PENGUKURAN KINERJA BEBERAPA SISTEM BASIS DATA RELASIONAL DENGAN KEMAMPUAN MENYIMPAN DATA BERFORMAT GML (*GEOGRAPHY MARKUP LANGUAGE*) YANG DAPAT DIGUNAKAN UNTUK MENDASARI APLIKASI-APLIKASI SISTEM INFORMASI GEOGRAFIS

Adi Nugroho[1], Sri Hartati[2]
[1] Fakultas Teknologi Informasi – Universitas Kristen Satya Wacana Salatiga – Jawa Tengah, dan Program S3 Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam - Universitas Gadjah Mada Jogyakarta.
[2] Program S3 Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam - Universitas Gadjah Mada Jogyakarta.
E-mail: adi.nugroho@staff.uksw.edu, shartati@ugm.ac.id

ABSTRAK: Dalam menampilkan data yang bersifat spasial ke arah para pengguna, Sistem-sistem Informasi Geografis yang ada saat ini memiliki 2 pilihan berkaitan dengan sistem basis data yang mendasarinya, yaitu menggunakan sistem basis data relasional yang bersifat umum, yang dapat digunakan untuk menyimpan data spasial yang memiliki tipe data yang sifatnya umum (*number, char, varchar*, dan sebagainya), atau menyimpan data terkait dalam bentuk GML (*Geography Markup Language*), yang pada dasarnya merupakan suatu kosa kata XML (*eXtensible Mark Language*) yang bersifat khas untuk data spasial. Dalam hal ini, jika Sistem Informasi Geografis yang bersangkutan memilih menggunakan GML, sesungguhnya ada 2 pilihan lagi yang bisa diambil berkaitan dengan hal ini, yaitumenyimpan data berformat GML tersebut menggunakan sistem basis data relasional biasa yang memiliki kemampuan untuk menyimpan data berformat XML (*XML Enabled Database*)atau menggunakan sistem basis data yang memang dirancang untuk menyimpan data berformat XML (*Native XML Database*) (*NXD*). Dengan alasan bahwa *XML Enabled Database* lebih bersifat umum di Indonesia, tulisan ini bermaksud membandingkan kinerja berbagai *XML Enabled Database* ditinjau dari sudutpandang pengembang aplikasi Sistem Informasi Geografis. Kinerja yang dimaksudkan adalah efisiensi terukur dalam satuan waktu dari beberapa*XML Enabled Database* dalam menerima operasi-operasi basis data yang sifatnya umum, yang biasa disebut dengan operasi-operasi CRUD (*Create-Read/Retrieve-Update-Delete*),untuk data yang berformat GML. Selain itu, tulisan ini juga mencoba membahas fleksibilitas *XML Enabled Database* terpilih dari sudutpandang pemrogramannya.

Kata kunci: Sistem basis data XML, XML *enabled database,* GML.

ABSTRACT: *If we want to represent spatial data to user using GIS (Geographical Information System) applications, we have 2 choices about the underlying database, that is general RDBMS (Relational Database Management System) for saving general spatial data (number, char, varchar, etc.), or saving spatial data in GML (Geography Markup Language) format. (GML is an another XML's special vocabulary for spatial data.) If we choose GML for saving spatial data, we also have 2 choices, that is saving spatial data in XML Enabled Database (relational databases that can be use for saving XML data) or we can use Native XML Database (NXD), that is special databases that can be use for saving XML data. In this paper, we try to make performance comparison for several XML Enabled Database when we do GML's CRUD (Create-Read-Update-Delete) operations to these databases. On the other side, we also want to see flexibility of XML Enabled Database from programmers view.*

Keywords*: Database Systems XML, XML Enabled Database, GML.*

PENDAHULUAN

Sistem Informasi Geografis (SIG), atau yang dalam bahasa aslinya disebut sebagai GIS (*Geographical Information System*), sesungguhnya merupakan sistem yang mencakup di dalamnya perangkat keras dan perangkat lunak pemetaan, aplikasi-aplikasi pencitraan jarak jauh (*remote sensing*), pemetaan tanah (*land surveying*), pemotretan udara (*aerial photography*), model-model matematika, model-model geografi, dan sebagainya, dan kakas-kakas (*tool*) yang dapat diimplementasikan secara terintegrasi di dalamnya [1, 2]. Secara lebih umum, aplikasi-aplikasi SIG pada dasarnya merupakan kakas yang memungkinkan pengguna untuk membuat *query* interaktif, melakukan analisis informasi-informasi yang berkaitan dengan keruangan/ spasial, melakukan penyuntingan data dan peta, kemudian menampilkan hasil dari semua operasi itu ke hadapan pengguna yang bersangkutan melalui

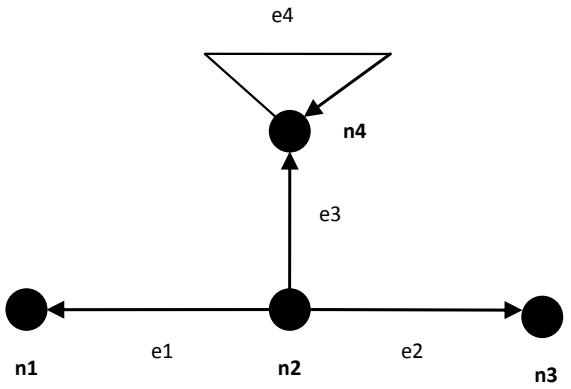
layar monitor komputer atau menampilkannya dalam bentuk suatu peta tercetak [1, 2].

Ada berbagai teknologi representasi dalam SIG modern, yaitu vektor, raster, serta TIN (*Triangular Irregular Network*), dan semuanya melibatkan data yang tersimpan dalam sistem basis data relasional (RDBMS-*Relational Database Management System*) [3]. Sistem basis data relasional pada umumnya memiliki tipe-tipe data tertentu, misalnya *number, char, varchar*, dan sebagainya, yang implementasinya secara internal sangat bergantung pada masing-masing *vendor* [4, 5]. Hal ini tentunya dapat digunakan secara cukup memadai jika SIG yang bersangkutan bersifat mandiri. Meski demikian, saat SIG yang bersangkutan perlu diintegrasikan dengan SIG-SIG yang lainnya, hal ini suatu saat mungkin akan menimbulkan kesulitan [6, 7], sebab mungkin saja SIG-SIG yang lainnya itu menggunakan sistem basis data relasional yang berbeda-beda, yang memiliki tipe-tipe data yang representasi internalnya mungkin berbeda-beda pula. Lalu bagaimana solusinya?

Beruntung sekali bahwa saat ini industri Teknologi Informasi sudah memiliki suatu format data yang bersifat netral terhadap *platform*, baik terhadap *platform* perangkat keras maupun *platform* perangkat lunak. Format data yang dimaksud adalah XML (*eXtensible Markup Language*) [6, 7, 8]. Sesungguhnya ada berbagai varian dari XML, yang masing-masing dirancang untuk kebutuhan-kebutuhan khusus, misalnya SOAP (*Simple Object Access Protocol*) dan WSDL (*Web Service Description Language*) untuk aplikasi-aplikasi *Web Service* [9], MusicXML yang digunakan sebagai sarana pertukaran data antarperangkat musik, SVG (*Scalable Vector Graphics*), dan banyak lagi. Sebagai salah satu contoh aplikasi SIG yang menggunakan berkas XML sebagai penyimpanan datanya adalah aplikasi-aplikasi SIG yang dikembangkan menggunakan teknologi Silverlight XAML (*eXtensible Application Markup Language*) dari Microsoft Corp. seperti yang dapat kita lihat di http://cadmaps. com/gisblog/ [8]. Secara khusus, saat ini beberapa SIG yang ada (sebagian besar masih dalam bentuk purwarupa [*prototype*] SIG berbasis Web yang bersifat riset untuk penggunaan masa depan [lihat http://www.iis.sinica.edu.tw] [10]) mengadopsi GML (*Geography Markup Language*) (yang pada dasarnya merupakan suatu kosa kata XML untuk data spasial) sebagai sarana pertukaran data di antara mereka [3]. Menurut dokumentasi-dokumentasi yang dikeluarkan oleh OGC (*Open Geography Consortium*) (2005) (perhatikan www.opengeospatial.org/specs/), GML pada dasarnya merupakan suatu kosa kata XML yang ditulis berdasarkan XML Schema dari W3C (*World Wide Web Consortium*), yang dapat digunakanuntuk memodelkan, mentransportasikan, serta menyimpan informasi-informasi yang bersifat spasial/keruangan [11, 10].

GML (*GEOGRAPHY MARKUP LANGUAGE*)

GML (*Geography Markup Language*), seperti telah disebutkan di atas, merupakan sebuah kosa kata XML yang ditulis berdasarkan XML Schema dari W3C (*World Wide Web Consortium*) yang dapat digunakan untuk memodelkan, mentransportasikan, serta menyimpan informasi-informasi yang bersifat geografis [11, 10, 12]. Konsep kunci yang digunakan oleh GML untuk memodelkan dunia nyata adalah OGC Abstract Spesification (perhatikan: www. opengeospatial.org/specs/) [13], dimana spesifikasi ini secara umum menyediakan berbagai jenis objek yang dapat digunakan untuk mendeskripsikan data geografi, termasuk di dalamnya data sistem rujukan kordinat, geometri, topologi, waktu, unit pengukuran, dan sebagainya [13]. GML juga dapat digunakan untuk mendeskripsikan fitur-fitur geografis seperti tutupan (*coverage*) yang bermanfaat untuk memodelkan relasi-relasi spasial antar berbagai fenomena yang memang dijumpai di dunia nyata [10, 13]. Selain itu, GML juga dapat digunakan untuk memodelkan data hasil observasi-observasi yang didapatkan secara langsung di lapangan, seperti data yang diperoleh melalui penggunaan kamera, citra satelit, dan sebagainya [10, 13]. Secara umum, spesifikasi GML bersifat terbuka, lintas-*platform*, sehingga sangat layak untuk digunakan sebagai sarana pertukaran data melintas berbagai SIG yang beragam jumlahnya di dunia ini [13].



Gambar 1. Jaringan Sederhana

Untuk memperlihatkan bagaimana GML dapat digunakan untuk mendeskripsikan data spasial/geografis, kita akan melihat contoh penerapannya berikut ini. Gambar 1 di atas merepresentasikan rekonstruksi simpul (*node*) dan garis (*edge*) yang, sebagai contoh saja, secara sederhana menggambarkan rute pemberhentian bus di suatu jaringan jalan raya. Simpul-

simpul, demi kemudahan, dianggap tidak memiliki struktur internal di dalamnya. Sementara itu, garis-garis memiliki simpul awal (yang memiliki nilai negatif) dan simpul akhir (yang memiliki nilai positif) yang dideskripsikan sebagai pasangan properti-properti simpul berarah, yang selanjutnya akan membentuk batasan garis yang bersangkutan. Dengan data di atas, kita bisa memberikan pertanyaan-pertanyaan yang berkaitan dengan "bagaimana urutan garis-garis yang diperlukan untuk melakukan pergerakan dari suatu simpul ke simpul yang lainnya?" Sebagai contoh, pergerakan dari n1 ke n3 akan memiliki urutan {+e1, +e2}, dan dari n3 ke n4 memiliki urutan {-e2, +e3}. Dalam hal ini, perhatikanlah bahwa kita dapat merujuk garis-garis dalam 2 arah. Karena suatu garis pada dasarnya memiliki arah yang bersifat implisit (sebagai contoh, e1 dapat dikatakan berarah dari n1 ke n2), garis itu dapat dilintasi baik pada arah itu maupun pada arah sebaliknya. Dengan demikian, -e1 dapat dikatakan bergerak dari n2 ke n1.

Jika kita melihat jaringan sederhana (*simple network*) seperti yang diperlihatkan Gambar 1, maka kita dapat merepresentasikannya dalam bentuk simpleNetwork.xml menggunakan spesifikasi yang dikenal oleh GML [13].

```
<?xml version="1.0" encoding="UTF-8"?>
<Topology
xmlns="http://www.opengis.net/app"
xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gml="http://www.opengis.net/gml"
xmlns:app="http://www.opengis.net/app"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/appnetworkExamples.xsd">
<gml:Node gml:id="n1"/>
<gml:Node gml:id="n2"/>
<gml:Node gml:id="n3"/>
<gml:Node gml:id="n4"/>
<gml:Edge gml:id="e1">
<gml:directedNode orientation="-"
xlink:href="#n1"/>
<gml:directedNode orientation="+"
xlink:href="#n2"/>
</gml:Edge>
<gml:Edge gml:id="e2">
<gml:directedNode orientation="-"
xlink:href="#n2"/>
<gml:directedNode orientation="+"
xlink:href="#n3"/>
</gml:Edge>
<gml:Edge gml:id="e3">
<gml:directedNode orientation="-"
xlink:href="#n2"/>
<gml:directedNode orientation="+"
xlink:href="#n4"/>
</gml:Edge>
<gml:Edge gml:id="e4">
<gml:directedNode orientation="-"
xlink:href="#n4"/>
<gml:directedNode orientation="+"
xlink:href="#n4"/>
</gml:Edge>
</Topology>
```

Berkas GML di atas adalah representasi dalam bentuk XML jaringan sederhana (*simple network*) seperti yang diperlihatkan dalam Gambar 1. Dokumentasi spesifikasi GML yang diluncurkan oleh OGC tentunya mendeskripsikan sintak-sintak untuk objek-objek yang jauh lebih kompleks dari yang diperlihatkan dalam jaringan sederhana di atas. Selain itu, simpul-simpul (*node*) yang ada mungkin saja dikembangkan lebih lanjut menjadi gambaran objek-objek kompleks yang sesungguhnya (misalnya stasiun pemberhentian bus yang memiliki banyak objek lain di dalamnya), yang menggambarkan fakta yang ada di dunia nyata; garis-garis (*edge*) mungkin juga dikembangkan lebih lanjut menjadi garis-garis majemuk yang juga menggambarkan suatu peta jalan sesungguhnya yang memang dijumpai di dunia nyata. Dalam hal ini, berkas GML sederhana di atas akan digunakan untuk membuat simulasi operasi-operasi CRUD (*Create-Read/Retrieve-Update-Delete*) atas berbagai sistem basis data XML (*XML Enabled Database*) seperti yang akan kita lihat di bagian selanjutnya tulisan ini.

SISTEM BASIS DATA XML (*XML ENABLED DATABASE*)

Sistem basis data XML pada dasarnya merupakan perangkat lunak penyimpan data yang memungkinkan data disimpan dalam format XML [14, 15]. Data dalam bentuk XML ini kemudian dapat di-*query*, diekspor dan selanjutnya dapat diserialisasi menjadi format yang diperlukan untuk menampilkannya ke arah para pengguna (misalnya menjadi format HTML [*Hypertext Markup Language*] untuk ditampilkan di perambah [*browser*] atau menjadi format WML [*Wireless Markup Language*] untuk ditampilkan di sarana-sarana bergerak [*mobile*]). Secara prinsip, sistem basis data XML ini sendiri dibagi dapat menjadi 2 jenis (meskipun sesungguhnya batasannya tidak terlalu jelas), yaitu:

1. Sistem basis data relasional yang mampu menyimpan dan mengeluarkan data berformat XML (*XML Enabled Database*), meski demikian tidak melakukan konversi-konversi apapun padanya [14, 15]. Konversi-konversi yang diperlukan

(misalnya untuk membentuk data berformat HTML atau WML) dilakukan oleh program aplikasi yang mengaksesnya. Contoh dari *XML Enabled Database* ini adalah Oracle, Microsoft SQL Server, MySQL, dan sebagainya.

2. Sistem basis data XML (*Native XML Database*) (NXD) yang mampu menyimpan data berformat XML dan menggunakan XML sebagai unit paling dasar penyimpanan data dan tidak menyimpannya dalam bentuk berkas-berkas teks, sehingga konversi-konversi yang diperlukan mungkin saja dilakukan di dalam sistem basis data yang bersangkutan [10, 14, 15, 16]. Beberapa contoh dari *Native XML Database* adalah BaseX, eXist, MonetDB/ Xquery, Sedna, Tamino, TeXtML, EMC xDB (ex-XHive), Qizx (XMLmind), dan sebagainya (beberapa di antaranya bersifat *opensource*).

Dalam tulisan ini, kita tidak akan membahas *Native XML Database*, alih-alih akan membahas kinerja (*performance*) beberapa sistem basis data relasional yang mampu menyimpan data berformat XML (*XML Enabled Database*) yang paling populer di Indonesia [16], yaitu Oracle, Microsoft SQL Server, serta MySQL.

Dalam hal di atas, beberapa sistem basis data XML mendukung penggunaan suatu API (*Application Programming Interface*) yang dinamakan sebagai XML: DB API (atau XAPI) sebagai suatu bentuk akses yang bersifat mandiri terhadap penyimpanan data yang berformat XML [14, 15]. Pada sistem-sistem basis data XML, XAPI sesungguhnya bertindak dengan cara yang serupa dengan ODBC (*Open Database Connectivity*) atau JDBC (*Java Database Connectivity*) saat kedua API ini digunakan untuk mengakses sistem basis data relasional [14, 15]. Pada tahun 2009, JCP (*Java Community Process*) meluncurkan sebuah versi final dari XSJ (*XQuery API for Java Spesification*) yang pada dasarnya merupakan suatu API yang memungkinkan aplikasi-aplikasi mengirimkan *query* yang sesuai dengan spesifikasi X Query 1.0 dari W3C (*World Wide Web Consortium*) untuk berkomunikasi dengan data yang berformat XML [14, 15]. Dalam hal ini, kita -seperti layaknya aplikasi-aplikasi SIG pada umumnya akan mencoba melakukan pengukuran kinerja dari arah aplikasi-aplikasi yang dikembangkan menggunakan bahasa pemrograman Java dengan pertolongan XAPI yang telah diadaptasi mulai dari JDBC (*Java Database Connectivity*) versi 3.0.

PENGUKURAN KINERJA SISTEM BASIS DATA XML

Untuk melakukan pengukuran kinerja operasi-operasi CRUD atas data XML (tepatnya GML) yang tersimpan di dalam beberapa sistem basis data XML (*XML Enabled Database*) yang dipilih (Oracle, Microsoft SQL Server, serta MySQL), pertama kali kita harus mengetahui tipe-tipe data dalam sistem-sistem basis data tersebut, yang dapat digunakan untuk menyimpan data berformat XML/GML [4, 2]. Dalam hal ini, sistem basis data Oracle 10g Express Edition yang digunakan dalam penelitian memiliki tipe data XML Type (mulai diperkenalkan pada Oracle 9i release 1) [14, 15], Microsoft SQL Server 2008 Express Edition memiliki tipe data xml [17], sementara MySQL 5.0 Community Edition menggunakan tipe data yang sifatnya relatif masih konvensional, yaitu text, varchar atau (jika ukuran berkas XML yang digunakan sangat besar) bisa juga menggunakan tipe data CLOB (*Character Large Object*) [4]. Selanjutnya untuk melakukan operasi-operasi CRUD terhadap masing-masing sistem basis data XML, kita bisa mengandalkan prosedur-prosedur tersimpan (*stored procedure*) dan fungsi-fungsi yang sifatnya khas untuk masing-masing sistem basis data XML (*XML Enabled Database*) yang dipilih. Masing-masing prosedur tersimpan dan fungsi diperlihatkan dalam Tabel 1.

Tabel 1. Prosedur-prosedur Tersimpan untuk Operasi-operasi CRUD Terhadap Berkas—berkas XML [3, 6, 8, 10]

SISTEM BASIS DATA	Operasi CREATE	Operasi LOAD	Operasi READ (SELECT)	Operasi UPDATE	Operasi DELETE
Oracle	CREATE TABLE … OF XMLType	DBMS_XDB. CREATERESOURCE()	XMLQUERY() GETCLOBVAL() EXISTNODE() EXTRACT() EXTRACTVALUE()	UPDATEXML()	
Microsoft SQL Server	SP_XML_ PREPAREDOCUMENT()	COPY	OPENXML() FOR XML RAW FOR XML AUTO FOR XML EXPLICIT FOR XML PATH QUERY() VALUE() EXIST() NODES()	MODIFY()	
MySQL	CREATE TABLE …	LOAD DATA	EXTRACTVALUE()	UPDATEXML()	

Catatan:

- Bagian-bagian yang kosong menunjukkan bahwa sistem basis data yang bersangkutan tidak menyediakan prosedur tersimpan untuk operasi yang bersangkutan. Kita bisa melakukannya menggunakan perintah-perintah SQL (*Structured Query Language*) baku yang umum digunakan (SQL-1999). Selain itu, beberapa sistem basis data seperti Oracle [14, 15] dan Microsoft SQL Server mengerjakan operasi-operasi penghapusan (*delete*) menggunakan operasi-operasi pembaruan (*update*) yang dimilikinya.
- Prosedur-prosedur tersimpan serta fungsi-fungsi yang terdaftar di atas adalah prosedur-prosedur tersimpan serta fungsi-fungsi yang terkait langsung dengan operasi CRUD terhadap berkas-berkas data berformat XML. Sesungguhnya spesifikasi XQuery mencakup juga di dalamnya prosedur-prosedur tersimpan serta fungsi-fungsi lain yang mampu melakukan ekstraksi dan mampu juga melakukan manipulasi sumber-sumber data lain yang dapat dilihat sebagai berkas XML (misalnya basis data relasional atau dokumen-dokumen lainnya, seperti lembar kerja [*spreadsheet*], serta berbagai format data lainnya) [18]. Sebagai contoh, sistem basis data Oracle 10g memiliki prosedur-prosedur tersimpan dan fungsi-fungsi yang mampu mengubah/mengkonversi tabel-tabel yang tersimpan di dalam basis data menjadi dokumen-dokumen yang berformat XML [19]. Untuk hal-hal yang terakhir ini, kita tidak memasukkannya ke dalam Tabel 1.

Selanjutnya, seperti telah kita singgung sebelumnya, kita akan melakukan pengukuran kinerja masing-masing sistem basis data XML (*XML Enabled Database*) melalui aplikasi sederhana yang dibuat menggunakan bahasa pemrograman Java. Untuk mengakses masing-masing sistem basis data dari arah aplikasi Java, kita bisa menggunakan JDBC-API (*Java Database Connectivity–Application Programming Interface*) sebab pada versinya yang terakhir (JDBC 4.0), API ini sudah mampu bekerja dengan data berformat XML/GML. Dalam hal akses ke sistem basis data XML (*XML Enabled Database*)ini, sesungguhnya ada 7 langkah dasar dalam menggunakan JDBC-API untuk mengakses basis data (*database*). Ketujuh langkah dasar itu dapat didaftarkan sebagai berikut [20].

1. Mengimpor paket java.sql.
2. Memuat dan mendaftarkan *driver* JDBC sesuai dengan sistem basis data yang akan diakses.
3. Menetapkan koneksi (*connection*) ke server basis data (*database server*).
4. Membuat pernyataan-pernyataan (*statement*) SQL (atau prosedur tersimpan).
5. Mengeksekusi pernyataan-pernyataan (*statement*) SQL (atau prosedur tersimpan).
6. Menerima hasilnya dari *resultset* (suatu kumpulan *record* di sisi komputer pengguna) kemudian melakukan pemrosesan-pemrosesan tertentu sesuai dengan apa yang diharapkan oleh para pengguna aplikasi.
7. Menutup pernyataan dan menutup hubungan (*connection*).

Itulah langkah-langkah dasar yang harus kita lakukan untuk dapat membuat aplikasi-aplikasi yang dapat mengakses sistem basis data XML. Selanjutnya, untuk mengetahui efisiensi (dalam waktu) masing-masing operasi CRUD, maka sebelum dan sesudah masing-masing operasi CRUD dilaksanakan, dalam aplikasi Java yang saya buat, saya juga harus menambahkan beberapa objek pewaktuan (objek Timer) yang berfungsi seperti layaknya *stopwatch* untuk masing-masing operasi CRUD atas sistem-sistem basis data XML (*XML Enabled Database*) yang dipilih.

Hasil pengukuran (dalam mikrodetik) untuk operasi-operasi CRUD atas berkas-berkas GML di dalam *XML Enabled Database* yang dipilih diperlihatkan dalam Tabel 2. Hasil pengukuran di atas didapatkan melalui percobaan pada komputer dengan prosesor Intel Core 2 Duo 2,8 GHz dengan memori 2 GB. Angka-angka yang ada di bagian atas-tengah tabel (1, 25, 50, 100) merupakan ukuran berkas GML yang digunakan dalam percobaan (dalam satuan KB). Selanjutnya, hasil-hasil pengukuran pada Tabel 2 di atas merupakan hasil rata-rata dari 5 percobaan untuk tiap operasi CRUD dengan ukuran berkas XML/GML yang sama dengan mengabaikan waktu yang diperlukan oleh aplikasi Java untuk melakukan koneksi terhadap sistem basis data XML yang dipilih.

Tabel 2. Perhitungan Waktu (Dalam Mikrodetik) Operasi-operasi CRUD Berkas-berkas XML

SISTEM BASIS DATA	Operasi INSERT				Operasi READ (SELECT)				Operasi UPDATE				Operasi DELETE			
	1	25	50	100	1	25	50	100	1	25	50	100	1	25	50	100
Oracle	1	4	8	16	1	2	3	4	1	2	4	5	1	1	2	3
Microsoft SQL Server	0	1.5	2	2.5	0	2	4.8	5.2	0.2	1.8	2	6	0	1	1	2.5
MySQL	0.3	2.2	43.7	85.9	2	1.6	3	5	2	3.1	3.1	4.7	1	1.6	1.1	1.6

ANALISIS

Data pengukuran pada Tabel 2 di atas menunjukkan Microsoft SQL Server secara umum (untuk operasi-operasi CRUD atas berkas-berkas XML/GML) memiliki kinerja yang terbaik, sementara MySQL memiliki kinerja yang paling tidak baik. Tentu saja istilah 'terbaik' ini bersifat relatif sebab hanya didasarkan pengukuran terhadap waktu semata. Jika kita mempertimbangkan kestabilan, maka kita dapat melihat bahwa Oracle mungkin merupakan sistem basis data yang paling stabil, sebab kita melihat kenaikan waktunya yang relatif bersifat linier terhadap ukuran berkas XML/GML yang diakses, dimana hal ini membuat pemrogram-pemrogram aplikasi dapat lebih mudah meramalkan kinerja aplikasi dalam berubahnya ukuran berkas XML/GML yang diakses. Selain itu, Oracle juga (dibandingkan dengan Microsoft SQL Server dan MySQL) memiliki kemampuan untuk menyimpan dan menangani jumlah data yang relatif lebih banyak, sehingga sangat sesuai untuk aplikasi-aplikasi Sistem Informasi Geografis (SIG) yang menggunakan data yang berukuran sangat besar dan berjumlah sangat banyak.

Dalam hal di atas, sistem basis data Oracle dan Microsoft SQL Server juga memiliki tipe data khusus untuk menangani data berformat XML/GML sehingga kita dapat memperkirakan bahwa kedua sistem basis data ini memiliki algoritma-algoritma yang telah dioptimasi untuk menangani data berformat XML/GML. Dalam hal ini, MySQL tidak memiliki tipe data khusus untuk menangani data XML/GML (MySQL memperlakukan berkas data XML/GML sebagai teks), sehingga (terbukti dari hasil pengukuran) kinerja MySQL tidak terlalu baik untuk aplikasi-aplikasi berbasis XML/GML (khususnya untuk aplikasi-aplikasi SIG) yang dikembangkan di atasnya (terutama untuk operasi-operasi penyisipan data [*insert*]). Selain itu, sistem basis data MySQL juga tidak memperlihatkan kinerja operasi CRUD yang konstan, yang dapat diramalkan, sepanjang berubahnya ukuran berkas XML/GML.

Selanjutnya, jika kita melihat pada Tabel 1, kita bisa melihat bahwa Oracle dan Microsoft SQL Server memiliki begitu banyak prosedur tersimpan serta fungsi terkait dengan data XML/GML, sehingga pemrogram-pemrogram aplikasi bisa bekerja dengan lebih fleksibel saat mereka mengembangkan aplikasi (khususnya aplikasi-aplikasi SIG) yang memanfaatkan data XML/GML yang disimpan di dalam kedua sistem basis data tersebut. Di sisi lain, sistem basis data MySQL tidak memiliki banyak prosedur tersimpan serta fungsi terkait dengan data XML/GML, sehingga pemrogram-pemrogram aplikasi harus membuat sendiri prosedur tersimpan serta fungsi-fungsi terkait dengan data XML/GML, jika mereka mau menggunakannya untuk menangani data dengan format-format XML/GML itu. Tentu saja ini bukan 'kekalahan' mutlak MySQL dibandingkan dengan baik Oracle maupun Microsoft SQL Server. Jika aplikasi-aplikasi yang akan dikembangkan tidak terlalu memerlukan pemrosesan data XML/GML yang kompleks, tentu saja MySQL sudah sangat memadai.

KESIMPULAN

Tulisan ini mencoba membandingkan kinerja sistem-sistem basis data XML (*XML Enabled Database*) terutama dari sudut efisiensi waktu serta fleksibilitas pemrogram dalam mengembangkan aplikasi-aplikasi SIG dengan data berformat XML/GML di atasnya. Hasil pengukuran dan analisis menunjukkan bahwa Oracle dan Microsoft SQL Server merupakan sistem-sistem basis data yang sesuai untuk digunakan sebagai tempat penyimpanan data untuk aplikasi-aplikasi SIG yang relatif kompleks. Dalam hal ini, MySQL juga mungkin untuk digunakan sebagai sistem yang mendasari SIG yang relatif tidak kompleks, dengan catatan MySQL tidak secara khusus dirancang untuk menangani data berformat XML/GML. Dari sudutpandang pemrograman, Oracle dan Microsoft SQL Server juga merupakan sistem-sistem basis data yang mungkin paling sesuai untuk aplikasi-aplikasi berbasis XML (khususnya GML), karena kedua sistem basis data ini memiliki sejumlah prosedur tersimpan (*stored procedure*) dan fungsi yang akan mempermudah pekerjaan pengembang aplikasi.

DAFTAR PUSTAKA

1. Kjenstad, K., 2006, *On the integration of object-based models and field-based models in GIS*. International Journal of Geographical Information Science Vol. 20, No. 5, May 2006, 491–509.
2. Zeiler, M., 1999, *Modeling Our World: The ESRI's Guide to Geodatabase Design.* Environmental System Research Institute Inc., Redland, California.
3. _____, 2002, *Metadata and GIS.* Environmental System Research Institute Inc., Redland, California.
4. Lans, R.F. van der, 2007, *SQL for MySQL Developer: A Comprehensive Tutorial and Refference.* Addison-Wesley, Upper-Saddle River, New York.
5. Price, J., Mikolaitis, S., 2008, *Oracle Database 11g: Master SQL and PL/SQL in the Oracle Database.* Oracle Press-McGraw-Hill Companies, Inc., New York.
6. Amirian, P., Alesheikh, A.A., 2008, *Hybrid Architecture for Implementing Efficient Geospatial Web Service: Integrating .NET Remoting and Web Services Technology.* American Journal of Applied Sciences 8 (5): 730-742, 2008ISSN 1812-5654.

7. Amirian, P., Alesheikh, A.A., 2008, *Publishing Geospatial Data through Geospatial Web Service and XML Database System*. American Journal of Applied Sciences 5 (10): 1358-1368, 2008ISSN 1546-9239.
8. Aplikasi SIG yang dikembangkan menggunakan teknologi Silverlight XAML. http://cadmaps.com/gisblog/.
9. Manoj, P. and Gosh, S.K., 2008, A Service-Oriented Approach for Integrating Heterogeneous Spatial Data Sources Realization of a Virtual Geo-Data Repository**.** *International Journal of Cooperative Information Systems* Vol. 17, No. 1 (2008) 111–153.
10. Huang, C.H., Chuang, T.R., Deng, D.P., and Lee, H.M.. *Efficient GML-native Processors for Web-GIS: Techniques and Tools*. Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. http://www.iis.sinica.edu.tw/~trc/public/publications/ACM-GIS06/ACM-GIS06HuangChuang DengLee.pdf. Diakses 6 Juni 2011.
11. Cox, S., Daisey, P., Lake, R., Portele, C., and Whiteside, A., 2005, *OpenGIS Geography Markup Language (GML) Encoding Spesification.* Open Geospatial Consortium, Inc., www.opengeospatial.org/legal/. Diakses November 2010.
12. Michaelis, D., Ames, D.P., Christopher, 2009, *Evaluation and Implementation of the OGC Web Processing Service for Use in Client-Side GIS*. Geoinformatica (2009) 13:109–120DOI 10.1007/s10707-008-0048-1.
13. Spesifikasi OGC. www.opengeospatial.org/specs/. Diakses 6 Desember 2010.
14. Higgins, S., 2003, *Oracle XML DB Developer's Guide.* Oracle Corporation, Redwood City.
15. Melnick, J., 2001, *Oracle XML Developer's Kit Programmers Guide*. Oracle Corporation, Redwood City.
16. Format data untuk aplikasi-aplikasi Sistem Informasi Geografis. www.wikipedia.com. Diakses 8 Desember 2010.
17. Coles, M., *Pro T-SQL 2008 Programmer's Guide,* Apress, New York. www.apress.com.
18. Spesifikasi XQuery. www.wikipedia.com. Diakses 8 Desember 2010.
19. Riset tentang penggunaan GML untuk penyimpanan data yang mendasari aplikasi-aplikasi SIG. http://www.itc.nl/library/papers_2004/msc/gfm/shrestha.pdf.
20. Nugroho, A., 2010, *Mengembangkan Aplikasi Basis Data Dengan Teknik XP Menggunakan IDE Eclipse*. Penerbit ANDI OFFSET, Jogyakarta.