**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](#_Toc1641579983)

[ISI LAPORAN 3](#_Toc1145355642)

[A. Studi Literatur 3](#_Toc2002929354)

[B. Deskripsi Percobaan 4](#_Toc1314928150)

[HASIL DAN PEMBAHASAN 7](#_Toc806664933)

[A. Database mbdrevan 7](#_Toc985836022)

[B. Database mbdrevan2 7](#_Toc983710092)

[C. Database mbdrevan3 8](#_Toc1916249985)

[D. Database mbdrevan4 9](#_Toc1067042836)

[DAFTAR PUSTAKA 10](#_Toc696916644)

# ISI LAPORAN

## Studi Literatur

1. Tunning 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,...);

1. Tunning DBMS Configuration (**key\_buffer\_size & innodb\_buffer\_pool\_size)**

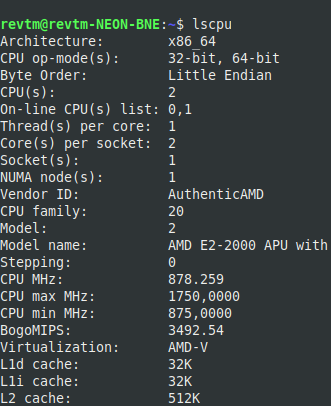
**key\_buffer\_size,** bagian ini adalah bagian terpenting ketika anda menggunakan tabel dengan jenis MyISAM. Gunakan sekitar 30 sampai 40% dari total memory server database anda agar mendapatkan performa maksimal. Tabel dengan jenis MyISAM menggunakan Sistem Operasi cache untuk melakukan penyimpanan 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.

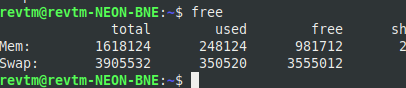
## Deskripsi Percobaan

1. Persiapan Database

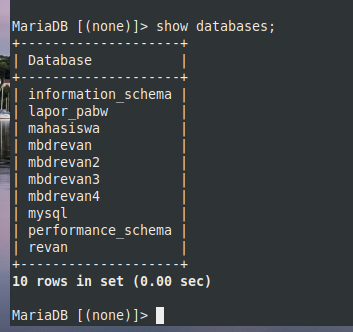
Pada percobaan ini menggunakan komputer dengan spesifikasi sebagai berikut:



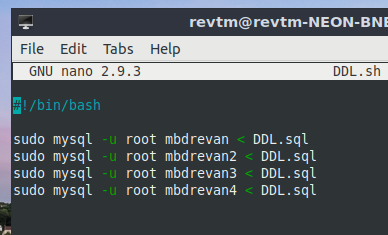
Sedangkan kondisi free memory saat penelitian dilakukan adalah sebagai berikut:



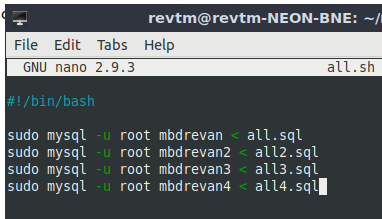
Jumlah database yang digunakan ada empat buah, yaitu ‘mbdrevan’,’mbdrevan2’, ’mbdrevan3’, dan ’mbdrevan4’.



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



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.



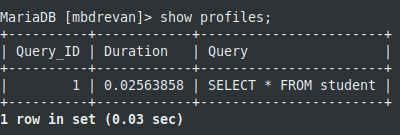
|  |  |
| --- | --- |
| all.sql | advisor = 100, student = 100, section = 200,takes = 200 |
| all2.sql | advisor = 200, student = 200, section = 400,takes = 400 |
| all3.sql | advisor = 500, student = 500, section = 1000,takes = 1000 |
| all4.sql | advisor = 700, student = 700, section = 20000,takes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



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 tunning lainny.

1. Tunning Index

Tunning index dilakukan pada tabel ‘student’ dan ‘takes’ dengan query:

create index cred4 on student(tot\_cred);

create index stud\_id on takes(ID);

Tunning index dilakukan dengan konfigurasi innoDB *default*.

1. Tunning Setting DBMS

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

# HASIL DAN PEMBAHASAN

## Database mbdrevan

|  |  |  |  |
| --- | --- | --- | --- |
|  | Sebelum tunning | 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 tunning 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 tunning dbms setting dan tabel yang menggunakan index. Berikut visualisasi data dengan menggunakan *line chart*.

## Database mbdrevan2

|  |  |  |  |
| --- | --- | --- | --- |
|  | Sebelum tunning | 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 tunning 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 tunning dbms setting dan tabel yang menggunakan index. Berikut visualisasi data dengan menggunakan *line chart*.

## Database mbdrevan3

|  |  |  |  |
| --- | --- | --- | --- |
|  | Sebelum tunning | tunning dbms setting | tunning 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 tunning 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 tunning yang menggunakan index. Berikut visualisasi data dengan menggunakan *line chart*.

## Database mbdrevan4

|  |  |  |  |
| --- | --- | --- | --- |
|  | Sebelum tunning | tunning dbms setting | tunning 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 tunning menunjukkan hasil waktu eksekusi yang lebih lama dibanding percobaan di database ketiga. Hal yang perlu dperhatikan adalah tunning dengan konfigurasi dbms justru menghasilkan waktu eksekusi yang lebih lama dibandingkan percobaan tanpa tunning. 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>