

**TUGAS BESAR**  
**MANAJEMEN BASIS DATA**  
**“Tuning Database System”**



**Disusun oleh :**

**Mia Audina**

**14117032**

**Dosen Pengampu:**

Arief Ichwani S.Kom, M.Cs

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

**2019**

## DAFTAR ISI

DAFTAR GAMBAR .....	ii
BAB I .....	1
STUDI LITERATUR.....	1
1.1 Index Tuning.....	1
1.2 Setting Database Configuration .....	1
BAB II .....	2
DESKRIPSI PERCOBAAN .....	2
BAB III HASIL DAN PEMBAHASAN .....	3
3.1 Hasil Sebelum Query Tuning .....	3
3.2 Hasil Setelah Query Tuning .....	3
3.3 Tabel Hasil Percobaan .....	4
BAB IV KESIMPULAN .....	5

## DAFTAR GAMBAR

Gambar 1. Pengaturan cache pada Database .....	1
Gambar 2. Hasil data set 1 .....	3
Gambar 3. Hasil data set 2 .....	3
Gambar 4. Hasil data set 3 .....	3
Gambar 5. Hasil data set 1 .....	3
Gambar 6. Hasil data set 2 .....	3

# BAB I

## STUDI LITERATUR

### 1.1 Index Tuning

*Index Tuning* merupakan bagian penyetelan basis data untuk memilih dan membuat indeks. Tujuan *Index Tuning* adalah untuk mengurangi waktu pemrosesan *query*. Tanpa indeks, mesin SQL Server menemukan data dengan cara mencari di setiap record, dapat dibayangkan jika record mencapai ribuan bahkan hingga miliaran. Dengan menggunakan indeks, transaksi dapat dilakukan dalam waktu yang jauh lebih singkat. (Allen, 2004)

### 1.2 Setting Database Configuration

*Setting Database Configuration* merupakan kegiatan yang dilakukan untuk melakukan peningkatan performa basis data dengan mengubah pengaturan dalam server DB, parameter yang sering dijadikan acuan adalah query cache size, maksimal file upload, maksimal ukuran post, dan memory limit. Pengaturan ini dapat dijalankan melalui command line dan pengaturan pada file php.ini.

```
MariaDB [query-tuning-1] > SET GLOBAL query_cache_size = 40000;
Query OK, 0 rows affected, 2 warnings (0.000 sec)

MariaDB [query-tuning-1] > SET GLOBAL query_cache_size = 41984;
Query OK, 0 rows affected, 1 warning (0.000 sec)







MariaDB [query-tuning-1] > SHOW VARIABLES LIKE 'query_cache_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| query_cache_size | 0     |
+-----+-----+
1 row in set (0.001 sec)
```

Gambar 1. Pengaturan cache pada Database

## BAB II

### DESKRIPSI PERCOBAAN

Pada percobaan ini menggunakan data set yang generate dari program java yang tersedia pada deskripsi tugas yang telah diberikan. Dengan melakukan compile file tableGen.java, akan menghasilkan file all.sql, all2.sql, all3.sql, all4.sql dan all5.sql sebagai data set yang akan dilakukan uji coba.

Name	Date modified	Type	Size
 .DS_Store	05/12/2019 9:26	DS_STORE File	7 KB
 all	22/12/2019 20:46	SQL File	63 KB
 all2	22/12/2019 13:55	SQL File	101 KB
 all3	22/12/2019 14:06	SQL File	214 KB
 all4	22/12/2019 14:22	SQL File	144 KB
 all5	22/12/2019 14:25	SQL File	144 KB

Untuk pengujian pertama dilakukan dengan query data dengan perintah sql sebagai berikut :

1. `SELECT * FROM student`
2. `SELECT * FROM student WHERE tot_cred > 30;`
3. `SELECT dept_name 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_idAS  
pengambilan,takes.semester,section.room_number,section.building,course.course_id,c  
ourse.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;`

Perintah sql tersebut akan dilakukan untuk setiap data set. Untuk setiap data set, pertama akan dilakukan uji coba secara langsung tanpa menggunakan tuning database. Setelah itu akan dilakukan tuning pada data set dan akan dilakukan uji coba lagi terhadap data set tersebut. Setelah itu dapat dilakukan perbandingan antara data set sebelum tuning, dan setelah dilakukan tuning.

## BAB III

### HASIL DAN PEMBAHASAN

Berikut merupakan hasil setelah dilakukan uji coba.

#### 3.1 Hasil Sebelum Query Tuning

```
MariaDB [mial]> show profiles;
```

Query_ID	Duration	Query
1	0.00043618	select * from student
2	0.00051444	select * from student where tot_cred > 30
3	0.00034638	select dept_name from student where tot_cred > 30
4	0.00316574	select * from takes join student on takes.ID = student.ID join section on takes.c
5	0.00513411	SELECT student.'name',student.dept_name,takes.sec_id AS pengambilan,takes.semeste

5 rows in set (0.000 sec)

**Gambar 2. Hasil data set 1**

8	0.00065556	select * from student
9	0.00112637	select * from student where tot_cred > 30
10	0.00051615	select dept_name from student where tot_cred > 30
11	0.00515587	select * from takes join student on takes.ID = student.ID join section on takes.course_id = section.c
12	0.00420830	SELECT student.'name',student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_numbe

12 rows in set (0.000 sec)

**Gambar 3. Hasil data set 2**

15	0.00115588	select * from student
16	0.00090957	select * from student where tot_cred > 30
17	0.00152706	select dept_name from student where tot_cred > 30
18	0.02606655	select * from takes join student on takes.ID = student.ID join section on takes.course_id = sectio
19	0.01209677	SELECT student.'name',student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_nu

15 rows in set (0.000 sec)

**Gambar 4. Hasil data set 3**

#### 3.2 Hasil Setelah Query Tuning

23	0.00050161	select * from student
24	0.00048664	select * from student where tot_cred > 30
25	0.00041737	select dept_name from student where tot_cred > 30
26	0.00247212	select * from takes join student on takes.ID = student.ID join section on takes.course
27	0.00283732	SELECT student.'name',student.dept_name,takes.sec_id AS pengambilan,takes.semester,sec

15 rows in set (0.000 sec)

**Gambar 5. Hasil data set 1**

31	0.00066282	select * from student
32	0.00088861	select * from student where tot_cred > 30
33	0.00046954	select dept_name from student where tot_cred > 30
34	0.00485188	select * from takes join student on takes.ID = student.ID join section on takes.course_id =
35	0.00599023	SELECT student.'name',student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.r

5 rows in set (0.000 sec)

**Gambar 6. Hasil data set 2**

39	0.00144368	select * from student
40	0.00152706	select * from student where tot_cred > 30
41	0.00150782	select dept_name from student where tot_cred > 30
42	0.02087941	select * from takes join student on takes.ID = student.ID join section on takes.cc
43	0.01206854	SELECT student.'name',student.dept_name,takes.sec_id AS pengambilan,takes.semester

15 rows in set (0.000 sec)

### 3.3 Tabel Hasil Percobaan

Data	Waktu Sebelum Tuning (s)					Waktu Sesudah Tuning (s)				
	Q1	Q2	Q3	Q4	Q5	Q1	Q2	Q3	Q4	Q5
advisor = 100, student = 100, section = 200,takes = 200	0.00043618	0.00051444	0.00034638	0.00316574	0.00513411	0.00050161	0.00048664	0.00041737	0.00247212	0.00283732
advisor = 200, student = 200, section = 400,takes = 400	0.00065556	0.00112637	0.00051615	0.00515507	0.00420830	0.00066282	0.00088861	0.00046954	0.00485188	0.00599023
advisor = 500, student = 500, section = 1000,takes = 1000	0.00115588	0.00090957	0.00152706	0.02606655	0.01209677	0.00144368	0.00152706	0.00150782	0.02087941	0.01206854
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 = 3000000,takes = 30000000										

## **BAB IV**

### **KESIMPULAN**

Tuning database dapat meningkatkan performa pada database. Pengolahan data menggunakan indexing dapat mempercepat proses transaksi data.