

TUGAS BESAR MANAJEMEN BASI DATA (IF3144)
INDEXING TUNING



DOSEN PENGAJAR

Arief Ichwani S.Kom., M.C

Disusun Oleh

Muhammad Irpansyah Damanik (14116058)

PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA

2019

DAFTAR ISI

DAFTAR ISI.....	i
BAB I STUDI LITERATUR	1
1.1 Tunning : Index.....	1
1.2 Tunning : Setting Configuration DBMS.....	1
BAB II DESKRIPSI PERCOBAAN	2
2.1 Tunning : Index	2
2.2 Tunning: Setting Configuration DBMS.....	4
BAB III HASIL DAN PEMBAHASAN.....	5
3.1 Tabel Hasil	5
3.2 Grafik Hasil	8
DAFTAR PUSTAKA	10

BAB I

STUDI LITERATUR

3.3 Tuning : Indexing

Index adalah sebuah objek dalam sebuah *database* yang dapat mempercepat proses pencarian (*query*) data. Saat *database* dibuat tanpa menggunakan *index*, maka kinerja *server database* dapat menurun secara signifikan. Hal ini dikarenakan *resource* CPU banyak digunakan untuk pencarian data atau pengaksesan *query* SQL dengan metode *table-scan*. *Index* membuat pencarian data akan lebih cepat dan tidak banyak menghabiskan *resource* CPU.^[1]

Index merupakan objek struktur data tersendiri yang tidak bergantung kepada struktur tabel. Setiap *index* terdiri dari nilai kolom dan penunjuk (atau ROWID) ke baris yang berisi nilai tersebut. Penunjuk tersebut secara langsung menunjuk ke baris yang tepat pada tabel, sehingga menghindari terjadinya *full table-scan*. Akan tetapi lebih banyak *index* pada tabel tidak berarti akan mempercepat *query*. Semakin banyak *index* pada suatu tabel menyebabkan kelambatan pemrosesan perintah-perintah DML (*Data Manipulation Language*), karena setiap terjadi perubahan data maka *index* juga harus disesuaikan. Berikut ini adalah beberapa alasan kenapa *index* diperlukan:^[1]

1. Kolom sering digunakan dalam klausa *WHERE* atau dalam kondisi *join*
2. Kolom berisi nilai dengan jangkauan yang luas
3. Kolom berisi banyak nilai *null*
4. Tabel berukuran besar dan sebagian besar *query* menampilkan data kurang dari 2-4%

Perlu kita perhatikan bahwa terdapat beberapa kondisi dimana tidak diperlukan kehadiran *index*, yaitu ketika:^[1]

1. Tabel kecil
2. Kolom tidak sering digunakan sebagai kondisi dalam *query*
3. Kebanyakan *query* menampilkan data lebih dari 2-4% dari seluruh data
4. Tabel sering di-*update*

1.2 Tuning : Setting Configuration DBMS

Secara default MySQL hadir dengan konfigurasi yang sudah cukup mumpuni melayani request database puluhan orang, apalagi jika hanya sekedar melayani testing environment oleh sang Developer saja. Namun jika pengguna aplikasi anda ini ratusan atau bahkan ribuan, anda harus melakukan *tuning* ulang untuk server bertrafik tinggi.^[2]



BAB II

DESKRIPSI PERCOBAAN

2.1 Tuning: Setting Configuration DBMS

Setting Configuration DBMS tanpa menggunakan indexing :

- a. Create Database DDL-MySQL

 Create database 

Create

- b. Import -> choose file -> pilih database sql yang ada pada rar (IF3144-1920-master) atau folder yang sudah di ekstrak IF3144-1920-master -> sql -> DDL-MySQL.sql (menggunakan software XAMPP)



Importing into the database "DDL-MySQL2"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Browse your computer: (Max: 2,048KiB)

You may also drag and drop a file on any page.

Character set of the file:

- c. Kemudian OK
- d. Buka tableGen.java edit sesuai data di bawah ini

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

- e. Jalankan tableGen.java dengan command javac tableGen.java -> ketik java tableGen.
Hasilnya akan seperti dibawah ini

```
D:\data irpan\tugas\mbd\IF3144-1920-master\IF3144-1920-master\sql\tableGen>javac tableGen.java
D:\data irpan\tugas\mbd\IF3144-1920-master\IF3144-1920-master\sql\tableGen>java tableGen
'E', 996, 10
'I', 501, 114
'B', 349, 78
'D', 298, 33
'D', 416, 11
'C', 438, 10
'H', 749, 74
'I', 9, 10
'C', 312, 81
'H', 800, 10
'GT', 'J', 835328.88
DE/GT
'DE', 'D', 690834.90
```

- f. Setelah langkah e sudah dikerjakan. Buka folder IF3144-1920-master-> sql -> tableGen -> sql -> all.sql. copy-kan semua data yang ada
- g. Buka cmd ketikkan mysql -u root

```
C:\Users\THOSHIBA>cd c:/xampp/mysql/bin
c:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
```

- h. Use database ddl-mysql dan paste-kan semua data yang di tahap f (all.sql) pastikan jangan sampai terjadi error

```
MariaDB [(none)]> use ddl-mysql
Database changed
MariaDB [ddl-mysql]> insert into classroom values('E', 996, 10);
Query OK, 1 row affected (0.01 sec)

MariaDB [ddl-mysql]> insert into classroom values('I', 501, 114);
Query OK, 1 row affected (0.00 sec)
```

- i. Langkah terakhir ketikkan perintah sesuai dengan ketentuan di bawah ini dan tulis jawabannya di bab 3.

j. Ulangi dari tahap d hingga ke Query ke 5

```
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.
```

2.2 Tuning menggunakan indexing :

Untuk melakukan konfigurasi tersebut, silakan buka file my.cnf pada direktori /etc/mysql/my.cnf (jika anda menggunakan linux) atau C:\xampp\mysql\bin\my.ini.^[2]

Silakan pakai dan ubah konfigurasi yang saya rekomendasikan berikut.^[2]

```
1 [mysqld]
2 datadir=/var/lib/mysql
3 tmpdir=/var/lib/mysqltmp
4 socket=/var/lib/mysql/mysql.sock
5 skip-locking
6 skip-name-resolve
7 table_cache=2048
8 thread_cache_size=32
9 back_log=100
10 max_connect_errors=10000
11 open_files=10000
12 interactive_timeout=400
13 wait_timeout=300
14 max_connections=500
15 skip-bdb
16 log-slow-queries=/var/lib/mysqllogs/slow-log
17 long_query_time=2
18 log-queries-not-using-indexes
19 max_allowed_packet=128M
20 tmp_table_size=256M
21 max_heap_table_size=256M
22 query_cache_size=32M
23 query_cache_limit = 6M
24 sort_buffer_size=4M
25 read_buffer_size=4M
26 read_rnd_buffer_size=16M
27 join_buffer_size=2M
28 default-storage-engine=InnoDB
29 key_buffer_size=128M
30 myisam_sort_buffer_size=64M
31 innodb_log_file_size=100M
32 innodb_buffer_pool_size=6G
33 innodb_additional_mem_pool_size=20M
34 innodb_support_xa = 0
35 [mysql.server]
36 user=mysql
37 [mysqld_safe]
```

Simpan dan restart server MySQL untuk menikmati hasilnya.

BAB III
HASIL DAN PEMBAHASAN

3.1. hasil Tuning Tanpa Indexing

Dengan keterangan :

kode Query	Query
Q1	SELECT * FROM student
Q2	SELECT * FROM student WHERE tot_cred > 30;
Q3	SELECT `name`, department FROM student WHERE tot_cred > 30;
Q4	SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id
Q5	SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan, takes.semester,section.room_number,section.building,course.course_id,course.dep t_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 Sebelum Tuning (ms)				
	Q1	Q2	Q3	Q4	Q5
advisor = 100, student = 100, section = 200,takes = 200	0.00 (100 r)	0.00 (77 r)	0.00 (77 r)	0.00 (414 r)	0.00 (414 r)
advisor = 200, student = 200, section = 400,takes = 400	0.00 (200 r)	0.00 (200 r)	0.00 (147r)	0.00 (147r)	0.00 (1165 r)
advisor = 500, student = 500, section = 1000,takes = 1000	0.00 (500 r)	0.00 (384 r)	0.00 (384 r)	0.02 (5989 r)	0.01 (5989 r)
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	-	-	-	-	-

Untuk advisor = 700, student = 700, section = 20000,takes = 20000, advisor = 1000, student = 1000, section = 100000,takes = 1000000 dan advisor = 10000, student = 10000, section = 30000000,takes = 30000000 tidak dapat diproses akan menampilkan error seperti dibawah ini

```
'735', '14', 'Spring', 2006, 'G', '241', 'I'
'946', '10', 'Spring', 2005, 'I', '461', 'G'
'977', '9', 'Spring', 2001, 'I', '461', 'C'
'149', '10', 'Spring', 2004, 'E', '473', 'P'
'361', '13', 'Spring', 2007, 'E', '473', 'J'
'405', '10', 'Spring', 2010, 'A', '862', 'J'
'612', '12', 'Fall', 2002, 'C', '150', 'E'
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 2000
    at tableGen.main(tableGen.java:315)
D:\data irpan\tugas\mbd\IF3144-1920-master\IF3144-1920-master\sql\tableGen>
```

Dikarenakan ada data yang ditetapkan oleh soal yang dibuat dan juga memerlukan banyak waktu. Batasanya dapat dilihat dibawah ini bahwa, max nilai dari section adalah 2000 melebihi dari ketentuan diatas. Begitu juga dengan data yang lainnya.


```

6 public class tableGen {
7
8     private static int maxClassroom = 100;
9     private static int maxStudent = 4000;
10    private static int maxDepartment = 50;
11    private static int maxRoom = 1000;
12    private static int maxCourse = 1000;
13    private static int maxSection = 2000;
14    private static int maxAdvisor = 4000;
15    private static int maxInstructor = 1000;
16    private static int maxTeaches = 5000;
17    private static int maxTakes = 40000;
18    private static int maxPrereq = 1000;
19    private static int maxBuilding = 100;
20    private static int maxName = 4200;
21    private static int maxDept = 100;
22    private static int maxTitle = 1000;
23    private static double maxSalary = 100000.0;
24    private static double maxBudget = 999999.0;
25    private static int maxID = 99999;

```

3.2 hasil Tuning Menggunakan Indexing

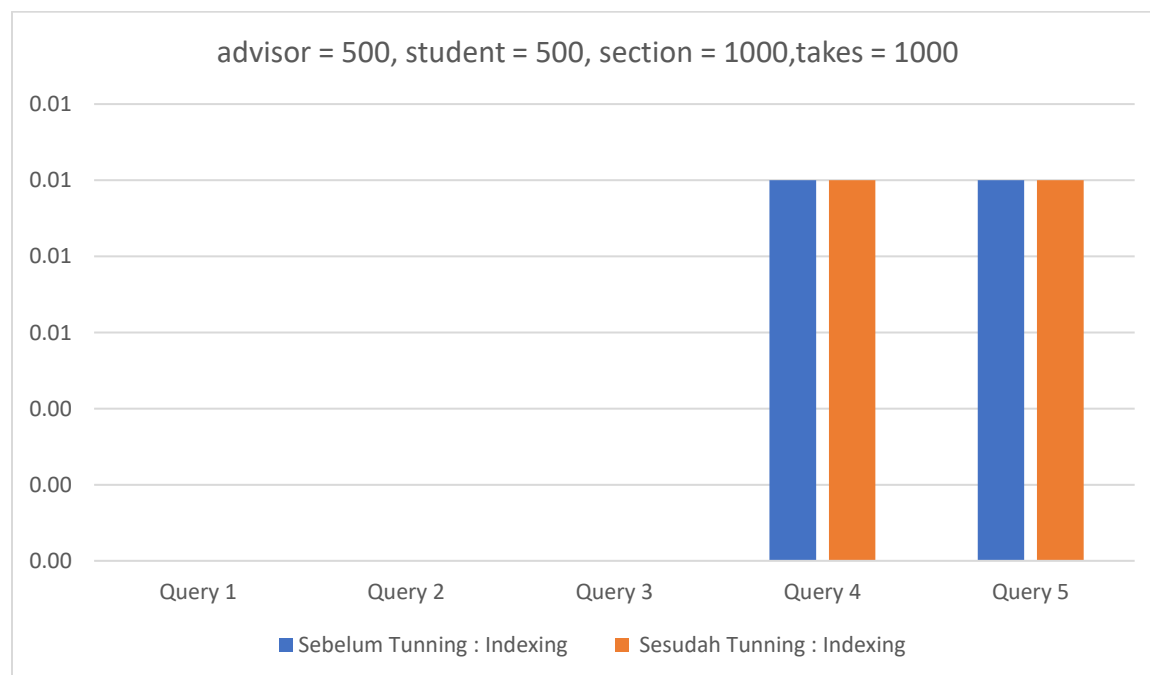
Data	Waktu sesudah Tuning (ms) r = rows				
	Q1	Q2	Q3	Q4	Q5
advisor = 100, student = 100, section = 200,takes = 200	0.00 (100 r)	0.00 (73 r)	0.00 (73 r)	0.00 (378 R)	0.00 (378 R)
advisor = 200, student = 200, section = 400,takes = 400	0.00 (200 r)	0.00 (158 r)	0.00 (158 r)	0.00 (1257 r)	0.00 (1257 r)
advisor = 500, student = 500, section = 1000,takes = 1000	0.00 (500 r)	0.00 (389 r)	0.00 (389 r)	0.02 (5968 r)	0.01 (5968 r)
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 = 3000000,takes = 30000000	-	-	-	-	-

Untuk advisor = 700, student = 700, section = 20000,takes = 20000, advisor = 1000, student = 1000, section = 100000,takes = 1000000 dan advisor = 10000, student = 10000, section = 30000000,takes = 30000000 tidak dapat diproses akan menampilkan error seperti dibawah ini

```
'735', '14', 'Spring', 2006, 'G', '241', 'I'
'946', '10', 'Spring', 2005, 'I', '461', 'G'
'977', '9', 'Spring', 2001, 'I', '461', 'C'
'149', '10', 'Spring', 2004, 'E', '473', 'P'
'361', '13', 'Spring', 2007, 'E', '473', 'J'
'405', '10', 'Spring', 2010, 'A', '862', 'J'
'612', '12', 'Fall', 2002, 'C', '150', 'E'
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 2000
    at tableGen.main(tableGen.java:315)
D:\data irpan\tugas\mbd\IF3144-1920-master\IF3144-1920-master\sql\tableGen>
```

Dikarenakan ada data yang ditetapkan oleh soal yang dibuat dan juga memerlukan banyak waktu. Batasanya dapat dilihat dibawah ini bahwa, max nilai dari section adalah 2000 melebihi dari ketentuan diatas. Begitu juga dengan data yang lainnya.

3.3 Grafik Hasil



Dari Query Q1, Q2 dan Q3 dengan Q4 dan Q5 dapat dikatakan mereka memiliki estimasi waktu yang sama sebelum atau sesudah di indexing, hanya saja yang membedakan diantara keduanya yaitu baris yang diproses . sesudah di indexing lebih sedikit memproses baris daripada sebelum di indexing. Tetapi sangat diuntungkan jika data yang diproses sangat banyak karena dapat diproses dengan cepat dengan data yang sudah di indexing begitu sebaliknya jika data yang diproses sedikit akan sama hal nya dengan data yang belum di indexing.

Berdasarkan hasil pengujian, dapat disimpulkan bahwa

1. Indexed view terbukti meningkatkan kinerja query dalam mengakses data secara signifikan.
2. Indexed view bekerja meningkatkan kinerja secara optimal pada jenis query agregasi dan join

DAFTAR PUSTAKA

- [1] admin, Inovasi Informatika Indonesia <https://www.i-3.co.id/2016/10/07/index-pada-database/> (hlm. 1).
- [2] admin satoe June 11, 2012
<https://emka.web.id/programming/database/mysql/2012/konfigurasi-mysql-untuk-trafik-tinggi/> (hlm. 2 dan 4)