

TUGAS BESAR
TUNING DATABASE SYSTEM
MATAKULIAH MANAJEMEN BASIS DATA

Disusun oleh:

Eron Wahyu /14117094



PROGRAM STUDI TEKNIK INFORMATIKA JURUSAN
TEKNOLOGI PRODUKSI, INDUSTRI DAN INFORMASI
INSTITUT TEKNOLOGI SUMATERA 2019

BAB I

PENDAHULUAN

1.1 Latar Belakang

manajemen basis data adalah suatu sistem atau perangkat lunak yang dirancang untuk mengelola suatu basis data dan menjalankan operasi terhadap data yang diminta banyak pengguna. Contoh tipikal MBD adalah akuntansi, sumber daya manusia, dan sistem pendukung pelanggan, MBD telah berkembang menjadi bagian standar di bagian pendukung (back office) suatu perusahaan. Contoh MBD adalah Oracle, SQL server 2000/2003, MS Access, MySQL dan sebagainya. DBMS merupakan perangkat lunak yang dirancang untuk dapat melakukan utilisasi dan mengelola koleksi data dalam jumlah yang besar. DBMS juga dirancang untuk dapat melakukan manipulasi data secara lebih mudah. Sebelum adanya DBMS, data pada umumnya disimpan dalam bentuk flat file, yaitu file teks yang ada pada sistem operasi. Sampai sekarangpun masih ada aplikasi yang menyimpan data dalam bentuk flat secara langsung.

Tujuan utama basis data merupakan untuk dapat memudahkan kita dalam memperoleh data yang kita cari. Selain itu ada beberapa tujuan basis data sebagai berikut yakni :

- Kecepatan ataupun kemudahan dalam menyimpan, mengedit ataupun memanipulasi data serta dapat melihat data.
- Efisiensi ruang penyimpanan dengan database, kita bisa dapat menghindari terjadinya kesamaan data dengan yang menggunakan kode tertentu sehingga ruang penyimpanan menjadi lebih efisien.

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. Dalam artikel ini, kita akan mencoba melakukan Query Tuning dengan bantuan Database Index.

1.2 Tujuan

Tujuan pembuatan tugas besar ini adalah sebagai berikut:

1. Memenuhi tugas besar matakuliah manajemen basis data.
2. Mencari fungsi tuning basis data dengan cara membandingkan waktu proses/respond time sebelum dan sesudah dituning
3. 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
3. Meningkatkan pengalaman mahasiswa dalam performa basis data

BAB II

LANDASAN TEORI

2.1 Database Management System (DBMS)

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. 10 2. Fasilitas untuk mengizinkan pengguna untuk menambah, mengedit, menghapus dan mendapatkan kembali data dari basis data, biasanya menggunakan Data Manipulation Language (DML).

Ada pula suatu fasilitas yang melayani pengaksesan data yang disebut query language. Bahasa yang diakui adalah Structured Query Language (SQL), yang merupakan standard dari DBMS. 3. Fasilitas untuk mengontrol ke basis data (DCL).

Contoh :

- a. Sistem keamanan yang mencegah user yang tidak punya otoritas untuk akses data.
- b. Suatu sistem terintegrasi yang memelihara konsistensi penyimpanan data.
- c. Suatu Sistem kontrol pengembalian data yang mana dapat mengembalikan data ke keadaan sebenarnya apabila terjadi kegagalan perangkat keras atau perangkat lunak.
- d. Terdapat suatu katalog yang dapat di akses oleh pengguna, yang menjelaskan data didalam basis data tersebut. Menurut Connolly dan Begg (2010: p18),

Komponen DBMS terbagi menjadi lima yaitu :

1. Hardware (perangkat keras) Hardware dapat berkisar dari komputer tunggal, mainframe tunggal, hingga jaringan komputer. Hardware yang dipakai tergantung pada kebutuhan organisasi dan Database Management System (DBMS) yang digunakan. Sebuah DBMS memerlukan jumlah minimum memori dan hardisk untuk 11 bekerja, tetapi konfigurasi yang minimum tidak memberikan performa yang handal.

2. Software (perangkat lunak) Komponen perangkat lunak terdiri dari perangkat lunak DBMS dan program aplikasi beserta sistem operasi (OS), termasuk jaringan perangkat lunak jika DBMS digunakan melalui jaringan.

3. Data Data merupakan data terpenting dalam DBMS khususnya sudut pandang dari end user mengenai data, dimana data berfungsi sebagai jembatan antara komponen mesin dengan komponen manusia.

4. Procedures Prosedur merupakan panduan dan aturan dalam membuat dan menggunakan basis data. Prosedur didalam basis data berupa : login ke dalam basis data, penggunaan fasilitas DBMS atau aplikasi program, cara menjalankan dan menghentikan DBMS, membuat backup database, menangani kerusakan hardware atau software, mengubah

struktur table, mengumpulkan basis data dari beberapa disk, meningkatkan kinerja atau membuat arsip data pada secondary storage.

5. People (manusia) Komponen terakhir yaitu manusia yang terlibat dengan sistem tersebut.

2.2 Tuning Database System

Tuning adalah cara agar performa transaksi data pada suatu sistem database dapat ditingkatkan dengan cara mengubah jalannya query database, menaikkan hardware dsb. Tuning data base dapat dikumpulkan dalam beberapa level: Hardware, Sistem Operasi, Database Management System, Schema basis data, Indexing, Query language.

2.3 Spesifikasi Alat

1. XAMPP ([[unsupported input](#)]'zæmp/ atau /'eks.æmp/^[1]) adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program.

Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

2. Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM).

3. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multialur, multipengguna, dengan sekitar 6 juta instalasi di seluruh dunia.

BAB III

PERCOBAAN

3.1 Tuning Database INDEXING

Tuning indexing yaitu mengoptimalkan waktu respond time dengan cara mengurutkan index.

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.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;`

Mengindexing table

```
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
```

DATA 1

Sebelum Indexing

```
1 | 0.70700216 | 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)
```

Setelah Indexing

```

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)

```

Perbandingan waktu sebelum dan sesudah tuning

Query	Non-Indexing(ms)	Indexing(ms)
1.	7,07	4,3
2.	1,547	1,2
3.	8,75	0,8
4.	4,16	3,6
5.	8,50 1	7,9

Data 2

Sebelum Tunning

```

1 | 0.06860060 | 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)

```

Sesudah Indexing

```

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)

```

Perbandingan waktu sebelum dan sesudah tuning

Query	Non-Indexing(ms)	Indexing(ms)
1.	69	57,3
2.	38,3	10,5
3.	1,44	0,95
4.	1,050	0,73
5.	49,9	15,5

Data 3

Sebelum Tuning

```
0.00116020 | SELECT * FROM student

0.00223750 | SELECT * FROM student WHERE tot_cred > 30

0.00111840 | SELECT `name`, dept_name FROM student WHERE tot_cred > 30

0.18902910 | SELECT * FROM takes JOIN student ON takes.ID = student.ID J

0.04922280 | SELECT student.`name`,student.dept_name,takes.sec_id AS peng
pt_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON t
```

Sesudah tuning

```
0.00484500 | SELECT * FROM student

0.00225590 | SELECT * FROM student WHERE tot_cred > 30

0.00050110 | SELECT `name`, dept_name FROM student WHERE tot_cred > 30

0.02184320 | SELECT * FROM takes JOIN student ON takes.ID = student.ID J

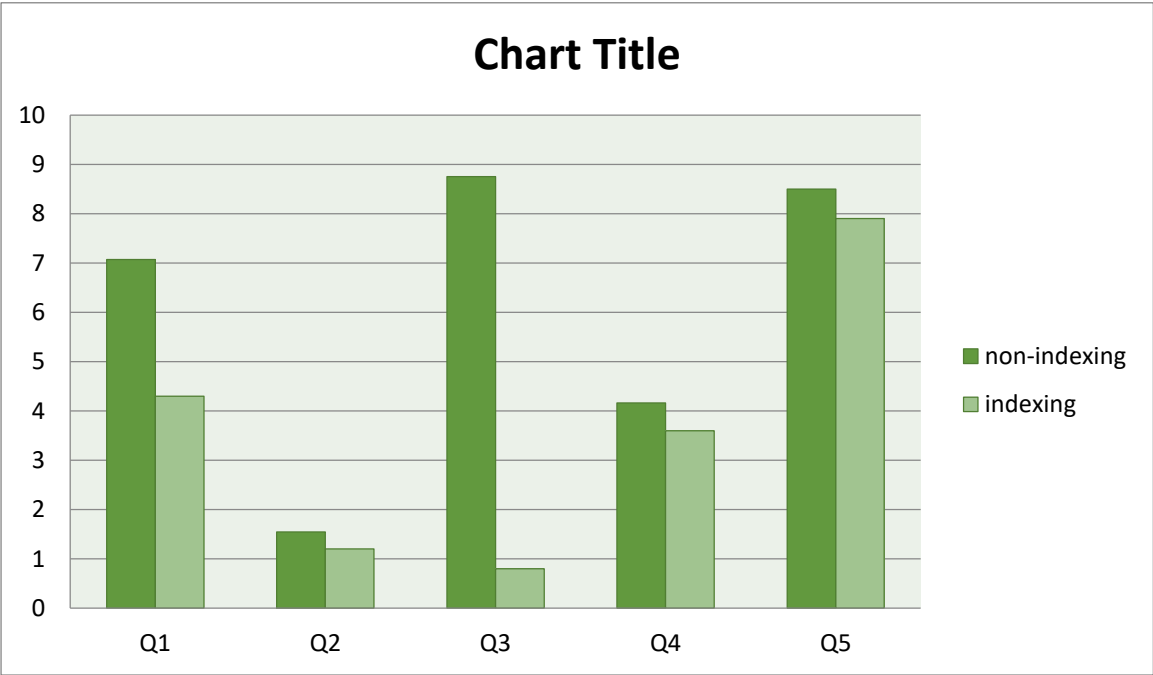
0.21341150 | SELECT student.`name`,student.dept_name,takes.sec_id AS peng
ept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON t
```

Perbandingan waktu sebelum dan sesudah tuning

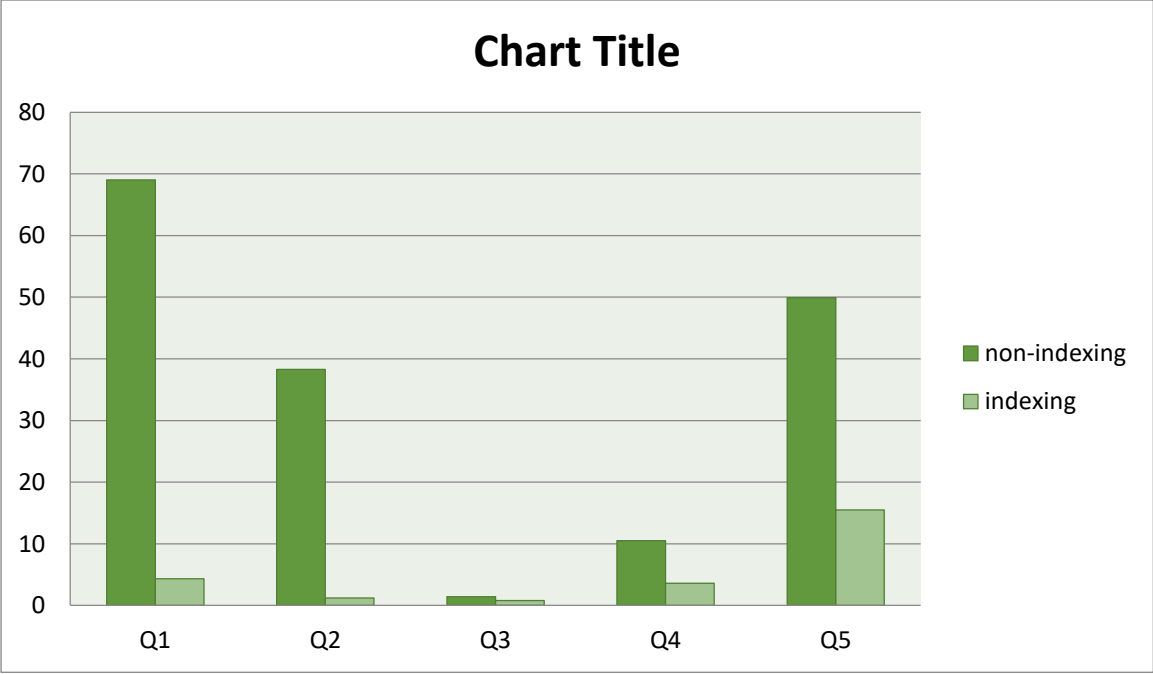
Query	Non-Indexing(ms)	Indexing(ms)
1.	150	127,2
2.	10,5	8,42
3.	370,5	151,3
4.	637,33	503,3
5.	630,3	583,7

Grafik

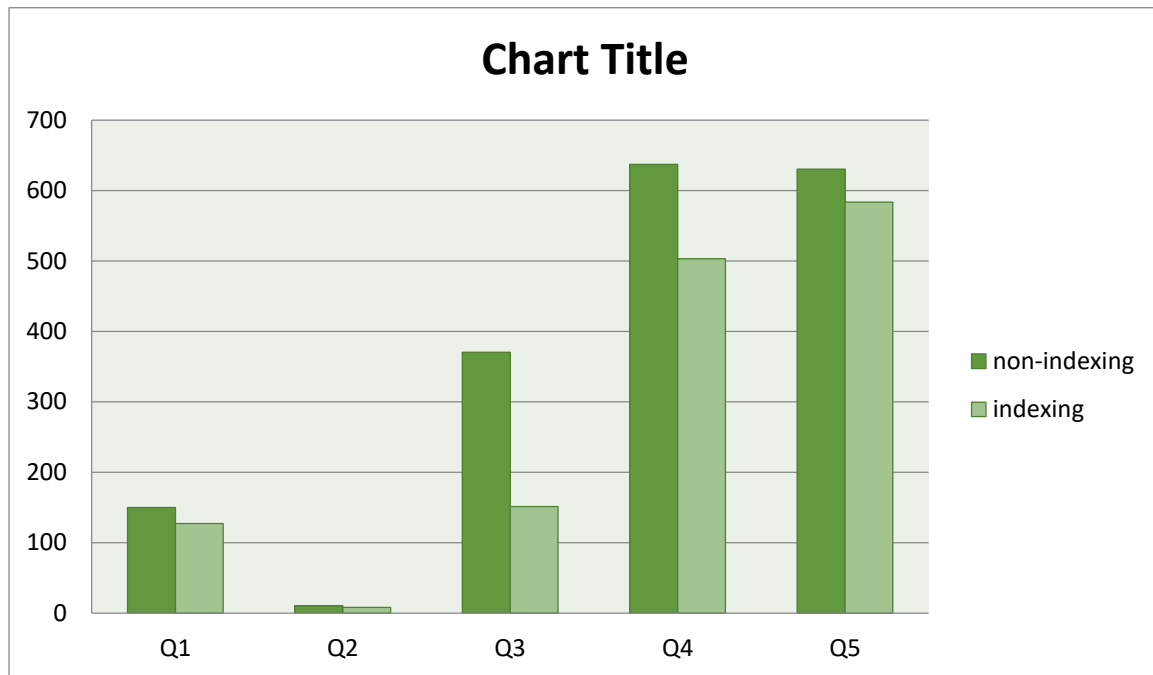
Data 1



Data 2



DATA 3



BAB IV

PENUTUP

4.1 Kesimpulan

Hasil yang didapatkan pada query 1 sampai 3 didapatkan hasil yang menunjukkan waktu yang dibutuhkan database untuk mengeksekusi data sesudah tuning lebih cepat daripada eksekusi data sesudah tuning . kesimpulan ini sesuai dengan fungsi tuning yang digunakan untuk indexing. Ini terjadi karena data pada indexing lebih kecil daripada data aslinya.

4.2 Saran

- Proses query ditentukan oleh performance computer yang digunakan apabila tanpa dituning.
- Apabila data terlalu banyak akan sulit dieksekusi oleh laptop menengah kebawah.

