

Makalah Seminar Tugas Akhir

PEMBUATAN PROGRAM UNTUK MENGUBAH FILE DATABASE MENJADI XML

Oleh : Darmawan (L2F 097 619)
Jurusan Teknik Elektro Universitas Diponegoro Semarang

Abstrak

Seiring dengan perkembangan teknologi informasi, khususnya internet, berkembang pula sebuah bahasa pemrograman web yang dapat mendukung kerja HTML yaitu XML (*Extensible Markup Language*). XML adalah sebuah markup language yang menyediakan format untuk mendeskripsikan data yang terstruktur. Data pada dokumen XML dimungkinkan berasal dari sebuah sumber data, misalnya database. Dalam tugas akhir ini, dibuat sebuah program aplikasi yang dapat menghasilkan dokumen XML dari sebuah file database. Program aplikasi dibuat memakai bahasa pemrograman Borland Delphi 6, sedangkan format database yang digunakan sebagai contoh adalah Paradox (*.db), dbase (*.dbf) dan Interbase server. Pengujian terhadap dokumen XML hasil konversi dari file database dilakukan dengan menampilkan dokumen tersebut dalam browser Internet Explorer 6.

1. Pendahuluan

Dunia teknologi informasi pada akhir abad 20 telah berkembang dengan pesat. Internet sebagai salah satu hasil dari perkembangan teknologi informasi telah memberikan dampak perubahan yang sangat mendasar dalam kehidupan manusia, terutama perubahan perilaku dan cara berpikir yang semakin berorientasi pada efektifitas dan efisiensi.

Perkembangan internet tersebut juga berdampak pada halaman-halaman web. Halaman web sekarang tidak hanya berisikan informasi saja, akan tetapi telah berubah menjadi halaman web aplikasi untuk memenuhi kebutuhan user yang semakin banyak dan kompleks.

Kebutuhan untuk web aplikasi ini berimplikasi pada bahasa yang sering digunakan untuk pembuatan web yaitu HTML (*Hypertext Markup Language*). HTML merupakan *markup language* untuk *document encoding* dan *display* yang banyak digunakan untuk halaman web. Akan tetapi HTML tidak memiliki struktur dalam dokumennya sehingga menyebabkan kesulitan pada aplikasi untuk membaca dan menganalisa sebuah data.. Untuk mengatasi hal tersebut digunakanlah XML (*Extensible Markup Language*).

XML adalah suatu *markup language* yang menyediakan format untuk mendeskripsikan data terstruktur. Sehingga pemakaian XML terutama digunakan untuk web yang memakai database dengan ukuran besar.

XML memiliki beberapa keunggulan apabila dibandingkan dengan HTML diantaranya :

- **Ekstensibilitas**

Pengguna XML atau *user* bebas menentukan tag-tag sendiri sesuai dengan kebutuhannya..

- **Pemisahan data dengan presentasi**

Pada sebuah dokumen XML, data yang terdapat di dalamnya menerangkan data itu sendiri demikian pula tag-tagnya menjelaskan mengenai isi datanya, sedangkan pada HTML antara data dengan cara *browser* menampilkan pada web dicampur adukkan sehingga isi data menjadi tidak jelas.

- **Fungsi search yang lebih tepat**

Apabila sebuah web dibuat dalam format XML dimana XML menjelaskan arti yang sebenarnya dari sebuah isi maka orang akan lebih mudah menemukan informasi yang dibutuhkan.

- **Dapat ditukar/ digabung dengan dokumen XML lain**

Karena di dalam XML terdapat pemisahan antara isi dan cara menampilkannya maka akan mempermudah dalam penggabungan data dari berbagai macam sumber menjadi satu dokumen XML tanpa perlu memperhatikan susunan tag ataupun susunan data.

- **Penulisan sekali untuk bermacam pemakaian**

Keuntungan lain dari pemisahan ini adalah sebuah file XML dapat ditulis sekali kemudian ditampilkan dengan cara yang bermacam-macam seperti di *web browser*, *handphone*, diubah ke suara dan lain-lain.

Dokumen atau file XML mengandung suatu data sehingga dimungkinkan untuk menghasilkan

dokumen tersebut dari sebuah sumber data, misalnya *database*.

2. Struktur XML

Secara umum, struktur XML berupa data yang tersusun dalam bentuk pohon (*tree*). Tag-tag yang terdapat dalam XML dapat didefinisikan sesuai dengan keinginan dari pembuatnya tanpa harus terikat pada aturan tertentu. Seperti pada HTML, tag XML diawali oleh tag pembuka `<tag>` dan diakhiri dengan tag penutup `</tag>`. Standar penulisannya adalah :

```
<?xml version="1.0" encoding =
"UTF-8" standalone = "yes"?>
<tag>
isi dokumen
</tag>
```

Penulisan baris pertama pada dokumen XML harus sesuai dengan urutan seperti diatas dan tidak boleh terbalik. Baris pertama `<?xml version="1.0">` menunjukkan bahwa file tersebut ditulis sesuai dengan aturan XML versi 1.0. Angka 1.0 merupakan atribut sehingga harus diberi tanda petik. *Encoding* menyatakan bahwa file XML ditulis dengan kode UTF-8. Penulisan *standalone* = *yes* menunjukkan bahwa file XML tersebut berdiri sendiri dan tidak ada file lain yang mendukung. File yang mendukung biasanya adalah file DTD (*Document Type Definition*). Sebuah file XML minimal harus mempunyai satu elemen utama. Elemen terdiri dari tag pembuka, tag penutup serta semua yang berada diantara tag pembuka dan tag penutup.

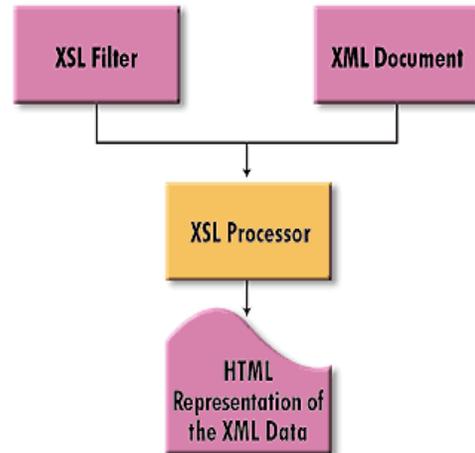
Dokumen XML dikatakan *well-formed* apabila telah memenuhi syarat – syarat :

- Setiap ada tag pembuka, harus ada tag penutup
- Tidak ada elemen yang overlapping
- Memiliki satu elemen utama
- Atribut harus diberi tanda petik

Dokumen XML yang *well-formed* dapat difaktorkan dengan menggunakan XML parser tanpa memunculkan pesan kesalahan (*error message*). XML parser adalah sebuah program untuk menguji apakah suatu dokumen XML benar atau tidak, sudah memenuhi aturan-aturan XML atau tidak. Untuk saat sekarang, ada beberapa browser seperti *Internet Explorer 5.0*, *Netscape 6.0/Mozilla* dan *Opera 4.0 beta* yang telah membuat XML parser di dalam aplikasi mereka, sehingga hasilnya dapat langsung dilihat dalam halaman web.

Untuk menampilkan dokumen XML dalam halaman web, dibutuhkan XSL (*Extensible Stylesheet Language*). XSL adalah sebuah bahasa yang berfungsi menjelaskan arti dari tag-tag yang digunakan dalam XML sehingga dapat dikenali oleh browser untuk kemudian ditampilkan sebagai

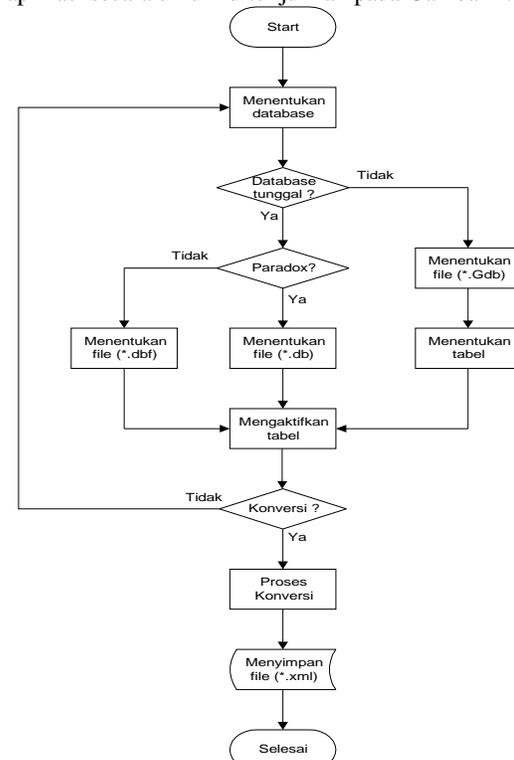
halaman web. Bagan proses untuk menampilkan XML dengan *web browser* ditunjukkan pada Gambar 1.



Gambar 1. Proses menampilkan XML dengan *web browser*

3. Perancangan Program

Perancangan program aplikasi untuk mengubah format *database* menjadi XML terdiri atas tiga prosedur utama, yaitu prosedur untuk menentukan *database*, prosedur mengaktifkan file *database* dan prosedur mengkonversi file *database* menjadi dokumen xml. Diagram alir program aplikasi secara umum ditunjukkan pada Gambar 2.



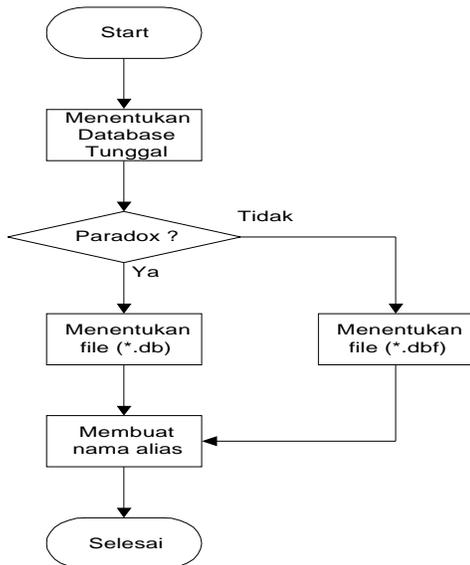
Gambar 2. Diagram alir program aplikasi secara umum

3.1 Prosedur Menentukan Database

Prosedur untuk menentukan *database* terbagi menjadi dua bagian, yaitu prosedur untuk menentukan *database tunggal* dan prosedur untuk menentukan *database server*.

3.1.1 Database Tunggal

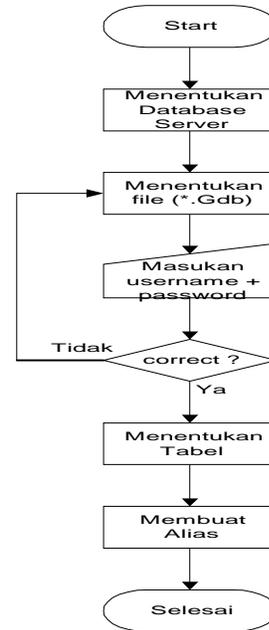
Pada saat menu *database tunggal* dipilih, aplikasi akan membuka menu untuk menentukan format *database* yaitu Paradox (*.db) dan dbase (*.dbf). Setelah pemilihan format *database* dilakukan, program akan masuk dalam menu *browse* untuk menentukan file *database* yang akan dikonversi. Pada proses *browse* ini juga dilakukan filter sehingga hanya file dengan ekstensi .db atau .dbf saja yang dapat dipilih. Setelah menentukan file yang akan dikonversi, aplikasi akan membuat nama alias untuk file tersebut sehingga dapat dikenali oleh Delphi. Diagram alir untuk prosedur menentukan database tunggal ditunjukkan pada Gambar 3.



Gambar 3. Diagram alir prosedur menentukan database tunggal

3.1.2 Database Server

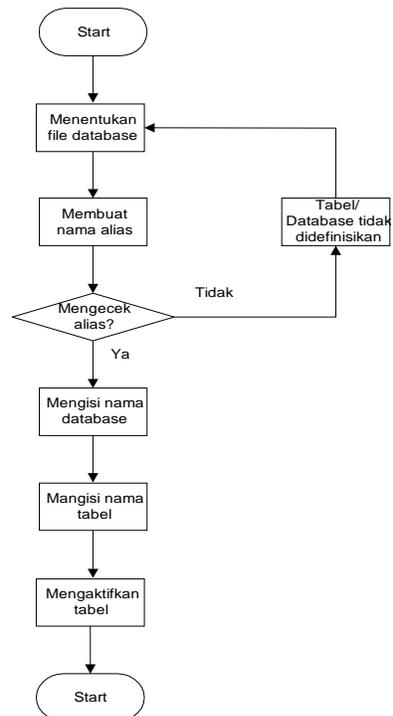
Untuk menu pilihan *database server*, pada saat dijalankan, aplikasi akan membuat sebuah daftar tabel dari file *database server*. Hal ini dikarenakan untuk sebuah file *database server* mempunyai jumlah tabel yang lebih dari satu. Proses pembuatan daftar tabel ini bersamaan dengan menu *browse* untuk memilih file *database server* dengan format .Gdb. Setelah nama file dipilih, kemudian file tersebut dijadikan sebagai *server* dengan username = SYSDBA dan password = masterkey. Diagram alir untuk prosedur memilih *database server* ditunjukkan pada Gambar 4.



Gambar 4. Diagram alir prosedur menentukan database server

3.2 Prosedur Mengaktifkan File Database

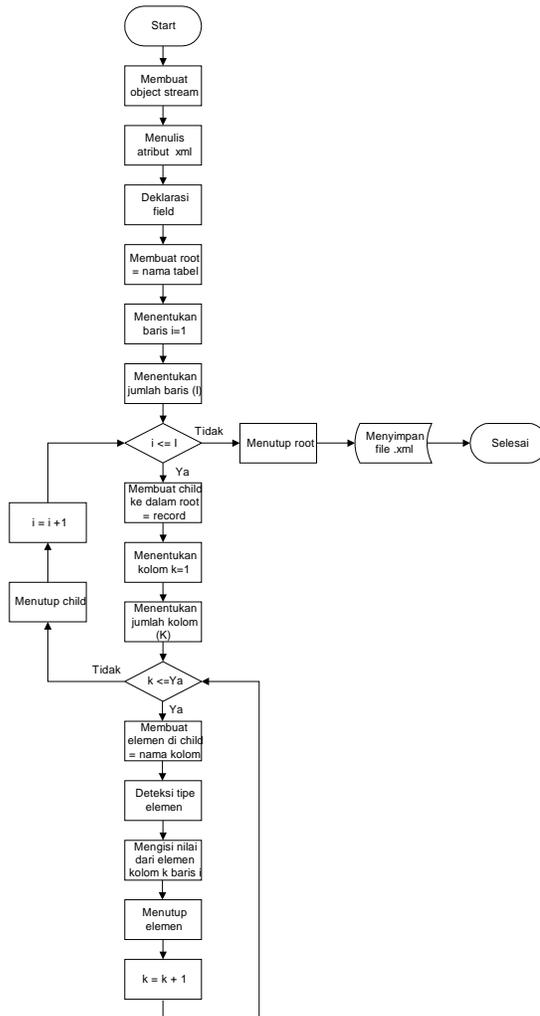
File *database* yang telah dipilih untuk dikonversi, harus diaktifkan terlebih dahulu sebelum dikonversi menjadi dokumen XML. Diagram alir prosedur untuk mengaktifkan file *database* ditunjukkan pada Gambar 5.



Gambar 5. Diagram alir prosedur mengaktifkan file database

3.3 Prosedur mengkonversi database menjadi dokumen XML

Prosedur utama dari program aplikasi ini adalah prosedur melakukan konversi atau perubahan dari suatu format *database* menjadi sebuah dokumen XML. Diagram alir untuk prosedur konversi dari format *database* menjadi dokumen xml ditunjukkan pada Gambar 6.



Gambar 6. Diagram alir prosedur konversi

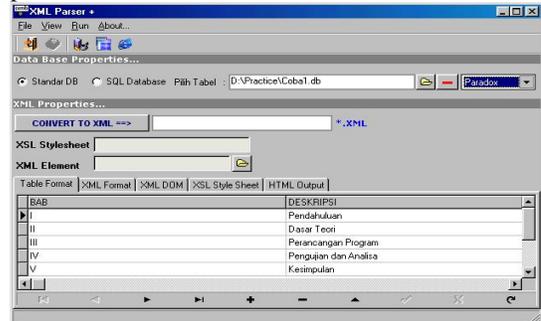
4. Pengujian dan Analisis

Proses pengujian terhadap program aplikasi untuk mengubah format *database* menjadi xml dilakukan dengan dua cara. Cara pertama adalah dengan melakukan klik ganda pada dokumen xml hasil konversi. Cara ini sering disebut juga memparser dokumen xml. Hasilnya akan terlihat pada *browser* Internet Explorer 6 dalam bentuk susunan yang terurut ke bawah (*tree*). Apabila setelah dilakukan klik ganda, dokumen xml hasil konversi dari format *database* dapat tampil di *browser* tanpa pesan kesalahan, menunjukkan

bahwa dokumen XML tersebut telah well-formed. Cara kedua adalah dengan mengubah dokumen xml ke dalam bentuk HTML, sehingga tampilan dari dokumen xml hasil konversi (*.xml) pada *browser* Internet Explorer 6 menjadi lebih jelas.

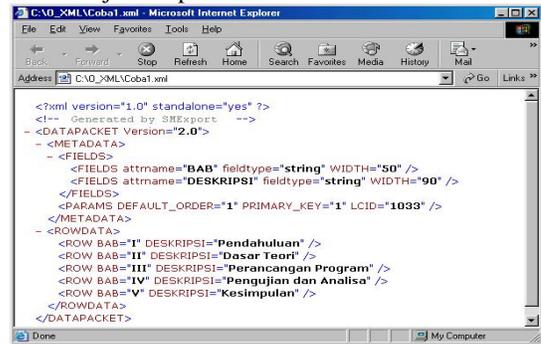
4.1 Pengujian Database format Paradox (*.db)

Untuk pengujian *database* dengan format Paradox (*.db), file yang digunakan sebagai contoh pengujian adalah file Coba1.db, seperti ditunjukkan pada Gambar 7.



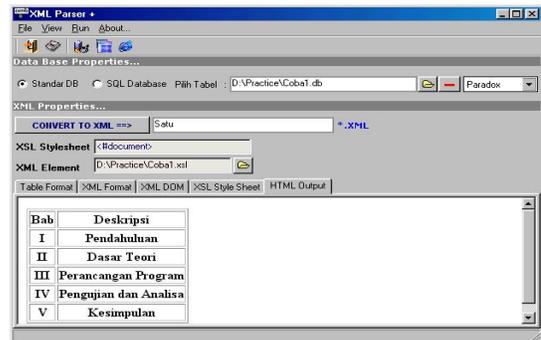
Gambar 7. File Coba1.db

Hasil konversi dari file Coba1.db ke dalam dokumen XML disimpan dengan nama Coba1.xml dan ditunjukkan pada Gambar 8.



Gambar 8. File Coba1.xml

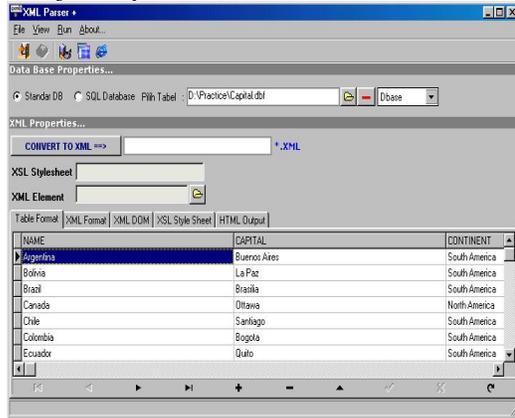
Tampilan dokumen XML (Coba1.xml) pada *web browser* seperti terlihat diatas menunjukkan bahwa dokumen tersebut telah *well-formed*. Dengan menambahkan XSL, dokumen XML tersebut dapat ditampilkan dalam HTML seperti ditunjukkan pada Gambar 9.



Gambar 9. Tampilan file Coba1.xml pada web browser

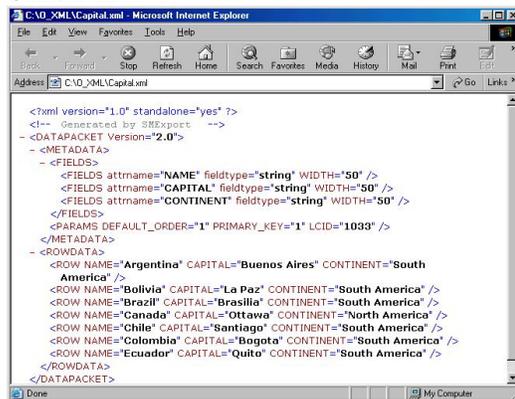
4.2 Pengujian Database format dbase (*.dbf)

Pengujian untuk database dengan format dbase (*.dbf) dilakukan seperti pada pengujian database dengan format Paradox. File yang digunakan adalah file Capital.dbf, seperti ditunjukkan pada Gambar 10.

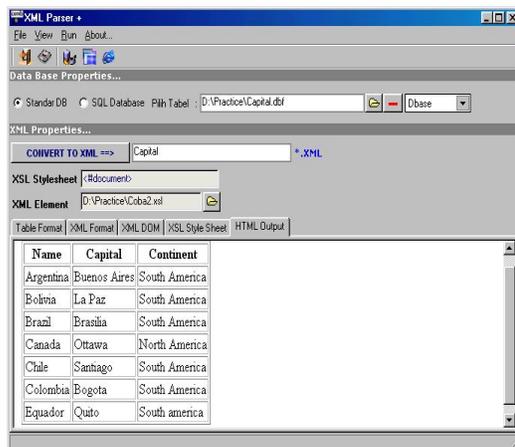


Gambar 10. File Capital.dbf

Hasil konversi dalam dokumen XML yang *well-formed* ditunjukkan pada Gambar 11, sedangkan tampilannya pada web browser ditunjukkan pada Gambar 12.



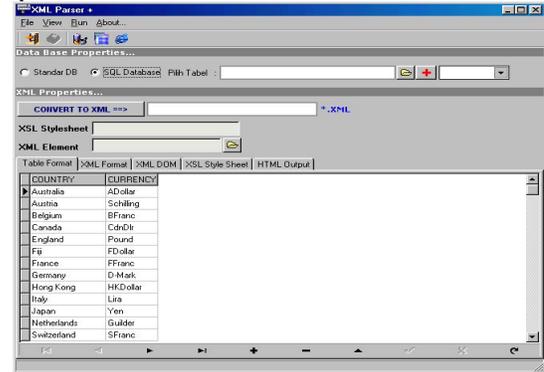
Gambar 11. File Capital.xml yang *well-formed*



Gambar 12. Tampilan file Capital.xml pada web browser

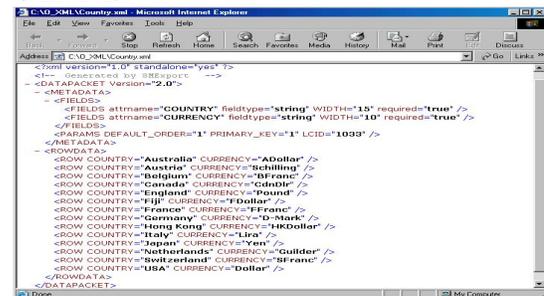
4.3 Pengujian Database server

Pada pengujian *database server*, file yang digunakan adalah Employee.gdb. File ini terdiri atas beberapa buah tabel yaitu *country*, *customer*, *department*, *employee*, *employee project*, *items*, *job*, *phone list*, *project*, *proj_dept_budget*, *salary history* dan *sales*. Untuk pengujian, dipilih satu buah tabel yaitu tabel *country* yang ditunjukkan pada Gambar 13.

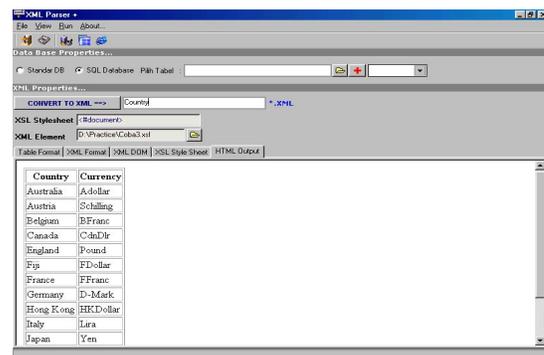


Gambar 13. Tampilan dari tabel country

Dokumen xml yang merupakan hasil konversi dari tabel *country* diatas disimpan dengan nama file Country.xml. Cara pengujian untuk mengetahui dokumen xml tersebut *well-formed* atau tidak, sama seperti pada pengujian dengan *database* tunggal yaitu dengan melakukan klik ganda pada dokumen yang bersangkutan. Untuk file Country.xml, hasilnya ditunjukkan pada Gambar 14 dan tampilan file tersebut pada web browser ditunjukkan pada Gambar 15.



Gambar 14. Tampilan file Country.xml yang *well-formed*



Gambar 15. Tampilan file Country.xml pada web browser

5. Kesimpulan

Dari pengujian dan analisa yang telah dilakukan, maka dapat disimpulkan beberapa hal berikut ini :

1. Program aplikasi untuk mengubah format *database* menjadi XML terbukti dapat digunakan sesuai dengan fungsinya.
2. Dokumen XML sebagai hasil konversi dari file *database* merupakan dokumen XML yang *well-formed*.
3. *Database* yang memiliki *field* dengan type data *binary large object* tidak dapat dikonversi menjadi dokumen XML yang *well-formed*.
4. Dokumen XML yang akan ditampilkan sebagai halaman *web* memerlukan XSL (*Extensible Stylesheet Language*).

6. Daftar Pustaka

1. Kadir. Abdul, "*Konsep dan Tuntunan Praktis Basis Data*", Yogyakarta, Andi Offset, 1999.
2. Post. Gerald V. "*Database Management Systems*", San Francisco, McGraw-Hill, 1990.
3. Pranata. Antony, "*Pemrograman Borland Delphi*", Yogyakarta, Andi, 2000.
4. Sidharta. Lani, "*Internet Informasi Bebas Hambatan 1*", Jakarta, Elex Media Komputindo, 1996.
5. Sidik. Betha dan Husni I. Pohan. "*HTML dan XML*", Bandung, Informatika, 2002.
6. Simpson. John E, "*Just XML*", Yogyakarta, Andi, 2002.
7. Wikanta. Prasaja, "*Mendesain Web dengan Pemrograman XML*", Jakarta, Elex Media Komputindo, 2001.
8. Young. Michael J, "*Microsoft XML Step by step*", Jakarta, Elex Media Komputindo, 2002.
9.XML parser, <http://www.xmlsoftware.com>
10.XML Validator, <http://msdn.microsoft.com/xml>.
11.XML ver 1.0, <http://www.w3c.org/TR/1998/REC-xml-19980210>

Mengetahui / Mengesahkan,

Pembimbing I

Pembimbing II

Ir. Kodrat I.S, MT
NIP. 132 046 696

Wahyudi, ST, MT
NIP. 132 086 662

| | |
|--|-----------------------------------------------------------------------------------------------------|
| | Darmawan. (L2F 097 619) Mahasiswa jurusan Teknik Elektro Universitas Diponegoro Semarang. |
|--|-----------------------------------------------------------------------------------------------------|