

TUGAS BESAR
TUNING DATABASE SYSTEM
MATAKULIAH MANAJEMEN BASIS DATA

Disusun oleh:

Reza Pahlevi /14116002



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI PRODUKSI, INDUSTRI DAN INFORMASI
INSTITUT TEKNOLOGI SUMATERA
2019

BAB I

PENDAHULUAN

1.1 Latar Belakang

Manajemen basis data merupakan sebuah metode yang memudahkan para pengguna untuk mengakses, menyimpan, memanipulasi serta dipanggil oleh penggunaannya pada suatu kumpulan data yang saling berhubungan. Sedangkan basis data sendiri berarti kumpulan data yang terdiri dari satu atau lebih tabel yang terintegrasi satu sama lain, dimana setiap user diberi wewenang untuk dapat mengakses (seperti mengubah, menghapus) data dalam tabel-tabel tersebut.

Tujuan basis data sendiri adalah untuk mempercepat dan memudahkan dalam penyimpanan dan manipulasi data, efisiensi ruang penyimpanan, keakuratan data, ketersediaan data, kelengkapan data, keamanan data dan kebersamaan pemakai. Sehingga basis data atau manajemen basis data sangat dibutuhkan oleh perusahaan dan transaksi-transaksi yang dilakukan sehari-hari untuk memudahkan dalam pencarian data tersebut.

Pada manajemen basis data terdapat salah satu cara pada database sistem untuk meningkatkan performa kinerja database yang disebut *tunning*. Pada tugas besar kali ini penulis membuat sebuah proses tuning pada level *indexing* dan *setting* sistem manajemen database (DBMS). Dimana terdapat suatu data yang akan diproses dengan cara tuning, lalu akan dibandingkan hasilnya antara waktu sebelum tuning dan waktu setelah tuning, menggunakan mysql dan java.

1.2 Tujuan

Tujuan pembuatan tugas besar ini adalah sebagai berikut:

1. Memenuhi tugas besar matakuliah manajemen basis data.
2. Melatih akademik mengenai pemahaman konsep dan cara melakukan tuning.
3. Mampu membandingkan *response time* sebelum dan sesudah melakukan *tuning* konfigurasi *my.ini* pada database
4. Mampu membandingkan *response time* sebelum dan sesudah melakukan *tuning indexing* pada database

1.3 Manfaat

Adapun manfaat dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Memudahkan dalam proses pencarian dalam kumpulan data.
2. Meningkatkan performa kerja database.
3. Dapat meningkatkan kemampuan akademik mengenai konsep dan cara melakukan tuning.

BAB II

LANDASAN TEORI

2.1 Database Management System (DBMS)

Menurut Turban (2010:94), “Database Management System adalah program software atau kumpulan program yang menyediakan akses ke database”. Berdasarkan teori ahli dapat disimpulkan bahwa, Database Management System (DBMS) adalah suatu program software yang menyediakan akses ke database dan memungkinkan user untuk mendefinisikan, membuat, memelihara, dan mengontrol akses ke database tersebut.

Menurut Connolly dan Begg (2010:68), komponen DBMS terdiri dari 5 komponen yaitu:

- **Hardware (Perangkat Keras):** DBMS dan aplikasi membutuhkan hardware untuk dapat dijalankan. Perangkat keras tersebut dapat berkisar dari komputer pribadi/tunggal atau jaringan komputer. Perangkat keras tersebut tergantung pada kebutuhan organisasi dan DBMS yang digunakan.
- **Software (Perangkat Lunak):** Perangkat-perangkat lunak yang dibutuhkan dalam menjalankan DBMS. Komponen perangkat lunak tersebut terdiri dari perangkat lunak DBMS itu sendiri dan program aplikasi, bersama dengan sistem operasi termasuk perangkat lunak ja
- **Data:** Data adalah bagian yang penting dalam DBMS karena data berperan sebagai jembatan antara komponen perangkat lunak dengan komponen manusia
- **Procedure:** Procedure adalah bagaimana aturan-aturan dalam menggunakan database. Pengguna sistem dan staff yang mengelola database memerlukan prosedur terdokumentasi mengenai cara menggunakan atau menjalankan sistem
- **People:** Merupakan user atau orang yang terlibat dalam sebuah system

A. Fungsi database manajemen system adalah sebagai berikut:

- Kontrol atas redundansi atau pengulangan data
- Konsistensi data

- Informasi yang diperoleh dari data yang sama lebih banyak
- Data dapat dibagikan
- Meningkatkan integrasi data
- Meningkatkan keamanan data
- Penetapan standarisasi pelaksanaan
- Skala ekonomi
- Kesimbangan dari kebutuhan yang saling bertentangan
- Meningkatkan aksesibilitas dan respon data
- Meningkatkan produktivitas
- Meningkatkan concurrency
- Meningkatkan layanan backup dan recovery

B. Kelebihan dan Kekurangan

- Keuntungan DBMS menurut Connolly dan Begg (2005:26), yaitu :
 1. Penggunaan data bersama (sharing of data)
 2. Mengurangi kerangkapan data
 3. Menghindari ketidakkonsistenan data
 4. Integritas data terpelihara
 5. Keamanan terjamin
 6. Penyimpanan data dalam jumlah yang besar
 7. Penetapan standarisasi
 8. Meningkatkan produktivitas
 9. Layanan Back up dan Recovery semakin baik

Keuntungan DBMS menurut Ramakrishnan dan Gehrke (2003:9) adalah sebagai berikut :

1. Data Independence
2. Efficient Data Access
3. Integritas dan keamanan data
4. Data Administration
5. Concurrent Access and Crash Recovery
6. Reduced Application Development Time

Keuntungan DBMS menurut Abdul Kadir (2001:17) adalah :

1. Kepraktisan: Menggunakan media penyimpanan sekunder yang berukuran kecil tetapi padat informasi
2. Kecepatan: Mesin dapat mengambil atau mengubah data jauh lebih cepat dari pada manusia
3. Mengurangi Kejemuan: Dengan menggunakan tangan dalam mengganti informasi cenderung bisa membuat kejemuan, namun dengan DBMS ini bisa mengurangi kejemuan
4. Kekinian: Informasi yang tersedia pada DBMS akan bersifat mutakhir dan akurat setiap saat

2.2 Tuning Database System

Tuning merupakan salah satu cara pada database system untuk meningkatkan performa kinerja database Tuning database system dapat dilakukan dengan pendekatan beberapa level:

1. Hardware
2. Sistem Operasi
3. Database Manajement System
4. Schema basis data
5. Indexing
6. Query

Pada tugas besar ini hanyaakan melakukan tunning pada level *indexing* dan *setting* Database Manajement System (DBMS).

2.3 Spesifikasi Alat

1. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread, multi-user. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

2. Java

Java merupakan bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya di suatu perusahaan perangkat lunak yang bernama Sun Microsystems, pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama “Oak”, namun pada tahun 1995 diganti namanya menjadi “Java”.

3. Laptop

Pada percobaan ini penulis menggunakan laptop ASUS VivoBook A442UR yang memiliki spesifikasi sebagai berikut :

- Processor : Intel® Core™ i5-8250U CPU @ 1.60GHz 1.80GHz
- RAM : 4.00 GB
- Storage : 250 GB SSD + 1 TB HDD
- OS : Windows 10 Home Single Language

BAB III

PERCOBAAN

3.1 Generate Data

Generate data menggunakan tableGen.java dilakukan dengan perintah

- javac tableGen.java
- java tableGen

3.2 Perubahan Nilai

Mengubah nilai pada tiap data dengan nilai

advisor = 100, student = 100, section = 200,takes = 200
advisor = 200, student = 200, section = 400,takes = 400
advisor = 500, student = 500, section = 1000,takes = 1000
advisor = 700, student = 700, section = 20000,takes = 20000
advisor = 1000, student = 1000, section = 100000,takes = 1000000
advisor = 1800, student = 1800, section = 180000,takes = 1800000
advisor = 10000, student = 10000, section = 30000000,takes = 30000000

3.3 Tuning

a. Tuning dengan mengubah konfigurasi pada *my.ini*. Nilai yang diubah yaitu :

- innodb_buffer_pool_size dari 16M menjadi 2700M
- innodb_additional_mem_pool_size dari 2M menjadi 20M
- sort_buffer_size dari 20M menjadi 512K

b. Tuning dengan menambah index pada kolom table :

- Kolom name pada tabel student
- Kolom tot_cred pada tabel student
- Kolom grade pada tabel takes

Kolom time_slot_id pada tabel section

3.4 Percobaan

- `SELECT * FROM student;`

No	Jumlah data	sebelum (second)	config (second)	index (second)
1	advisor = 100, student = 100, section = 200,takes = 200	0.00	0.00	0.00
2	advisor = 200, student = 200, section = 400,takes = 400	0.00	0.00	0.00
3	advisor = 500, student = 500, section = 1000,takes = 1000	0.00	0.00	0.00
4	advisor = 700, student = 700, section = 20000,takes = 20000	0.00	0.00	0.00
5	advisor = 1000, student = 1000, section = 100000,takes = 1000000	0.00	0.00	0.00
6	advisor = 1800, student = 1800, section = 180000,takes = 1800000	0.00	0.00	0.00
7	advisor = 10000, student = 10000, section = 30000000,takes = 30000000	0.02	0.01	0.01

Pada jumlah data nomor 1 sampai 6 hasilnya yaitu 0.00 second, jadi sebelum tuning maupun setelah tuning mendapatkan repond time 0.00 second. Pada jumlah data nomor 7 sebelum tuning yaitu 0.02 second, tuning dengan config 0.01 second, dan tuning dengan index yaitu 0.01 second.

- `SELECT * FROM student WHERE tot_cred > 30;`

No	Jumlah data	sebelum (second)	config (second)	index (second)
1	advisor = 100, student = 100, section = 200,takes = 200	0.00	0.00	0.00
2	advisor = 200, student = 200, section = 400,takes = 400	0.00	0.00	0.00
3	advisor = 500, student = 500, section = 1000,takes = 1000	0.00	0.00	0.00
4	advisor = 700, student = 700, section = 20000,takes = 20000	0.00	0.00	0.00
5	advisor = 1000, student = 1000, section = 100000,takes = 1000000	0.00	0.00	0.00
6	advisor = 1800, student = 1800, section = 180000,takes = 1800000	0.00	0.00	0.00
7	advisor = 10000, student = 10000, section = 30000000,takes = 30000000	0.01	0.01	0.01

Pada jumlah data nomor 1 sampai 6 hasilnya yaitu 0.00 second, jadi sebelum tuning maupun setelah tuning mendapatkan repond time 0.00 second. Pada jumlah data nomor 7 sebelum tuning yaitu 0.01 second, tuning dengan config 0.01 second, dan tuning dengan index yaitu 0.01 second.

- `SELECT name, dept_name FROM student WHERE tot_cred > 30;`

No	Jumlah data	sebelum (second)	config (second)	index (second)
1	advisor = 100, student = 100, section = 200,takes = 200	0.00	0.00	0.00
2	advisor = 200, student = 200, section = 400,takes = 400	0.00	0.00	0.00
3	advisor = 500, student = 500, section = 1000,takes = 1000	0.00	0.00	0.00
4	advisor = 700, student = 700, section = 20000,takes = 20000	0.00	0.00	0.00
5	advisor = 1000, student = 1000, section = 100000,takes = 1000000	0.00	0.00	0.00
6	advisor = 1800, student = 1800, section = 180000,takes = 1800000	0.00	0.00	0.00
7	advisor = 10000, student = 10000, section = 30000000,takes = 30000000	0.01	0.00	0.00

Pada jumlah data nomor 1 sampai 6 hasilnya yaitu 0.00 second, jadi sebelum tuning maupun setelah tuning mendapatkan repond time 0.00 second. Pada jumlah data nomor 7 sebelum tuning yaitu 0.01 second, tuning dengan config 0.00 second, dan tuning dengan index yaitu 0.00 second.

- `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id;`

No	Jumlah data	sebelum (second)	config (second)	index (second)
1	advisor = 100, student = 100, section = 200,takes = 200	0.01	0.01	0.00
2	advisor = 200, student = 200, section = 400,takes = 400	0.01	0.01	0.01
3	advisor = 500, student = 500, section = 1000,takes = 1000	0.02	0.02	0.02
4	advisor = 700, student = 700, section = 20000,takes = 20000	5.77	2.80	2.10
5	advisor = 1000, student = 1000, section = 100000,takes = 1000000			
6	advisor = 1800, student = 1800, section = 180000,takes = 1800000			
7	advisor = 10000, student = 10000, section = 30000000,takes = 30000000			

Pada jumlah data nomor 1 sampai nomor 3 didapatkan hasil sebelum tuning dan setelah tuning tidak berbeda jauh karena data yang masih sedikit. Untuk jumlah data nomor 4 didapatkan hasil yang cukup signifikan yaitu sebelum tuning 5.77 second, setelah tuning config 2.80 second, dan paling efisien dengan tuning index yaitu 2.10 second. Dari query ini didapatkan hasil yang sangat banyak yaitu sebanyak 2.019.367 row. Untuk nomor 5 sampai 7 tidak didapatkan hasil karena jumlah data yang sangat banyak dan komputer dari penulis tidak mampu untuk menjalankan query ini.

- `SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id;`

No	Jumlah data	sebelum (second)	config (second)	index (second)
1	advisor = 100, student = 100, section = 200,takes = 200	0.00	0.00	0.00
2	advisor = 200, student = 200, section = 400,takes = 400	0.00	0.00	0.00
3	advisor = 500, student = 500, section = 1000,takes = 1000	0.01	0.01	0.01
4	advisor = 700, student = 700, section = 20000,takes = 20000	2.77	2.65	1.31
5	advisor = 1000, student = 1000, section = 100000,takes = 1000000			
6	advisor = 1800, student = 1800, section = 180000,takes = 1800000			
7	advisor = 10000, student = 10000, section = 30000000,takes = 300000000			

Pada jumlah data nomor 1 sampai nomor 3 didapatkan hasil sebelum tuning dan setelah tuning tidak berbeda jauh karena data yang masih sedikit. Untuk jumlah data nomor 4 didapatkan hasil yaitu sebelum tuning 2.77 second, setelah tuning config 2.80 second, dan paling efisien dengan tuning index yaitu 1.31 second. Dari query ini didapatkan hasil yang sangat banyak yaitu sebanyak 2.019.367 row. Untuk nomor 5 sampai 7 tidak didapatkan hasil karena jumlah data yang sangat banyak dan komputer dari penulis tidak mampu untuk menjalankan query ini

BAB IV

PENUTUP

4.1 Kesimpulan

Hasil yang didapatkan pada query 1 sampai 3 mendapatkan respon time yang cepat yaitu 0.00 second hingga 0.02 second karena query ini ringan dan tidak membebani kinerja DBMS karena hanya operasi SELECT untuk menampilkan data. Sedangkan untuk query 4 dan 5 mendapatkan respon time yang lambat yaitu lebih dari 1 second pada data yang besar, hal itu karena query tersebut menggunakan operasi JOIN yang sangat membebani kinerja DBMS.

Hasil tuning dengan index lebih cepat dibandingkan dengan config. Untuk data yang besar sangat besar index sangat membantu kinerja DBMS karena data disimpan dalam bentuk pohon sehingga membantu pembacaan data.

4.2 Saran

Adapun saran dari penulis yaitu :

- Menggunakan komputer yang spesifikasinya lebih bagus untuk mempercepat proses query.
- Lebih banyak mengubah config yang lain untuk mendapatkan hasil yang maksimal dengan referensi ahli.
- Mengindex semua field untuk mendapatkan respond time yang lebih cepat.

LAMPIRAN

```
Command Prompt - mysql -u root
| 99975 | 938 | 62 | Spring | 2009 | C | 99975 | yuyun | DK |
| 117 | 938 | 95 | Spring | 2006 | C | 99975 | yuyun | DK | P
| 99975 | 938 | 62 | Spring | 2009 | C | 99975 | yuyun | DK |
| 117 | 938 | 96 | Spring | 2006 | B | 99975 | yuyun | DK | I
| 99975 | 938 | 62 | Spring | 2009 | C | 99975 | yuyun | DK |
| 117 | 938 | 97 | Fall | 2001 | B | 99975 | yuyun | DK | K
| 99975 | 938 | 62 | Spring | 2009 | C | 99975 | yuyun | DK |
| 117 | 938 | 98 | Spring | 2009 | C | 99975 | yuyun | DK | E
| 99975 | 938 | 62 | Spring | 2009 | C | 99975 | yuyun | DK |
| 117 | 938 | 99 | Spring | 2005 | G | 99975 | yuyun | DK | J
+-----+
2019367 rows in set (2.80 sec)

MariaDB [dbms1]>
Administrator Command Prompt - mysql -u root
| RQ | yuyun | DK | 62 | Spring | 910 | F | 938
| RQ | yuyun | DK | 62 | Spring | 191 | D | 938
| RQ | yuyun | DK | 62 | Spring | 666 | B | 938
| RQ | yuyun | DK | 62 | Spring | 489 | C | 938
| RQ | yuyun | DK | 62 | Spring | 212 | B | 938
| RQ | yuyun | DK | 62 | Spring | 666 | B | 938
| RQ | yuyun | DK | 62 | Spring | 477 | G | 938
| RQ | yuyun | DK | 62 | Spring | 477 | G | 938
+-----+
2019367 rows in set (1.31 sec)

Administrator Command Prompt - mysql -u root
| Adri | HH | 94 | Spring | 642 | J | 968
| SS | Yohan | HH | 97 | Fall | 642 | J | 968
| SS | Budi | GO | 98 | Fall | 642 | J | 968
| SS | Ahmad | HH | 98 | Fall | 642 | J | 968
| SS | Ande | RR | 99 | Fall | 642 | J | 968
| SS | rahmat | SS | 99 | Fall | 642 | J | 968
| SS |
+-----+
2019367 rows in set (2.65 sec)

MariaDB [dbms1]>
```