

ANALISA PERBANDINGAN RESPONSE TIME DAN THROUGHPUT PADA XML DAN DBMS SEBAGAI MEDIA PENYIMPANAN DATA

Ragil Martha¹⁾, Yanuar Firdaus²⁾, Kusuma Ayu Laksitowening³⁾

Fakultas Informatika Institut Teknologi Telkom

Jalan Telekomunikasi No.1 Terusan Buah Batu Bandung 40257

(022) 7564108

E-mail :¹mr.ragil@gmail.com, ²yanuar@ittelkom.ac.id, ³kal@ittelkom.ac.id

ABSTRAKS

In the environment where information exchanges running rapidly, performance becomes a matter that is absolutely necessary to ensure the accuracy and speed of information delivery. Response time and throughput are two of the performance parameters that could be considered in choosing data storage media. A DBMS is a set of software programs that controls the organization, storage, management, and retrieval of data in a database. Along with the development of information technology, the needs of DBMS as storage media would be crucial. In addition to DBMS, there is another data storage format -had a similar structure to that relational-named XML. Nowadays Native XML Databases had been introduced as a new technology of XML. In Native XML Databases, XML files could be related one to another just like in DBMS. The objective of the research was to analyze what is best between those two data storage format. The analysis result concluded that DBMS give lower response time and greater throughput.

Keywords: Native XML Database, XML, DBMS, response time, throughput

1. PENDAHULUAN

1.1 Latar Belakang

Saat ini performansi menjadi suatu hal yang mutlak diperlukan dalam teknologi informasi, mengingat kebutuhan akan ketepatan dan kecepatan penyampaian informasi yang semakin meningkat. Yang menjadi penilaian dalam performansi sebuah basis data di antaranya adalah *response time* dan *throughput*[6]. *Response time* adalah waktu yang dibutuhkan untuk mengeksekusi satu *statement query*. Sedangkan *throughput* adalah jumlah *statement* atau perintah yang dapat dieksekusi selama satu satuan waktu. Semakin kecil *response time* dan semakin besar *throughput* sebuah media penyimpanan data, semakin baik pula performansinya.

Seiring berkembangnya dunia teknologi informasi tersebut, peranan DBMS sebagai perangkat lunak untuk menyimpan dan mengelola basis data akan semakin terasa. Namun selain DBMS, ada sebuah format penyimpanan data yang memiliki struktur hierarkis yang sama dengan database relasional dan mudah dalam pertukaran informasi yaitu XML. Dalam perkembangannya, muncul sebuah teknologi yang disebut *Native XML database*. *Native XML database* adalah suatu relational database yang didesain untuk menyimpan data dalam file format XML[4]. Dengan *Native XML database*, file – file XML dapat saling berelasi sama seperti relational database pada umumnya, sehingga bisa dilakukan *query* di dalamnya.

Pada penelitian ini dilakukan analisa perbandingan antara DBMS dan XML yang digunakan sebagai media penyimpanan data dengan parameter *response time* dan *throughput*. Perbandingan ini bertujuan untuk mengetahui

apakah penyimpanan data dengan format XML lebih baik dari pada di DBMS atau sebaliknya.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka permasalahan yang dibahas dalam penelitian ini adalah :

- Bagaimana perbandingan *response time* dan *throughput* antara data yang disimpan dalam format XML dengan data di DBMS.
- Faktor-faktor apa yang menyebabkan *response time* dan *throughput* format penyimpanan data yang satu lebih baik daripada format penyimpanan data yang lain. Dalam hal ini adalah antara format XML dengan DBMS.

1.3 Tujuan Penelitian

Tujuan dari dilakukannya penelitian ini adalah

- Menganalisis performansi data di DBMS dan data dengan format XML menggunakan parameter *response time* dan *throughput*.
- Menganalisis faktor - faktor yang menyebabkan salah satu format data lebih unggul dari sisi performansi dibanding format data yang lain. Format data yang dimaksud adalah format XML dan DBMS.

2. XML DAN NATIVE XML DATABASE

XML dapat disebut juga sebagai '*metalanguage*', yaitu suatu bahasa yang digunakan untuk mendefinisikan bahasa lain[12]. Kata *extensible* menunjukkan suatu fasilitas yang memungkinkan kita untuk menambah atau mendefinisikan 'bahasa markup' sendiri, bukan seperti HTML yang hanya memiliki tag untuk mendefinisikan elemen dokumen, yang menunjukkan tanda bagaimana

ditampilkan dalam browser web. XML disimpan sebagai file teks (ASCII) biasa[12].

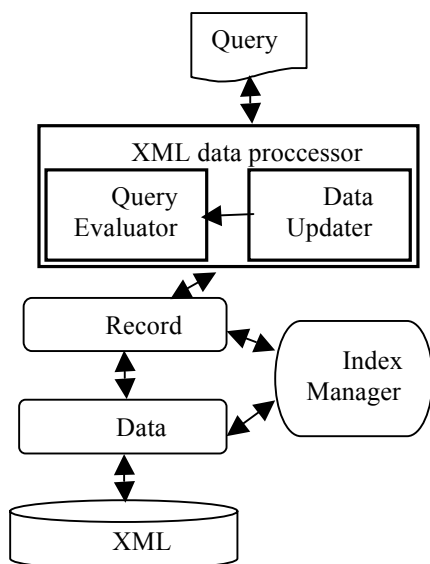
Elemen-elemen yang terdapat dalam XML meliputi:

- Blok utama yang dibangun secara terstruktur dengan XML[12].
- Tag mulai (start tag) dan akhir (end tag)[12].
- Harus secara tepat berulang[12].

XML memiliki keunggulan dalam hal kemudahan pertukaran data dan susunan struktur hierarkisnya yang mirip *database relasional*. Sistem ini disebut dengan *native XML database*. Di sini file-file XML saling direlasikan sama seperti sistem basis data relasional. Terbukti juga bahwa ia mampu bertindak sebagai suatu basis data yang sanggup menerima query dan menampilkan sesuai dengan query tersebut. Selain itu, format XML yang sederhana, makin menambah kemudahan untuk diimplementasikan dalam suatu aplikasi data. Pengembang atau *developer* dapat memberikan tag sesuai dengan yang diinginkan.

Native XML database adalah suatu relational database yang didesain untuk menyimpan data dalam file format XML[4]. Dengan *Native XML database*, file – file XML dapat saling berelasi sama seperti relational database pada umumnya, sehingga bisa dilakukan *query* di dalamnya. Dan seperti database lainnya, *Native XML Database* mendukung adanya transaksi, *locking*, *concurrency*, *security*, *query languages*, *programmatic APIs*, *indexing*, dan lain sebagainya[4].

Di bawah ini adalah arsitektur Native XML Database pada umumnya :



Gambar 1. Arsitektur Native XML Database

3. DBMS

Pada berbagai macam aplikasi, basis data (*database*) merupakan bagian yang penting, karena dengan basis data suatu aplikasi dapat menyimpan, memproses dan menampilkan informasi - informasi

yang dapat dimanfaatkan oleh *user*. Basis data adalah sebuah kumpulan dari data yang disusun secara *logic* dan diatur oleh sekumpulan aturan - aturan, prosedur - prosedur dan fungsionalitas - fungsionalitas yang membantu menjamin konsistensi dan interpretasi data tersebut. Oleh sebab itu, dibutuhkan suatu *tool* yang dapat digunakan untuk mengelola, menjamin konsistensi dan interpretasi data[11].

Berdasarkan kebutuhan di atas, maka munculah sebuah *tools* yang dapat digunakan untuk memenuhi kebutuhan tersebut yang bernama *Database Management System (DBMS)*. *Database Management System* adalah kumpulan dari *software program* yang kompleks yang berguna untuk mengontrol dan mengatur storage dan mengambil sekumpulan data, menyusun data secara *logic* dan menyediakan fungsi - fungsi bagi *user* untuk menjamin bahwa data tersebut disusun secara *logic* dan menjamin kekonsistenan data[14].

Software Program ini juga meliputi fasilitas *backup* yang kuat, kemampuan untuk *recovery*, *logging*, *locking*, *concurrency control* serta pengamanan terhadap data yang ada di dalamnya [11].

Secara umum, sebuah DBMS terdiri dari empat komponen, yaitu :

- Modeling Language* untuk mendefinisikan skema dari tiap basis data yang disimpan di dalam basis DBMS berdasarkan model dari DBMS[14].
- Data Structure (field, record, file* dan objek) yang dioptimasi untuk berhubungan dengan sejumlah data berukuran besar yang disimpan pada *permanent storage*[14].
- Database query language* dan *report writer* yang memungkinkan user berinteraksi dengan basis data, menganalisis data dan memanipulasi data sesuai dengan hak-hak yang dimiliki oleh user tersebut[14].
- Mekanisme transaksi yang menjamin *ACID property*, untuk menjamin integritas data walaupun diakses secara bersamaan oleh *users* dan terjadi kesalahan (*fault*)[14].

Sebuah Oracle Database server terdiri dari Oracle instance dan Oracle database. Oracle Instance terdiri dari memory areas dan background process[11]. Di dalam memory areas terdapat beberapa komponen, yaitu :

- Database buffer cache yang digunakan untuk menampung blok data dari database yang akan diproses[11].
- Redo log buffer yang digunakan untuk menampung informasi redo (yang akan digunakan dalam proses recovery) sebelum ditulis di disk[11].
- Shared pool yang digunakan untuk menampung statement SQL yang dapat dipakai bersama oleh user dari basis data[11].

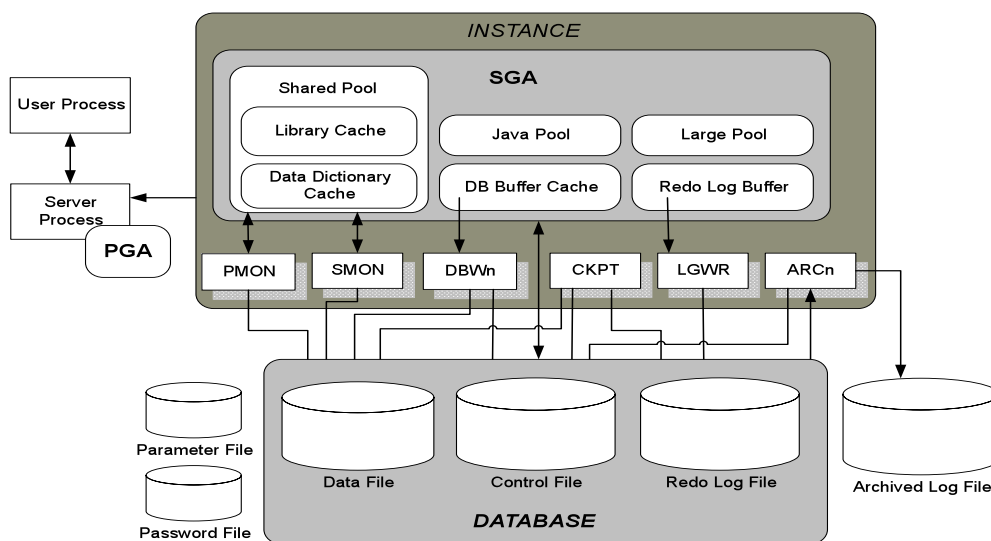
- d. Large pool yang digunakan untuk menampung I/O request dan sebagai penampung dari backup yang dilakukan recovery manager (RMAN)[11].
- e. Java pool yang digunakan untuk menampung kode-kode program java[11].

Di dalam Oracle database terdapat file-file fisik yaitu :

- a. Data file yang digunakan untuk menyimpan data yang dimiliki oleh user maupun aplikasi[11].
- b. Control File yang digunakan untuk menyimpan data atau informasi dari basis data itu sendiri (informasi struktur fisik basis data)[11].
- c. Redo Log file yang merupakan tempat penyimpanan informasi redo yang akan digunakan dalam proses recovery[11].

Pada saat sebuah query diproses, Oracle server proses akan memeriksa apakah blok data yang dibutuhkan ada di database buffer cache. Jika blok yang dibutuhkan tidak terdapat di database buffer cache, maka server process akan melakukan pembacaan untuk mencari blok yang dibutuhkan di data file untuk kemudian meletakkan salinannya di database buffer cache. Jika saat pemrosesan query berikutnya memerlukan blok yang sama dengan query sebelumnya, maka pemrosesan query tersebut tidak memerlukan pembacaan data ke file fisik. Blok data yang berada di memory dan tidak sering diakses akan dikeluarkan dari database buffer cache[7].

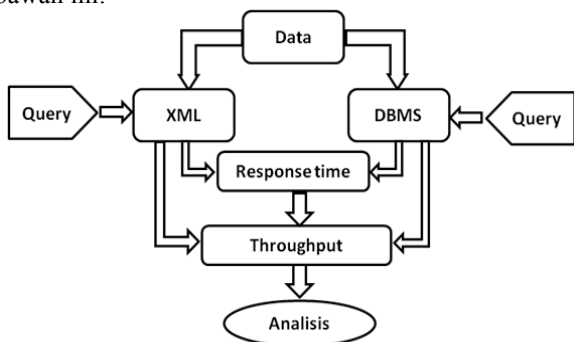
Pada penelitian ini DBMS yang digunakan adalah Oracle 10g R 2. Adapun arsitektur dari Oracle 10g R 2 dapat dilihat pada gambar berikut:



Gambar 2: Arsitektur Oracle 10g R 2

4. GAMBARAN UMUM PERANGKAT UJI

Aplikasi yang dibangun adalah perangkat simulasi untuk menganalisis response time dan throughput XML dan DBMS, secara garis besar perangkat yang dibangun dapat direpresentasikan dalam gambar di bawah ini:



Gambar 3: Gambaran umum sistem

5. PENGUJIAN DAN ANALISA

5.1 Analisa Response Time

Pada analisis response time, ada dua faktor yang mempengaruhi performansi XML dan DBMS sebagai media penyimpanan data. Faktor yang pertama adalah masalah arsitektur dari masing - masing media penyimpanan tersebut. Hal ini sangat berarti karena beda arsitektur, beda juga dalam penanganan query. Untuk DBMS sendiri khususnya oracle yang digunakan pada penelitian ini, pada saat sebuah query diproses, Oracle server proses akan memeriksa apakah blok data yang dibutuhkan ada di database buffer cache. Jika blok yang dibutuhkan tidak terdapat di database buffer cache, maka server process akan melakukan pembacaan untuk mencari blok yang dibutuhkan di data file untuk kemudian meletakkan salinannya di database buffer cache. Jika saat pemrosesan query berikutnya memerlukan blok yang sama dengan query sebelumnya, maka pemrosesan query tersebut tidak memerlukan pembacaan data ke file fisik. Blok data yang berada di memory dan tidak sering diakses akan dikeluarkan dari database buffer cache[7].

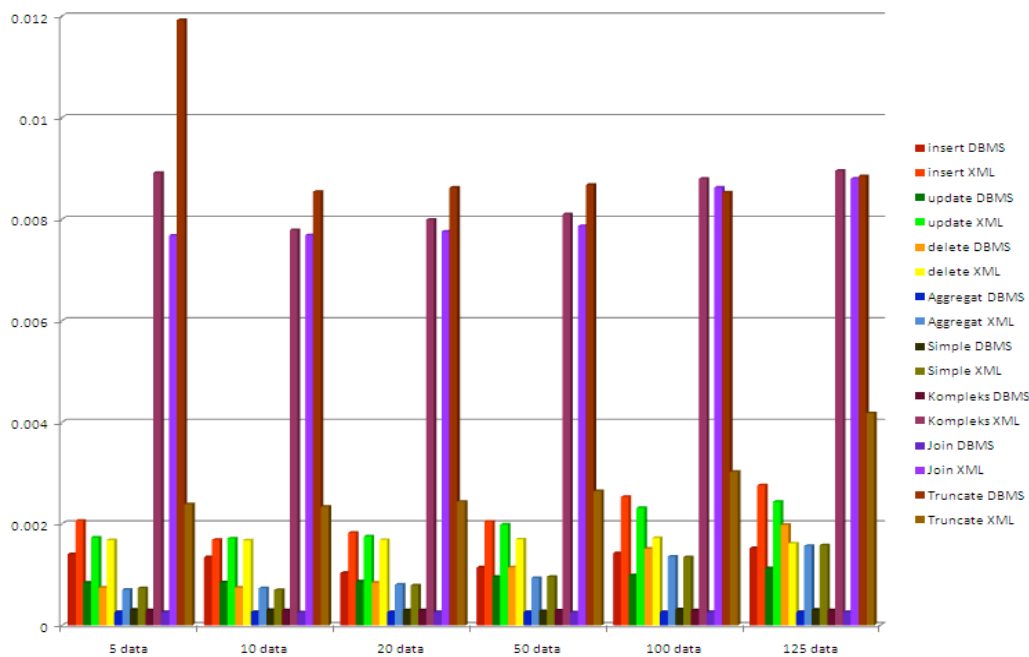
Sedangkan untuk XML, pada saat *query* diproses, akan masuk ke XML data processor. Di XML data processor yang berperan adalah *Link to XML*. Di sini *Query* akan dievaluasi. Kemudian ke bagian *record buffer* untuk memeriksa apakah blok data yang dibutuhkan ada di *record buffer*. Jika blok yang dibutuhkan tidak terdapat di *record buffer*, maka *data manager* akan melakukan pembacaan dengan bantuan *index manager* untuk mencari blok yang dibutuhkan di data file untuk kemudian meletakkannya di *record buffer*. Jika saat pemrosesan *query* berikutnya memerlukan blok yang sama dengan *query* sebelumnya, maka pemrosesan *query* tersebut tidak memerlukan pembacaan data ke file fisik. Blok data yang berada di *memory* dan tidak sering diakses akan dikeluarkan dari *record buffer*. [8]

Faktor kedua adalah masalah *indexing*. Di DBMS khususnya oracle 10g R, terdapat proses

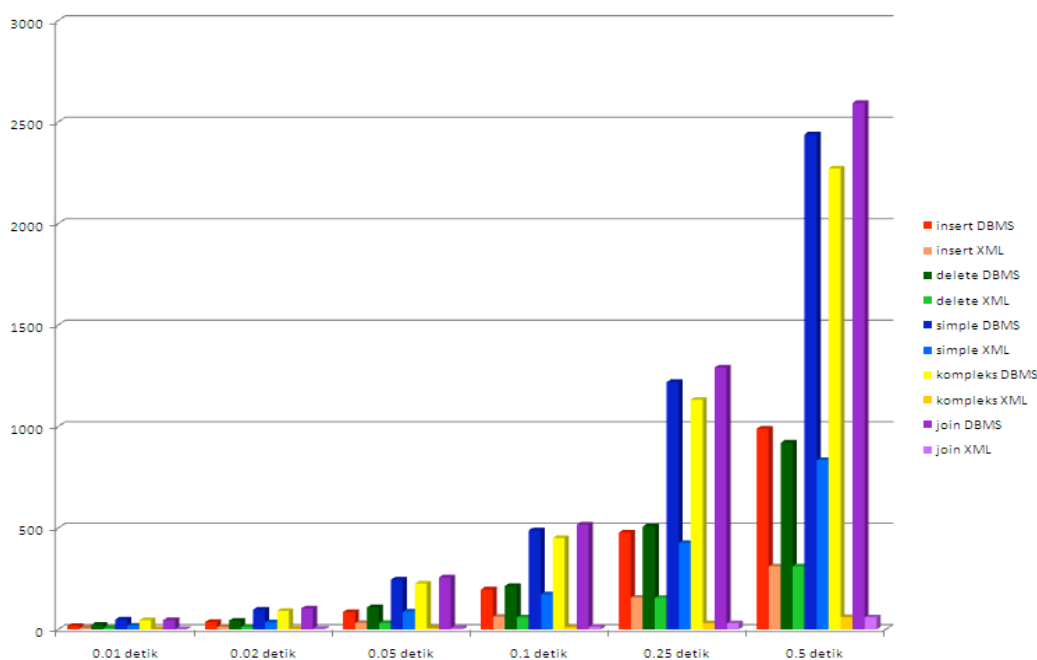
indexing. Ada dua macam *indexing* di oracle, yaitu *B-Tree* dan *Bitmap*. Namun yang dipakai pada penelitian ini dan merupakan *default* dari oracle adalah *B-Tree*[7]. Sedangkan untuk XML sendiri menerapkan *indexing* dengan tipe *value and structural indexes*. Index dalam file xml ini bertujuan untuk meningkatkan "*query speed*" dalam pemrosesan *query*[4].

5.2 Analisa Throughput

Melalui perangkat simulasi yang telah dibuat, dapat dilihat tabel perbandingan *throughput* antara DBMS dengan XML dari setiap *statement query* yang ada. Pada analisis *throughput* ada dua faktor yang mempengaruhi performansi XML dan DBMS sebagai media penyimpanan data. Faktor tersebut adalah faktor arsitektur dan faktor *indexing*. Kedua faktor tersebut telah dijelaskan dalam analisis *response time* di atas.



Gambar 4: Grafik rata - rata hasil pengujian response time



Gambar 5: Grafik rata-rata hasil pengujian throughput

6. KESIMPULAN

Response time dan throughput DBMS sebagai media penyimpanan data lebih baik daripada XML, karena DBMS telah memiliki arsitektur dan indexing yang lebih baik daripada XML. Dari hasil simulasi didapat bahwa DBMS unggul dengan perbandingan sekurang - kurangnya dua kali lipat. Throughput XML menurun bila dibandingkan dengan DBMS pada saat statement kompleks query dan join query karena karakter indexing DBMS yang lebih baik daripada XML.

PUSTAKA

- [1] Aditya Bayu Samudra Siregar, 2002, "Dukungan XML pada pengembangan perangkat lunak untuk sistem pembelajaran jarak jauh di Magister Teknologi Informasi (MTI) UGM", Skripsi Jurusan Teknik Elektro fakultas teknik UGM
- [2] Belajar Tanpa batas. 2008. XML. Dikutip : 28 Agustus 2008, [online]. Available : <http://mbro.belajar.net/2005/01/01/xml>
- [3] Budi Andanasari, 2009, "Analisis performansi pada Object-Oriented Database", Tugas akhir Jurusan Teknik Informatika IT Telkom
- [4] Bourret, Ronald, 2005, "XML and Databases", <http://www.rpbourret.com/xml/XMLAndDatabases.htm>, didownload tanggal 29 Agustus 2008
- [5] Forum Diskusi Teknologi XML. XML Database. Dikutip : 27 Agustus 2008, [online]. Available:<http://www.diskusiweb.com/viewthread.php?tid=22968&page=1#pid149744>
- [6] Krishna, S. 1992, "Introduction to database and knowledge-base systems", Jakarta, World Scientific
- [7] Oracle, 2005, "Oracle Database 10g Administration Workshop I", Oracle
- [8] Orientx, 2005, "Architecture Native XML Database", <http://idke.rur.edu.cn/orientx/orientx3.0.html>
- [9] Rajagukguk Ronald, 2008, Perjalanan ke desa LINQ, Bandung, PT. Elex Media Komputindo
- [10] Ramakrishnan, Raghu and Johannes Gehrke, 2003, "Database Management Systems", The McGraw-Hill Companies, Inc
- [11] Rob Mattison, 1997, "Understanding Database Management System Second Edition", McGraw Hill
- [12] Samsyiar, Evara, 2006, Belajar Sendiri Administrasi Database Oracle 10g, Jakarta, PT. Elex Media Komputindo
- [13] Silberschatz, Korth, Sudharsan, 2002, *Database System Concepts*, Mc Graw Hill
- [14] Wikipedia, 2008, "Database Management System", <http://wikipedia.com/>, didownload tanggal 20 September 2008
- [15] XML Schema Part 1: Structures Second Edition. *XML Schema*. Dikutip, 29 Agustus 2008, [online]. Available : <http://www.w3.org/TR/xmlschema-1/>