

**Tugas Besar Manajemen Basis Data**  
**DATABASE TUNNING**



**Dosen Pengampu :**

Arief Ichwani S.Kom, M.Cs

**Disusun oleh:**

Genta Ari Anggoro

14117078

**Program Studi Teknik Informatika**  
**Jurusan Teknologi Produksi dan Informasi - Institut Teknologi Sumatera**  
**Jl. Terusan Ryacudu, Gedung D215**  
**2019**

## DAFTAR ISI

DAFTAR ISI.....	ii
BAB I.....	1
STUDI LITERATUR.....	1
1.1    Tunning : Indexing .....	1
1.2    Tunning : Setting DBMS .....	2
BAB II.....	2
DESKRIPSI PERCOBAAN .....	2
2.1    Tunning Indexing .....	2
BAB III .....	3
HASIL DAN PEMBAHASAN.....	3
3.1    Hasil Pecobaan .....	3
3.2    Data Hasil .....	5
3.3    Grafik Data.....	6
3,4    Deskripsi.....	7
DAFTAR PUSTAKA .....	7

# BAB I

## STUDI LITERATUR

Menurut Rich Schiesser dalam bukunya yang berjudul IT System Mangement, *Performance and Tuning* adalah sebuah metodologi untuk memaksimalkan *throughput* dan meminimalkan *response time* untuk proses-proses seperti transaksi *online*, aktivitas-aktivitas internet dan *batch job*. Dengan kata lain, *Performance and Tuning* adalah sebuah upaya untuk memaksimalkan infrastruktur baik *software* ataupun *hardware* yang ada agar dapat berfungsi dengan optimal. Performance and Tuning sebenarnya memiliki dua tugas yang dijalankan berdampingan, *performance* bertujuan untuk memonitor kinerja dari infrastruktur. Hasil monitor ini akan ditindaklanjuti dengan *tuning* sebagai upaya untuk mengoptimalkan kinerja.

Beberapa area infrastruktur yang dapat diukur kinerjanya untuk dilakukan tuning adalah:

1. *Server*
2. *Disk Storage*
3. *Databases*
4. *Networks*
5. *Desktop Computers*

### **1.1 Tuning : Indexing**

Pada database, index merupakan sebuah struktur data yang berisi kumpulan *keys* beserta referensinya ke *actual data* di table. Tujuannya untuk mempercepat proses penentuan lokasi data tanpa melakukan pencarian secara penuh ke seluruh data (full scan).

#### **A. Benefit**

Fitur ini bisa menjawab 3 problem yang disampaikan sebelumnya dengan benefit-benefit berikut :

##### **1. Sorted & Less**

Jumlah data pada index jauh lebih kecil daripada data aslinya (tergantung kondisi datanya). Dan data-data ini juga tersimpan secara berurutan dalam bentuk struktur apapun (bisa *linear list*, *tree*, *dsb*). Tentu ini menjawab *problem #1 & problem #2*.

Dengan index, pencarian dimulai dengan menelusuri data pada index (yang jauh lebih kecil & teratur), kemudian index akan memberikan referensi ke posisi data yang asli. Dari posisi tersebut, pencarian pada *dataset* asli akan menjadi lebih singkat.

##### **2. (Mostly) Stored in Memory (RAM)**

Dengan ukuran *dataset* yang jauh lebih kecil, index besar kemungkinan dapat disimpan di RAM, dimana cost untuk baca/tulis pada RAM jauh lebih kecil daripada pada Harddisk. Sehingga tidak bisa dipungkiri proses pencarian *keys* pada index sangatlah cepat.

## 1.2 Tuning : Setting DBMS

MySQL dan MariaDB menggunakan InnoDB sebagai mesin penyimpanan default.

InnoDB menyediakan pendekatan yang lebih fleksibel dan setiap informasi basis data disimpan dalam file data .ibd . Setiap file **.ibd** mewakili tablespace sendiri. Dengan begitu operasi database seperti " **TRUNCATE** " dapat diselesaikan lebih cepat dan Anda juga dapat mengklaim kembali ruang yang tidak digunakan saat menjatuhkan atau memotong tabel database.

Manfaat lain dari konfigurasi ini adalah kenyataan bahwa Anda dapat menyimpan beberapa tabel database di perangkat penyimpanan yang terpisah. Ini dapat sangat meningkatkan beban **I / O** pada disk Anda. .

### Meningkatkan ukuran kumpulan buffer InnoDB

Mesin **InnoDB** memiliki kolam penyangga yang digunakan untuk menyimpan data dan indeks dalam memori. Ini tentu saja akan membantu permintaan MySQL / MariaDB Anda dieksekusi secara signifikan lebih cepat. Memilih ukuran yang tepat di sini membutuhkan beberapa keputusan yang sangat penting dan pengetahuan yang baik tentang konsumsi memori sistem Anda.

**Aturan** : Anda harus mengatur **ukuran buffer pool InnoDB = 80% RAM**

### Atur Koneksi MySQL Max

Arahan **max\_connections** memberi tahu server Anda berapa banyak koneksi konkuren diizinkan. Koneksi dibuka hanya untuk saat query MySQL dieksekusi - setelah itu ditutup dan koneksi baru dapat menggantikannya.

Pengaturan default untuk max\_connections adalah 100. Anda dapat melihat apa pengaturan saat ini dengan menjalankan perintah SQL berikut dari perintah:

## BAB II

### DESKRIPSI PERCOBAAN

#### 2.1 Tuning Indexing

Pada percobaan ini akan menggunakan data sebagai berikut :

Percobaan 1	advisor = 100, student = 100, section = 200,takes = 200
Percobaan 2	advisor = 200, student = 200, section = 400,takes = 400
Percobaan 3	advisor = 500, student = 500, section = 1000,takes = 1000

## 1. Membuat Index

```
CREATE INDEX IDX_STUDENT ON student(name,tot_cred) USING BTREE;  
CREATE INDEX IDX_takes ON takes(grade) USING BTREE;  
CREATE INDEX IDX_department ON department(building,budget) USING BTREE;
```

Disini saya membuat index pada tabel student, takes, dan department. Karena kita akan banyak menggunakan tabel dan field di atas untuk eksekusi querynya.

Lalu menjalankan Query :

1. SELECT \* FROM student;
2. SELECT \* FROM student WHERE tot\_cred > 30;
3. SELECT `name`, `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\_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 Percobaan

##### 1. Percobaan 1

advisor = 100, student = 100, section = 200, takes = 200

- Waktu eksekusi sebelum tuning

Query_ID	Duration	Query
1	0.70760216	select * from student
2	1.54762648	SELECT * FROM student WHERE tot_cred > 30
3	0.07753976	SELECT `name`, `dept_name` FROM student WHERE tot_cred > 30
4	0.41635444	SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id
5	0.08501264	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
5 rows in set (0.00 sec)		

- Waktu eksekusi sesudah tuning

```

MariaDB [ddl-mysql]> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.00435598 | SELECT * FROM student |
| 2 | 0.00127492 | SELECT * FROM student WHERE tot_cred > 30 |
| 3 | 0.00080643 | SELECT 'name', 'dept_name' FROM student WHERE tot_cred > 30 |
| 4 | 0.00662613 | SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id |
| 5 | 0.00896541 | 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 |
+-----+-----+-----+
5 rows in set (0.05 sec)

```

## 2. Percobaan 2

advisor = 200, student = 200, section = 400, takes = 400

- Waktu eksekusi sebelum tuning

```

MariaDB [mbd2]> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.06868060 | SELECT * FROM student |
| 2 | 0.03806202 | SELECT * FROM student WHERE tot_cred > 30 |
| 3 | 0.00144645 | SELECT 'name', 'dept_name' FROM student WHERE tot_cred > 30 |
| 4 | 0.10359115 | SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id |
| 5 | 0.04973343 | 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 |
+-----+-----+-----+
5 rows in set (0.04 sec)

```

- Waktu eksekusi sesudah tuning

```

+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.00573586 | SELECT * FROM student |
| 2 | 0.00106179 | SELECT * FROM student WHERE tot_cred > 30 |
| 3 | 0.00088515 | SELECT 'name', 'dept_name' FROM student WHERE tot_cred > 30 |
| 4 | 0.01738936 | SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id |
| 5 | 0.01157413 | 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 |
+-----+-----+-----+
5 rows in set (0.00 sec)

```

## 3. Percobaan 3

advisor = 500, student = 500, section = 1000, takes = 1000

- Waktu eksekusi sebelum tuning

```

MariaDB [mbd3]> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.14566546 | SELECT * FROM student |
| 2 | 0.00175430 | SELECT * FROM student WHERE tot_cred > 30 |
| 3 | 0.36638357 | SELECT `name`, `dept_name` FROM student WHERE tot_cred > 30 |
| 4 | 0.65585139 | SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id |
| 5 | 0.62519761 | 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 |
+-----+-----+-----+
5 rows in set (0.00 sec)

```

- Waktu eksekusi sesudah tuning

```

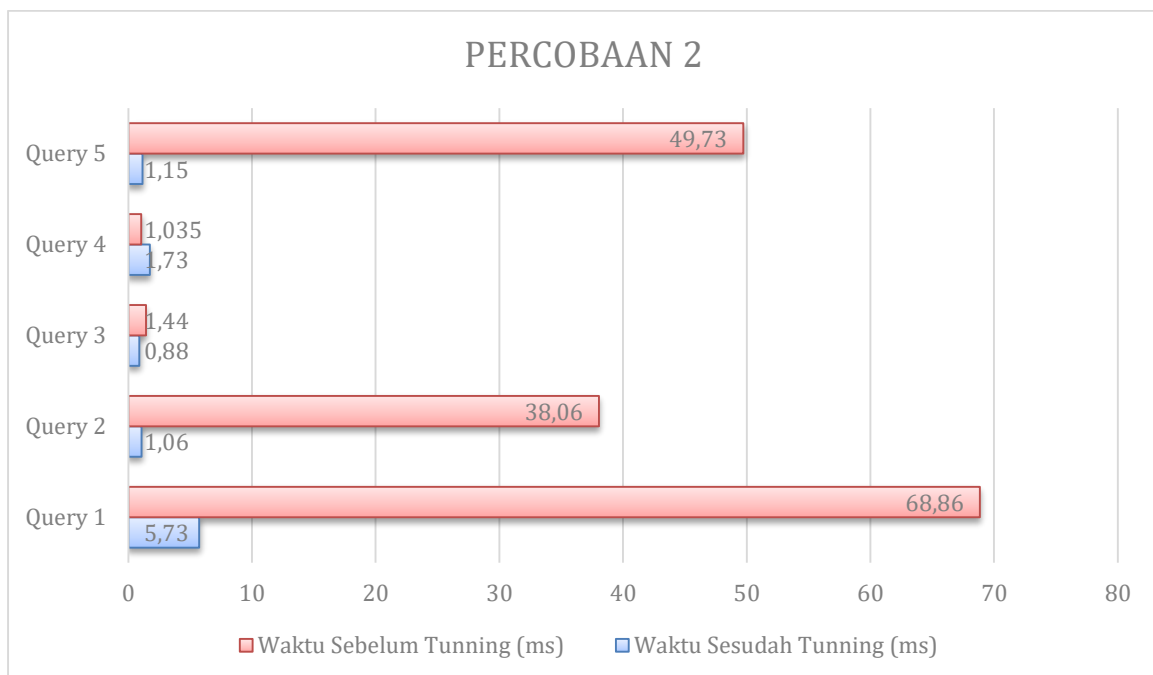
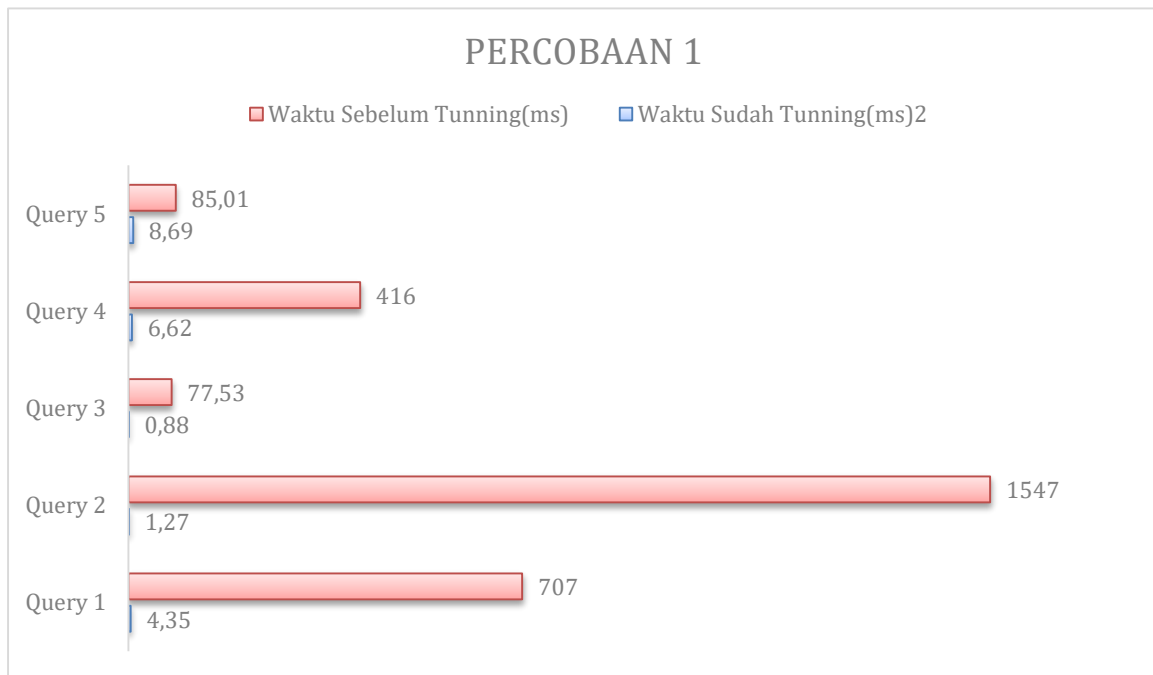
MariaDB [mbd3]> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.11720119 | SELECT * FROM student |
| 2 | 0.00229447 | SELECT * FROM student WHERE tot_cred > 30 |
| 3 | 0.00160901 | SELECT `name`, `dept_name` FROM student WHERE tot_cred > 30 |
| 4 | 0.47132196 | SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id |
| 5 | 5.38306826 | 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 |
+-----+-----+-----+

```

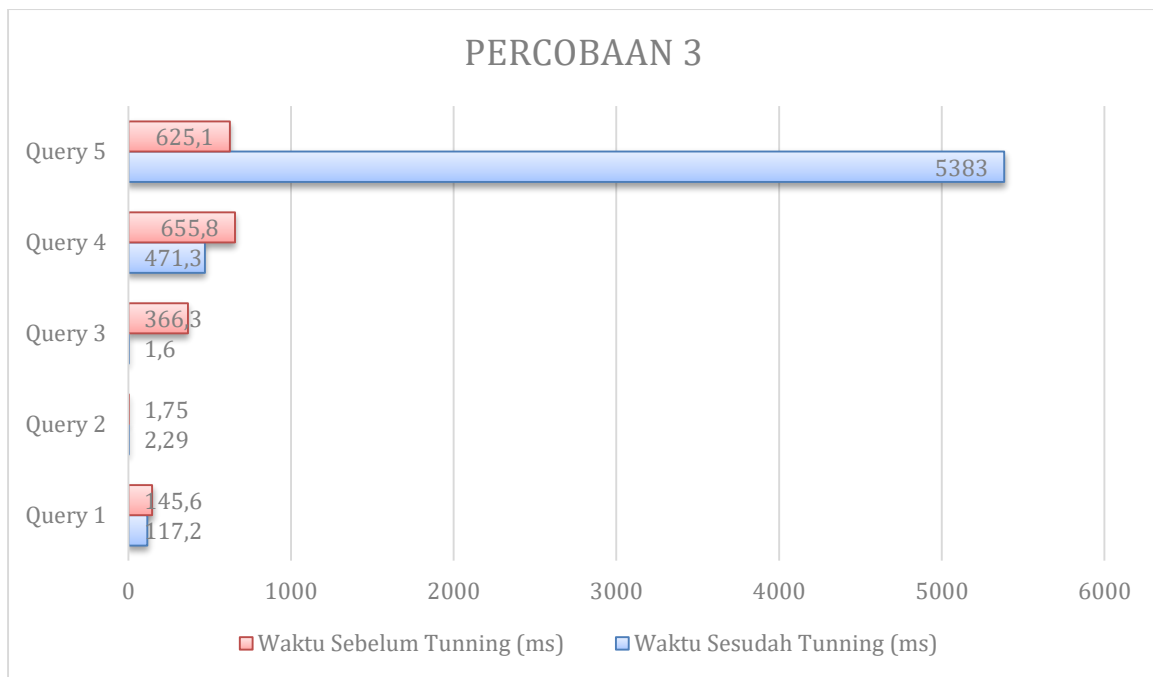
### 3.2 Data Hasil

DATA	Waktu Sebelum Tunning (ms)					Waktu Sesudah Tuning (ms)				
	Q1	Q2	Q3	Q4	Q5	Q1	Q2	Q3	Q4	Q5
advisor = 100, student = 100, section = 200,takes = 200	707	1547	77,53	416	85,0 1	4,3	1,2	0,8	6,6	8,9
advisor = 200, student = 200, section = 400,takes = 400	68,86	38,06	1,44	1,035	49,73	5,73	1,06	0,88	1,73	1,15
advisor = 500, student = 500, section = 1000,takes = 1000	145,6	1,75	366,3	655,8	625,1	117,2	2,29	1,6	471,3	5383

### 3.3 Grafik Data







### 3,4 Deskripsi

Dari hasil tuning index yang di dapatkan dari percobaan 1, percobaan 2, dan percobaan 3 pada database mendapatkan waktu eksekusi yang lebih cepat dari waktu eksekusi sebelum melakukan tuning. Hal ini sesuai dengan fungsi dari *Tunning and Performance* dengan mengunakan teknik Indexing, ini dapat terjadi karena jumlah data pada index jauh lebih kecil daripada data aslinya. Dan data-data ini juga tersimpan secara berurutan dalam bentuk struktur.

## DAFTAR PUSTAKA

- HAI, R. (2019, Agustus 12). *How to optimize MySQL/MariaDB Performance on Server*. Retrieved from HARIANACEH INDONESIA:  
<https://www.harianaceh.co.id/2019/08/12/how-to-optimize-mysql-mariadb-performance-on-server/>
- Imawan, A. (2018 ., Agustus 29). *Database Optimization #1 : Indexing*. Retrieved from DOT Lab: <https://medium.com/dot-lab/database-optimization-1-indexing-649e7837a22a>
- Wibowo, A. (2019, Juni 18). *Seberapa Pentingkah Performance dan Tuning Pada SQL Database?* Retrieved from BINUS UNIVERSITY GRADUTE PROGRAM :  
<https://mti.binus.ac.id/2019/06/18/seberapa-pentingkah-performance-dan-tuning-pada-sql-database/>