

TUGAS BESAR MANAJEMEN BASIS DATA



**Disusun Oleh :
14116033 ARIF WICAKSONO**

**Kelas :
Manajemen Basis Data RD**

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

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

Daftar Isi

Daftar Isi.....	2
BAB I.....	3
STUDI LITERATUR.....	3
1.1 Tuning : Indexing.....	3
1.2 Tuning : Setting Configuration DBMS.....	3
1. DBMS.....	3
2. JAVA.....	3
3. PC.....	3
BAB II.....	4
DESKRIPSI PERCOBAAN.....	4
2.1 Tuning : Indexing.....	4
2.2 Tuning : Setting Configuration DBMS.....	5
BAB III.....	6
HASIL PERCOBAAN.....	6
DAFTAR PUSTAKA.....	10

BAB I

STUDI LITERATUR

1.1 Tuning : Indexing

Pengindeksan adalah cara untuk mengoptimalkan kinerja database dengan meminimalkan jumlah akses disk yang diperlukan saat permintaan diproses. Ini adalah teknik struktur data yang digunakan untuk mencari dan mengakses data dalam database dengan cepat. Aktivitas 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. Umumnya programmer, pada saat mengembangkan sebuah aplikasi database, pasti akan membuat sebuah database dan memasukkan beberapa data contoh ke dalamnya. Pada saat pengembangan, tidak ada masalah dan kelihatannya semua baik-baik saja, namun seiring berjalannya waktu, aplikasi mulai terkesan lambat dan bisa jadi sampai *hang*. Hal ini terjadi karena biasanya programmer tidak pernah melakukan load testing menggunakan data besar, dalam artian, jumlah row yang banyak

1.2 Tuning : Setting Configuration DBMS

Database server adalah program komputer yang menyediakan layanan basis data untuk program komputer lain. Database server didefinisikan sebagai client server model. Database management system menyediakan fungsi-fungsi database server dan beberapa DBMS (seperti mysql) sangat eksklusif untuk client-server model database access. Disini kita melakukan performance tuning untuk meningkatkan kinerja pencarian pada data all.sql. Dengan memakai :

1. DBMS

Database yang digunakan Mysql. Mysql sendiri adalah perangkat lunak sistem manajemen basis data yang multithread maupun multi-user.

2. JAVA

Bahasa pemrograman yang digunakan adalah Java. Dengan menggunakan environment JDK untuk membuka all.sql melalui perintah javac dan java.

3. PC

Percobaan ini menggunakan PC dengan spek :

- Processor : Intel®Core™ i3-3240T CPU @2.90GHz (4CPUs)
- Memory : 4096 MB
- Storage : HDD 500 Gb
- OS : Windows 10 Pro

BAB II DESKRIPSI PERCOBAAN

2.1 Tuning : Indexing

Pertama yang harus dilakukan generate data pada file tableGen.java menggunakan environment java, sebelumnya pastikan ada java telah terinstal kemudian buka melalui Command Prompt dan lakukan dengan perintah :

- Javac tableGen.java
- Java tableGen

Maka akan terbentuk all.sql dari hasil generate tabel. Terlihat seperti gambar 1.

```

Directory of D:\Xampp\htdocs\IF3144-1920\sql\tableGen
21-Dec-19 23:25 <DIR> .
21-Dec-19 23:25 <DIR> ..
04-Dec-19 00:05 <DIR> ref
21-Dec-19 23:25 <DIR> sql
21-Dec-19 23:25 9,622 tableGen.class
05-Dec-19 05:19 15,274 tableGen.java
05-Dec-19 09:27 <DIR> _MACOSX
2 File(s) 24,896 bytes
5 Dir(s) 66,370,355,200 bytes free

D:\Xampp\htdocs\IF3144-1920\sql\tableGen>javac tableGen.java

D:\Xampp\htdocs\IF3144-1920\sql\tableGen>java tableGen.java
Error: Could not find or load main class tableGen.java

D:\Xampp\htdocs\IF3144-1920\sql\tableGen>java tableGen
'J', 910, 18
'E', 194, 10
'G', 928, 10
'H', 567, 20
'F', 417, 25
'D', 101, 29
'E', 676, 34
'E', 648, 94
'G', 272, 43
'A', 417, 10
'DE', 'F', 448052.44
  
```

gambar 1 Tabel generate

Kemudian lakukan pengujian query dengan pembaharuan data sebagai berikut :

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

Tabel 1. Data uji

Adapun query pengujian yang dilakukan adalah :

1. `SELECT * FROM student;`
2. `SELECT * FROM student WHERE tot_cred > 30;`
3. `SELECT `name` department 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;`

2.2 Tuning : Setting Configuration DBMS

Untuk percobaan tuning dapat dilakukan dengan mengubah file my.ini pada direktori Xampp/mysql/bin atau dapat menggunakan index pada kolom table.

◆ Tuning Dengan mengubah file my.ini

- innodb_buffer_pool_size dari 16M menjadi 6G
- innodb_additional_mem_pool_size dari 2M menjadi 20M
- sort_buffer_size dari 20M menjadi 4M

◆ Tuning dengan menambahkan index

- Kolom name pada tabel student
- Kolom tot_cred pada tabel student
- Kolom semester pada tabel takes
- Kolom time_slot_id pada tabel section

BAB III HASIL PERCOBAAN

3.1 Percobaan

Melakukan percobaan dengan query berikut :

1. SELECT * FROM student;

Data	Waktu Tanpa Tuning (ms)	Waktu Tuning config (ms)	Waktu Tuning index (ms)
advisor = 100, student = 100, section = 200,takes = 200	0	0	0
advisor = 200, student = 200, section = 400,takes = 400	0	0	0
advisor = 500, student = 500, section = 1000,takes = 1000	0	0	0
advisor = 700, student = 700, section = 20000,takes = 20000	0	0	0
advisor = 1000, student = 1000, section = 100000,takes = 1000000	0	0	0
advisor = 1800, student = 1800, section = 180000,takes = 1800000	0	0	0
advisor = 10000, student = 10000, section = 30000000,takes = 30000000	20	10	10

Tabel 2. hasil Query 1

Didapat dari hasil uji, data 1 hingga 6 memiliki 0 ms response time, dan data ke 7 memiliki response time 30ms dan setelah di tuning menjadi 10ms.

2. SELECT * FROM student WHERE tot_cred > 30;

Data	Waktu Tanpa Tuning (ms)	Waktu Tuning config (ms)	Waktu Tuning index (ms)
advisor = 100, student = 100, section = 200,takes = 200	0	0	0
advisor = 200, student = 200, section = 400,takes = 400	0	0	0

advisor = 500, student = 500, section = 1000,takes = 1000	0	0	0
advisor = 700, student = 700, section = 20000,takes = 20000	0	0	0
advisor = 1000, student = 1000, section = 100000,takes = 1000000	0	0	0
advisor = 1800, student = 1800, section = 180000,takes = 1800000	0	0	0
advisor = 10000, student = 10000, section = 30000000,takes = 30000000	10	10	0

Tabel 3. hasil Query 2

Dapat dilihat, query 2 tidak terlalu berbeda jauh dengan query 1, Query ini menghasilkan keluaran yang tidak terlalu banyak makan pencarian. Data ke 7 memiliki waktu response time 10ms, ketika sebelum maupun sesudah dituning.

3. SELECT `name` department FROM student WHERE tot_cred > 30;

Data	Waktu Tanpa Tuning (ms)	Waktu Tuning config (ms)	Waktu Tuning index (ms)
advisor = 100, student = 100, section = 200,takes = 200	0	0	0
advisor = 200, student = 200, section = 400,takes = 400	0	0	0
advisor = 500, student = 500, section = 1000,takes = 1000	0	0	0
advisor = 700, student = 700, section = 20000,takes = 20000	0	0	0
advisor = 1000, student = 1000, section = 100000,takes = 1000000	0	0	0
advisor = 1800, student = 1800, section = 180000,takes = 1800000	0	0	0
advisor = 10000, student = 10000, section = 30000000,takes = 30000000	20	10	10

Tabel 4. hasil Query 3

Kemudian query 3 tidak terlalu berbeda jauh juga dengan query 2, Ketika Data ke 7 memiliki waktu response time 20ms, dan 10ms sesudah dituning.

4. `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id;`

Data	Waktu Tanpa Tuning (ms)	Waktu Tuning config (ms)	Waktu Tuning index (ms)
advisor = 100, student = 100, section = 200,takes = 200	0	0	0
advisor = 200, student = 200, section = 400,takes = 400	10	0	0
advisor = 500, student = 500, section = 1000,takes = 1000	20	20	15
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			

Tabel 5. hasil Query 4

Pada query ke 4, PC yang digunakan hanya dapat menjalankan sampai data ke 3, karena data selanjutnya PC tidak sanggup mengeksekusi dengan data yang sangat banyak.

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;`

Data	Waktu Tanpa Tuning (ms)	Waktu Tuning config (ms)	Waktu Tuning index (ms)
advisor = 100, student = 100, section = 200,takes = 200	0	0	0
advisor = 200, student = 200, section = 400,takes = 400	10	10	10

advisor = 500, student = 500, section = 1000,takes = 1000	25	20	20
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 = 300000000			

Tabel 6. hasil Query 5

Pada query ke 5, di dapat response time hingga data ke 3, ketika mengeksekusi data selanjutnya, PC Juga tidak dapat melakukan eksekusi dan dikawatirkan jika di paksa akan Blue Screen.

3.2 Pembahasan

Hasil dari percobaan didapat bahwa, Melakukan tuning dapat dengan 2 cara yaitu dengan config php.ini dan melakukan indexing pada database. Tuning dapat mempercepat kinerja pencarian yang menghasilkan response time yang lebih cepat dari pada database yang tidak melakukan tuning. Dapat dilihat juga, tuning menggunakan indexing lebih cepat dari yang config.

DAFTAR PUSTAKA

Silberschatz, Abraham., Korth, Henry F., dan Sudarshan, S., 2006, "Database System Concepts Sixth edition", New York. : McGraw-Hill.

"Performance Tuning Sederhana di MySQL Menggunakan Index". Pojok Programmer. Di akses 21 Desember 2019.

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

"Konfigurasi MySQL untuk Trafik Tinggi" Blog Emka .Di akses 21 Desember 2019.

<https://emka.web.id/programming/database/mysql/2012/konfigurasi-mysql-untuk-trafik-tinggi/>