

1 Pendahuluan

1.1 Latar belakang

Selama dekade terakhir ini, hampir semua sektor telah sangat bergantung pada teknologi informasi. Jumlah data yang disimpan dalam sistem informasi tersebut secara kontinyu terus membesar. Seiring dengan semakin besarnya jumlah data munculah kebutuhan akan pengaksesan data yang cepat. Pengaksesan data yang cepat akan sangat berguna dalam rangka analisis, pengambilan keputusan, dan hal-hal penting lainnya. Hal-hal tersebut sangat penting bagi suatu organisasi agar dapat bertahan di tengah-tengah persaingan yang semakin ketat dewasa ini. Untuk mendukung sistem informasi di atas data yang semakin besar jumlahnya, *database management system*(DBMS) merupakan jawaban yang paling tepat dalam menjawab persoalan kecepatan akses data.

Untuk menyimpan data yang besar jumlahnya di DBMS dibutuhkan tabel dengan jumlah kolom dan baris yang besar. Baris merepresentasikan jumlah data, sedangkan kolom merepresentasikan detail informasi dari sebuah data. Pada saat *retrieve*, data akan dibaca dari tabel, dan pada kebanyakan proses *retrieve* tidak semua kolom yang ada dibutuhkan. Walaupun tidak semua kolom diambil akan tetapi pada pemrosesannya semua kolom akan dibaca(*full table scan*). Hal tersebut tentunya tidak efektif sehingga dibutuhkan sebuah cara untuk mengatasi persoalan tersebut. Sebuah cara diperkenalkan yaitu dengan memodifikasi arsitektur penyimpanannya melalui mekanisme *column-stores*.

Kebanyakan DBMS konvensional mengimplementasikan *row-stores storage system* dalam penyimpanan datanya, dimana semua atribut dari sebuah baris ditempatkan secara berdampingan dalam media penyimpanan data, sedangkan pada arsitektur *column-stores* yang akan disimpan adalah nilai dari tiap atribut, dan terpisah tiap atributnya. Dewasa ini, dengan semakin berkembang dan besarnya kebutuhan akan data dalam aplikasi-aplikasi besar, maka dibutuhkan pula tingkat pengaksesan yang cepat untuk pengambilan keputusan dan analisis. Dengan arsitektur *column-stores*, DBMS hanya perlu membaca nilai dari kolom yang dibutuhkan untuk memproses query yang diberikan, sehingga dapat menghindari membawa atribut-atribut yang *irrelevant* kedalam memori untuk diproses, seperti yang jamak terjadi pada arsitektur *row-stores* DBMS. Hal tersebut tentunya akan meningkatkan performansi dari DBMS tersebut. Dengan meningkatnya performansi tentu kecepatan akses akan data-data yang besar meningkat pula.

1.2 Perumusan masalah

Berdasarkan latar belakang tersebut, maka permasalahan yang akan dibahas adalah sebagai berikut :

1. Bagaimana tingkat perbandingan performansi antara arsitektur *column-stores* database dengan *row-stores* database.
2. Faktor-faktor yang menyebabkan performansi DBMS yang satu lebih baik daripada DBMS yang lain.

3. Adakah situasi dimana *column-stores* database atau *row-stores* database efisien atau tidak efisien untuk digunakan.

Untuk menghindari meluasnya meteri pembahasan Tugas Akhir ini, maka Penulis membatasi permasalahan sebagai berikut :

1. Menggunakan data akademik IT Telkom.
2. Pemodelannya menggunakan relasional model.
3. Yang diamati hanyalah performansi dengan parameter *throughput* dan *response time*.
4. Implementasi *column-stores* database menggunakan MonetDB 5.
5. Implementasi *row-stores* database menggunakan Oracle 10g.

1.3 Tujuan

Tujuan yang ingin dicapai dalam Tugas Akhir ini adalah :

1. Membangun perangkat lunak yang dapat digunakan untuk menguji performansi kedua DBMS sehingga bisa dijadikan sebagai sarana menarik kesimpulan.
2. Mengukur tingkat performansi antara kedua DBMS pada skenario-skenario yang diambil.
3. Menganalisis faktor-faktor yang menyebabkan salah satu DBMS lebih unggul dari sisi performansi dibanding yang lain.
4. Mencari situasi dimana *column-stores* database atau *row-stores* database efisien atau tidak efisien untuk digunakan.

1.4 Metodologi penyelesaian masalah

Berikut ini adalah metodologi penyelesaian masalah yang dipergunakan dalam menyelesaikan Tugas Akhir ini:

1.4.1 Study Literatur

Pada tahap ini yang dilakukan adalah mencari referensi yang dapat digunakan sebagai acuan dalam pembangunan basis data dalam DBMS MonetDB 5 atau dalam DBMS Oracle 10g, meliputi DDL, DML maupun query yang akan diterapkan pada skenario pengujian. Selain itu referensi lainnya juga digunakan untuk membantu pembuatan aplikasi dalam Java.

1.4.2 Analisis dan Desain

Tahapan ini adalah tahapan yang meliputi analisis dan perancangan terhadap basis data yang akan dibangun, merancang kebutuhan dan mendesain perangkat lunak yang akan digunakan untuk menguji performansi DBMS.

1.4.3 Implementasi Basis Data dan Perangkat Lunak

Tahap ini adalah tahap pembangunan basis data baik di tradisional DBMS maupun di *column-stores* DBMS. Selain membangun basis data, pada tahap ini dibangun pula perangkat lunak yang akan digunakan sebagai alat untuk membandingkan performansi dari kedua DBMS. Pada tahap ini DBMS yang digunakan untuk *row-stores* adalah Oracle 10g, untuk *column-stores* menggunakan MonetDB 5, sedangkan perangkat lunaknya menggunakan bahasa pemrograman Java.

1.4.4 Testing

Pada tahap ini akan dilakukan pengujian terhadap perangkat lunak yang telah dibangun, apakah sudah bekerja dengan benar. Pengujian dilakukan dengan memberikan statement SQL yang sama ke dua DBMS tersebut.

1.4.5 Analisis Hasil

Output dari perangkat lunak berupa *throughput* dan *response time*, baik dari konvensional DBMS maupun *column-stores* DBMS akan dibandingkan, sehingga akan diketahui DBMS mana yang lebih baik dari segi performansi. Setelah didapat DBMS mana yang lebih baik, kemudian akan dianalisis apa faktor-faktor apa yang menyebabkan salah satu DBMS lebih baik dari yang lainnya.

1.4.6 Pengambilan Kesimpulan dan Penyusunan Laporan

Pada tahap terakhir ini diambil kesimpulan dari hasil analisis yang telah dilakukan pada tahap sebelumnya untuk kemudian disusun laporan terhadap analisis yang telah dilakukan.

1.5 Sistematika Penulisan

Tugas akhir ini disusun dengan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Berisi pemaparan mengenai latar belakang masalah, tujuan, perumusan masalah, batasan masalah, metodologi tugas akhir, dan sistematika penulisan.

BAB II LANDASAN TEORI

Berisi uraian mengenai landasan teori yang akan digunakan, meliputi teori tentang konsep *column-stores* dan *row-stores* database dan teori-teori pendukung yang berkaitan dengan penelitian tugas akhir ini.

BAB III DESAIN DAN PERANCANGAN SISTEM

Berisi tentang *use case*, *sequence diagram*, *class diagram* dan *entity relationship diagram* dari aplikasi dan database.

BAB IV ANALISIS DAN EVALUASI SISTEM

Berisi hasil analisa dari penelitian dan pengujian yang dilakukan.

BAB V PENUTUP

Berisi kesimpulan dan saran untuk pengembangan lebih lanjut terhadap hasil penelitian ini.