

TUGAS BESAR
MANAGEMEN BASIS DATA
TUNING DATABASE



Disusun oleh

Muhammad Muttaqin 14117138

INFORMATICS ENGINEERING
INSTITUT TEKNOLOGI SUMATERA

2019

DAFTAR ISI

BAB I.....	i
PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Tujuan.....	1
BAB II	2
LANDASAN TEORI	2
2.1. Database Management System (DBMS).....	2
2.2. Index Tuning	2
2.3. Tuning dengan Setting Konfigurasi DBMS	2
BAB III.....	1
PERCOBAAN DAN ANALISIS DATA	1
3.1. Data yang dipakai.....	1
3.2. Query yang digunakan	1
3.3. Tuning.....	1
3.4. Data Hasil Percobaan	2
3.5. Analisis Percobaan	4
BAB IV	5
PENUTUP.....	5
Kesimpulan.....	5
DAFTAR PUSTAKA.....	6

BAB I

PENDAHULUAN

1.1. Latar Belakang

Basis data adalah kumpulan data mentah yang memiliki entitas dan relasi yang kemudian dapat diolah agar menjadi data yang memiliki makna. Basis data memiliki aplikasi untuk mengelola data-data, termasuk membuat, membaca, memperbarui, dan menghapus data yang ada. Pengelola basis data disebut Database Manajemen Sistem (DBMS).

Fitur-fitur yang diberikan oleh sebuah database berbagai macam, seperti keamanan, efisiensi, kesediaan data, kelengkapan data, keamanan data dan *durability*. Sehingga database tersebut menjadi mudah digunakan, namun beberapa database yang bersifat gratis, tidak memiliki fitur yang lengkap, terutama pada hal efisiensi data.

Efisiensi pengambilan data pada database biasa disebut *tuning*. Efisiensi data pada database dapat dilakukan dengan cara indexing, yaitu *tuning* data. *Tuning* pada pengaturan atau konfigurasi DBMS juga dapat dilakukan. Maka kali ini, akan dilakukan percobaan untuk *tuning* database pada index data dan konfigurasi DBMS.

1.2. Tujuan

1. Memahami konsep dan cara melakukan tuning database dengan indexing.
2. Memahami konsep dan cara melakukan tuning database dengan konfigurasi DBMS.
3. Memahami perbedaan akses data setelah tuning index dan konfigurasi DBMS.

BAB II

LANDASAN TEORI

2.1. Database Management System (DBMS)

DBMS adalah koleksi data yang besar yang terintegrasi, yang memodelkan data nyata, yang memiliki entitas dan relasi. Yang disimpan di paket software dan *manage* database.

2.2. Index Tuning

Index tuning adalah membuat indeks yang tepat untuk mempercepat query dalam mengakses data dengan menghapus indeks berlebih

2.3. Tuning dengan Setting Konfigurasi DBMS

Tuning dengan melakukan perubahan/setting konfigurasi DBMS guna mempercepat kinerja DBMS, contohnya menambahkan cache.

BAB III

PERCOBAAN DAN ANALISIS DATA

3.1. Data yang dipakai

Data yang dipakai adalah sebagai berikut:

1. advisor = 100, student = 100, section = 200, takes = 200
2. advisor = 200, student = 200, section = 400, takes = 400
3. advisor = 500, student = 500, section = 1000, takes = 1000
4. advisor = 700, student = 700, section = 20000, takes = 20000
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

3.2. Query yang digunakan

Query yang digunakan adalah sebagai berikut:

1. `SELECT * FROM student`
2. `SELECT * FROM student WHERE tot_cred > 30;`
3. `SELECT `name`, department FROM student WHERE tot_cred > 30;`
4. `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id`
5. `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`

3.3. Tuning

Tuning database yang digunakan terbagi menjadi dua bagian, yaitu:

1. Tuning dengan mengubah konfigurasi setting *my.ini* pada DBMS yaitu dengan mengubah nilai-nilai antara lain:
 - a. `innodb_buffer_pool_size` dari 16M menjadi 2500M
 - b. `innodb_log_file_size`, size dari 5M menjadi 100M
 - c. `innodb_log_buffer_size`, size dari 8M menjadi 100M

2. Tuning dengan melakukan indexing pada beberapa kolom table:
 - a. Data name di table student
 - b. Data tot_cred di table student
 - c. Data grade di table takes
 - d. Data time_slot_id di table section

3.4. Data Hasil Percobaan

Berikut ini adalah data yang didapatkan sebelum dan setelah melakukan ujicoba tuning pada index table dan tuning melalui konfigurasi.

Query = SELECT * FROM student

Data	Waktu Sebelum Tuning (s)	Waktu Sesudah Tuning (s)	
		konfigurasi	indexing
advisor = 100, student = 100, section = 200,takes = 200	0.00162	0.00091	0.00150
advisor = 200, student = 200, section = 400,takes = 400	0.00154	0.00124	0.00141
advisor = 500, student = 500, section = 1000,takes = 1000	0.00217	0.01922	0.00211
advisor = 700, student = 700, section = 20000,takes = 20000	0.00710	0.00337	0.04319

Query = SELECT * FROM student WHERE tot_cred > 30;

Data	Waktu Sebelum Tuning (ms)	Waktu Sesudah Tuning (ms)	
		konfigurasi	indexing
advisor = 100, student = 100, section = 200,takes = 200	0.00162	0.00091	0.00150
advisor = 200, student = 200, section = 400,takes = 400	0.00254	0.00164	0.00141
advisor = 500, student = 500, section = 1000,takes = 1000	0.00351	0.00225	0.00245
advisor = 700, student = 700, section = 20000,takes = 20000	0.00902	0.00771	0.00699

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

Data	Waktu Sebelum Tuning (ms)	Waktu Sesudah Tuning (ms)	
		konfigurasi	indexing
advisor = 100, student = 100, section = 200,takes = 200	0.00125	0.00078	0.00094
advisor = 200, student = 200, section = 400,takes = 400	0.00209	0.00152	0.00192
advisor = 500, student = 500, section = 1000,takes = 1000	0.00188	0.00135	0.00159
advisor = 700, student = 700, section = 20000,takes = 20000	0.00614	0.00407	0.00622

Query = SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id

Data	Waktu Sebelum Tuning (ms)	Waktu Sesudah Tuning (ms)	
		konfigurasi	indexing
advisor = 100, student = 100, section = 200,takes = 200	0.00875	0.00442	0.00537
advisor = 200, student = 200, section = 400,takes = 400	0.01747	0.01387	0.1635
advisor = 500, student = 500, section = 1000,takes = 1000	0.05681	0.03691	0.04129
advisor = 700, student = 700, section = 20000,takes = 20000	0.01697	0.00832	0.00954

Query = 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

Data	Waktu Sebelum Tuning (ms)	Waktu Sesudah Tuning (ms)	
		konfigurasi	indexing
advisor = 100, student = 100, section = 200,takes = 200	0.00663	0.00524	0.00635
advisor = 200, student = 200, section = 400,takes = 400	0.01251	0.01226	0.01382
advisor = 500, student = 500, section = 1000,takes = 1000	0.03708	0.03347	0.03562
advisor = 700, student = 700, section = 20000,takes = 20000	0.00981	0.00871	0.00882

3.5. Analisis Percobaan

Pada percobaan diatas, penulis melakukan pengujian pada beberapa query yang telah dilakukan pada sejumlah data terkecil sampai data terbesar. Eksekusi setiap query sebelum dilakukan tuning cukup lama, kemudian penulis melakukan tuning pada 2 aspek yaitu tuning pada setting konfigurasi DBMS dan tuning dengan Teknik indexing. Pada tuning dengan Teknik indexing menghasilkan waktu eksekusi query lebih baik dari eksekusi pertama, pada beberapa query diatas perubahan eksekusi data cukup signifikan. Kemudian pada tuning kedua yaitu tuning pada setting konfigurasi DBMS, eksekusi query memiliki waktu yang lebih baik dari tuning indexing. Sehingga tuning pada setting konfigurasi DBMS lebih baik ketimbang tuning indexing.

Pada kedua tuning yang dilakukan, dapat dilihat tuning konfigurasi DBMS lebih baik karena tuning konfigurasi DBMS akan mempengaruhi kinerja akses data. Seperti tuning pada buffer pool size untuk memperbesar cache sehingga akses data lebih cepat. Pada tuning yang penulis lakukan, penulis melakukan tuning pada buffer pool dari 16 M menjadi 2000 M, log file size dari 5 M menjadi 100 M dan buffer size dari 8 M menjadi 100 M. Menghasilkan waktu akses query yang baik dibandingkan sebelum dilakukan tuning konfigurasi DBMS.

Tuning dengan indexing lebih baik ketimbang akses pertama sebelum dilakukan tuning, namun tidak lebih baik dari tuning konfigurasi DBMS karena tuning pada indexing mempercepat akses data sesuai dengan field yang diberikan tuning indexing. Seperti tuning pada indexing table student dengan memberikan indexing pada field nama dan tot_cred. Menghasilkan waktu akses query yang baik dibandingkan sebelum dilakukan tuning sama sekali.

BAB IV

PENUTUP

Kesimpulan

1. Waktu akses query pada suatu database dipengaruhi oleh jumlah data, query yang dilakukan, tuning yang dilakukan serta spesifikasi computer.
2. Tuning pada setting konfigurasi DBMS menghasilkan waktu query lebih baik dibanding tuning indexing.

DAFTAR PUSTAKA

Silberschatz, H.F. Korth, S. Sudarshan, 'Database System Concepts', McGraw-Hill