

**LAPORAN TUGAS
MANAJEMEN BASIS DATA**



Oleh:
M Iqbal Revantama
14117003
MBD RB

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI PRODUKSI, INDUSTRI DANINFORMASI
INSTITUT TEKNOLOGI SUMATERA
2019**

DAFTAR ISI

DAFTAR ISI.....	2
ISI LAPORAN.....	3
A. Studi Literatur.....	3
B. Deskripsi Percobaan.....	4
HASIL DAN PEMBAHASAN.....	7
A. Database mbdrean.....	7
B. Database mbdrean2.....	7
C. Database mbdrean3.....	8
D. Database mbdrean4.....	9
DAFTAR PUSTAKA.....	10

ISI LAPORAN

A. Studi Literatur

1. Tuning Index

Index adalah objek pada MySQL yang berisi data yang terurut - dari nilai-nilai pada satu atau lebih field dalam suatu table.

Sama seperti daftar isi pada sebuah buku, index terutama digunakan untuk mempercepat pencarian terhadap suatu set data dengan kondisi tertentu - yang melibatkan kombinasi field yang sudah didefinisikan dalam suatu index.

Tanpa index, pencarian data biasanya akan memakan waktu lama, terutama jika data sudah dalam skala jumlah yang sangat besar.

Syntax pada MySQL untuk membuat suatu index adalah sebagai berikut :

```
CREATE INDEX index_name ON tbl_name (index_col_name,...);
```

2. Tuning DBMS Configuration (key_buffer_size & innodb_buffer_pool_size)

key_buffer_size, bagian ini adalah bagian terpenting ketika anda menggunakan tabel dengan jenis MyISAM. Gunakan sekitar 30 sampai 40% dari total memory server database anda agar mendapatkan performa maksimal. Tabel dengan jenis MyISAM menggunakan Sistem Operasi cache untuk melakukan penyimpanan data sementara jadi menggunakan sebagian besar memory untuk cache adalah pilihan yang tepat. Sedangkan variabel key_buffer_size bisa diisi sesuai dengan besaran file .MYI yang anda miliki, menggunakan key buffer size terlalu besar sedangkan file .MYI yang anda miliki hanya 1GB maka akan sia sia, jika file .MYI anda hanya berkisar antara 1GB anda bisa menggunakan key buffer size sebesar 16-32Mb saja.

innodb_buffer_pool_size, bagian ini sangat penting jika anda menggunakan tabel dengan jenis Innodb, tabel ini lebih sensitif jika dibandingkan dengan jenis tabel MyISAM. Perlu diperhatikan untuk tabel berjenis Innodb ini adalah kebalikan dari key buffer size tabel MyISAM. Untuk isi dari variabel ini anda bisa menggunakan 10 sampai 20% saja.

B. Deskripsi Percobaan

1. Persiapan Database

Pada percobaan ini menggunakan komputer dengan spesifikasi sebagai berikut:

```
revtm@revtm-NEON-BNE:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                 2
On-line CPU(s) list:   0,1
Thread(s) per core:    1
Core(s) per socket:    2
Socket(s):              1
NUMA node(s):          1
Vendor ID:              AuthenticAMD
CPU family:             20
Model:                  2
Model name:             AMD E2-2000 APU with
Stepping:               0
CPU MHz:                878.259
CPU max MHz:           1750,0000
CPU min MHz:           875,0000
BogoMIPS:               3492.54
Virtualization:         AMD-V
L1d cache:              32K
L1i cache:              32K
L2 cache:               512K
```

Sedangkan kondisi free memory saat penelitian dilakukan adalah sebagai berikut:

```
revtm@revtm-NEON-BNE:~$ free
              total        used        free      shared  buff/cache   available
Mem:           1618124       248124       981712           0       388288       981712
Swap:          3905532       350520       3555012           0           0       3555012
revtm@revtm-NEON-BNE:~$
```

Jumlah database yang digunakan ada empat buah, yaitu 'mbdrevan', 'mbdrevan2', 'mbdrevan3', dan 'mbdrevan4'.

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| lapor_pabw |
| mahasiswa |
| mbdrevan |
| mbdrevan2 |
| mbdrevan3 |
| mbdrevan4 |
| mysql |
| performance_schema |
| revan |
+-----+
```

Kemudian semua database di-import dengan query DDL yang telah disiapkan, untuk mempermudah *import* maka proses dilakukan dengan skrip bash.

```
#!/bin/bash

sudo mysql -u root mbdrevan < DDL.sql
sudo mysql -u root mbdrevan2 < DDL.sql
sudo mysql -u root mbdrevan3 < DDL.sql
sudo mysql -u root mbdrevan4 < DDL.sql
```

Begitu juga dengan proses import isi tabel yang juga menggunakan skrip bash. Setiap file all.sql, all2.sql, all3.sql, dan all4.sql mewakili jumlah *record* pertabel yang berbeda.

```
#!/bin/bash

sudo mysql -u root mbdrevan < all.sql
sudo mysql -u root mbdrevan2 < all2.sql
sudo mysql -u root mbdrevan3 < all3.sql
sudo mysql -u root mbdrevan4 < all4.sql
```

all.sql	advisor = 100, student = 100, section = 200,takes = 200
all2.sql	advisor = 200, student = 200, section = 400,takes = 400
all3.sql	advisor = 500, student = 500, section = 1000,takes = 1000
all4.sql	advisor = 700, student = 700, section = 20000,takes = 20000

Untuk menampilkan durasi dari setiap query, maka fitur profiling diaktifkan dengan perintah:

```
SET session profiling = 1;
```

Untuk melihat durasi query maka dapat menggunakan perintah:

```
SHOW profiles;
```

```
MariaDB [mbdreva]> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.02563858 | SELECT * FROM student |
+-----+-----+-----+
1 row in set (0.03 sec)
```

Query yang akan diberikan untuk diambil data durasi eksekusinya adalah sebagai berikut:

Quey1	SELECT * FROM student;
Quey2	SELECT * FROM student WHERE tot_cred > 30;
Quey3	SELECT name, dept_name FROM student WHERE tot_cred > 30;
Quey4	SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id;
Quey5	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;

Setiap percobaan pada satu database selesai, maka database akan didaur ulang dengan cara *drop-create* untuk digunakan lagi pada kondisi tuning lainnya.

2. Tuning Index

Tuning index dilakukan pada tabel 'student' dan 'takes' dengan query:

```
create index cred4 on student(tot_cred);
create index stud_id on takes(ID);
```

Tuning index dilakukan dengan konfigurasi *innnoDB default*.

3. Tuning Setting DBMS

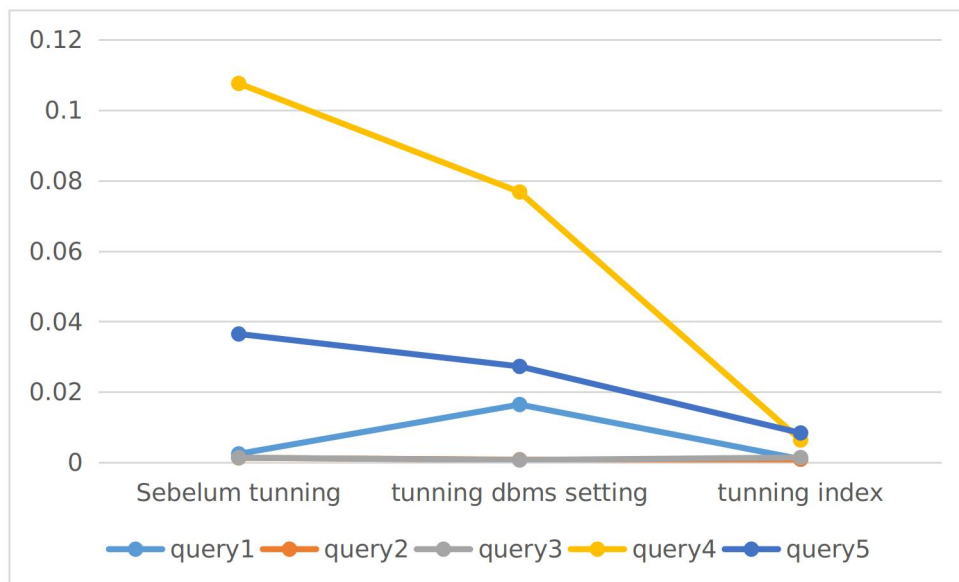
Tuning dilakukan dengan mengubah size dari buffer size sesuai dengan tinjauan studi literatur. Tuning ini dilakukan tanpa adanya penggunaan metode index.

HASIL DAN PEMBAHASAN

A. Database mbdrevan

	Sebelum tuning	tunning dbms setting	tunning index
query1	0,00248968	0,01647488	0,00098886
query2	0,00135438	0,00082632	0,00096652
query3	0,00140527	0,00074703	0,00144813
query4	0,10755662	0,07676174	0,00640474
query5	0,03648824	0,02729108	0,00841333

Percobaan query pada kondisi sebelum tuning menunjukkan hasil yang masih terbilang cepat karena data record yang sedikit. Namun hasil dari perbandingan waktu eksekusi yang lebih teliti menunjukkan adanya peningkatan kecepatan eksekusi setelah adanya tuning dbms setting dan tabel yang menggunakan index. Berikut visualisasi data dengan menggunakan *line chart*.

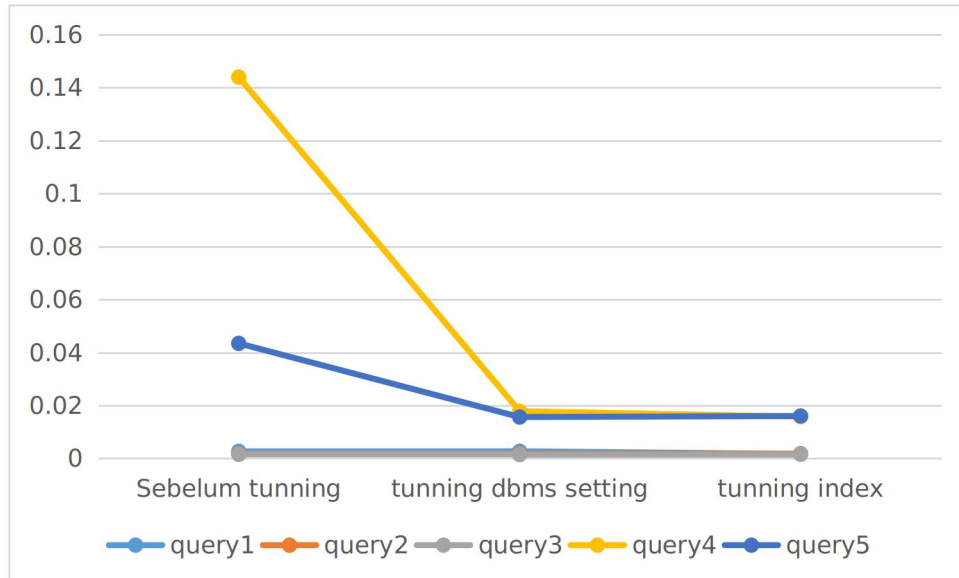


B. Database mbdrevan2

	Sebelum tuning	tunning dbms setting	tunning index
query1	0,00268559	0,00271444	0,00167931
query2	0,00178054	0,00174816	0,00181142
query3	0,00154781	0,00145611	0,00158284
query4	0,14397482	0,01782867	0,01581975
query5	0,04345314	0,01563759	0,01601459

Percobaan query pada kondisi sebelum tuning menunjukkan hasil waktu eksekusi yang lebih lama dibanding percobaan di database pertama, namun masih

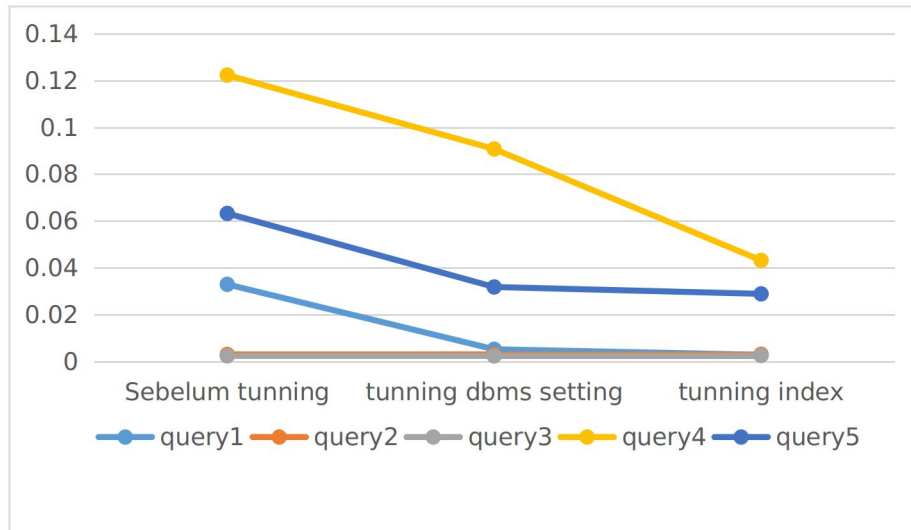
terbilang cepat karena data record yang juga sedikit. Hasil dari perbandingan waktu eksekusi yang lebih teliti menunjukkan adanya peningkatan kecepatan eksekusi yang cukup drastis setelah adanya tuning dbms setting dan tabel yang menggunakan index. Berikut visualisasi data dengan menggunakan *line chart*.



C. Database mbdrevan3

	Sebelum tuning	tuning dbms setting	tuning index
query1	0,03291907	0,00517383	0,00282484
query2	0,00299141	0,00286044	0,00305579
query3	0,00235844	0,00231944	0,00251345
query4	0,12231995	0,09074755	0,04314169
query5	0,06321586	0,03179085	0,02885908

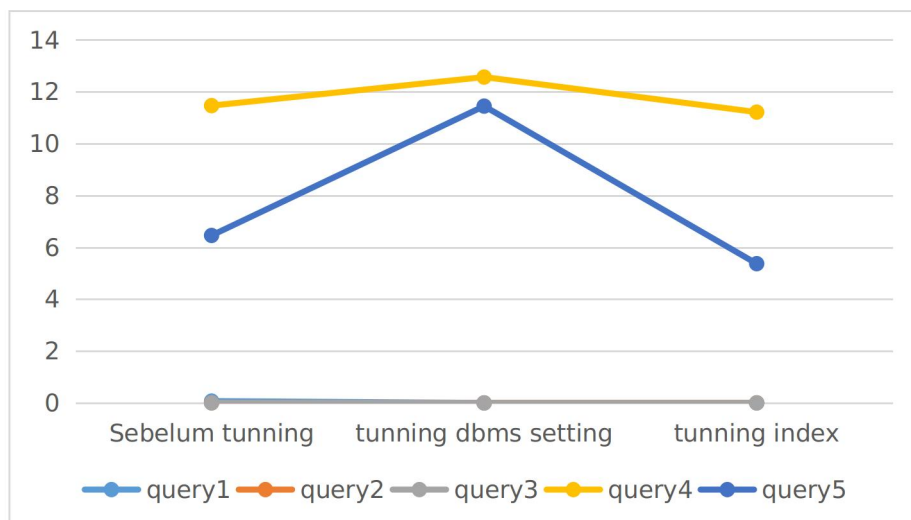
Percobaan query pada kondisi sebelum tuning menunjukkan hasil waktu eksekusi yang lebih lama dibanding percobaan di database kedua. Hasil dari perbandingan waktu eksekusi yang lebih teliti menunjukkan adanya peningkatan kecepatan eksekusi yang cukup drastis terutama pada tuning yang menggunakan index. Berikut visualisasi data dengan menggunakan *line chart*.



D. Database mbdrevan4

	Sebelum tuning	tuning dbms setting	tuning index
query1	0,07397061	0,00202476	0,00347641
query2	0,00352679	0,00235035	0,00362966
query3	0,00273295	0,0015597	0,00290029
query4	11,46827414	12,56973348	11,21836405
query5	6,4611866	11,44968153	5,37284423

Percobaan query pada kondisi sebelum tuning menunjukkan hasil waktu eksekusi yang lebih lama dibanding percobaan di database ketiga. Hal yang perlu diperhatikan adalah tuning dengan konfigurasi dbms justru menghasilkan waktu eksekusi yang lebih lama dibandingkan percobaan tanpa tuning. Diperkirakan ruang RAM tersedia pada saat tersebut sangat kecil sehingga data sementara dari proses select sebelumnya tidak lagi tersedia dan DBMS harus mengakses hardisk kembali. Berikut visualisasi data dengan menggunakan *line chart*.



DAFTAR PUSTAKA

<https://sites.google.com/a/phi-integration.com/mysql-tutorial/sql/membuat-index-pada-mysql>

<https://jagoanit.com/index.php/2018/01/04/apa-yang-harus-di-tuneup-dari-server-mysql/>

<https://www.linux.com/tutorials/5-commands-checking-memory-usage-linux/>

<https://dev.mysql.com/doc/refman/5.5/en/show-profile.html>