

TUGAS BESAR
MANAJEMEN BASIS DATA
TUNNING DATABASE



DISUSUN OLEH :
Yopan Eko Situmorang (14117161)

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN

2019

DAFTAR ISI

TUGAS BESAR MANAJEMEN BASIS DATA TUNNING DATABASE	i
DAFTAR ISI	ii
BAB I STUDI LITERATUR.....	1
1.1. TUNNING : INDEXING.....	1
1.2. TUNNING : SETTING CONFIGURATION DBMS.....	1
BAB II DESKRIPSI PERCOBAAN	3
2.1. TUNNING INDEXING.....	3
2.2. TUNNING SETTING CONFIGURATION DBMS.....	3
BAB III HASIL DAN PEMBAHASAN	5
3.1. HASIL QUERY SEBELUM INDEXING.....	5
• Data Set 1 • Data Set 2	5
Data Set 2.....	7
3.3. KESIMPULAN.....	8
DAFTAR PUSTAKA	9

BAB I STUDI LITERATUR

1.1. TUNNING : INDEXING

Index adalah sebuah objek dalam sistem database yang dapat mempercepat proses pencarian (*query*) data. Saat database dibuat tanpa menggunakan index, maka kinerja server database dapat menurun secara drastis. Hal ini dikarenakan resource CPU banyak digunakan untuk pencarian data atau pengaksesan query SQL dengan metode table-scan. Index membuat pencarian data akan lebih cepat dan tidak banyak menghabiskan resource CPU.

Index merupakan objek struktur data yang tidak bergantung kepada struktur tabel. Setiap index terdiri dari nilai kolom dan penunjuk (atau ROWID) ke baris yang berisi nilai tersebut. Penunjuk tersebut secara langsung menunjuk ke baris yang tepat pada tabel, sehingga menghindari terjadinya full table-scan. Akan tetapi lebih banyak index pada tabel tidak berarti akan mempercepat query. Semakin banyak index pada suatu tabel menyebabkan kelambatan pemrosesan perintah-perintah DML (*Data Manipulation Language*), karena setiap terjadi perubahan data maka index juga harus disesuaikan.

Index tidak dapat dimodifikasi. Kita harus menghapusnya terlebih dahulu dan menciptakannya kembali. Kita dapat menghapus definisi index dari data dictionary dengan perintah DROP INDEX.

Contoh: DROP INDEX employee_last_name_idx;

1.2. TUNNING : SETTING CONFIGURATION DBMS

Database Tuning adalah sejumlah aktifitas yang dilakukan untuk memperbaiki atau meningkatkan kinerja atau performance sebuah database. Aktifitas tuning ini meliputi banyak aspek dari software hingga hardware, antara lain I/O Tuning, DBMS Tuning, Query Tuning, dan Database Maintenance. Masing-masing memiliki tekniknya sendiri-sendiri, dan membutuhkan skill yang mumpuni. Namun kita tetap bisa mempelajari teknik-teknik dasarnya. Dalam artikel ini, kita akan mencoba melakukan Query Tuning dengan bantuan

database index. Ini dilakukan untuk mengoptimalkan kinerja dari basis data. Database tuning tidak hanya meliputi tuning query, tapi juga pada rancangan basis data, pemilihan DBMS, sistem operasi dan CPU yang menjalankan DBMS. Tujuan dari database tuning yaitu memaksimalkan penggunaan sumber daya sistem untuk melakukan kerja secara efisien dan secepat mungkin. Kebanyakan sistem telah dirancang untuk bekerja secara efisien, tapi bisa dilakukan peningkatan kinerja MySQL secara signifikan. dengan mengkostumisasi pengaturan, konfigurasi database dan meng-tuning DBMS. Hasilnya, kinerja database MySQL lebih cepat dan produktifitas semakin meningkat.

MySQL secara default menggunakan konfigurasi moderate, dalam arti bahwa konfigurasi yang diberikan merupakan konfigurasi yang menyesuaikan dengan sistem server secara umum. Ada penyesuaian antara performa dengan stabilitas. Tidak diset performa tinggi agar tidak mempengaruhi stabilitas sistem.mysql-performance-tuning. Kita bisa dengan mudah melakukan ujicoba tuning sistem dengan cara membackup file konfigurasi standard dan kemudian mengubah beberapa opsi/parameter yang tersedia

BAB II DESKRIPSI PERCOBAAN

2.1. TUNNING INDEXING

Lakukan Tuning Query database dengan melakukan indexing pada beberapa filed yang anda pilih. Index pada data set 1 sampai data set 3 dilakukan pada table student dan pada table section.

```
MariaDB [tubes1]> create index section_idx on section(sec_id);
```

Pada table student indexing dilakukan pada kolom id dengan nama index adalah student_idx. Kemudian pada table section, indexing dilakukan pada kolom sec_id dengan nama index section_idx.

```
MariaDB [tubes1]> create index student_idx on student(id);
```

Hal berikut dilakukan pada data set satu sampai data set tiga

2.2. TUNNING SETTING CONFIGURATION DBMS

Database yang digunakan pada kasus ini adalah Mysql. Untuk pengaturan konfigurasi database manajemen system. Hal pertama yang harus dilakukan adalah mengatur ukuran file yang bisa diupload pada file C://xampp/mysql/bin/my.ini dengan mengganti ketiga parameter berikut :

- innodb_log_file_size=25M
- innodb_log_buffer_size=32M
- innodb_flush_log_at_trx_commit=1
- innodb_lock_wait_timeout=50

```

# Comment the following if you are using InnoDB tables
#skip-innodb
innodb_data_home_dir="C:/xampp/mysql/data
innodb_data_file_path=ibdata1:10M:autoextend
innodb_log_group_home_dir="C:/xampp/mysql/data
innodb_log_arch_dir="C:/xampp/mysql/data
## You can set innodb_buffer_pool_size up to 50 - 80 %
## of RAM but beware of setting memory usage too high
innodb_buffer_pool_size=2500M
## Set innodb_log_file_size to 25 % of buffer pool size
innodb_log_file_size=25M
innodb_log_buffer_size=32M
innodb_flush_log_at_trx_commit=1
innodb_lock_wait_timeout=50

## UTF 8 Settings
#init-connect='SET NAMES utf8'
#collation-set=utf8_unicode_ci
#character-set=utf8
#skip-character-set-client-handshake
#character-sets-dir="C:/xampp/mysql/share/charsets

```



BAB III HASIL DAN PEMBAHASAN

3.1. HASIL QUERY SEBELUM INDEXING.

Pada permasalahan yang didapat, saya hanya menggunakan 3 data yang digunakan sebagai percobaan tuning, adapun data yang digunakan adalah :

- advisor = 100, student = 100, section = 200,takes = 200 sebagai Q1
- advisor = 200, student = 200, section = 400,takes = 400 sebagai Q2
- advisor = 500, student = 500, section = 1000,takes = 1000 sebagai Q3

Query yang dijalankan untuk percobaan :

- SELECT * FROM student;
- SELECT * FROM student WHERE tot_cred > 30;
- SELECT `name`, department FROM student WHERE tot_cred > 30;
- SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id
- 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 Set 1

Query	Waktu
Q1	0.00078210
Q2	0.04373350
Q3	0.00216120
Q4	0.12805690
Q5	0.15726640

• Data Set 2

Query	Waktu
Q1	0.13768350
Q2	0.04900220
Q3	0.00190140
Q4	1.14076090
Q5	0.23884500

- **Data Set 3**

Query	Waktu
Q1	0.04472140
Q2	0.01244250

Q3	0.00183460
Q4	0.07187720
Q5	0.15822370

The image shows two screenshots of a MySQL command-line interface (CLI) window. The window title is "XAMPP for Windows - mysql -u root". The interface displays a list of queries and their execution times, formatted as a table with columns: Query_ID, Duration, and Query.

Top Screenshot: Shows queries 4 through 18. Query 18 is highlighted, showing a complex JOIN query. The output indicates 15 rows in set (0.447 sec).

Bottom Screenshot: Shows queries 15 through 29. Query 29 is highlighted, showing a complex JOIN query. The output indicates 15 rows in set (0.000 sec).

```

XAMPP for Windows - mysql -u root
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 4 | 0.04373350 | select *from student where tot_cred>30 |
| 5 | 0.00216120 | select name,dept_name from student where tot_cred>30 |
| 6 | 0.12805690 | select *from takes join student on takes.id=student.id join section on takes.course_id=section.course_id |
| 7 | 0.15726640 | SELECT student.'name',student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,cour |
| 8 | 5.20339500 | se.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.cou |
| 9 | 0.00142380 | drop database tubes2 |
| 10 | 15.50643440 | SELECT DATABASE() |
| 11 | 0.09969740 | drop database tubes2 |
| 12 | 0.15464180 | SELECT DATABASE() |
| 13 | 0.00074070 | create database tubes2 |
| 14 | 0.13768350 | SELECT DATABASE() |
| 15 | 0.04900220 | select *from student |
| 16 | 0.00190140 | select *from student where tot_cred>30 |
| 17 | 1.14076090 | select name,dept_name from student where tot_cred>30 |
| 18 | 1.14076090 | select *from takes join student on takes.id=student.id join section on takes.course_id=section.course_id |
| 19 | 0.23884500 | SELECT student.'name',student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,cour |
| 20 | 0.23884500 | se.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.cou |
+-----+-----+-----+
15 rows in set (0.447 sec)

MariaDB [tubes2]>

```

```

XAMPP for Windows - mysql -u root
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 15 | 0.04900220 | select *from student where tot_cred>30 |
| 16 | 0.00190140 | select name,dept_name from student where tot_cred>30 |
| 17 | 1.14076090 | select *from takes join student on takes.id=student.id join section on takes.course_id=section.course_id |
| 18 | 0.23884500 | SELECT student.'name',student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,cours |
| 19 | 0.16039780 | e.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.cou |
| 20 | 0.16039780 | create database tubes3 |
| 21 | 0.00037490 | SELECT DATABASE() |
| 22 | 4.63905500 | drop database tubes3 |
| 23 | 0.00112900 | SELECT DATABASE() |
| 24 | 0.00170230 | create database tubes3 |
| 25 | 0.00031910 | SELECT DATABASE() |
| 26 | 0.04472140 | select *from student |
| 27 | 0.01244250 | select *from student where tot_cred>30 |
| 28 | 0.00183460 | select name,dept_name from student where tot_cred>30 |
| 29 | 0.07187720 | select *from takes join student on takes.id=student.id join section on takes.course_id=section.course_id |
| 30 | 0.15822370 | SELECT student.'name',student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,cours |
| 31 | 0.15822370 | e.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.cou |
+-----+-----+-----+
15 rows in set (0.000 sec)

MariaDB [tubes3]>

```


3.2. HASIL QUERY SETELAH INDEXING

Pada permasalahan yang didapat, saya hanya menggunakan 3 data yang digunakan sebagai percobaan tuning, adapun data yang digunakan adalah :

- advisor = 100, student = 100, section = 200,takes = 200 sebagai Q1
- advisor = 200, student = 200, section = 400,takes = 400 sebagai Q2
- advisor = 500, student = 500, section = 1000,takes = 1000 sebagai Q3

Sedangkan query yang dijalankan adalah :

- SELECT * FROM student;
- SELECT * FROM student WHERE tot_cred > 30;
- SELECT `name`, department FROM student WHERE tot_cred > 30;
- SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id
- 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 Set 2

Data Set 1

Query	Waktu
Q1	0.00076720
Q2	0.00076950
Q3	0.00069920
Q4	0.00260760
Q5	0.00255510

Query	Waktu
Q1	0.84778850
Q2	0.00081550
Q3	0.00071780
Q4	1.00071780
Q5	0.55329740

Data Set 3

Query	Waktu
Q1	0.49915200

Q2	0.00168410
Q3	0.00088310
Q4	1.02908130
Q5	0.78525050

Dari sejumlah query yang dijalankan dari ketiga data set tersebut, semakin banyak data yang dihasilkan semakin lama response time yang dibutuhkan dalam memproses query sql. Setelah dilakukan indexing pada kolom id table student dan kolom sec_id pada table section, hasilnya query yang dijalankan lebih cepat dari sebelum dilakukan indexing.

Mak dari sejumlah query yang dijalankan dapat terlihat jika semakin banyak data yang dihasilkan semakin lama response time yang dibutuhkan dalam memproses query sql. Setelah indexing dilakukan output yang dihasilkan jauh lebih cepat dari query yang sebelum di indexing

3.3. KESIMPULAN

- Database tuning merupakan sejumlah aktifitas yang dilakukan untuk mengoptimalkan kinerja dari basisdata. Database tuning tidak hanya meliputi tuning query, tapi juga pada rancangan basis data, pemilihan DBMS, sistem operasi dan CPU yang menjalankan DBMS, tujuan dari database tuning yaitu memaksimalkan penggunaan sumber daya sistem untuk melakukan kerja secara efisien dan secepat mungkin
- Hasil tuning tidak akan terlihat, jika query yang dijalankan sangat sederhana.
- Tuning dengan indexing tidak selalu menghasilkan waktu yang cepat dalam eksekusi query. Hal ini sesuai dengan studi literatur tentang indexing. Indexing memungkinkan DBMS mengakses record-record secara lebih cepat, namun dikarenakan query yang sederhana dan data yang sedikit, sehingga record-record yang dilakukan indexing tidaklah bervariasi.
- Tuning dengan config DBMS selalu berhasil membuat query dapat dieksekusi lebih cepat. Hal ini sesuai dengan studi literatur tentang config DBMS. Config DBMS memungkinkan DBMS untuk mengeksekusi query dengan cepat karena pada settingan DBMS terdapat buffer dan juga cache yang dapat mempercepat eksekusi query. Semakin besar tempat penyimpanan tersebut, maka semakin baik pula hasilnya.

DAFTAR PUSTAKA

1. <https://www.teknojempol.com/2017/12/mysql-database-tuning-bagian-1.html>
2. <http://rani-irsan.blogspot.com/2014/06/934-mysql-create-index.html>
3. <https://www.i-3.co.id/2016/10/07/index-pada-database/>
4. <https://www.teknojempol.com/2017/12/mysql-database-tuning-bagian-1.html>chord