# TUGAS BESAR MANAJEMEN BASIS DATA TUNNING DATABASE



DOSEN PENGAMPU: Ahmad Luky Ramdani, S.Kom., M.Kom.

DISUSUN OLEH: Agus Michael Pangihutan Sianipar (14117066)

PROGRAM STUDI
TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA
2019

# **DAFTAR ISI**

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-

membuat pencarian data akan lebih cepat Index dan tidak banyak

menghabiskan resource CPU.

Index merupakan objek struktur data tersendiri 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 merupakan sejumlah aktifitas yang 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 yang diberikan merupakan konifgurasi 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.

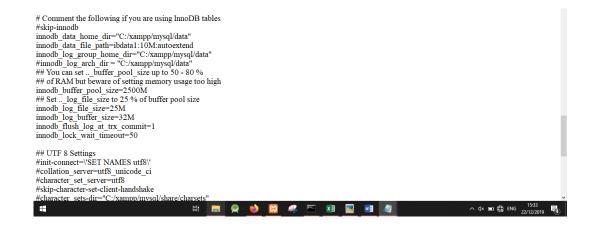
```
MariaDB [tubes1]> create index student_idx on student(<u>i</u>d);
```

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



### BAB III HASIL DAN PEMBAHASAN

#### 3.1. HASIL QUERY SEBELUM INDEXING.

Pada kasus kali ini, hanya ada tiga data set yang digunakan karena dari segi spesifikasi laptop yang tidak mendukung banyak data, sehingga data set 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 1

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

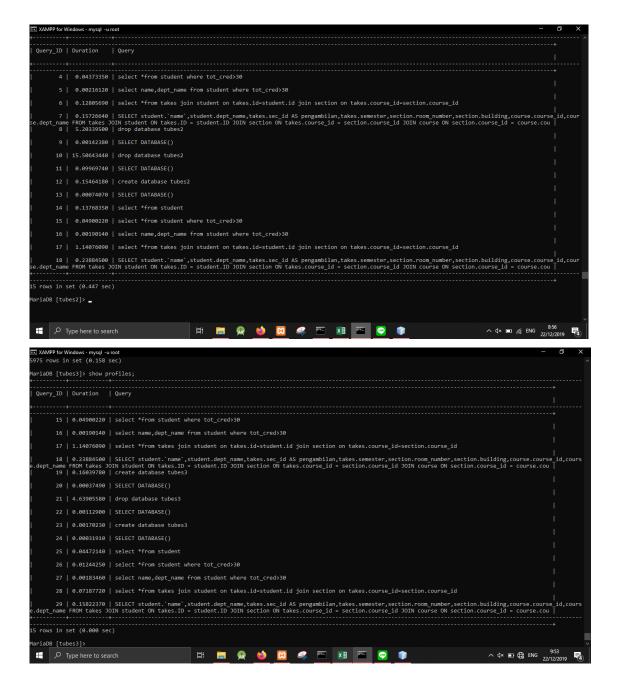
#### • Data Set 2

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



#### 3.2. HASIL QUERY SETELAH INDEXING

Pada kasus kali ini, hanya ada tiga data set yang digunakan karena dari segi spesifikasi laptop yang tidak mendukung banyak data, sehingga data set 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 1

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

Data Set 2

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.

#### 3.3. KESIMPULAN

Database tuning merupakan sejumlah aktifitas yang 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.

Dari percobaan yang telah dilakukan dengan menggunakan tiga buah set data yang disimpan kedalam tiga buah database berbeda dapat diketahui bahwa waktu pengeksekusian sebuah query terbukti lebih cepat di proses ketika dilakukan indexing. Indexing terbukti dapat mempermudah dan mempercepat proses pencarian namun apabila kita melihat pada data set 3 yang tidak mempercepat proses eksekusi query maka indexing tak selalu dapat membantu namun dapat mempercepat pencarian dan eksekusi query.

## **DAFTAR PUSTAKA**

- 1. https://www.teknojempol.com/2017/12/mysql-database-tuning-bagian-1.html
- 2. <a href="http://rani-irsan.blogspot.com/2014/06/934-mysql-create-index.html">http://rani-irsan.blogspot.com/2014/06/934-mysql-create-index.html</a>