

LAPORAN AKHIR

Diajukan Kepada Pengampu Matakuliah Manajemen Basis Data
untuk Memenuhi Tugas Besar Manajemen Basis Data



Dosen Pengampu :

Arief Ichwani, S.kom., M.Cs.

Diusulkan oleh :

Titis Prawilas Sari

(14117065)

**PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA
2019/2020**

DAFTAR ISI

DAFTAR ISI	ii
DAFTAR GAMBAR	iii
DAFTAR TABEL.....	iv
BAB I	1
STUDI LITERATUR.....	1
1. Tunning : Indexing	1
2. Tunning : Setting configuration	1
BAB II.....	1
DESKRIPSI PERCOBAAN.....	2
1. Tunning : Indexing	2
BAB III.....	3
HASIL DAN PEMBAHASAN	4
1. Hasil Pengujian dan Pembahasan Kuisisioner dan Problem Space	4
BAB IV	7
KESIMPULAN.....	7
DAFTAR PUSTAKA	8

DAFTAR GAMBAR

Gambar 1 Syntax Query Indexing	2
Gambar 2 Setting Configuration.....	3

DAFTAR TABEL & GRAFIK

Table 1 Create Unique Index.....	1
Table 2 Tuning : Setting Configuration	3
Table 3 Data Uji.....	4
Table 4 Percobaan dengan Data1.....	4
Table 5 Percobaan dengan Data2.....	4
Table 6 Percobaan dengan Data3.....	5
Grafik 1 Perubahan waktu eksekusi.....	5
Grafik 2 Perubahan waktu eksekusi.....	5
Grafik 3 Perubahan waktu eksekusi.....	6

BAB I

STUDI LITERATUR

1. Tuning : Indexing

Index adalah objek pada MySQL yang berisi data yang terurut dari nilai-nilai pada satu atau lebih field dalam suatu tabel. Penggunaan index pada database merupakan salah satu teknik pembuatan database yang baik. Hal ini terutama sangat berguna pada implementasi database dengan skala VLDB (Very Large Database) atau OLDB (Online Large Database). Saat database dibuat tanpa menggunakan index, maka kinerja server database dapat menurun secara drastis. Hal ini dikarenakan resource komputer banyak digunakan untuk pencarian data atau pengaksesan query SQL dengan metode table-scan. Cecilia dan Mihai (2011) mengemukakan bahwa index pada tabel memungkinkan SQL server untuk mendapatkan hasil pencarian tanpa mencari seluruh data di dalam tabel. Tabel dengan menggunakan index adalah cara terbaik untuk mengurangi logical read and disk input/output karena menyediakan mekanisme pencarian terstruktur. Ada dua cara berbeda untuk mendefinisikan index yaitu: a. Like a dictionary: kamus merupakan daftar kata-kata yang disusun menurut abjad. Indeks didefinisikan seperti kamus adalah sekumpulan data di-order secara leksikografi. Untuk alasan ini pencarian di indeks tidak akan mencakup semua baris tapi akan lebih mudah berdasarkan data yang di-order. b. Like a book index: pendekatan ini membuat indeks tidak akan mengubah tata letak data dalam tabel, tetapi hanya seperti indeks buku akan posisi data dalam tabel untuk posisi yang sesuai dalam tabel. Indeks didefinisikan dengan cara ini akan berisi data di kolom diindeks dan jumlah baris yang sesuai dari data. Cara penulisan untuk membuat index dalam database dapat dilihat seperti berikut:

Table 1 Create Unique Index

<pre>CREATE UNIQUE INDEX index_name ON table_name (column1, column2, ..);</pre>

2. Tuning : Setting configuration

Untuk melakukan performance tuning pada PostgreSQL dengan skala bisnis dengan konfigurasi manajemen database digunakan database administrator yang akan menganalisis parameter konfigurasi database PostgreSQL dan merekomendasikan konfigurasi optimal sesuai dengan workload yang dibutuhkan.

BAB II

DESKRIPSI PERCOBAAN

1. Tuning : Indexing

Pada database yang telah dibuat lakukan query 1 sampai query 5, catat waktunya. Lalu buat tuning indexing , dan lakukan kembali query 1 sampai query5

Keterangan Query:

- `SELECT * FROM student;`
- `SELECT * FROM student WHERE tot_cred > 30;`
- `SELECT `name`, department 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;`

Syntax Tuning Indexing

```
MariaDB [dbms]> create index index_tot_cred on student(tot_cred);
Query OK, 0 rows affected (1.513 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [dbms]> create index index_course_dept_name on course(dept_name);
Query OK, 0 rows affected (0.908 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [dbms]> create index index_student_id on student(id);
Query OK, 0 rows affected (0.238 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [dbms]> create index index_takes_id on takes(id);
Query OK, 0 rows affected (1.237 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [dbms]> create index index_takes_course_id on takes(course_id);
Query OK, 0 rows affected (0.697 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [dbms]> create index index_section_course_id on section(course_id);
Query OK, 0 rows affected (0.708 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [dbms]> create index index_course_id on course(course_id);
Query OK, 0 rows affected (0.769 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [dbms]>
```

Gambar 1 Syntax Query Indexing

2. Tuning: Setting Configuration

Atur setting pada file my.ini terletak di C:\xampp\mysql\bin. Lakukan perubahan pada nilai nilai yang ada, ubah sesuai dengan kebutuhan. Pada saat ini hanya 2 nilai yang diubah.

Table 2 Tuning : Setting Configuration

Sebelum diubah	Sesudah diubah
innodb_lock_wait_timeout=50	innodb_lock_wait_timeout=50
innodb_log_file_size=5M	innodb_log_file_size=10M

```
## Set .._log_file_size to 25 % of buffer pool size
innodb_log_file_size=10M
innodb_log_buffer_size=8M
innodb_flush_log_at_trx_commit=1
innodb_lock_wait_timeout=100
```

Gambar 2 Setting Configuration

BAB III

HASIL DAN PEMBAHASAN

i.

1. Hasil Pengujian dan Pembahasan

Terdapat 7 data uji:

Table 3 Data Uji

Data
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 = 100000,takes = 1000000
advisor = 1800, student = 1800, section = 180000,takes = 1800000
advisor = 10000, student = 10000, section = 30000000,takes = 30000000

- Data 1 : advisor = 100, student = 100, section = 200,takes = 200
Data mengalami perubahan waktu yang signifikan setelah dilakukan tuning pada query1 dan query4.

Table 4 Percobaan dengan Data1

Query	Waktu sebelum tuning	Waktu sesudah tuning
Q1	0.077	0.001
Q2	0.001	0.001
Q3	0.001	0.001
Q4	0.082	0.003
Q5	0.003	0.003

- Data 2 : advisor = 200, student = 200, section = 400,takes = 400
Data mengalami perubahan waktu yang signifikan setelah dilakukan tuning pada query1, query4 dan query5.

Table 5 Percobaan dengan Data2

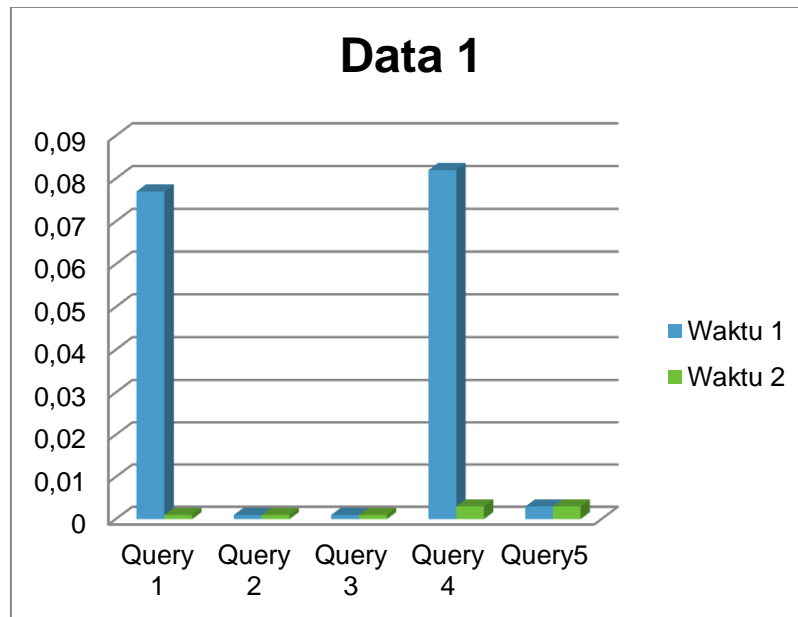
Query	Waktu sebelum tuning	Waktu sesudah tuning
Q1	0.001	0.001
Q2	0.001	0.001
Q3	0.001	0.001
Q4	0.692	0.006
Q5	0.004	0.004

- Data 3 : advisor = 500, student = 500, section = 1000,takes = 1000
Data mengalami perubahan waktu yang signifikan setelah dilakukan tuning pada query1, query4 dan query5.

Table 6 Percobaan dengan Data3

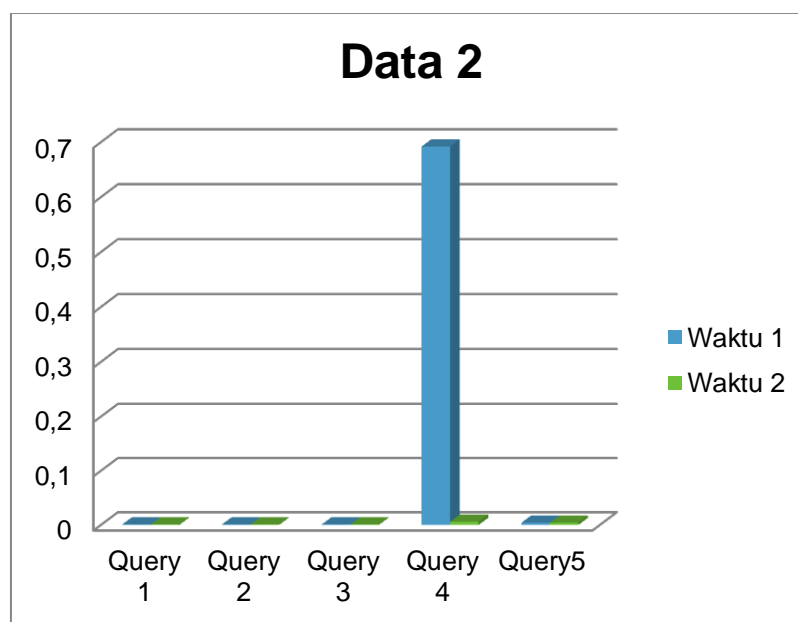
Query	Waktu sebelum tuning	Waktu sesudah tuning
Q1	0.180	0.001
Q2	0.001	0.001
Q3	0.001	0.001
Q4	0.205	0.025
Q5	0.073	0.015

Dari dibawah dapat dilihat terdapat dua query yang mengalami perubahan waktu eksekusi yang signifikan



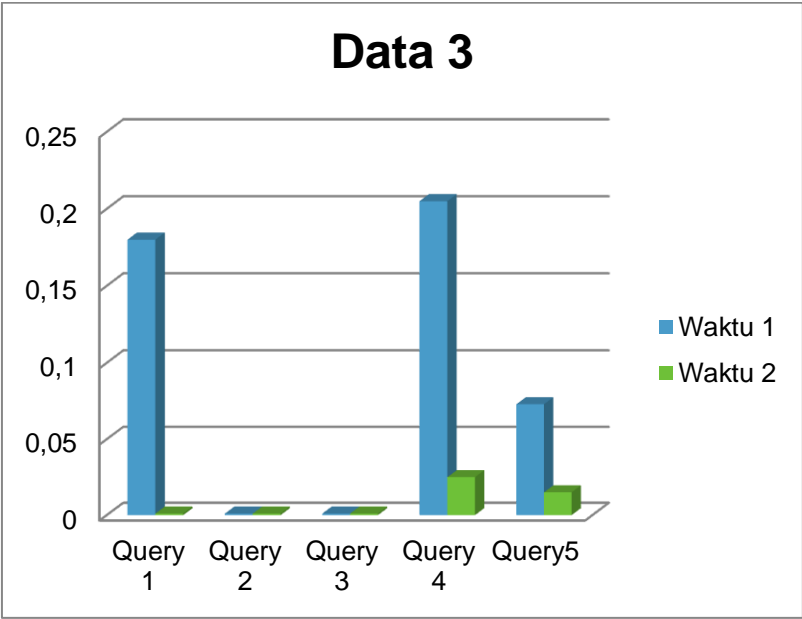
Grafik 1 Perubahan waktu eksekusi

Dari dibawah dapat dilihat terdapat satu query yang mengalami perubahan waktu eksekusi yang signifikan



Grafik 2 Perubhaan waktu eksekusi

Dari dibawah dapat dilihat terdapat tiga query yang mengalami perubahan waktu eksekusi yang signifikan



Grafik 3 Perubahan waktu eksekusi

BAB IV

KESIMPULAN

Tuning adalah aktifitas yang dilakukan untuk meningkatkan kinerja atau performance dari perangkat lunak dan perangkat keras, pada saat ini dilakukan query tuning pada suatu dbms dengan menambahkan index pada setiap table yang akan digunakan dalam sebuah query. Dari percobaan yang telah dilakukan dengan metode tuning indexing didapatkan hasil bahwa terdapat peningkatan waktu eksekusi yang signifikan pada beberapa query yang telah di uji coba, hal ini sesuai dengan tujuan dilakukannya tuning indexing yaitu untuk meningkatkan performance dalam hal ini meningkatkan waktu eksekusi sebuah query.

DAFTAR PUSTAKA

Nilaliliana Prihatin (2017). Optimasi Query pada System Informasi Pencatatan Aktifitas Perubahan Data Nilai Mahasiswa.