

# **TUGAS BESAR MANAJEMEN BASIS DATA**

## **Tuning Database Management System**



### **Dosen Pengampu :**

Ahmad Luky Ramdani, S.Kom., M.Kom

### **Disusun Oleh :**

Putri Maulidya Yasshiro      14117022

**PROGRAM STUDI TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI SUMATERA  
2019**

## DAFTAR ISI

COVER.....	1
DAFTAR ISI.....	2
DAFTAR GAMBAR.....	3
DAFTAR TABEL.....	4
BAB I.....	5
Studi Literatur.....	5
1.1 Tunning Indexing.....	5
1.2 Tunning Setting Configuration DBMS.....	5
BAB II.....	6
Deskripsi Percobaan.....	6
2.1 Tunning Indexing.....	6
BAB III.....	7
Hasil dan Pembahasan.....	7
3.1 Hasil Pengujian dan Pembahasan.....	7
DAFTAR PUSTAKA.....	9
LAMPIRAN.....	10

## DAFTAR GAMBAR

Gambar 1 Gambar Indexing Data 1.....	10
Gambar 2 Gambar Indexing Data 2.....	10
Gambar 3 Gambar Indexing Data 3.....	11

## DAFTAR TABEL

Tabel 1 Data Sampel.....	7
Tabel 2 Hasil data 1.....	7
Tabel 3 Hasil data 2.....	8
Tabel 4 Hasil data 3.....	8

# **BAB I**

## **Studi Literatur**

### **1.1 Tunning Indexing**

Tunning database adalah untuk meningkatkan performance dari database tersebut, sehingga respon dari database server bisa lebih cepat. Pada dasarnya tunning tidak akan terasa jika pada aplikasi database yang kecil, namun pada database dengan data yang banyak dan aktivitas pengambilan data yang cukup padat akan sangat terasa. Parameter yang bisa diubah pada tunning database antara lain :

- Config Start up database
- Mempercepat query
- Indexing
- Strukur tabel

Pada pengujian ini, kita akan melakukan tunning poin ke tiga, yaitu tunning dengan *indexing*. Ada lima hal yang mempengaruhi performace database, antara lain :

1. Workload (beban kerja)
2. Throughput (kapasitas komputer dalam mengolah data)
3. Resource (hardware)
4. Optimazation (optimalisasi yang berhubungan dengan query)
5. Contention (isi database)

*Indexing* adalah sebuah data struktur yang menyimpan nilai spesifik sebuah kolom pada sebuah tabel. Fungsi *indexing* adalah untuk membantu mempercepat proses eksekusi sebuah query ke sebuah database yang sudah berisi banyak data.

### **1.2 Tunning Setting Configuration DBMS**

Untuk melakukan setting configuration DBMS kita dapat beralih menuju file my.ini yang terletak di xampp. Apabila kita membuka xampp.exe pilih menu config pada mysql nya. Kemudian lakukan perubahan pada nilai-nilai yang ada, pada percobaan ini hanya tiga yang diubah yaitu sebagai berikut:

- Innodb\_buffer\_pool\_size=3M
- Innodb\_log\_file\_size=15M
- Innodb\_log\_buffer\_size=30M

## **BAB II**

### **Deskripsi Percobaan**

#### **2.1 Tuning Indexing**

Kita ingat bahwa percobaan ini memerintahkan kita untuk melakukan Query SQL sebanyak 5 Query. Kita anggap saja Q1 sampai Q5. Perhatikan pembagian Query yang dimaksud sebagai berikut :

- Q1 : SELECT \*FROM Student;
- Q2 : SELECT \*FROM Student WHERE tot\_cred>30;
- Q3 : SELECT 'name', dept\_name FROM student WHERE tot\_cred>30;
- Q4 : SELECT \*FROM takes JOIN student ON takes.ID=student.ID JOIN section JOIN section ON takes.course\_id=section.course\_id
- Q5 : 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;

## BAB III

### Hasil dan Pembahasan

#### 3.1 Hasil Pengujian dan Pembahasan

Data yang diujikan ada sebanyak 7 data. Namun pada kesempatan percobaan kali ini, Saya hanya mengujikan 3 data saja. Untuk menguji ketiga data tersebut kita dapat membuat masing-masing databasenya. Pengubahan nilai yang sesuai data yang ada dilakukan pada tableGen.java di struktur void main programnya. Dari data-data tersebut akan dilakukan pengujian Query dari Q1 sampai Q5 pada masing-masing database. Berikut ini uraian data yang diketahui :

*Table 1 Data Sampel*

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=20000, takes=20000
Advisor=1800, student=1800, section=180000, takes=180000
Advisor=10000, student=10000, section=30000000, takes=30000000

Berikut hasil indexing tunning Q1 sampai Q5 dalam satuan waktu second:

1. Data 1, diketahui Advisor=100, student=100, section=200, takes=200

*Table 2 Hasil data 1*

Query	Sebelum Tuning (s)	Setelah Tuning (s)
Q1	0.001	0.001
Q2	0.002	0.001
Q3	0.001	0.001
Q4	0.003	0.003
Q5	0.003	0.003

Dari data di atas terlihat bahwa perbedaan waktu akses antara sebelum dan setelah tuning tidaklah jauh berbeda. Hal ini dikarenakan database masih melibatkan jumlah data yang sedikit.

2. Data 2, diketahui Advisor=200, student=200, section=400, takes=400

Table 3 Hasil data 2

Query	Sebelum Tunning (s)	Setelah Tunning (s)
Q1	0.001	0.001
Q2	0.001	0.001
Q3	0.001	0.001
Q4	0.001	0.003
Q5	0.003	0.003

Dari data di atas terlihat bahwa perbedaan waktu akses antara sebelum dan setelah tuning juga tidaklah jauh berbeda. Hal ini dikarenakan database masih melibatkan jumlah data yang tidak terlalu banyak.

3. Data 3, diketahui Advisor=500, student=500, section=1000, takes=1000

Table 4 Hasil data 3

Query	Sebelum Tunning (s)	Setelah Tunning (s)
Q1	0.001	0.001
Q2	0.001	0.001
Q3	0.003	0.001
Q4	0.299	0.005
Q5	0.431	0.010

Dari data di atas terlihat bahwa perbedaan waktu akses antara sebelum dan setelah tuning tidaklah jauh berbeda. Hal ini dikarenakan database masih melibatkan jumlah data yang sedikit.



## DAFTAR PUSTAKA

[https://pojokprogramer.net/Performance\\_tunning\\_sederhana\\_MySQL](https://pojokprogramer.net/Performance_tunning_sederhana_MySQL)

[https://mti.binus.ac.id/Seberapa\\_pentingkah\\_performance\\_dan\\_tuning\\_pada\\_SQL\\_database](https://mti.binus.ac.id/Seberapa_pentingkah_performance_dan_tuning_pada_SQL_database)

[https://id.scribd.com/tuning\\_database](https://id.scribd.com/tuning_database)

[https://medium.com/database\\_optimization#1:indexing\\_dot\\_lab\\_medium](https://medium.com/database_optimization#1:indexing_dot_lab_medium)

[https://www.slideshare.net/penjelasan\\_indexing\\_database\\_1411510850](https://www.slideshare.net/penjelasan_indexing_database_1411510850)

## LAMPIRAN

```
MariaDB [dbms-1]> create index indexing_student on student(tot_cred);
Query OK, 0 rows affected (0.344 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [dbms-1]> create index indexing_student_name on student(name);
Query OK, 0 rows affected (0.390 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [dbms-1]> create index indexing_dept_name on student(dept_name);
Query OK, 0 rows affected (0.467 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [dbms-1]> create index indexing_course on course(dept_name);
Query OK, 0 rows affected (1.564 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

*Gambar 2 Gambar Indexing Data 1*

```
MariaDB [dbms-2]> create index indexing_student on student(tot_cred);
Query OK, 0 rows affected (0.281 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [dbms-2]> create index indexing_student_name on student(name);
Query OK, 0 rows affected (0.300 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [dbms-2]> create index indexing_dept_name on student(dept_name);
Query OK, 0 rows affected (0.401 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [dbms-2]> create index indexing_course on course(dept_name);
Query OK, 0 rows affected (0.413 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

*Gambar 3 Gambar Indexing Data 2*

```
MariaDB [dbms-3]> create index indexing_student on student(tot_cred);  
Query OK, 0 rows affected (0.392 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
MariaDB [dbms-3]> create index indexing_student_name on student(name);  
Query OK, 0 rows affected (0.299 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
MariaDB [dbms-3]> create index indexing_course on course(dept_name);  
Query OK, 0 rows affected (0.431 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

*Gambar 4 Gambar Indexing Data 3*