

PENERAPAN RELATIONAL DATA MENGGUNAKAN XQUERY PADA PEMROGRAMAN XML

Heri Sismoro¹ dan Ahmad Luthfi²

¹Dosen STMIK AMIKOM Yogyakarta

²Dosen Universitas Bina Darma Palembang.

Abstraksi

Saat ini hampir setiap jalur bisnis data dikirim menggunakan Relational Database System dan Structure Query Language (SQL). Dengan menggunakan kedua fungsi tersebut maka banyak hal yang akan dapat dilakukan seperti pemanggilan, dan manipulasi data. eXtensible Markup Language (XML) adalah salah satu pemenang dari sekian banyak nominator bahasa yang dapat melakukan semua hal tersebut diatas. Evolusi dan perkembangan XML tidak dapat di bendung lagi karena kebutuhan para vendor yang menuntut sebuah bahasa pemrograman tidak hanya sebatas dapat memanggil, dan memanipulasi data namun juga dapat melakukan fungsi-fungsi khusus seperti pencarian (search), integrasi antar proses, dan publikasi (publishing).

Kata Kunci: *XML, Xquery, Relational Data, Structure Query Language*

Pendahuluan

XML adalah himpunan aturan atau dapat juga diartikan sebagai panduan atau konvensi untuk mendesain format teks yang memungkinkan untuk membuat data menjadi terstruktur. XML bukanlah bahasa pemrograman, dan tidak perlu menjadi seorang programmer untuk belajar atau menggunakannya. XML

mempermudah komputer dalam menghasilkan data, membaca data, dan meyakinkan bahwa struktur data tidak bersifat ambigu. XML menghindari kekurangan-kekurangan yang umum terjadi dalam bahasa pemrograman: XML bersifat ekstensibel, tak tergantung pada suatu platform tertentu, serta mendukung internasionalisasi dan lokalisasi. XML sepenuhnya mendukung unicode.

Seperti halnya HTML, XML memanfaatkan penggunaan *tag* (kata-kata yang dikurung oleh '<' dan '>') dan *atribut* (dalam bentuk nama="nilai"). Bila HTML mendefinisikan arti dari tiap tag dan atribut, dan seringkali juga mendefinisikan bagaimana teks di antara tag-tag tersebut harus ditampilkan dalam sebuah browser, XML menggunakan tag hanya untuk membatasi sebuah data, dan menyerahkan sepenuhnya interpretasi data pada aplikasi yang membacanya. Dengan kata lain, jika menemukan "<p>" pada suatu file XML, jangan terburu-buru mengasumsikan bahwa itu adalah sebuah paragraf. Tergantung pada konteksnya, mungkin itu berarti parameter, pengarang, penyanyi, atau sesuatu yang mengharuskan artinya wajib mengandung huruf "p"?).

Program untuk membuat pengolah angka (*spreadsheet*), buku alamat, dan data terstruktur lainnya seringkali menyimpan data pada disk dengan format teks atau biner. Salah satu kelebihan penggunaan format teks adalah karena format ini memungkinkan seseorang, jika perlu, melihat isi data tanpa harus menggunakan program pembuat data tersebut; Anda dapat membacanya dengan editor teks kesukaan Anda. Format teks juga memungkinkan developer men-*debug* aplikasi dengan mudah. Seperti halnya HTML, file XML adalah file teks yang tidak dimaksudkan untuk dibaca begitu saja, tapi ia dapat dibaca jika memang dibutuhkan. Aturan-aturan untuk file XML jauh lebih ketat daripada HTML.

Jika sebuah tag lupa ditulis, atau jika sebuah atribut tidak menyertakan tanda petik, maka file tersebut tidak dapat digunakan, sedangkan pada HTML kesalahan-kesalahan semacam itu masih ditolerir dan bahkan

seringkali diperbolehkan secara eksplisit, atau paling tidak masih ditolerir. Spesifikasi resmi XML tidak melarang aplikasi untuk menebak maksud pembuat file XML yang salah itu; jika file itu mengandung kesalahan, aplikasi harus berhenti memprosesnya, serta menampilkan pesan kesalahan.

XML memperbolehkan mendefinisikan format dokumen baru dengan mengombinasikan dan menggunakan ulang format lain. Karena dua format yang dikembangkan secara independen memungkinkan elemen atau atribut mempunyai nama yang sama, maka harus berhati-hati saat mengombinasikan format-format tersebut (apakah "<p>" berarti "paragraf" dari format ini atau "pengarang" dari format itu?). Untuk menghilangkan kebingungan saat mengombinasikan beberapa format, XML menyediakan mekanisme *namespace*. XSL dan RDF adalah contoh yang baik dari format berbasis XML yang menggunakan namespaces. *XML Schema* didesain untuk menyediakan dukungan serupa untuk modularitas pada tingkat pendefinisian struktur dokumen XML, dengan mempermudah pengombinasian dua *schema* untuk menghasilkan *schema* ketiga yang melingkupi struktur dokumen gabungan.

Pembahasan

Evolusi Pemrograman XML

Perkembangan XML dimulai pada tahun 1996 dan menjadi standar W3C sejak Februari 1998, yang mungkin membuat orang berpikir bahwa teknologi ini masih belum begitu matang. Namun kenyataannya teknologi ini bukan merupakan barang baru. Sebelum XML sudah ada apa yang dinamakan SGML, yang dikembangkan pada awal tahun 80-an, menjadi standar ISO sejak 1986, dan digunakan secara luas untuk berbagai proyek dokumentasi berskala besar. Pengembangan HTML dimulai pada tahun 1990.

Desainer XML mengambil bagian terbaik dari SGML, dan didukung pengalaman dengan HTML, menghasilkan sesuatu yang

tidak kalah hebat dibandingkan SGML, namun jauh lebih umum dan lebih sederhana untuk digunakan. Kadang-kadang memang sulit untuk membedakan evolusi dan revolusi. Sebagai tambahan, SGML lebih banyak digunakan untuk dokumentasi teknis dan jarang digunakan untuk data jenis lain, sementara XML sebaliknya.

Dari perspektif bisnis, hampir semua tipe data dapat dipresentasikan sebagai XML, dengan suatu grammar yang mendeskripsikan strukturnya. XML memungkinkan pemisahan arti semantik dari bagaimana penampilannya pada layar atau perangkat keluaran. Ini sebagai analogi pemisahan model/view pada perancangan berorientasi objek. Sebagai contoh dari pemisahan model/view adalah komponen Java Swing.

Selanjutnya perkembangan XML digunakan pada beberapa lapangan bisnis seperti:

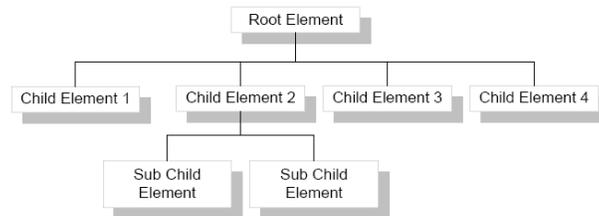
1. e-Commerce, dimana peranan XML dapat secara spesifik menyatakan pemesanan, pembelian, transaksi, inventori, dan billing.
2. Searching pada platform independence, jika data web dikodekan ke dalam XML, dengan mendeskripsikan makna dari isi sebenarnya, seseorang dapat menemukan layanan sumber daya yang diinginkan.
3. Kemampuan untuk mengakses data, dokumen XML bisa dikodekan ke tipe data informasi dan relasinya yang terdapat di database. Informasi database dapat di akses sebagai XML lewat web.
4. Penyederhanaan aplikasi, sifat XML yang terbuka dan eXtensible memungkinkan kita untuk mempresentasikan semua fungsi format data yang berbeda dalam satu XML yang spesifik dengan domain grammar tertentu.
5. Sharing data via internet, XML memiliki kemampuan untuk membuka akses dari end-user ke data, dan kemampuan untuk melakukan pilihan-pilihan yang cerdas, dengan demikian XML dapat lebih mempercepat koneksi antara bisnis dengan konsumen.

XML menyediakan sintaks yang tidak bersifat ambigu untuk RDF, bahasa untuk mengekspresikan metadata (pada kenyataannya, untuk pengetahuan secara umum). RDF seperti *hypertext* yang telah ditingkatkan sampai tahap berikutnya. Bila *hypertext* menghubungkan sepotong teks dan membiarkan relasi diantaranya tak jelas, RDF dapat menghubungkan apa saja dan memberikan nama untuk relasi tersebut: "A adalah harga dari B" mungkin merupakan relasi antara sebuah obyek dengan sejumlah uang; "A lebih berat daripada B" mungkin merupakan relasi antara dua pegulat sumo; "A adalah akibat dari B" mungkin merupakan relasi antara mandi dan menjadi basah.

Untuk mengkomunikasikan pengetahuan, baik dalam XML/RDF maupun dalam Bahasa Indonesia, manusia dan mesin harus menyetujui kata-kata apa yang digunakan. Himpunan kata-kata yang didefinisikan secara tepat untuk mendeskripsikan daerah kehidupan tertentu (dari "belanja" sampai "logika matematika") dinamakan "ontologi." RDF, ontologi, dan representasi arti sehingga komputer dapat membantu pekerjaan manusia, semuanya merupakan topik dalam Semantic Web Activity.

Relational Data dan XML

Penggunaan dan transmisi pada data dalam sebuah sistem membutuhkan representasi pada nilai data tersebut dan semantic atas keterhubungan nilai-nilai tersebut.



Gambar 1. Diagram Hirarki XML

Pada gambar 1 dapat dilihat bagaimana struktur atau diagram XML, yang menganut sistem hirarki dimana root (induk) elemen memiliki anak-anak elemen dan sub-sub elemen lainnya.

Berikut ini adalah perbandingan karakteristik data pada XML dan relational.

Tabel 1. Karakteristik XML dan Relational

<i>Data Characteristics</i>	<i>XML</i>	<i>Relational</i>
Flat Structured Data	✓	✓
Hierarchical Structured Data	✓	Not First Class: PK-FK with cascading delete
Semi-structured Data	✓	Not First Class
Mark-up Data	✓	Not First Class: FTS
Order preservation	✓	Not First Class
Recursion	✓	✓ (Recursive query)

Pada karakteristik data hierarchical structured data, untuk XML mempunyai status First class sedangkan pada relational, hirarki data bukanlah yang first class. Begitupun juga dengan untuk semi-structured data, markup data, order preservation, dan recursion untuk relational memiliki karakteristik bukan first class. Selanjutnya adalah perbandingan skenario pada XML dan relational.

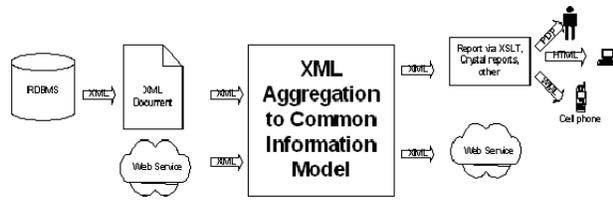
Tabel 2. Skenario XML dan Relational

<i>Scenarios</i>	<i>XML</i>	<i>Relational</i>
Relational Data Exchange	Use as transport, shred to relational	Storage and Query
Document Management	Use as markup, store natively	Provides framework to manage collections and relationships; provides Full-text search
Semi-structured Data	Represent semi-structured parts	Represent structured parts
Message audit	Store natively	Used for querying over promoted properties
Object serialization	Store natively	Used for querying over promoted properties

Pada skenario XML dan Relational, dapat dilihat bahwa beberapa perbedaan dalam hal Relational Data Exchange, Document Management, Semi-Structured Data, Message Audit, dan Object serialization.

XQuery pada Relational Data

Pada prinsipnya, XQuery pada relational data adalah sebuah bahasa ekspresi fungsional yang dapat digunakan untuk query atau proses XML data dan banyak data yang dapat di presentasikan pada model data yang sama sebagai XML. Ketika penggunaan ekspresi sebuah bahasa, program XQuery lebih mudah untuk dipahami dan di kontrol daripada XSLT, karena memiliki kompleksitas aturan sistem (rule-based-system).



Gambar 2. Using XML for Data Aggregation and Reporting

Pada Gambar 2 dapat dilihat bahwa ada dua XML input yaitu RDBMS dan Web Service. Untuk RDBMS, database digunakan sebagai format otomatis dari data ke XML.

```

<Report2003>
<ID>347</ID>
<Company>Progress Software</Company>
<Symbol>PRGS</Symbol>
<_x0035_2_x0020_Week_x0020_Change>0.036</_x0035_2_x0020_Week_x0020_Change>
<MC>588</MC>
<MC-Cash>419.2</MC-Cash>
<MC_x002F_Rev>2.09625668449198</MC_x002F_Rev>
<TR>1.49447415329768</TR>
<Cash>168.8</Cash>
<Rev>280.5</Rev>
<EPS>0.6</EPS>
<P_x002F_E>29.4</P_x002F_E>
<Price>17.61</Price>
<Shares>33.4</Shares>
<Employees>1291</Employees>
<Rev_x0020_000s>217.27343144849</Rev_x0020_000s>
<City>Bedford</City>
</Report2003>
  
```

Gambar 3. Historical Stock Data Retrieved From Access database

Penutup

Seiring dengan perkembangannya, XML dapat menyelesaikan permasalahan teknis dan strategis pada pemrograman aplikasi. Dengan menggabungkan unsur bahasa XQuery dan Relational Data, maka lengkaplah sudah kesempurnaan dari sebuah XML. Suatu format data terbuka seperti XML memungkinkan aplikasi berdasar pada kumpulan feature, dibandingkan pada format yang rahasia dan proprietary.

Daftar Pustaka

- Anders Møller, *The XML Revolution*,
www.brics.dk/NS/01/8/BRICS-NS-01-8.pdf, 2003.
- Daniela Florescu, *An XML Programming Language for Web Service Specification and Composition*,
www.w3.org/XML/2001/07/xmlpm-florescu.pdf, 2003
- J.E. Funderburk, *XML Programming with SQL and Query*,
www.research.ibm.com/journal/sj/414/reinwald.pdf, 2001.
- Jonathan Robie, *SQL/XML, XQuery, and Native XML Programming Languages*, www.idealliance.org/papers/dx_xmle04/papers/02-04-02/02-04-02.pdf, 2002