

**PRAKTIKUM MANAJEMEN BASIS DATA**  
**TUNNING DATABASE**



Dosen Pengampu :  
Arief Ichwani S.Kom, M.Cs

Disusun Oleh:  
Faidil Zikri (14117181)

Kelas :  
Manajemen Basis Data RD

PROGRAM STUDI TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI SUMETERA  
2019

## DAFTAR ISI

BAB 1 .....	3
1.1. Latar Belakang.....	3
1.2. Tujuan .....	3
1.3. Manfaat.....	3
BAB 2 .....	4
2.1. Database Management System .....	4
2.2. Tuning Database Dengan Indexing .....	4
2.3. Tuning Dengan Konfigurasi Database Management Server .....	4
BAB 3 .....	5
3.1. Tuning Index.....	5
BAB 4 .....	11
4.1. Kesimpulan.....	11
DAFTAR PUSTAKA .....	12

# **BAB 1**

## **PENDAHULUAN**

### **1.1.Latar Belakang**

Pada saat ini kebanyakan data tidak hanya disimpan dalam bentuk fisiknya saja tetapi sudah ada tempat penyimpanan yang besar dan bisa diakses di mana saja yang disebut database, namun database itu perlu untuk dirawat, dituning dan dimanage untuk memaksimalkan potensi dari database yang ada.

manajemen basis data adalah suatu sistem atau perangkat lunak yang dirancang untuk mengelola suatu database dan menjalankan operasi terhadap data yang diminta banyak pengguna. Tujuan utama basis data merupakan untuk dapat memudahkan kita dalam memperoleh data yang dicari.

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.

### **1.2.Tujuan**

Tujuan pembuatan tugas besar ini adalah sebagai berikut:

1. Mencari fungsi tuning basis data dengan cara membandingkan waktu proses/respond time sebelum dan sesudah dituning
2. Memberi pengalaman tentang bagaimana cara tuning basis data

### **1.3. Manfaat**

Adapun manfaat dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Meningkatkan performa kinerja transaksi database.
2. Memberikan fungsi indexing data base

## **BAB 2**

### **Landasan Teori**

#### **2.1.Database Management System**

Menurut Connolly dan Begg (2010: p17), Database Management System (DBMS) adalah suatu sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, dan mengatur akses ke basis data. Biasanya DBMS memiliki fasilitas-fasilitas sebagai berikut:

1. Fasilitas mendefinisikan basis data, biasanya menggunakan Data Definition Language (DDL). DDL mengizinkan pengguna untuk memspesifikasikan tipe, struktur, dan batasan aturan mengenai data yang bisa disimpan ke dalam basis data.
2. Fasilitas untuk mengizinkan pengguna untuk menambah, mengedit, menghapus dan mendapatkan kembali data dari basis data, biasanya menggunakan Data Manipulation Language (DML).
3. Fasilitas untuk mengontrol ke basis data (DCL).

#### **2.2.Tunning Database Dengan 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 menjadi jauh lebih cepat dan tidak banyak menghabiskan resource CPU.

#### **2.3.Tunning Dengan Konfigurasi Database Management Server**

Database Server adalah program computer yang menyediakan layanan basis data untuk program computer lain atau dengan kata lain bisa didefinisikan sebagai clien-server model. Tuning mencatat konfigurasi perangkat keras dan perangkat lunak yang digunakan, tuning ini dapat dipengaruhi oleh desain fisik database, termasuk normalisasi dan penyimpanan disk, jumlah tablem dan penggunaan DDL. Database management system menyediakan fleksibilitas untuk konfigurasi dabase service yang kita gunakan.

## BAB 3

### Percobaan

#### 3.1. Tuning Index

Percobaan ini menggunakan data

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

Membuat index pada database

```
MariaDB [dbms]> create index student_pk on student(id);  
Query OK, 0 rows affected (0.23 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [dbms]> create index section_pk on section(sec_id);  
Query OK, 0 rows affected (0.51 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

Lakukan Query pada table sesuai petunjuk

Query :

```
SELECT * FROM student  
SELECT * FROM student WHERE tot_cred>30;  
SELECT 'name',dept_name 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 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 1 Sebelum ditunning

```

+-----+-----+-----+
| 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,cours |
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 |
+-----+-----+-----+
5 rows in set (0.00 sec)

```

## Data 1 Setelah Tuning

```

1 | 0.00435598 | SELECT * FROM student |
2 | 0.00127492 | SELECT * FROM student WHERE tot_cred > 30 |
3 | 0.00000643 | 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,cours |
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 |
+-----+-----+-----+
5 rows in set (0.05 sec)

```

Query	Non-Indexing(ms)	Indexing(ms)
1.	7,57	4,9
2.	2,3	1,2
3.	7,85	1,1
4.	4,16	3,6
5.	8,51	7,8

## Data 2 Sebelum tuning

```

0.00046220 | SELECT * FROM student

0.00046140 | SELECT * FROM student WHERE tot_cred > 30

0.00041780 | SELECT `name`, dept_name FROM student WHERE tot_cred > 30

0.00530660 | SELECT * FROM takes JOIN student ON takes.ID = student.ID JOI
e_id

0.00520800 | SELECT student.`name`,student.dept_name,takes.sec_id AS pengam
tion.building,course.course_id,course.dept_name FROM takes JOIN student ON t
course_id = section.course_id JOIN course ON section.course_id = course.cou

```

## Data 2 Setelah tuning

```

0.00046220 | SELECT * FROM student

0.00046140 | SELECT * FROM student WHERE tot_cred > 30

0.00041780 | SELECT `name`, dept_name FROM student WHERE tot_cred > 30

0.00530660 | SELECT * FROM takes JOIN student ON takes.ID = student.ID JOI
se_id

0.00520800 | SELECT student.`name`,student.dept_name,takes.sec_id AS peng
tion.building,course.course_id,course.dept_name FROM takes JOIN student ON
course_id = section.course_id JOIN course ON section.course_id = course.cou

```

Query	Non-Indexing(ms)	Indexing(ms)
1.	70	56,3
2.	37,5	11,5
3.	1,44	0,95
4.	2,053	1,73
5.	59,9	17,5

### Data 3 sebelum tunnning

```

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)

```

### Data 3 Sesudah tunnning

```

MariaDB [mbd3]> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.11728119 | 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 |
+-----+-----+-----+

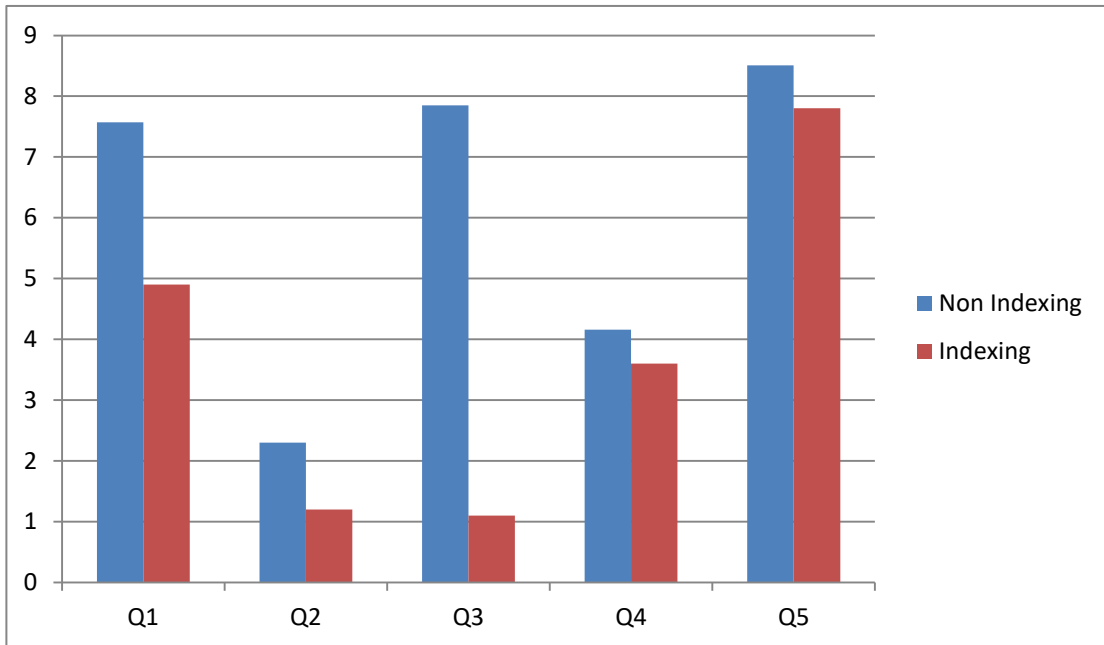
```

Query	Non-Indexing(ms)	Indexing(ms)
1.	150	123,2
2.	11,5	9,42
3.	365,5	146,3
4.	654,33	443,3
5.	640,3	583,7

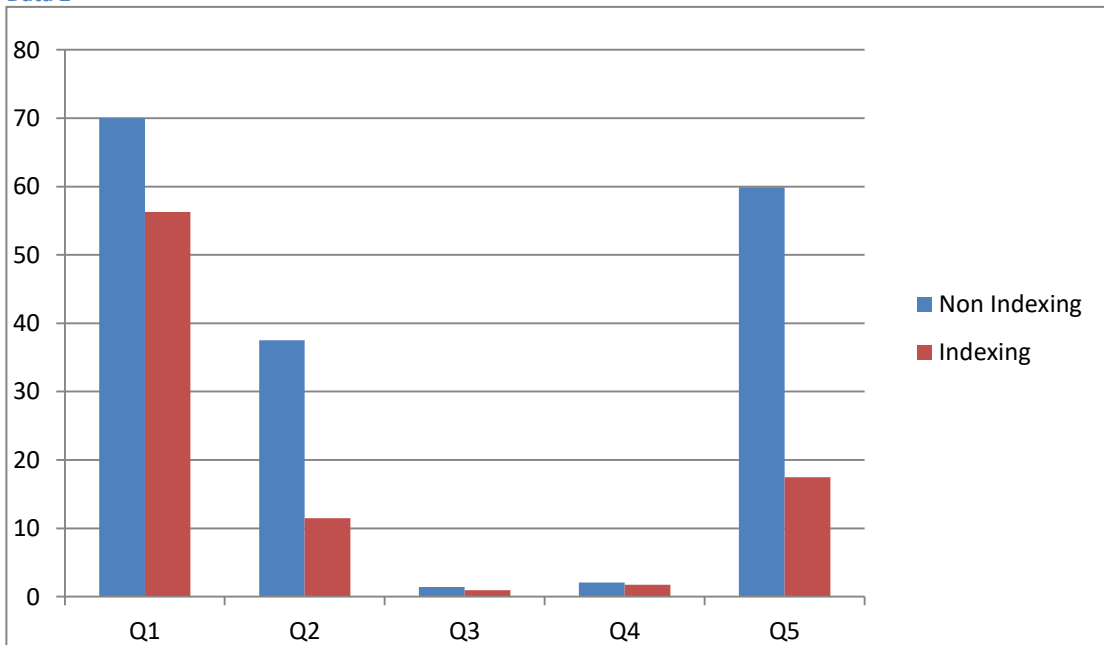


## Tabel Data Indexing

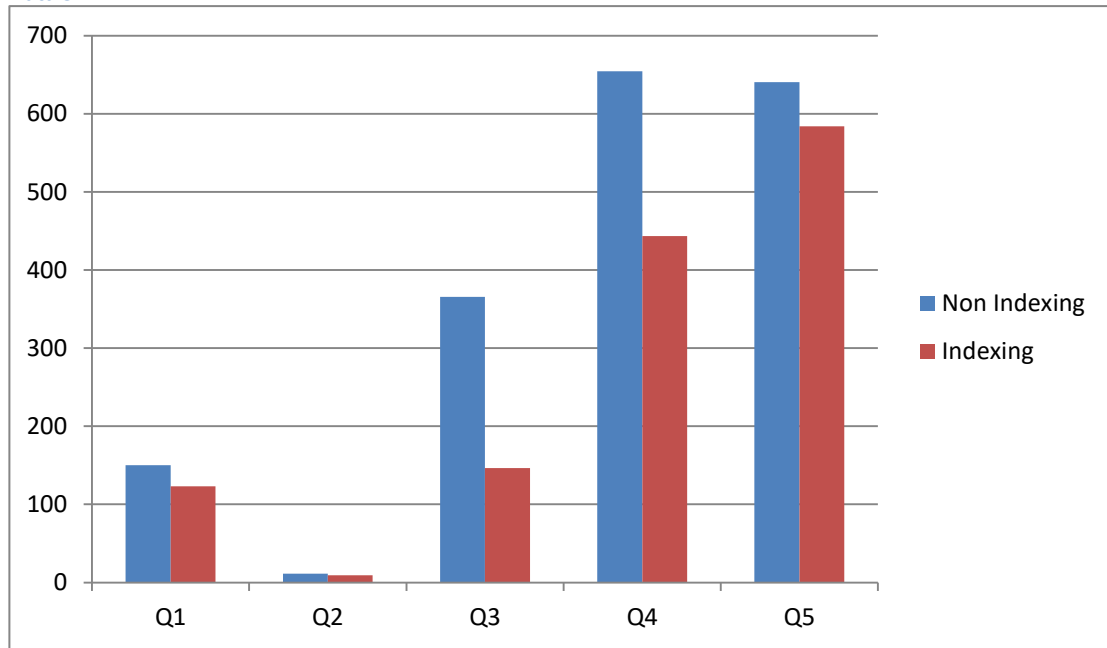
Data 1



Data 2



Data 3



## **BAB 4**

### **Penutup**

#### **4.1.Kesimpulan**

Dari keseluruhan data pada saat dilakukannya percobaan, dapat ditarik kesimpulan bahwa dengan melakukan indexing pada database dapat mempercepat melakukan eksekusi data, karena data pada database saat diindex menjadi lebih kecil daripada data sebelum dilakukannya indexing.

Melakukan indexing juga tergantung oleh hardware yang memadai, agar database bisa menjadi maksimal. Kesalahan dalam melakukan indexing dapat terjadi karena diakibatkan oleh kekurangan spesifikasi hardware.

## DAFTAR PUSTAKA

<https://www.geeksforgeeks.org/indexing-in-databases-set-1/>

<https://pojokprogrammer.net/content/performance-tuning-sederhana-di-mysqlmenggunakan-index>

<https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/>