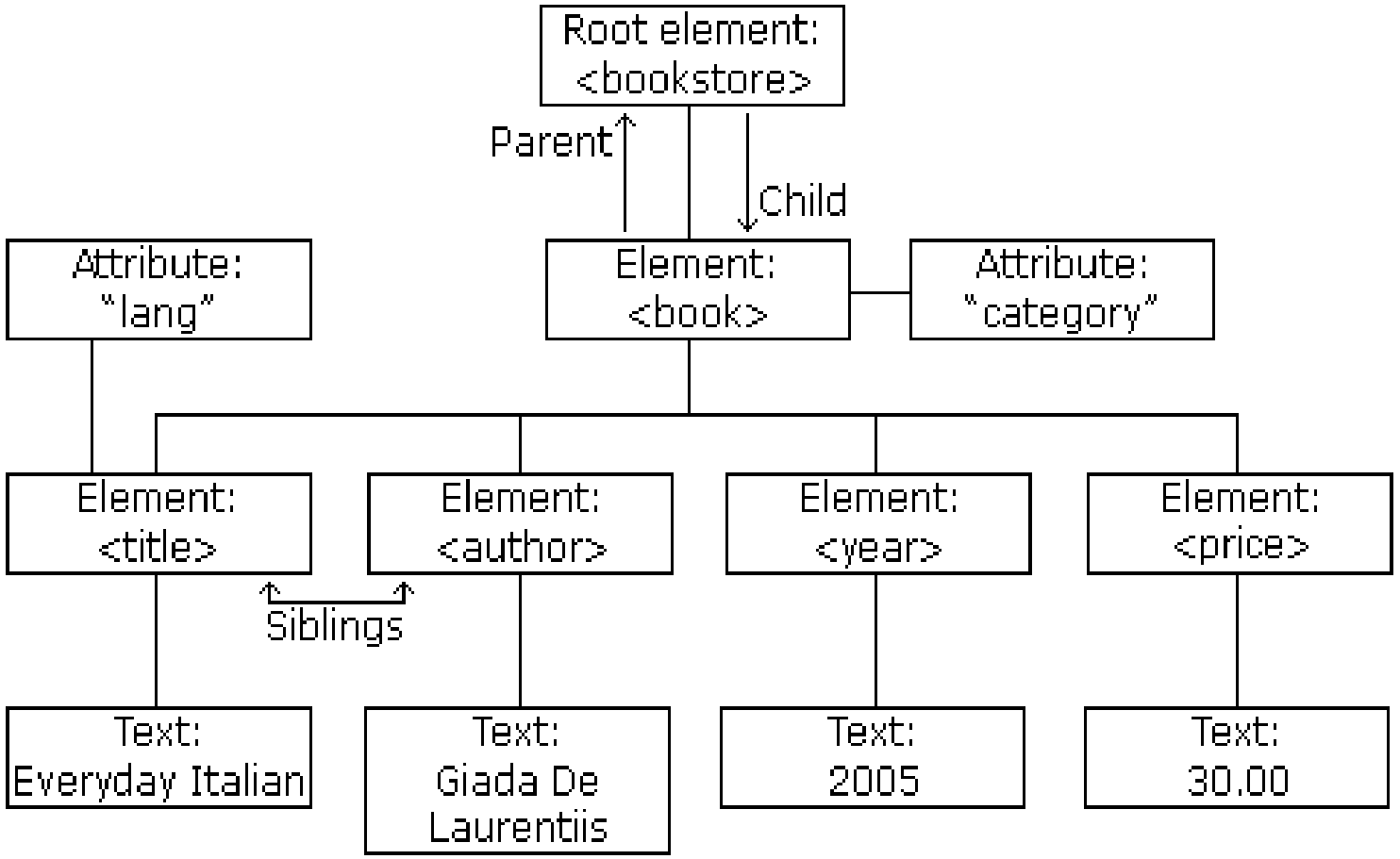
  (jk msh cukup waktu, akan dibahas di chapter yg lain)

XML

- Didesain untuk transport data dan penyimpanannya bukan menampilkannya
- Tags are not predefined. You must define your own tags, its designed to be self-descriptive
- XML is complement of HTML
- HTML dan JS yang akan mengambil data di XML dan menampilkan di Webpages, sehingga HTML dan XML terpisah but complemet each other
- With XML, machine also can understand data

- Its W3 recommendation
- Base of many new web techs:
  - XHTML
  - WSDL for describing available web services
  - WAP and WML as markup languages for handheld devices
  - RSS languages for news feeds
  - RDF and OWL for describing resources and ontology
  - SMIL for describing multimedia for the web

- XML documents form a tree structure that starts at "the root" and branches to "the leaves"
- Imagine, struktur XML spt tree, ada root, percabangan, dan anak cabang lagi dst
- Mendefinisikan tag sendiri, depend on the need, tdk spt HTML yg memiliki reserved keyword tag and properties


Root element:
<bookstore>
Parent
Child
Attribute:
"lang"
Element:
<book>
Attribute:
"category"
Element:
<title>
Element:
<author>
Element:
<year>
Element:
<price>
Siblings
Text:
Everyday Italian
Text:
Giada De
Laurentiis
Text:
2005
Text:
30.00

```
<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">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Element → book
Attribute → category

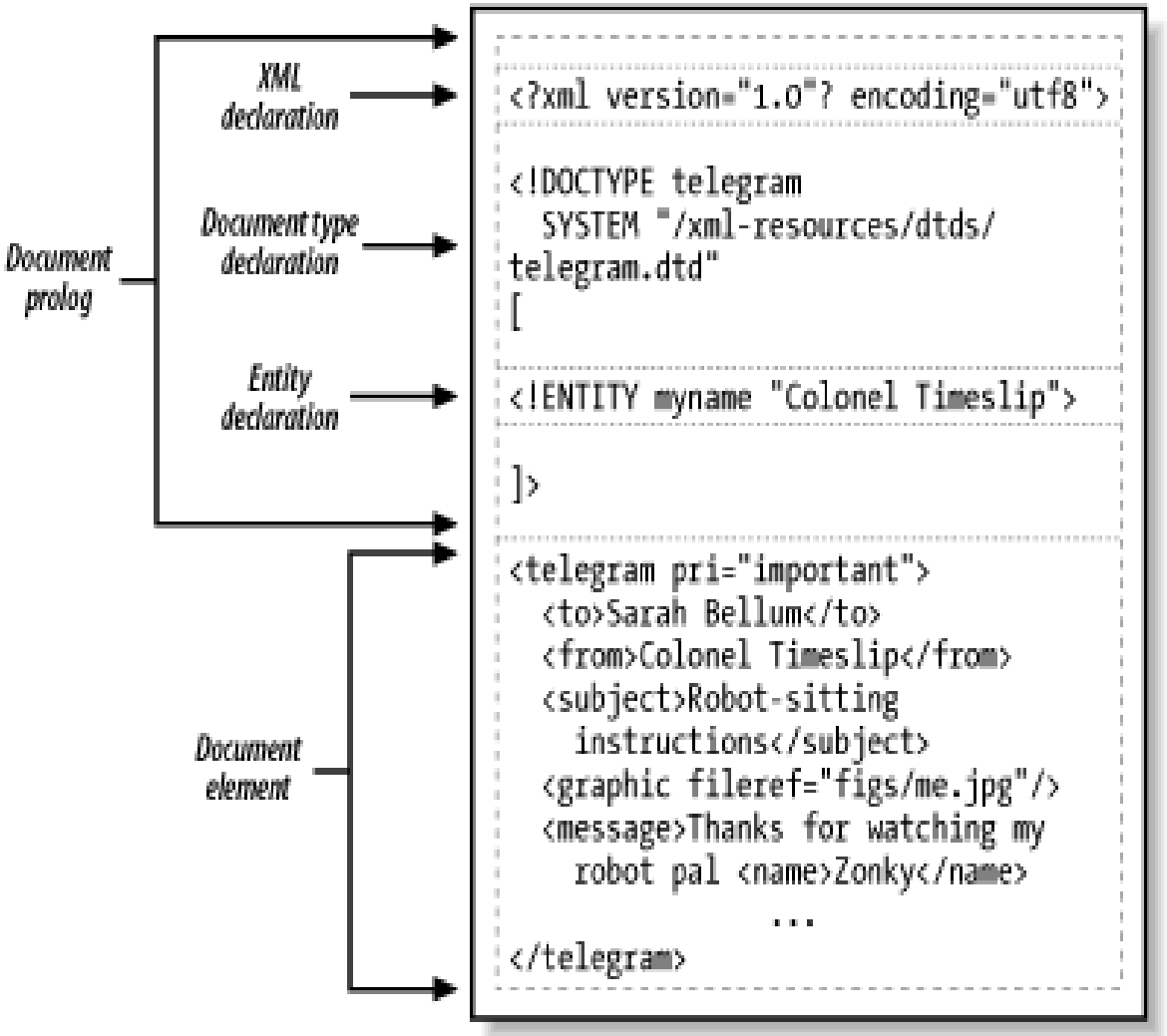
- Student information with HTML
- Data be written in HTML bundle then can be showed in browser
- Data from database

```
<table border="1">
<tr>
<td>Name : Jono Sujono</td>
<td>NIM : TI3670307</td>
<td>Birthplace : Sragen</td>
<td>Birthday : 12021986</td>
<td>Address : Jl.Senopati no.8, Sragen</td>
<td>Religion : Islam</td>
<td>Email : jono@uns.ac.id</td>
<td>Mobilephone : 123456789</td>
<td>IPK : 3.6</td>
</tr>
<tr>
<td>Name : Mono Sunono</td>
<td>NIM : TI3671307</td>
<td>Birthplace : Surakarta</td>
<td>Birthday : 02091986</td>
<td>Address : Mojosongo,Surakarta</td>
<td>Religion : Hindu</td>
<td>Email : mono@uns.ac.id</td>
<td>Mobilephone : 023456089</td>
<td>IPK : 3.5</td>
</tr>
</table>
```

Name : Jono Sujono	NIM : TI3670307	Birthplace : Sragen	Birthday : 12021986	Address : Jl.Senopati no.8, Sragen	Religion : Islam	Email : jono@uns.ac.id	Mo
Name : Mono Sunono	NIM : TI3671307	Birthplace : Surakarta	Birthday : 02091986	Address : Mojosongo,Surakarta	Religion : Hindu	Email : mono@uns.ac.id	Mo

- In XML, a simple one just want to show the structured of XML

```
<?xml version="1.0" encoding="UTF-8"?>
<student_information>
    <student>
        <name>Jono Sujono</name>
        <nim>TI3670307</nim>
        <birthplace>Sragen</birthplace>
        <birthday>12021986</birthday>
        <address>Jl.Senopati no.8, Sragen</address>
        <religion>Islam</religion>
        <email>jono@uns.ac.id</email>
        <mobilephone>123456789</mobilephone>
        <ipk>3.6</ipk>
    </student>
    <student>
        <name>Mono Sumono</name>
        <nim>TI3671307</nim>
        <birthplace>Surakarta</birthplace>
        <birthday>02091986</birthday>
        <address>Mojosongo, Surakarta</address>
        <religion>Hindu</religion>
        <email>mono@uns.ac.id</email>
        <mobilephone>223456789</mobilephone>
        <ipk>3.5</ipk>
    </student>
</student_information>
```



”id” can be used
for identifying
the different itemset

attributes cannot contain multiple values (elements can)
attributes cannot contain tree structures (elements can)
attributes are not easily expandable (for future changes)

- ”id” can be used for identifying the different itemset
-

```
<messages>
  <note id="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note id="502">
    <to>Jani</to>
    <from>Tove</from>
    <heading>Re: Reminder</heading>
    <body>I will not</body>
  </note>
</messages>
```

XML Naming

- Naming Rules
  - Names can contain letters, numbers, and other characters
  - Names cannot start with a number or punctuation character
  - Names cannot start with the keyword xml (or XML, or Xml, etc)
  - Names cannot contain spaces

"Well Formed" XML

- Adalah yang benar sintaknya
  - XML documents must have a root element
  - XML elements must have a closing tag
  - XML tags are case sensitive
  - XML elements must be properly nested
  - XML attribute values must be quoted
- Ex: above examples are well format XML

"Valid" XML

- Yang memenuhi aturan DTD (Data Type Definition)
- DTD is using to define the structure of an XML document.
-

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

- With a DTD, independent groups of people can agree to use a standard DTD for interchanging data.
- Your application can use a standard DTD to verify that the data you receive from the outside world is valid.
- Detail DTD, silakan dipelajari sendiri, banyak sumber

XML Schema

- XML Schema adalah alternatif dari fungsi DTD, better

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">

<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

XSLT

- XSLT (eXtensible Stylesheet Language Transformations)
- Style sheet language of XML, for displaying XML, seperti CSS but khusus untuk XML, bagian dari XSL, XSL, describes how the XML document should be displayed
  - XSLT - a language for transforming XML documents
  - XPath - a language for navigating in XML documents
  - XSL-FO - a language for formatting XML documents

- XSLT menjadi spt bagian yg plg penting dalam XSL
- XSLT transform an XML document into another XML document, or another type of document that is recognized by a browser, like HTML and XHTML
- XSLT transforms an XML source-tree into an XML result-tree
- XPath is a language for navigating in XML documents, define parts of the source document that should match one or more predefined templates

ex:another xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<?xml-stylesheet type="text/xsl" href="contoh.xsl"?>
<catalog>
        <cd>
                <title>Empire Burlesque</title>
                <artist>Bob Dylan</artist>
                <country>USA</country>
                <company>Columbia</company>
                <price>10.90</price>
                <year>1985</year>
        </cd>
        <cd>
                <title>Hide your heart</title>
                <artist>Bonnie Tyler</artist>
                <country>UK</country>
                <company>CBS Records</company>
                <price>9.90</price>
                <year>1988</year>
        </cd>
        <cd>
                <title>Greatest Hits</title>
                <artist>Dolly Parton</artist>
                <country>USA</country>
```

add xsl into XML syntac

And the xsl

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th>Title</th>
        <th>Artist</th>
<th>Country</th>
<th>Price</th>
<th>Year</th>
      </tr>
      <tr>
        <td><xsl:value-of select="catalog/cd/title" /></td>
        <td><xsl:value-of select="catalog/cd/artist" /></td>
<td><xsl:value-of select="catalog/cd/country" /></td>
<td><xsl:value-of select="catalog/cd/price" /></td>
<td><xsl:value-of select="catalog/cd/year" /></td>
      </tr>
    </table>
  </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

CD Collection

Title	Artist	Country	Price	Year
Empire Burlesque	Bob Dylan	USA	10.90	1985

- Add for-each

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>CD Collection</h2>
    <table border="1">
      <tr bgcolor="red">
        <th>Title</th>
        <th>Artist</th>
        <th>Country</th>
        <th>Company</th>
        <th>Price</th>
        <th>Year</th>
      </tr>
      <xsl:for-each select="catalog/cd">
      <tr>
        <td><xsl:value-of select="title" /></td>
        <td><xsl:value-of select="artist" /></td>
        <td><xsl:value-of select="country" /></td>
        <td><xsl:value-of select="company" /></td>
        <td><xsl:value-of select="price" /></td>
        <td><xsl:value-of select="/year" /></td>
      </tr>
      </xsl:for-each>
    </table>
  </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

CD Collection

Title	Artist	Country	Company	Price	Year
Empire Burlesque	Bob Dylan	USA	Columbia	10.90	1985
Hide your heart	Bonnie Tyler	UK	CBS Records	9.90	1988
Greatest Hits	Dolly Parton	USA	RCA	9.90	1982
Still got the blues	Gary Moore	UK	Virgin records	10.20	1990
Eros	Eros Ramazzotti	EU	BMG	9.90	1997
One night only	Bee Gees	UK	Polydor	10.90	1998
Sylvias Mother	Dr.Hook	UK	CBS	8.10	1973
Maggie May	Rod Stewart	UK	Pickwick	8.50	1990
Romanza	Andrea Bocelli	EU	Polydor	10.80	1996
When a man loves a woman	Percy Sledge	USA	Atlantic	8.70	1987
Black angel	Savage Rose	EU	Mega	10.90	1995
1999 Grammy Nominees	Many	USA	Grammy	10.20	1999
For the good times	Kenny Rogers	UK	Mucik Master	8.70	1995
Big Willie style	Will Smith	USA	Columbia	9.90	1997
Tupelo Honey	Van Morrison	UK	Polydor	8.20	1971

PHP and XML

- Need parser, ex:Expat Parser
- Make its possible to process XML documents in PHP
- Terdapat built in function untk menangani XML
- XML Parser:
  - Structure based
  - Content based

- Banyak cara parsing dengan PHP, berikut yang termudah

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
        <cd>
                <title>Empire Burlesque</title>
                <artist>Bob Dylan</artist>
                <country>USA</country>
                <company>Columbia</company>
                <price>10.90</price>
                <year>1985</year>
        </cd>
```

```php
<?php
$xml = simplexml_load_file("contoh.xml");

echo $xml->getName() . "<br />";
foreach($xml->children() as $child)
  {
  echo $child->getName() . ": " . $child . "<br />";
  }
?>
```

cd
title: Empire Burlesque
artist: Bob Dylan
country: USA
company: Columbia
price: 10.90
year: 1985

- Untuk data yang banyak, spt contoh sebelumnya

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<?xml-stylesheet type="text/xsl" href="contoh.xsl"?>
<catalog>
        <cd>
                <title>Empire Burlesque</title>
                <artist>Bob Dylan</artist>
                <country>USA</country>
                <company>Columbia</company>
                <price>10.90</price>
                <year>1985</year>
        </cd>
        <cd>
                <title>Hide your heart</title>
                <artist>Bonnie Tyler</artist>
                <country>UK</country>
                <company>CBS Records</company>
                <price>9.90</price>
                <year>1988</year>
        </cd>
        <cd>
                <title>Greatest Hits</title>
                <artist>Dolly Parton</artist>
                <country>USA</country>
```

- A bit modification

```php
<?php
$xml = simplexml_load_file("contoh.xml");

echo $xml->getName() . "<br />";
foreach($xml->children() as $child)
  {
  echo $child->getName() . ": " . $child . "<br />";//nested loop to parse XML doc
  foreach($child->children() as $child2)
  {
  echo $child2->getName() . ": " . $child2 . "<br />";
  }
  }
?>
```

- Hasilnya akan berulang

- A bit modification

```php
<?php
$xml = simplexml_load_file("contoh.xml");

echo $xml->getName() . "<br />";
foreach($xml->children() as $child)
  {
  echo $child->getName() . ": " . $child . "<br />";//nested loop to parse XML doc
  foreach($child->children() as $child2)
  {
  echo $child2->getName() . ": " . $child2 . "<br />";
  }
  }
?>
```

- Hasilnya akan berulang

catalog
cd:
title: Empire Burlesque
artist: Bob Dylan
country: USA
company: Columbia
price: 10.90
year: 1985
cd:
title: Hide your heart
artist: Bonnie Tyler
country: UK
company: CBS Records
price: 9.90
year: 1988
cd:
title: Greatest Hits
artist: Dolly Parton
country: USA
company: RCA
price: 9.90
year: 1982

cd:
title: Still got the blues
artist: Gary Moore
country: UK
company: Virgin records
price: 10.20
year: 1990
cd:
title: Eros
artist: Eros Ramazzotti
country: EU
company: BMG
price: 9.90
year: 1997
cd:
title: One night only
artist: Bee Gees
country: UK
company: Polydor
price: 10.90
year: 1998
cd:
title: Sylvias Mother
artist: Dr.Hook
country: UK
company: CBS
price: 8.10
year: 1973

XML and MySQL

- Database dalam penyajian data dapat menggunakan XML
- XML lebih menguntungkan drpd HTML, karena dengan XML, data dapat dipertukarkan, diambil atau digabungkan over internet
- Secara khusus XML menjadi aspek technical paling dasar dalam perkembangan evolusi Web 3.0 → Semantic, make machine understand data
- XML makes machine understand the meaning

- Konsep tdk berbeda dengan menampilkan data dgn HTML, but in XML format
- XML as a web delivery format, data lebih mudah be shared, be integrated by dififferent applications
- Dengan XML data spread easier over the web
- Lebih mudah extracts information from the document using standard XML tools
- Php already has built in function to handle it, a many different ways to express it

- Existing relational database
- Show on webpage in XML format

Id	Pwd
abc123	123abc
goes*	*goes
var999	999var

```php
<?php
header("Content-type: text/xml");//make sure to define
$host = "localhost";
$user = "root";
$pass = "dww";
$database = "Dataku";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($database, $linkID) or die("Could not find database.");

$query = "SELECT * FROM Member";
$resultID = mysql_query($query, $linkID) or die("Data not found.");

$xml_output = "<?xml version=\"1.0\"?>\n";//also these
$xml_output .= "<Anggota>\n";

for($x = 0 ; $x < mysql_num_rows($resultID) ; $x++){
    $row = mysql_fetch_assoc($resultID);
    $xml_output .= "\t<Data>\n";
    $xml_output .= "\t\t<Id>" . $row['Id'] . "</Id>\n";
    $xml_output .= "\t\t<Pwd>" . $row['Pwd'] . "</Pwd>\n";
    $xml_output .= "\t</Data>\n";
}
$xml_output .= "</Anggota>";
echo $xml_output;
?>
```

- Result

  tanpa xsl

```
−<Anggota>
  −<Data>
      <Id>abc123</Id>
      <Pwd>123abc</Pwd>
    </Data>
  −<Data>
      <Id>goes*</Id>
      <Pwd>*goes</Pwd>
    </Data>
  −<Data>
      <Id>var999</Id>
      <Pwd>999var</Pwd>
    </Data>
</Anggota>
```

- Add xsl

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>Anggota</h2>
    <table border="1">
      <tr bgcolor="red">
        <th>Id</th>
        <th>Pwd</th>
      </tr>
      <xsl:for-each select="Anggota/Data">
      <tr>
        <td><xsl:value-of select="Id" /></td>
        <td><xsl:value-of select="Pwd" /></td>
      </tr>
      </xsl:for-each>
    </table>
  </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

- Add line xsl
-

```php
<?php
header("Content-type: text/xml");//make sure to define
$host = "localhost";
$user = "root";
$pass = "dww";
$database = "Dataku";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($database, $linkID) or die("Could not find database.");

$query = "SELECT * FROM Member";
$resultID = mysql_query($query, $linkID) or die("Data not found.");

$xml_output = "<?xml version=\"1.0\"?>\n";//also these
$xml_output .= "<?xml-stylesheet type='text/xsl' href='my.xsl'?>\n";// line xsl
$xml_output .= "<Anggota>\n";

for($x = 0 ; $x < mysql_num_rows($resultID) ; $x++){
    $row = mysql_fetch_assoc($resultID);
    $xml_output .= "\t<Data>\n";
    $xml_output .= "\t\t<Id>" . $row['Id'] . "</Id>\n";
    $xml_output .= "\t\t<Pwd>" . $row['Pwd'] . "</Pwd>\n";
    $xml_output .= "\t</Data>\n";
}
$xml_output .= "</Anggota>";
echo $xml_output;
?>
```

- And result

Anggota

Id	Pwd
abc123	123abc
goes*	*goes
var999	999var