

**LAPORAN TUGAS BESAR
TUNING DATABASE SYSTEM
MATA KULIAH MANAJEMEN BASIS DATA**

oleh :

Dewi Rahayu / 14116056

Dosen Pengajar:

Arief Ichwani, S.Kom.,M.Cs.



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

DAFTAR ISI

DAFTAR ISI.....	2
DAFTAR TABEL.....	3
DAFTAR GAMBAR.....	4
BAB I_PENDAHULUAN.....	5
1.1 Latar Belakang.....	5
1.2 Tujuan.....	5
1.3 Manfaat.....	6
BAB II_STUDI LITERATUR.....	7
2.1 DBMS (<i>Database Management System</i>).....	7
2.2 Tunning Indexing.....	8
BAB III_HASIL DAN PEMBAHASAN.....	10
3.1 Hasil.....	10
3.2 Pembahasan.....	11
BAB IV KESIMPULAN.....	29
DAFTAR PUSTAKA.....	30

DAFTAR TABEL

Tabel 1. Data 1 10

Tabel 2. Data 2 10

Tabel 3. Data 3 11

Tabel 4. Data 4 11

DAFTAR GAMBAR

Gambar 2. Insert Tabel Database	13
Gambar 3. Running Data 3 dengan Query 1 Sebelum Tunning	14
Gambar 4. Running Data 3 dengan Query 2 Sebelum Tunning	15
Gambar 5. Hasil Running Data 3 dengan Query 3 sebelum Tunning.....	16
Gambar 6. Hasil Running Data 3 dan Query 4 sebelum Tunning	17
Gambar 7. Hasil Running Data 3 dengan Query 5 sebelum Tunning.....	18
Gambar 8. Hasil Running Data 3 dengan Query 1 setelah tuning config	19
Gambar 9. Hasil Running Data 3 dengan Query 2 setelah tuning config	20
Gambar 10. Hasil Running Data 3 dengan query 3 setelah Tunning Config.....	21
Gambar 11. Hasil Running Data 3 dengan query 4 setelah Tunning Config.....	22
Gambar 12. Hasil Running Data 3 dengan query 5 setelah Tunning Config.....	23
Gambar 13. Hasil Running Data 3 dengan query 1 setelah indexing	24
Gambar 14. Hasil Running Data 3 dengan query 2 setelah indexing	25
Gambar 15. Hasil Running Data 3 dengan query 3 setelah indexing	26
Gambar 16. Hasil Running Data 3 dengan query 4 setelah indexing	27

BAB I

PENDAHULUAN

1.1 Latar Belakang

Semakin berkembangnya teknologi informasi pada saat ini sangat membantu setiap pekerjaan manusia. Seperti dalam hal pengumpulan data, setiap orang dalam suatu institusi atau perusahaan pasti tidak bisa lepas dari menggunakan DBMS (Database Management System). Dari yang sederhana seperti menggunakan Microsoft Access sampai dengan menggunakan DBMS yang cukup kompleks seperti Oracle. DBMS ini bertujuan untuk mempermudah dalam hal penyimpanan data maupun dalam hal manipulasi data, yang nantinya data tersebut dapat digunakan kembali apabila diperlukan. Selain teknologi pengumpulan data yang terus berkembang, teknologi penyimpanan data pun terus mengalami peningkatan. Dahulu biasanya suatu media penyimpanan seperti Harddisk mempunyai kapasitas dalam ukuran Giga, tetapi sekarang banyak ditemui kapasitas Harddisk yang sampai pada ukuran Tera. Hal ini sangat membantu suatu sekolah atau perusahaan yang akan menyimpan data yang mempunyai ukuran yang cukup besar [1].

Sistem manajemen basis data (*database management system*) adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan mengakses sumber data dengan cara praktis dan efisien. DBMS dapat digunakan untuk mengakomodasikan berbagai macam pemakai yang memiliki kebutuhan akses yang berbeda-beda. DBMS pada umumnya menyediakan fasilitas atau fitur-fitur yang memungkinkan data dapat diakses dengan mudah, aman, dan cepat yaitu independensi, konkurensi, integritas, pemulihan, katalog sistem [2]. Namun, untuk meningkatkan *performance* dari *database* salah satunya dengan melakukan *tunning*.

Pada tugas besar mata kuliah Manajemen Basis Data ini yaitu melakukan *tunning* pada sebuah *database* yang telah diberikan.

1.2 Tujuan

Adapun tujuan dari tugas besar ini adalah sebagai berikut:

1. Memberikan pelajaran bagaimana melakukan *tunning* terhadap sebuah *database*.
2. Mahasiswa dapat mengetahui waktu yang didapatkan ketika running sebelum dan

setelah *tunning database*.

1.3 Manfaat

Adapun manfaat yang bisa diambil adalah sebagai berikut:

1. *Tunning* bisa meningkatkan *performance* dari sebuah *database*.
2. Mampu meningkatkan waktu kerja dari sebuah *database*.

BAB II

STUDI LITERATUR

2.1 DBMS (*Database Management System*)

Menurut Date C.J. Date, definisi DBMS adalah tempat atau lokasi untuk sekumpulan berkas data yang sudah terkomputerisasi dengan tujuan untuk memelihara informasi, dan juga memuat informasi tersebut, terutama apabila informasi tersebut sedang dibutuhkan. penggunaan DBMS dalam jaringan komputer adalah untuk menghindari kekacauan dalam hal pengolahan data yang jumlahnya besar. DBMS merupakan perantara bagi pengguna dan *database*. Ada dua jenis bahasa komputer yang dapat digunakan dalam berinteraksi dengan DBMS, yaitu:

1. *Data Definition Language* (DDL); digunakan untuk menggambarkan desain dari basis data secara keseluruhan, mulai dari membuat tabel baru, memuat indeks, maupun mengubah tabel.
2. *Data Manipulation Language* (DML); digunakan untuk memanipulasi dan mengambil data dari *database*, menghapus data dari *database*, dan mengubah data pada suatu *database* [3].

Adapun komponen dari DBMS adalah sebagai berikut

- File Manager adalah mengelola ruang didalam suatu disk dan struktur data yang dipakai untuk mempresentasikan informasi yang disimpan didalam suatu disk.
- Database Manager adalah menyediakan interface antar data low-level yang terdapat pada basis data dengan program aplikasi serta query yang diberikan ke suatu sistem.
- Query Processor adalah menterjemahkan perintah dalam bahasa query ke instruksi low-level yang dapat dimengerti database manager.
- DML Precompiler adalah mengkonversi pernyataan atau perintah DML yang ditambahkan dalam suatu program aplikasi kepada pemain prosedur normal dalam bahasa induk.
- DDL Compiler adalah mengkonversi berbagai perintah DDL ke dalam sekumpulan tabel yang mengandung meta data [4].

2.2 Tuning Indexing

2.2.1 Tuning

Tuning adalah suatu tindakan untuk menemukan sesuatu yang menjadi penyebab masalah dan membuat suatu perbaikan yang diperlukan untuk mengurangi dampak dari masalah tersebut. Sesuai dengan pengertiannya tuning menurut Chan dan Ashdown (2009,p22) yaitu tindakan mengidentifikasi hambatan-hambatan paling signifikan dan membuat perubahan yang diperlukan untuk mengurangi dampak dari permasalahan tersebut. Ada beberapa factor yang dapat digunakan untuk mengukur efisiensi, yaitu:

a. Transaction Throughput

Yaitu berapa banyak transaksi yang dapat diproses dalam jangka waktu yang diberikan. Contoh: sebanyak 100 transaksi dapat diproses dalam waktu satu jam. Semakin banyak transaksi yang diproses dalam jangka waktu satu jam, semakin tinggi throughputnya.

b. Response Time

Response time adalah waktu yang digunakan untuk menyelesaikan satu transaksi. Dari titik pandang seorang pemakai, semakin singkat response time-nya semakin baik. Bagaimanapun, ada beberapa faktor mempengaruhi response time yang tidak bisa dikontrol oleh perancang seperti pada sistem loading. Response time dapat diminimalisasi dengan cara mengurangi waktu selisih dan waktu tunggu (contention dan wait times) terutama waktu tunggu disk I/O, mengurangi jumlah waktu yang diperlukan sumber daya, menggunakan komponen yang lebih cepat.

c. Penyimpanan Disk

Yaitu jumlah media penyimpanan yang dibutuhkan pada suatu disk. Semakin minim penggunaan media penyimpanan, maka akan semakin efisien.

2.2.2 Indexing

Menurut Ashdown dan Kyte (2011,p61), Index adalah sebuah struktur optional yang diasosiasikan dengan tabel atau table cluster yang dapat mempercepat

akses data. Sedangkan menurut Connolly dan Begg (2010,p242) index adalah sebuah struktur yang menyediakan percepatan akses kepada sebuah baris dalam tabel didasari pada nilai kesimpulan bahwa index adalah sebuah struktur yang digunakan untuk mempercepat akses data dan didasari dari sebuah nilai. Oleh karena itu,jika suatu tabel dengan jumlah data yang besar tidak menggunakan index, maka database harus melakukan full table scan dalam mencari nilai yang akan menyebabkan waktu akan semakin bertambah seiring dengan meningkatnya volume data. Adapun karakteristik index adalah sebagai berikut:

a. Usability

Index dapat berupa usable ataupun unusable, dimana suatu unusable index tidak diatur oleh operasi DML dan diabaikan oleh optimizer sehingga meningkatkan performa bulk loads yaitu kejadian dimana dilakukannya proses entry data dalam jumlah besar ke dalam tabel dimana jika index sedang dalam kondisi usable dapat memperlambat proses entry data tersebut. Selain itu penggunaan Unusable Index dan Index partitions tidak mengkonsumsi space dan ketika suatu Index dibuat menjadi unusable, database akan melakukan proses drop pada Index Segment.

b. Visibility

Index dapat berupa visible atau invisible, dimana suatu invisible Index akan diatur oleh operasi DML dan tidak digunakan secara default oleh optimizer. Selain itu suatu invisible Index merupakan suatu alternatif untuk membuat Index tersebut menjadi unusable atau men-drop nya. Kemudian, Invisible Index sangat berguna untuk melakukan testing terhadap suatu Index sebelum Index tersebut di drop atau digunakan secara temporer tanpa mengganggu kinerja aplikasi [5].

BAB III

HASIL DAN PEMBAHASAN

3.1 Hasil

- Data 1 : advisor=100, student = 100, section = 200,takes = 200

Tabel 1. Data 1

NO	Query	Sebelum Tuning	Sesudah Tunning Config	Sesudah Tunning Indexing
1	Q1	0,05 detik	0,00 detik	0,00 detik
2	Q2	0,04 detik	0,00 detik	0,00 detik
3	Q3	0,00 detik	0,00 detik	0,00 detik
4	Q4	0,00 detik	0,00 detik	0,00 detik
5	Q5	0,00 detik	0,00 detik	0,00 detik

- Data 2 : advisor = 200, student = 200, section = 400,takes = 400

Tabel 2. Data 2

NO	Query	Sebelum Tuning	Sesudah Tunning Config	Sesudah Tunning Indexing
1	Q1	0,00 detik	0,00 detik	0,00 detik
2	Q2	0,00 detik	0,00 detik	0,00 detik
3	Q3	0,00 detik	0,00 detik	0,00 detik
4	Q4	0,00 detik	0,00 detik	0,01 detik
5	Q5	0,02 detik	0,00 detik	0,00 detik

- Data 3 : advisor = 500, student = 500, section = 1000,takes = 1000

Tabel 3. Data 3

NO	Query	Sebelum Tuning	Sesudah Tuning Config	Sesudah Tuning Indexing
1	Q1	0,00 detik	0,00 detik	0,00 detik
2	Q2	0,00 detik	0,00 detik	0,00 detik
3	Q3	0,00 detik	0,00 detik	0,00 detik
4	Q4	0,03 detik	0,03 detik	0,04 detik
5	Q5	0,02 detik	0,02 detik	0,02 detik

- Data 4 : advisor = 700, student = 700, section = 20000,takes = 20000

Tabel 4. Data 4

NO	Query	Sebelum Tuning	Sesudah Tuning Config	Sesudah Tuning Indexing
1	Q1	0,00 detik	0,00 detik	0,00 detik
2	Q2	0,04 detik	0,00 detik	0,00 detik
3	Q3	0,00 detik	0,00 detik	0,00 detik
4	Q4	0,20 detik	0,20 detik	0,19 detik
5	Q5	0,17detik	0,11 detik	0,11 detik

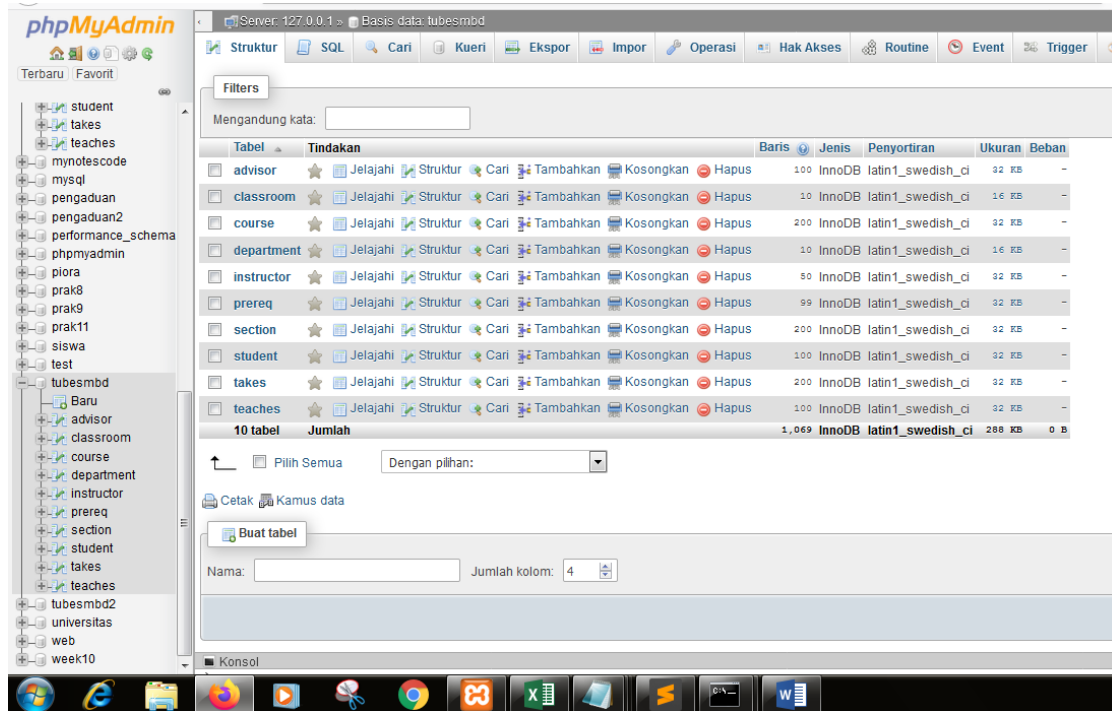
Percobaan yang dilakukan hanya bisa sampai data yang ke 4 yaitu : advisor = 700, student = 700, section = 20000,takes = 20000. Hal ini disebabkan spesifikasi laptop yang kurang memadai untuk melakukan tuning data yang ratusan ribu bahkan sampai puluhan juta. Dan hal ini menyebabkan running memakan waktu yang berjam-jam sehingga tuning tidak terselesaikan.

3.2 Pembahasan

Hasil diatas didapatkan dari hasil Tuning database. Dimana manfaat tuning salah satunya yaitu meningkatkan performance database itu sendiri. Pada percobaan kali ini

terdapat 7 data. Dimana masing-masing data terdapat nilai dari beberapa tabel yang berada didalam database.

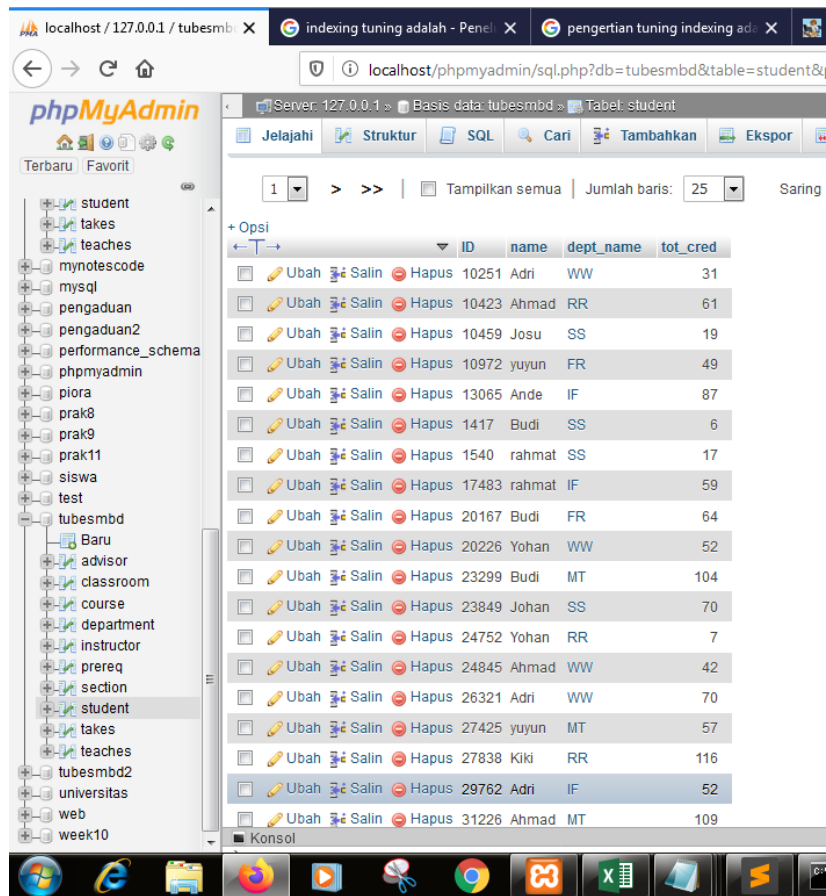
1. Langkah pertama kita harus membuat database baru terlebih dahulu dengan mengisi beberapa tabel yang telah diberikan.



Gambar 1. Database

Gambar diatas merupakan salah satu sample dari database yang telah dibuat dan berisi beberapa tabel yang telah diberikan.

2. Selanjutnya adalah mengetikkan perintah “javac tableGen.java” dan “java tableGen” di dalam command prompt, dan kemudian menginsrtkan data didalam table-table yang telah ada.



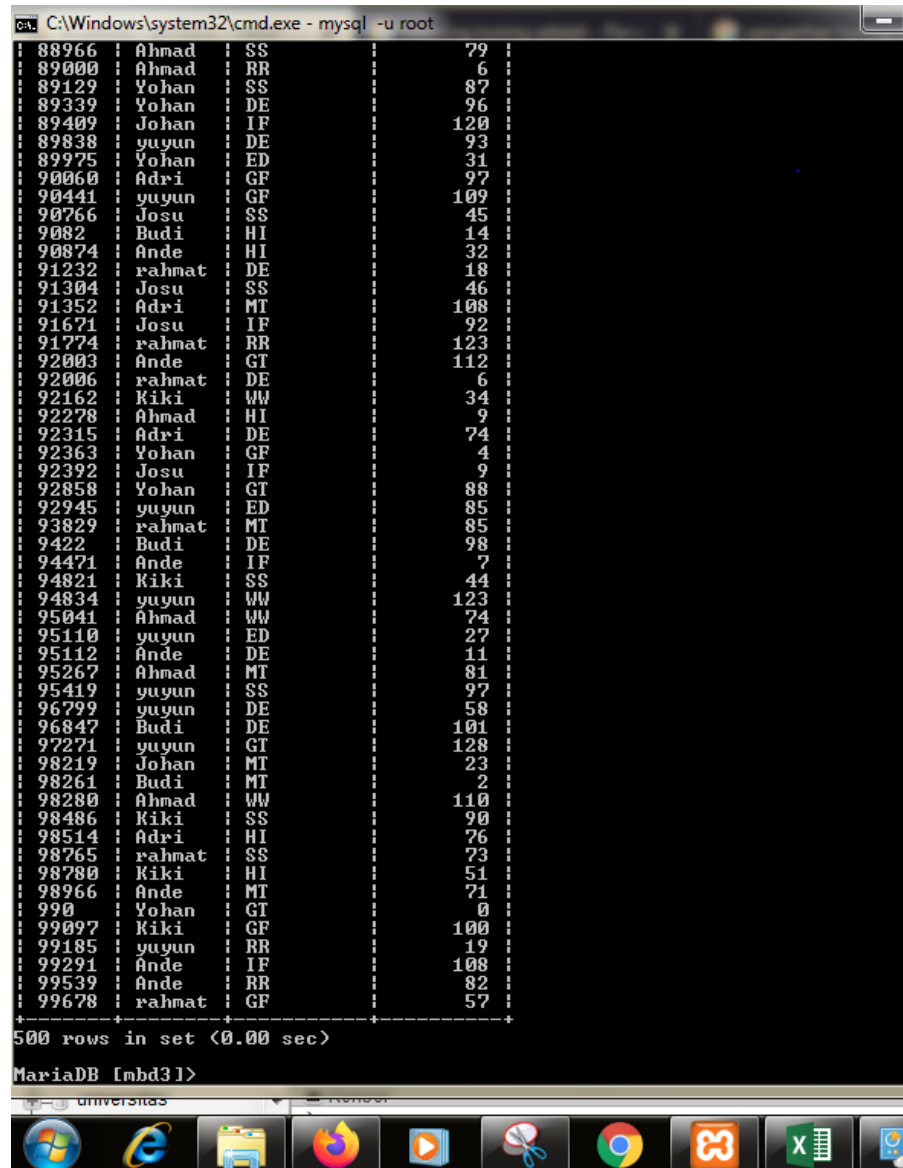
Gambar 2. Insert Tabel Database

Gambar diatas salah satu sampel memasukkan data ke dalam tabel-tabel yang ada di database.

3. Langkah selanjutnya adalah melakukan running pada query-query yang telah disediakan.

1. Sebelum Tunning

- Data 3 : advisor = advisor = 500, student = 500, section = 1000,takes = 1000
- Query 1 : SELECT * FROM student;



88966	Ahmad	SS	79
89000	Ahmad	RR	6
89129	Yohan	SS	87
89339	Yohan	DE	96
89409	Johan	IF	120
89838	yuyun	DE	93
89975	Yohan	ED	31
90060	Adri	GF	97
90441	yuyun	GF	109
90766	Josu	SS	45
9082	Budi	HI	14
90874	Ande	HI	32
91232	rahmat	DE	18
91304	Josu	SS	46
91352	Adri	MT	108
91671	Josu	IF	92
91774	rahmat	RR	123
92003	Ande	GT	112
92006	rahmat	DE	6
92162	Kiki	WW	34
92278	Ahmad	HI	9
92315	Adri	DE	74
92363	Yohan	GF	4
92392	Josu	IF	9
92858	Yohan	GT	88
92945	yuyun	ED	85
93829	rahmat	MT	85
9422	Budi	DE	98
94471	Ande	IF	7
94821	Kiki	SS	44
94834	yuyun	WW	123
95041	Ahmad	WW	74
95110	yuyun	ED	27
95112	Ande	DE	11
95267	Ahmad	MT	81
95419	yuyun	SS	97
96799	yuyun	DE	58
96847	Budi	DE	101
97271	yuyun	GT	128
98219	Johan	MT	23
98261	Budi	MT	2
98280	Ahmad	WW	110
98486	Kiki	SS	90
98514	Adri	HI	76
98765	rahmat	SS	73
98780	Kiki	HI	51
98966	Ande	MT	71
990	Yohan	GT	0
99097	Kiki	GF	100
99185	yuyun	RR	19
99291	Ande	IF	108
99539	Ande	RR	82
99678	rahmat	GF	57

500 rows in set (0.00 sec)

MariaDB [mhd31]>

Gambar 3. Running Data 3 dengan Query 1 Sebelum Tunning

Gambar diatas adalah salah satu sampel hasil running yang diambil dari data 3 dengan query 1 sebelum dilakukan tuning, dan menghasilkan waktu 0,00 detik.

- Data 3 : advisor = advisor = 500, student = 500, section = 1000,takes = 1000
- Query 2 : SELECT * FROM student WHERE tot_cred > 30;

Ca. C:\Windows\system32\cmd.exe - mysql -u root

86680	Ande	GF	54
86766	Budi	ED	77
87019	Budi	IF	115
87238	Kiki	WW	72
8733	yuyun	MT	94
87331	Johan	WW	63
8736	Ahmad	MT	75
87365	Ahmad	GT	129
87425	Adri	SS	43
87635	Johan	SS	123
87749	Yohan	GF	70
87762	yuyun	MT	64
87841	Ahmad	RR	86
88308	Yohan	WW	64
88966	Ahmad	SS	79
89129	Yohan	SS	87
89339	Yohan	DE	96
89409	Johan	IF	120
89838	yuyun	DE	93
89975	Yohan	ED	31
90060	Adri	GF	97
90441	yuyun	GF	109
90766	Josu	SS	45
90874	Ande	HI	32
91304	Josu	SS	46
91352	Adri	MT	108
91671	Josu	IF	92
91774	rahmat	RR	123
92003	Ande	GT	112
92162	Kiki	WW	34
92315	Adri	DE	74
92858	Yohan	GT	88
92945	yuyun	ED	85
93829	rahmat	MT	85
9422	Budi	DE	98
94821	Kiki	SS	44
94834	yuyun	WW	123
95041	Ahmad	WW	74
95267	Ahmad	MT	81
95419	yuyun	SS	97
96799	yuyun	DE	58
96847	Budi	DE	101
97271	yuyun	GT	128
98280	Ahmad	WW	110
98486	Kiki	SS	90
98514	Adri	HI	76
98765	rahmat	SS	73
98780	Kiki	HI	51
98966	Ande	MT	71
99097	Kiki	GF	100
99291	Ande	IF	108
99539	Ande	RR	82
99678	rahmat	GF	57

385 rows in set (0.00 sec)

MariaDB [mbd31]

Gambar 4. Running Data 3 dengan Query 2 Sebelum Tunning
 Gambar diatas salah satu sampel data 3 dirunning dengan query 2 sebelum dilakukan tuning, dan menghasilkan waktu sebanyak 0,00 detik.

- Data 3 : advisor = advisor = 500, student = 500, section = 1000,takes = 1000
- Query 3 : SELECT `name`, dept_name FROM student WHERE tot_cred > 30;

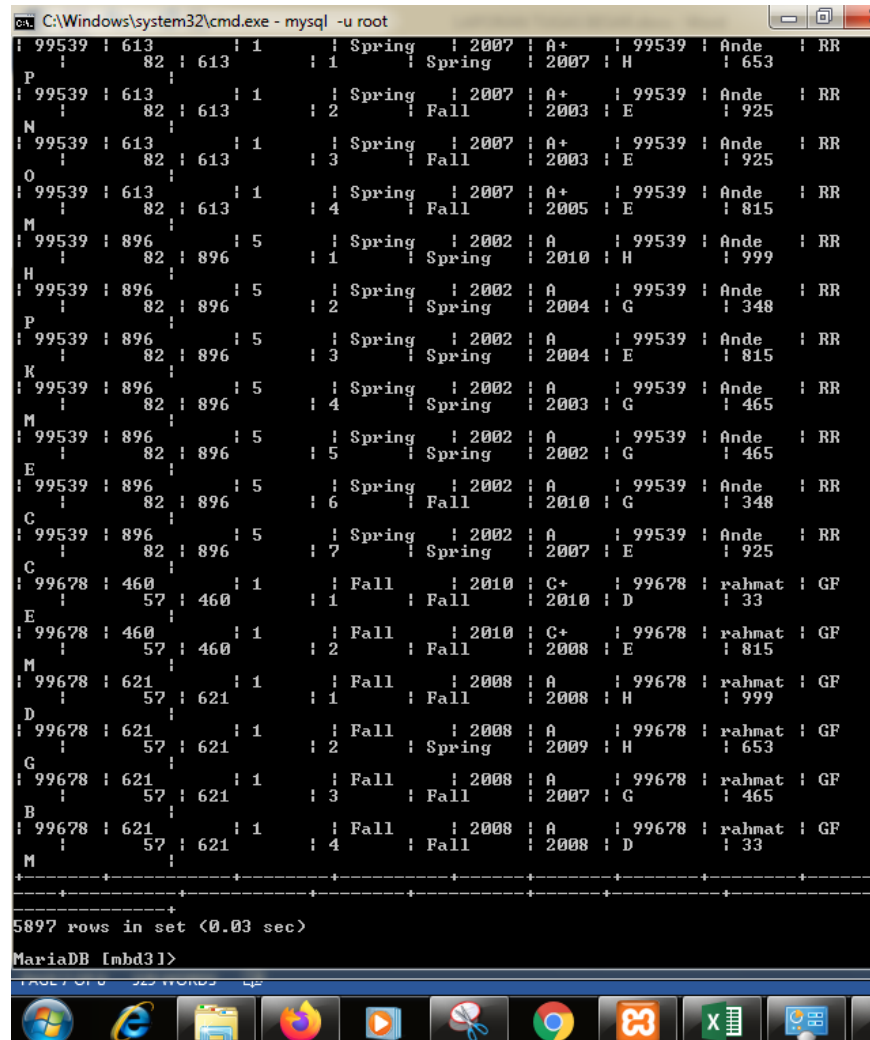
```
C:\Windows\system32\cmd.exe - mysql -u root

+-----+
| Ande   | GF |
| Budi   | ED |
| Budi   | IF |
| Kiki    | WW |
| yuyun   | MT |
| Johan   | WW |
| Ahmad   | MT |
| Ahmad   | GT |
| Adri    | SS |
| Johan   | SS |
| Johan   | GF |
| yuyun   | MT |
| Ahmad   | RR |
| Johan   | WW |
| Ahmad   | SS |
| Johan   | SS |
| Johan   | DE |
| Johan   | IF |
| yuyun   | DE |
| Johan   | ED |
| Adri    | GF |
| yuyun   | GF |
| Josu    | SS |
| Ande    | HI |
| Josu    | SS |
| Adri    | MT |
| Josu    | IF |
| rahmat  | RR |
| Ande    | GT |
| Kiki    | WW |
| Adri    | DE |
| Johan   | GT |
| yuyun   | ED |
| rahmat  | MT |
| Budi    | DE |
| Kiki    | SS |
| yuyun   | WW |
| Ahmad   | WW |
| Ahmad   | MT |
| yuyun   | SS |
| yuyun   | DE |
| Budi    | DE |
| yuyun   | GT |
| Ahmad   | WW |
| Kiki    | SS |
| Adri    | HI |
| rahmat  | SS |
| Kiki    | HI |
| Ande    | MT |
| Kiki    | GF |
| Ande    | IF |
| Ande    | RR |
| rahmat  | GF |
+-----+
385 rows in set (0.00 sec)
```

Gambar 5. Hasil Running Data 3 dengan Query 3 sebelum Tunning
Gambar diatas sala satu sampel data 3 dirunning dengan query 3 sebelum tuning dan

menghasilkan waktu sebanyak 0,00 detik.

- Data 3 : advisor = advisor = 500, student = 500, section = 1000,takes = 1000
- `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id;`



Gambar 6. Hasil Running Data 3 dan Query 4 sebelum Tunning

Gambar diatas salah satu sampel hasil running data 3 dengan query 4 sebelum tuning dan menghasilkan waktu sebanyak 0,03 detik.

- Data 3 : advisor = advisor = 500, student = 500, section = 1000,takes = 1000
- `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;

Kiki	GF	5	Spring	712	E	891
Kiki	GF	5	Spring	712	E	891
Yohan	RR	7	Fall	653	H	891
Yohan	RR	7	Fall	999	H	891
Yohan	RR	7	Fall	999	H	891
Yohan	RR	7	Fall	925	E	891
Yohan	RR	7	Fall	925	E	891
Yohan	RR	7	Fall	815	E	891
Yohan	RR	7	Fall	712	E	891
Yohan	RR	7	Fall	712	E	891
Ande	VW	7	Fall	653	H	891
Ande	VW	7	Fall	999	H	891
Ande	VW	7	Fall	999	H	891
Ande	VW	7	Fall	925	E	891
Ande	VW	7	Fall	925	E	891
Ande	VW	7	Fall	815	E	891
Ande	VW	7	Fall	712	E	891
Ande	VW	7	Fall	712	E	891
Adri	IF	7	Fall	653	H	891
Adri	IF	7	Fall	999	H	891
Adri	IF	7	Fall	999	H	891
Adri	IF	7	Fall	925	E	891
Adri	IF	7	Fall	925	E	891
Adri	IF	7	Fall	815	E	891
Adri	IF	7	Fall	712	E	891
Adri	IF	7	Fall	712	E	891

5897 rows in set (0.02 sec)

MariaDB [mbd31]>

Gambar 7. Hasil Running Data 3 dengan Query 5 sebelum Tunning
Gambar diatas salah satu sampel hasil running data 3 dengan query 5 sebelum tuning dan menghasilkan waktu sebanyak 0,02 detik.

2. Setelah Tunning Config

- Data 3 : advisor = 500, student = 500, section = 1000,takes = 1000
- Query 1 : SELECT * FROM student;

```
C:\Windows\system32\cmd.exe - mysql -u root
+-----+-----+-----+-----+
88966 | Ahmad | SS | 79 |
89000 | Ahmad | RR | 6 |
89129 | Yohan | SS | 87 |
89339 | Yohan | DE | 96 |
89409 | Johan | IF | 120 |
89838 | yuyun | DE | 93 |
89975 | Yohan | ED | 31 |
90060 | Adri | GF | 97 |
90441 | yuyun | GF | 109 |
90766 | Josu | SS | 45 |
9082 | Budi | HI | 14 |
90874 | Ande | HI | 32 |
91232 | rahmat | DE | 18 |
91304 | Josu | SS | 46 |
91352 | Adri | MT | 108 |
91671 | Josu | IF | 92 |
91774 | rahmat | RR | 123 |
92003 | Ande | GT | 112 |
92006 | rahmat | DE | 6 |
92162 | Kiki | WW | 34 |
92278 | Ahmad | HI | 9 |
92315 | Adri | DE | 74 |
92363 | Yohan | GF | 4 |
92392 | Josu | IF | 9 |
92858 | Yohan | GT | 88 |
92945 | yuyun | ED | 85 |
93829 | rahmat | MT | 85 |
9422 | Budi | DE | 98 |
94471 | Ande | IF | 7 |
94821 | Kiki | SS | 44 |
94834 | yuyun | WW | 123 |
95041 | Ahmad | WW | 74 |
95110 | yuyun | ED | 27 |
95112 | Ande | DE | 11 |
95267 | Ahmad | MT | 81 |
95419 | yuyun | SS | 97 |
96799 | yuyun | DE | 58 |
96847 | Budi | DE | 101 |
97271 | yuyun | GT | 128 |
98219 | Johan | MT | 23 |
98261 | Budi | MT | 2 |
98280 | Ahmad | WW | 110 |
98486 | Kiki | SS | 90 |
98514 | Adri | HI | 76 |
98765 | rahmat | SS | 73 |
98780 | Kiki | HI | 51 |
98966 | Ande | MT | 71 |
990 | Yohan | GT | 0 |
99097 | Kiki | GF | 100 |
99185 | yuyun | RR | 19 |
99291 | Ande | IF | 108 |
99539 | Ande | RR | 82 |
99678 | rahmat | GF | 57 |
+-----+-----+-----+-----+
500 rows in set (0.00 sec)

MariaDB [mbd3]>
```

Gambar 8. Hasil Running Data 3 dengan Query 1 setelah tuning config
Gambar diatas salah satu sampel hasil running data 3 dengan query 1 sesudah tuning config dan menghasilkan waktu sebanyak 0,00 detik.

- Data 3 : advisor = 500, student = 500, section = 1000,takes = 1000
- Query 2 : SELECT * FROM student WHERE tot_cred > 30;

```

C:\Windows\system32\cmd.exe - mysql -u root

```

86680	Ande	GF	54
86766	Budi	ED	77
87019	Budi	IF	115
87238	Kiki	WW	72
87233	yuyun	MT	94
87331	Johan	WW	63
8736	Ahnad	MT	75
87365	Ahnad	GT	129
87425	Adri	SS	43
87635	Johan	SS	123
87749	Yohan	GF	70
87762	yuyun	MT	64
87841	Ahnad	RR	86
88308	Yohan	WW	64
88966	Ahnad	SS	79
89129	Yohan	SS	87
89339	Yohan	DE	96
89409	Johan	IF	120
89838	yuyun	DE	93
89975	Yohan	ED	31
90060	Adri	GF	97
90441	yuyun	GF	109
90766	Josu	SS	45
90874	Ande	HI	32
91304	Josu	SS	46
91352	Adri	MT	108
91671	Josu	IF	92
91774	rahmat	RR	123
92003	Ande	GT	112
92162	Kiki	WW	34
92315	Adri	DE	74
92858	Yohan	GT	88
92945	yuyun	ED	85
93829	rahmat	MT	85
9422	Budi	DE	98
94821	Kiki	SS	44
94834	yuyun	WW	123
95041	Ahnad	WW	74
95267	Ahnad	MT	81
95419	yuyun	SS	97
96799	yuyun	DE	58
96847	Budi	DE	101
97271	yuyun	GT	128
98280	Ahnad	WW	110
98486	Kiki	SS	90
98514	Adri	HI	76
98765	rahmat	SS	73
98780	Kiki	HI	51
98966	Ande	MT	71
99097	Kiki	GF	100
99291	Ande	IF	108
99539	Ande	RR	82
99678	rahmat	GF	57

385 rows in set (0.00 sec)

MariaDB [mbd31]

Gambar 9. Hasil Running Data 3 dengan Query 2 setelah tuning config
 Gambar diatas salah satu sampel hasil running data 3 dengan query 2 sesudah tuning config dan menghasilkan waktu sebanyak 0,00 detik.

- Data 3 : advisor = 500, student = 500, section = 1000,takes = 1000
- Query 3 : SELECT `name`, dept_name FROM student WHERE tot_cred > 30;

```

+-----+-----+
| Adri   | GF   |
| yuyun  | GF   |
| Josu   | SS   |
| Ande   | HI   |
| Josu   | SS   |
| Adri   | MT   |
| Josu   | IF   |
| rahmat | RR   |
| Ande   | GT   |
| Kiki   | WW   |
| Adri   | DE   |
| Yohan  | GT   |
| yuyun  | ED   |
| rahmat | MT   |
| Budi   | DE   |
| Kiki   | SS   |
| yuyun  | WW   |
| Ahmad  | WW   |
| Ahmad  | MT   |
| yuyun  | SS   |
| yuyun  | DE   |
| Budi   | DE   |
| yuyun  | GT   |
| Ahmad  | WW   |
| Kiki   | SS   |
| Adri   | HI   |
| rahmat | SS   |
| Kiki   | HI   |
| Ande   | MT   |
| Kiki   | GF   |
| Ande   | IF   |
| Ande   | RR   |
| rahmat | GF   |
+-----+-----+
385 rows in set (0.00 sec)

MariaDB [mbd3]>

```

Gambar 10. Hasil Running Data 3 dengan query 3 setelah Tunning Config

Gambar diatas salah satu sampel hasil running data 3 dengan query 3 sesudah tuning config dan menghasilkan waktu sebanyak 0,00 detik.

- Data 3 : advisor = 500, student = 500, section = 1000,takes = 1000
- Query 4 : SELECT * FROM takes JOIN student ON takes.ID = student.ID
JOIN section ON takes.course_id = section.course_id;

K	99539	896		5		Spring	2002	A	99539	Ande	RR
		82	896		3	Spring	2004	E		815	
	99539	896		5		Spring	2002	A	99539	Ande	RR
		82	896		4	Spring	2003	G		465	
M	99539	896		5		Spring	2002	A	99539	Ande	RR
		82	896		5	Spring	2002	G		465	
	99539	896		5		Spring	2002	A	99539	Ande	RR
		82	896		6	Fall	2010	G		348	
E	99539	896		5		Spring	2002	A	99539	Ande	RR
		82	896		7	Spring	2007	E		925	
	99539	896		5		Spring	2002	A	99539	Ande	RR
		82	896		1	Fall	2010	C+	99678	rahmat	GF
C	99678	460		1		Fall	2010	D	99678	33	
		57	460		2	Fall	2008	E		815	
	99678	460		1		Fall	2010	C+	99678	rahmat	GF
		57	460		1	Fall	2008	A	99678	rahmat	GF
M	99678	621		1		Fall	2008	A	99678	rahmat	GF
		57	621		2	Spring	2009	H		653	
	99678	621		1		Fall	2008	A	99678	rahmat	GF
		57	621		3	Fall	2007	G		465	
D	99678	621		1		Fall	2008	A	99678	rahmat	GF
		57	621		4	Fall	2008	D	99678	33	
	99678	621		1		Fall	2008	A	99678	rahmat	GF
		57	621								
G	99678	621		1		Fall	2008	A	99678	rahmat	GF
		57	621								
	99678	621		1		Fall	2008	A	99678	rahmat	GF
		57	621								
B	99678	621		1		Fall	2008	A	99678	rahmat	GF
		57	621								
	99678	621		1		Fall	2008	A	99678	rahmat	GF
		57	621								
M	99678	621		1		Fall	2008	A	99678	rahmat	GF
		57	621								
	99678	621		1		Fall	2008	A	99678	rahmat	GF
		57	621								

Gambar 11. Hasil Running Data 3 dengan query 4 setelah Tunning Config

Gambar diatas salah satu sampel hasil running data 3 dengan query 4 sesudah tuning config dan menghasilkan waktu sebanyak 0,03 detik.

- Data 3 : advisor = 500, student = 500, section = 1000,takes = 1000
- Query 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;

```

+----+-----+-----+-----+-----+-----+-----+
| WW | Ande | WW | 7 | Fall | 653 | H | 891 |
| WW | Ande | WW | 7 | Fall | 999 | H | 891 |
| WW | Ande | WW | 7 | Fall | 999 | H | 891 |
| WW | Ande | WW | 7 | Fall | 925 | E | 891 |
| WW | Ande | WW | 7 | Fall | 925 | E | 891 |
| WW | Ande | WW | 7 | Fall | 815 | E | 891 |
| WW | Ande | WW | 7 | Fall | 712 | E | 891 |
| WW | Ande | WW | 7 | Fall | 712 | E | 891 |
| WW | Adri | IF | 7 | Fall | 653 | H | 891 |
| WW | Adri | IF | 7 | Fall | 999 | H | 891 |
| WW | Adri | IF | 7 | Fall | 999 | H | 891 |
| WW | Adri | IF | 7 | Fall | 925 | E | 891 |
| WW | Adri | IF | 7 | Fall | 925 | E | 891 |
| WW | Adri | IF | 7 | Fall | 815 | E | 891 |
| WW | Adri | IF | 7 | Fall | 712 | E | 891 |
| WW | Adri | IF | 7 | Fall | 712 | E | 891 |
+----+-----+-----+-----+-----+-----+-----+
5897 rows in set (0.02 sec)
MariaDB [mbd3]>

```

Gambar 12. Hasil Running Data 3 dengan query 5 setelah Tunning Config

Gambar diatas salah satu sampel hasil running data 3 dengan query 5 sesudah tuning config dan menghasilkan waktu sebanyak 0,02 detik.

3. Setelah Indexing

- Data 3 : advisor = 500, student = 500, section = 1000,takes = 1000
- Query 1 : SELECT * FROM student;

```

92162 Kiki WW 34
92278 Ahmad HI 9
92315 Adri DE 74
92363 Yohan GF 4
92392 Josu IF 9
92858 Yohan GT 88
92945 yuyun ED 85
93829 rahmat MT 85
9422 Budi DE 98
94471 Ande IF 7
94821 Kiki SS 44
94834 yuyun WW 123
95041 Ahmad WW 74
95110 yuyun ED 27
95112 Ande DE 11
95267 Ahmad MT 81
95419 yuyun SS 97
96799 yuyun DE 58
96847 Budi DE 101
97271 yuyun GT 128
98219 Johan MT 23
98261 Budi MT 2
98280 Ahmad WW 110
98486 Kiki SS 90
98514 Adri HI 76
98765 rahmat SS 73
98780 Kiki HI 51
98966 Ande MT 71
990 Yohan GT 0
99097 Kiki GF 100
99185 yuyun RR 19
99291 Ande IF 108
99539 Ande RR 82
99678 rahmat GF 57
500 rows in set (0.00 sec)
MariaDB [mbd3]>

```

Gambar 13. Hasil Running Data 3 dengan query 1 setelah indexing
Gambar diatas salah satu sampel hasil running data 3 dengan query 1 sesudah indexing dan menghasilkan waktu sebanyak 0,00 detik.

- Data 3 : advisor = 500, student = 500, section = 1000,takes = 1000
- Query 2 : SELECT * FROM student WHERE tot_cred > 30;

90766	Josu	SS	45
90874	Ande	HI	32
91304	Josu	SS	46
91352	Adri	MT	108
91671	Josu	IF	92
91774	rahmat	RR	123
92003	Ande	GT	112
92162	Kiki	WW	34
92315	Adri	DE	74
92858	Yohan	GT	88
92945	yuyun	ED	85
93829	rahmat	MT	85
9422	Budi	DE	98
94821	Kiki	SS	44
94834	yuyun	WW	123
95041	Ahmad	WW	74
95267	Ahmad	MT	81
95419	yuyun	SS	97
96799	yuyun	DE	58
96847	Budi	DE	101
97271	yuyun	GT	128
98280	Ahmad	WW	110
98486	Kiki	SS	90
98514	Adri	HI	76
98765	rahmat	SS	73
98780	Kiki	HI	51
98966	Ande	MT	71
99097	Kiki	GF	100
99291	Ande	IF	108
99539	Ande	RR	82
99678	rahmat	GF	57

385 rows in set (0.00 sec)

MariaDB [mhd31]>

Gambar 14. Hasil Running Data 3 dengan query 2 setelah indexing
 Gambar diatas salah satu sampel hasil running data 3 dengan query 2 sesudah indexing dan menghasilkan waktu sebanyak 0,00 detik.

- Data 3: advisor = 500, student = 500, section = 1000,takes = 1000
- Query 3 : SELECT `name`, dept_name FROM student WHERE tot_cred > 30;

Josu	SS
Ande	HI
Josu	SS
Adri	MT
Josu	IF
rahmat	RR
Ande	GT
Kiki	WW
Adri	DE
Yohan	GT
yuyun	ED
rahmat	MT
Budi	DE
Kiki	SS
yuyun	WW
Ahmad	WW
Ahmad	MT
yuyun	SS
yuyun	DE
Budi	DE
yuyun	GT
Ahmad	WW
Kiki	SS
Adri	HI
rahmat	SS
Kiki	HI
Ande	MT
Kiki	GF
Ande	IF
Ande	RR
rahmat	GF

385 rows in set (0.00 sec)

MariaDB [mbd31]

Gambar 15. Hasil Running Data 3 dengan query 3 setelah indexing
 Gambar diatas salah satu sampel hasil running data 3 dengan query 3 sesudah indexing dan menghasilkan waktu sebanyak 0,00 detik.

- Data 3: advisor = 500, student = 500, section = 1000,takes = 1000
- Query 4 : SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id;

M	99539	613	82	613	1	4	Spring	Fall	2007	A+	2005	E	99539	Ande	815	RR
H	99539	896	82	896	5	1	Spring	Spring	2002	A	2010	H	99539	Ande	999	RR
P	99539	896	82	896	5	2	Spring	Spring	2002	A	2004	G	99539	Ande	348	RR
K	99539	896	82	896	5	3	Spring	Spring	2002	A	2004	E	99539	Ande	815	RR
M	99539	896	82	896	5	4	Spring	Spring	2002	A	2003	G	99539	Ande	465	RR
E	99539	896	82	896	5	5	Spring	Spring	2002	A	2002	G	99539	Ande	465	RR
C	99539	896	82	896	5	6	Spring	Fall	2002	A	2010	G	99539	Ande	348	RR
C	99539	896	82	896	5	7	Spring	Spring	2002	A	2007	E	99539	Ande	925	RR
E	99678	460	57	460	1	1	Fall	Fall	2010	C+	2010	D	99678	rahmat	33	GF
M	99678	460	57	460	1	2	Fall	Fall	2010	C+	2008	E	99678	rahmat	815	GF
D	99678	621	57	621	1	1	Fall	Fall	2008	A	2008	H	99678	rahmat	999	GF
G	99678	621	57	621	1	2	Fall	Spring	2008	A	2009	H	99678	rahmat	653	GF
B	99678	621	57	621	1	3	Fall	Fall	2008	A	2007	G	99678	rahmat	465	GF
M	99678	621	57	621	1	4	Fall	Fall	2008	A	2008	D	99678	rahmat	33	GF

5897 rows in set (0.04 sec)

MariaDB [mbd31]

Gambar 16. Hasil Running Data 3 dengan query 4 setelah indexing
Gambar diatas salah satu sampel hasil running data 3 dengan query 4 sesudah indexing dan menghasilkan waktu sebanyak 0,04 detik.

- Data 3: advisor = 500, student = 500, section = 1000,takes = 1000
- Query 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;

```
Yohan | RR | 7 | Fall | 712 | E | 891
Ande | WW | 7 | Fall | 653 | H | 891
Ande | WW | 7 | Fall | 999 | H | 891
Ande | WW | 7 | Fall | 999 | H | 891
Ande | WW | 7 | Fall | 925 | E | 891
Ande | WW | 7 | Fall | 925 | E | 891
Ande | WW | 7 | Fall | 815 | E | 891
Ande | WW | 7 | Fall | 712 | E | 891
Ande | WW | 7 | Fall | 712 | E | 891
Adri | IF | 7 | Fall | 653 | H | 891
Adri | IF | 7 | Fall | 999 | H | 891
Adri | IF | 7 | Fall | 925 | E | 891
Adri | IF | 7 | Fall | 925 | E | 891
Adri | IF | 7 | Fall | 815 | E | 891
Adri | IF | 7 | Fall | 712 | E | 891
Adri | IF | 7 | Fall | 712 | E | 891
5897 rows in set (0.02 sec)
MariaDB [mbd31]
```

Gambar 17. Hasil Running Data 3 dengan query 5 setelah indexing

Gambar diatas salah satu sampel hasil running data 3 dengan query 5 sesudah indexing dan menghasilkan waktu sebanyak 0,02 detik.

BAB IV

KESIMPULAN

Berdasarkan percobaan yang telah dilakukan, bahwasanya *tunning* dapat meningkatkan *performance* dari *database*. Namun, apabila terjadi waktu yang semakin lama setelah ditunning maka hal tersebut terjadi kegagalan *tunning*. Pada percobaan yang telah dilakukan hanya bisa me-*running* sampai data ke 4, dikarenakan spesifikasi laptop yang tidak memungkinkan untuk terus melanjutkan *tunning* tersebut. Pada percobaan ke 4 *running* telah memakan waktu yang lama, *response time* nya juga sangat lambat karena data yang digunakan sangat banyak sedangkan laptop tidak memadai serta dengan *query-query* join yang menggabungkan tabel-tabel yang sangat banyak sehingga membebani kinerja dari DBMS dalam melakukan *tunning*. Waktu yang didapatkan dari percobaan ke 4 mencapai 0,11 sampai dengan 0.20 detik setelah *tunning*, ini jauh lebih lambat dari percobaan pertama sampai ketiga dengan waktu mencapai 0,00 sampai 0,04 detik setelah *tunning*.

DAFTAR PUSTAKA

- [1] A. Hamid, "Makalah Database," 2012. [Online]. Available: <https://unindrapti.wordpress.com/2012/12/04/makalah-database/>.
- [2] A. Pramono, "Pengertian Sistem Manajemen Basis Data," 2017. [Online]. Available: <https://blud.co.id/wp/2017/10/pengertian-sistem-manajemen-basis-data/>.
- [3] Maxmanroe, "Pengertian DBMS: Fungsi, Manfaat, Komponen, Cara Kerja, dan Contohnya," 2019. [Online]. Available: <https://www.maxmanroe.com/vid/teknologi/komputer/pengertian-dbms.html>.
- [4] P. M. P. Online, "Pengertian DBMS, Tujuan, Fungsi, Macam, Komponen, Keuntungan, Kerugian dan Contohnya." [Online]. Available: <https://www.seputarpengetahuan.co.id/2017/08/pengertian-dbms-database-management-system-tujuan-fungsi-macam-komponen-keuntungan-kerugian-contoh.html>.
- [5] Axsaner, "Tuning dan Indexing," 2019. [Online]. Available: <http://axsaner.blogspot.com/2015/01/tuning-dan-indexing-di-oracle.html#.Xf3fqc6yTIU>.