LAPORAN TUGAS BESAR
MANAGEMEN BASIS DATA
TUNING

Dosen Pengampu :

Ahmad Luky Ramdani, S.Kom, M.Kom.

disusun oleh :

Fatin Delfiandita (14117087)

Kelas- RA

PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA
2019

DAFTAR ISI

****

BAB I
STUDI LITERATUR

1.1 Tuning : Indexing

Index adalah sebuah objek dalam sisem database yang dapat mempercepat proses pencarian (query) data. Index adalah *daftar isi* yang dapat dipakai oleh query tanpa perlu membaca isi tabel secara langsung. Tujuan adanya index dalam database antara lain adalah untuk mempercepat pencarian data berdasarkan kolom tertentu. Tanpa adanya sebuah index pada database, kinerja databse sangat menurun dengan sangat signifikan. Hal ini dikarenakan resoude 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.

Index diperlukan pada saat ada kondisi tabel dangan besar, kolom sering digunakan sebagai kondisi dalam query, kebanyakan query menampilkan data lebih dari 2-10% dari selutuh data dan tabel jarang di perbarui. Ada beberapa perbedan clustered index dan non clustered index pada sql server yaitu dalam clustered index hanya terdiri dalam satu tabel atau hanya dapat diterapkan satu kali pada satu tabel, sedangkan pada nonculstered index boleh lebih dari satu(banyak).

1.2 Tuning : Setting Configuration DBMS

Untuk melakukan performance tuning pada PostgreSQL dengan skala bisnis dengan konfigurasi manajemen database digunakan database administrator yang akan menganalisis parameter konfigurasi database PostgreSQL dan merekomendasikan konfigurasi optimal sesuai dengan workload Anda.

BAB II
DESKRIPSI PERCOBAAN

2.1 Tunning : Indexing

Dengan menggunakan database yang sudah ditentukan dan untuk melakukan tuning index, maka diperlukan data waktu sebelum di tuning dan sesudah dituning untuk dapat membandingkan keduanya, akan dijelaskan pada deskripsi di bawah ini :

```
MariaDB [mbd]> CREATE INDEX ind_student ON student(ID, TOT_CRED) USING BTREE;
Query OK, 0 rows affected (0.47 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [mbd]> CREATE INDEX ind_TAKES ON takes(ID) USING BTREE;
Query OK, 0 rows affected (0.23 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [mbd]> CREATE INDEX ind_takes ON takes(ID,course_id) USING BTREE;
ERROR 1061 (42000): Duplicate key name 'ind_takes'
MariaDB [mbd]> CREATE INDEX ind_takes2 ON takes(ID,course_id) USING BTREE;
Query OK, 0 rows affected (0.63 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [mbd]> CREATE INDEX ind_course ON section(course_id) USING BTREE;
Query OK, 0 rows affected (0.61 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

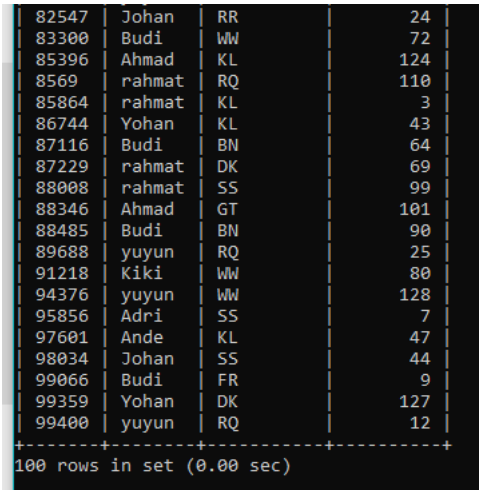
2.1.2 Data 1 (advisor = 100, student = 100, section = 200,takes = 200)

1. Dengan menggunakan query 1 (SELECT * FROM student)

   Waktu eksekusi sebelum dituning :

```
| 81639 | Ahmad   | GF         |        22 |
| 82155 | Ande    | MT         |         4 |
| 84337 | Ahmad   | DK         |       116 |
| 84701 | Kiki    | GF         |        28 |
| 85750 | Yohan   | GT         |        95 |
| 87494 | Adri    | GO         |        32 |
| 87531 | Adri    | GT         |        74 |
| 87719 | Johan   | GT         |        20 |
| 88778 | Adri    | DK         |        96 |
| 88987 | Ahmad   | MT         |        92 |
| 89069 | Yohan   | KL         |        51 |
| 90950 | Josu    | DE         |       103 |
| 90998 | Yohan   | RR         |        39 |
| 91655 | Ande    | GF         |        89 |
| 92405 | rahmat  | DK         |       129 |
| 95660 | Budi    | GF         |        75 |
| 96137 | Johan   | GF         |         0 |
| 9804  | Josu    | DE         |        29 |
| 99254 | Yohan   | SS         |        50 |
| 99845 | Yohan   | GO         |        84 |
+-------+---------+------------+-----------+
100 rows in set (0.16 sec)
```

Waktu eksekusi sesudah dituning :

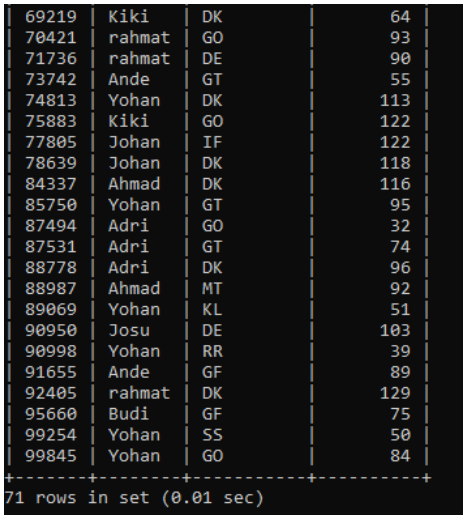


82547	Johan	RR	24
83300	Budi	WW	72
85396	Ahmad	KL	124
8569	rahmat	RQ	110
85864	rahmat	KL	3
86744	Yohan	KL	43
87116	Budi	BN	64
87229	rahmat	DK	69
88008	rahmat	SS	99
88346	Ahmad	GT	101
88485	Budi	BN	90
89688	yuyun	RQ	25
91218	Kiki	WW	80
94376	yuyun	WW	128
95856	Adri	SS	7
97601	Ande	KL	47
98034	Johan	SS	44
99066	Budi	FR	9
99359	Yohan	DK	127
99400	yuyun	RQ	12

100 rows in set (0.00 sec)

2. Dengan menggunakan query 2 (SELECT * FROM student WHERE tot_cred > 30)

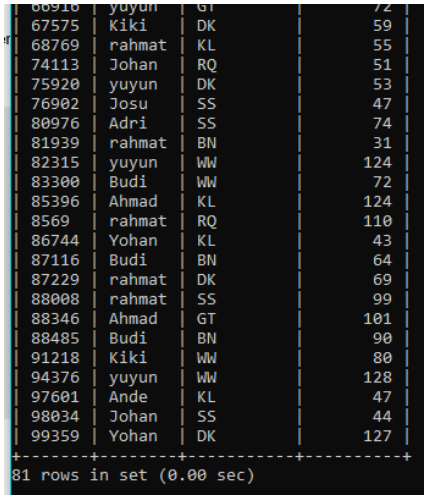
   Waktu eksekusi sebelum menggunakan tuning :



69219	Kiki	DK	64
70421	rahmat	GO	93
71736	rahmat	DE	90
73742	Ande	GT	55
74813	Yohan	DK	113
75883	Kiki	GO	122
77805	Johan	IF	122
78639	Johan	DK	118
84337	Ahmad	DK	116
85750	Yohan	GT	95
87494	Adri	GO	32
87531	Adri	GT	74
88778	Adri	DK	96
88987	Ahmad	MT	92
89069	Yohan	KL	51
90950	Josu	DE	103
90998	Yohan	RR	39
91655	Ande	GF	89
92405	rahmat	DK	129
95660	Budi	GF	75
99254	Yohan	SS	50
99845	Yohan	GO	84

71 rows in set (0.01 sec)

Sesudah menggunakan tuning :



67575	Kiki	DK	59
68769	rahmat	KL	55
74113	Johan	RQ	51
75920	yuyun	DK	53
76902	Josu	SS	47
80976	Adri	SS	74
81939	rahmat	BN	31
82315	yuyun	WW	124
83300	Budi	WW	72
85396	Ahmad	KL	124
8569	rahmat	RQ	110
86744	Yohan	KL	43
87116	Budi	BN	64
87229	rahmat	DK	69
88008	rahmat	SS	99
88346	Ahmad	GT	101
88485	Budi	BN	90
91218	Kiki	WW	80
94376	yuyun	WW	128
97601	Ande	KL	47
98034	Johan	SS	44
99359	Yohan	DK	127

81 rows in set (0.00 sec)

3. Dengan menggunakan query 3 (SELECT `name`, department FROM student WHERE tot_cred > 30)

   Sebelum menggunakan tuning :

Yohan	DK
Kiki	GO
Johan	IF
Johan	DK
Ahmad	DK
Yohan	GT
Adri	GO
Adri	GT
Adri	DK
Ahmad	MT
Yohan	KL
Josu	DE
Yohan	RR
Ande	GF
rahmat	DK
Budi	GF
Yohan	SS
Yohan	GO

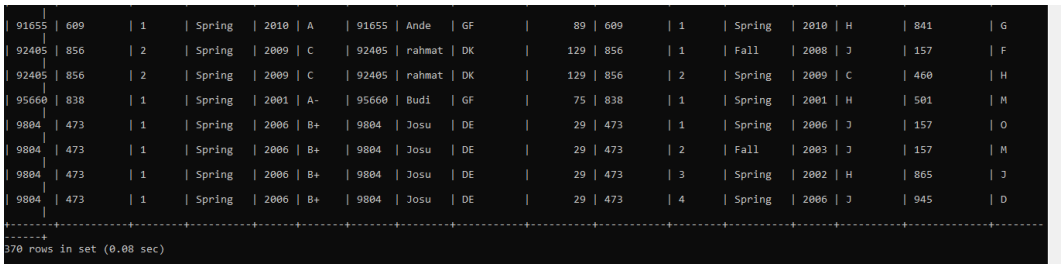
71 rows in set (0.00 sec)

Sesudah menggunakan tuning :

```
| Josu    | GO         |
| rahmat  | DF         |
| Budi    | GO         |
| Kiki    | GF         |
| rahmat  | HH         |
| yuyun   | DF         |
| Yohan   | GO         |
| Adri    | GO         |
| Johan   | HH         |
| Yohan   | MT         |
| Adri    | GO         |
| Ande    | BN         |
| Kiki    | KL         |
| Kiki    | DF         |
| Ande    | GT         |
| Josu    | DF         |
| Yohan   | MT         |
| yuyun   | HH         |
| Johan   | BN         |
| Budi    | GF         |
| yuyun   | DF         |
| Adri    | DE         |
| Ahmad   | GO         |
| Johan   | GO         |
| Josu    | HH         |
+---------+------------+
158 rows in set (0.00 sec)
```

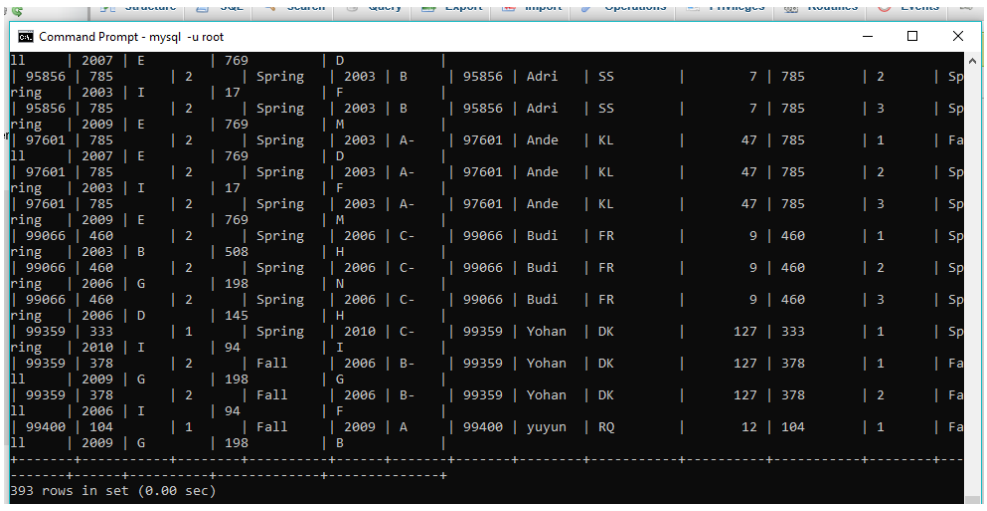
4. Dengan menggunakan query 4 (SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id)

Sebelum menggunakan tuning :



```
| 91655 | 609      | 1    | Spring | 2010 | A    | 91655 | Ande   | GF |      89 | 609      | 1    | Spring | 2010 | H    |  841 | G
| 92405 | 856      | 2    | Spring | 2009 | C    | 92405 | rahmat | DK |     129 | 856      | 1    | Fall   | 2008 | J    |  157 | F
| 92405 | 856      | 2    | Spring | 2009 | C    | 92405 | rahmat | DK |     129 | 856      | 2    | Spring | 2009 | C    |  460 | H
| 95660 | 838      | 1    | Spring | 2001 | A-   | 95660 | Budi   | GF |      75 | 838      | 1    | Spring | 2001 | H    |  501 | M
| 9804  | 473      | 1    | Spring | 2006 | B+   | 9804  | Josu   | DE |      29 | 473      | 1    | Spring | 2006 | J    |  157 | O
| 9804  | 473      | 1    | Spring | 2006 | B+   | 9804  | Josu   | DE |      29 | 473      | 2    | Fall   | 2003 | J    |  157 | M
| 9804  | 473      | 1    | Spring | 2006 | B+   | 9804  | Josu   | DE |      29 | 473      | 3    | Spring | 2002 | H    |  865 | J
| 9804  | 473      | 1    | Spring | 2006 | B+   | 9804  | Josu   | DE |      29 | 473      | 4    | Spring | 2006 | J    |  945 | D
+-------+----------+------+--------+------+------+-------+--------+----+---------+----------+------+--------+------+------+------+-------+
370 rows in set (0.08 sec)
```

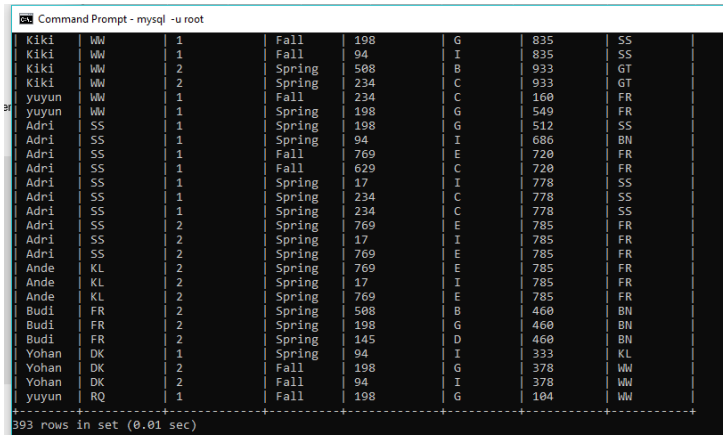
Sesudah menggunakan tuning :



```
Command Prompt - mysql  -u root
ll      | 2007 | E         | 769          | D
| 95856 | 785      | 2      | Spring   | 2003 | B    | 95856 | Adri   | SS        |       7 | 785      | 2      | Sp
ring    | 2003 | I         | 17           | F
| 95856 | 785      | 2      | Spring   | 2003 | B    | 95856 | Adri   | SS        |       7 | 785      | 3      | Sp
ring    | 2009 | E         | 769          | M
| 97601 | 785      | 2      | Spring   | 2003 | A-   | 97601 | Ande   | KL        |      47 | 785      | 1      | Fa
ll      | 2007 | E         | 769          | D
| 97601 | 785      | 2      | Spring   | 2003 | A-   | 97601 | Ande   | KL        |      47 | 785      | 2      | Sp
ring    | 2003 | I         | 17           | F
| 97601 | 785      | 2      | Spring   | 2003 | A-   | 97601 | Ande   | KL        |      47 | 785      | 3      | Sp
ring    | 2009 | E         | 769          | M
| 99066 | 460      | 2      | Spring   | 2006 | C-   | 99066 | Budi   | FR        |       9 | 460      | 1      | Sp
ring    | 2003 | B         | 508          | H
| 99066 | 460      | 2      | Spring   | 2006 | C-   | 99066 | Budi   | FR        |       9 | 460      | 2      | Sp
ring    | 2006 | G         | 198          | N
| 99066 | 460      | 2      | Spring   | 2006 | C-   | 99066 | Budi   | FR        |       9 | 460      | 3      | Sp
ring    | 2006 | D         | 145          | H
| 99359 | 333      | 1      | Spring   | 2010 | C-   | 99359 | Yohan  | DK        |     127 | 333      | 1      | Sp
ring    | 2010 | I         | 94           | I
| 99359 | 378      | 2      | Fall     | 2006 | B-   | 99359 | Yohan  | DK        |     127 | 378      | 1      | Fa
ll      | 2009 | G         | 198          | G
| 99359 | 378      | 2      | Fall     | 2006 | B-   | 99359 | Yohan  | DK        |     127 | 378      | 2      | Fa
ll      | 2006 | I         | 94           | F
| 99400 | 104      | 1      | Fall     | 2009 | A    | 99400 | yuyun  | RQ        |      12 | 104      | 1      | Fa
ll      | 2009 | G         | 198          | B
+-------+----------+--------+----------+------+------+-------+--------+-----------+---------+----------+--------+---
------+------+----------+------------+-------------+
393 rows in set (0.00 sec)
```

5. Dengan menggunakan 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).

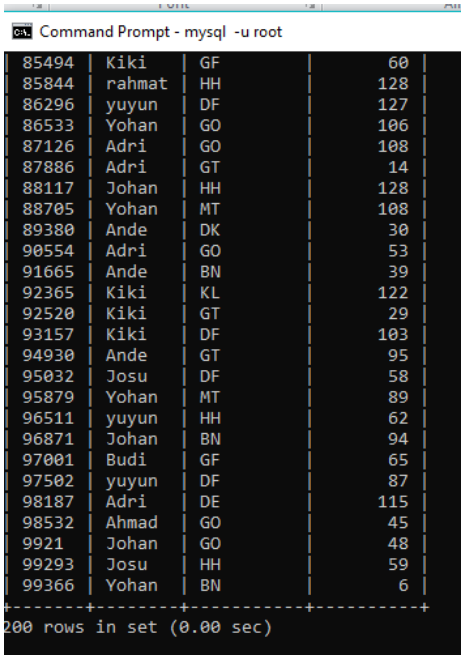
   Sesudah menggunakan tuning :



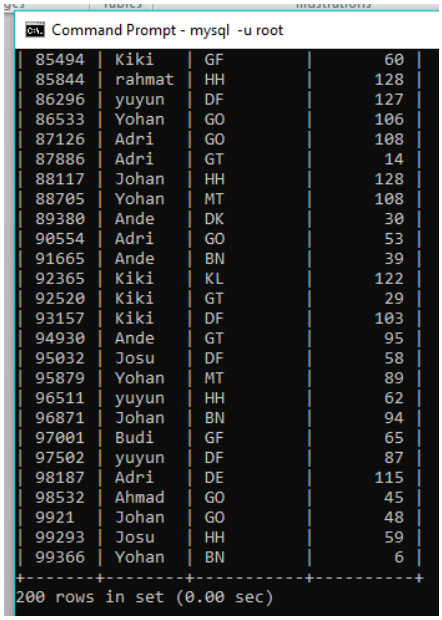
2.1.3 Data 2 (advisor = 200, student = 200, section = 400,takes = 400)

1. Dengan menggunakan query 1 (SELECT * FROM student)

   Waktu sebelum dituning :

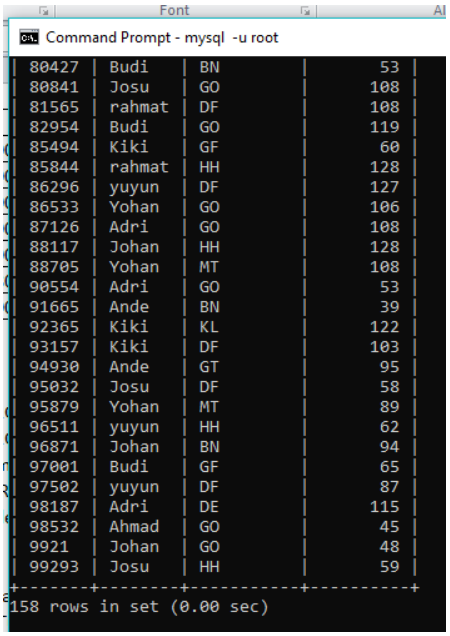


Waktu sesudah dituning :

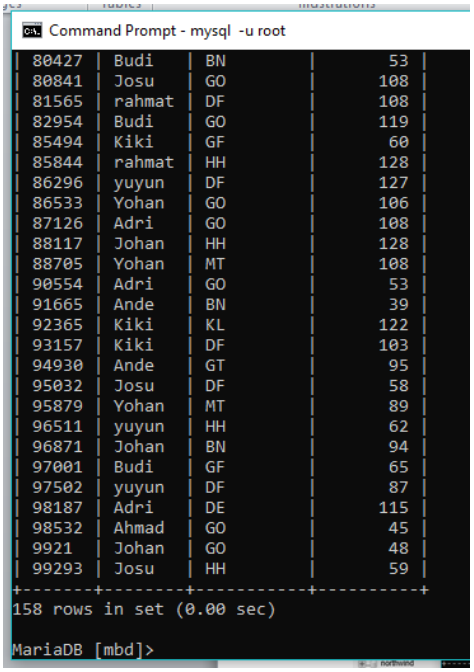


2. Dengan menggunakan query 2 (SELECT * FROM student WHERE tot_cred > 30)

   Waktu sebelum dituning :

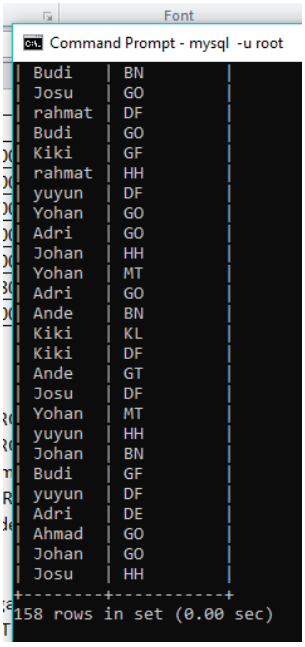


Waktu sesudah dituning :

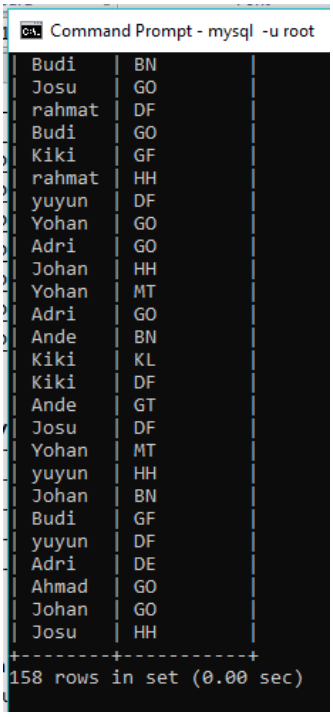


3. Dengan menggunakan query 3 (SELECT `name`, department FROM student WHERE tot_cred > 30)

   Sebelum menggunakan tuning :

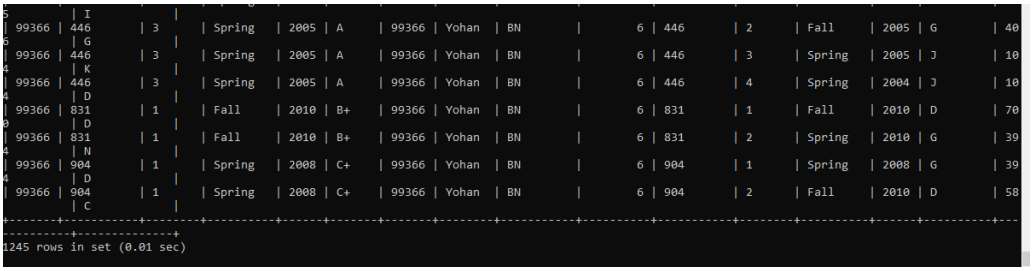


Sesudah menggunakan tuning :

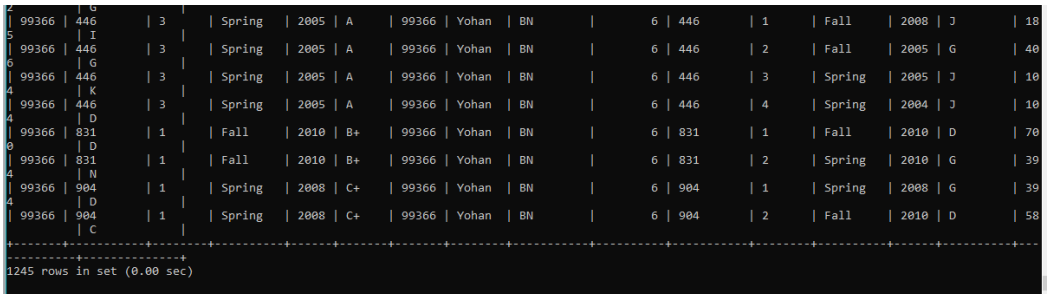


4. Dengan menggunakan query 4 (SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id)

Sebelum menggunakan tuning :

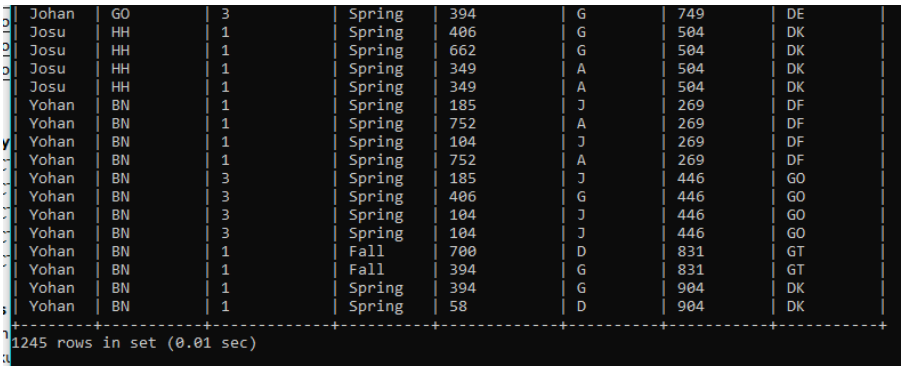


Sesudah menggunakan tuning :



5. Dengan menggunakan 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)

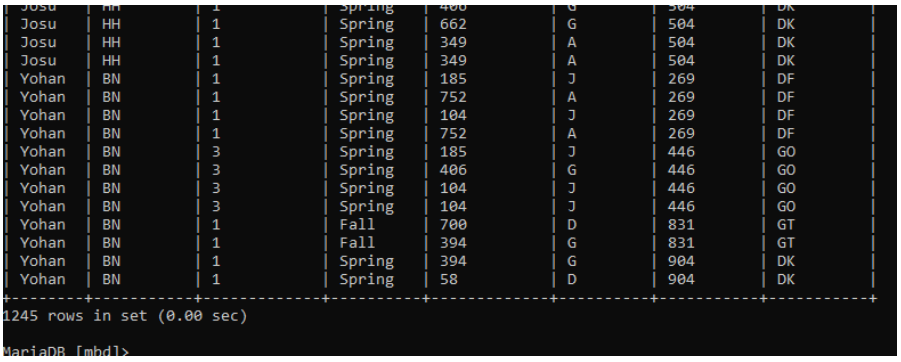
   Sebelum menggunakan tuning :



Johan	GO	3	Spring	394	G	749	DE
Josu	HH	1	Spring	406	G	504	DK
Josu	HH	1	Spring	662	G	504	DK
Josu	HH	1	Spring	349	A	504	DK
Josu	HH	1	Spring	349	A	504	DK
Yohan	BN	1	Spring	185	J	269	DF
Yohan	BN	1	Spring	752	A	269	DF
Yohan	BN	1	Spring	104	J	269	DF
Yohan	BN	1	Spring	752	A	269	DF
Yohan	BN	3	Spring	185	J	446	GO
Yohan	BN	3	Spring	406	G	446	GO
Yohan	BN	3	Spring	104	J	446	GO
Yohan	BN	3	Spring	104	J	446	GO
Yohan	BN	1	Fall	700	D	831	GT
Yohan	BN	1	Fall	394	G	831	GT
Yohan	BN	1	Spring	394	G	904	DK
Yohan	BN	1	Spring	58	D	904	DK

1245 rows in set (0.01 sec)

   Sesudah menggunakan tuning :



Josu	HH	1	Spring	662	G	504	DK
Josu	HH	1	Spring	349	A	504	DK
Josu	HH	1	Spring	349	A	504	DK
Yohan	BN	1	Spring	185	J	269	DF
Yohan	BN	1	Spring	752	A	269	DF
Yohan	BN	1	Spring	104	J	269	DF
Yohan	BN	1	Spring	752	A	269	DF
Yohan	BN	3	Spring	185	J	446	GO
Yohan	BN	3	Spring	406	G	446	GO
Yohan	BN	3	Spring	104	J	446	GO
Yohan	BN	3	Spring	104	J	446	GO
Yohan	BN	1	Fall	700	D	831	GT
Yohan	BN	1	Fall	394	G	831	GT
Yohan	BN	1	Spring	394	G	904	DK
Yohan	BN	1	Spring	58	D	904	DK

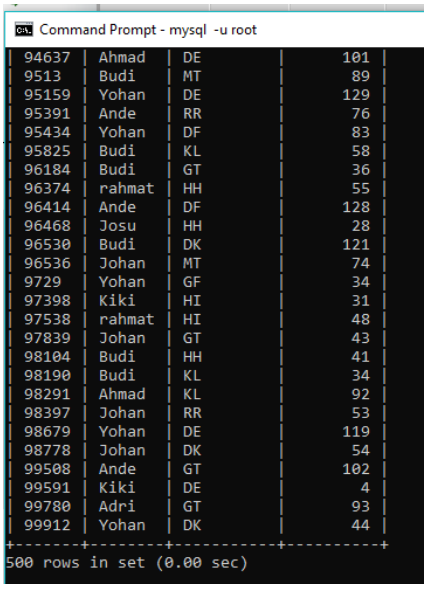
1245 rows in set (0.00 sec)

MariaDB [mbdl]>

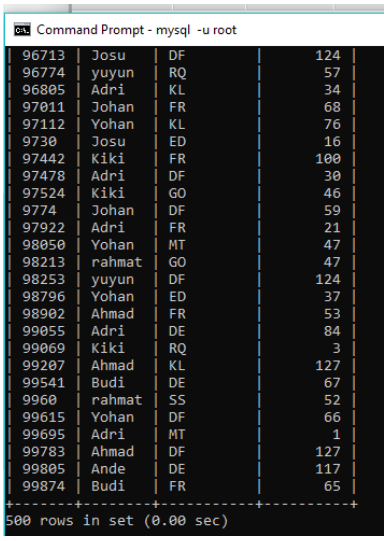
2.1.4 Data 3 (advisor = 500, student = 500, section = 1000,takes = 1000)

1. Dengan menggunakan query 1 (SELECT * FROM student)

   Waktu sebelum dituning :

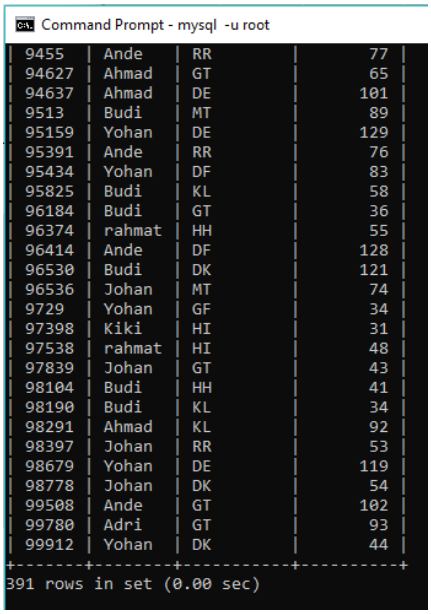


   Waktu sesudah dituning :

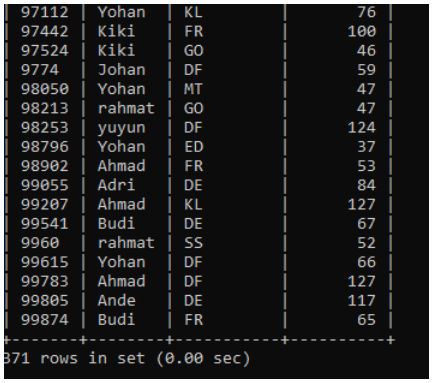


2. Dengan menggunakan query 2 (SELECT * FROM student WHERE tot_cred > 30)

   Waktu sebelum dituning :



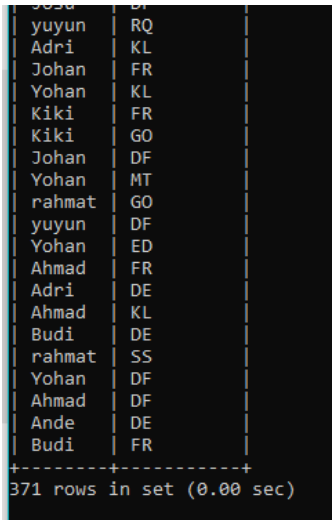
   Waktu sesudah dituning :



3. Dengan menggunakan query 3 (SELECT `name`, department FROM student WHERE tot_cred > 30)

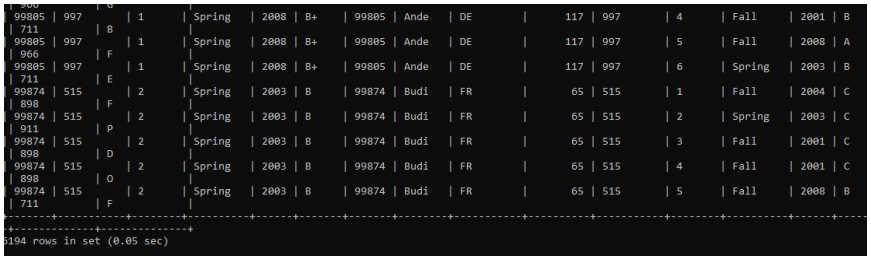
   Sebelum menggunakan tuning :

   Sesudah menggunakan tuning :



4. Dengan menggunakan query 4 (SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id)

   Sebelum menggunakan tuning :

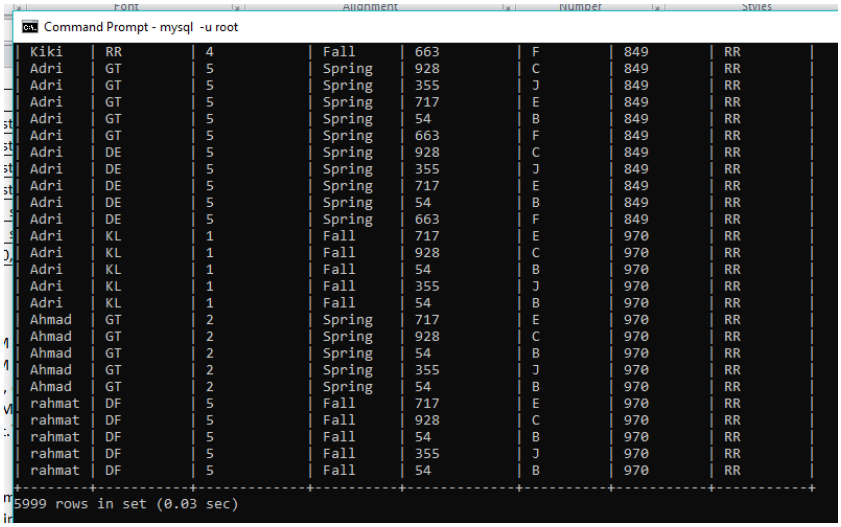


99805	997	1	Spring	2008	B+	99805	Ande	DE	117	997	4	Fall	2001	B	711	B
99805	997	1	Spring	2008	B+	99805	Ande	DE	117	997	5	Fall	2008	A	966	F
99805	997	1	Spring	2008	B+	99805	Ande	DE	117	997	6	Spring	2003	B	711	E
99874	515	2	Spring	2003	B	99874	Budi	FR	65	515	1	Fall	2004	C	898	F
99874	515	2	Spring	2003	B	99874	Budi	FR	65	515	2	Spring	2003	C	911	P
99874	515	2	Spring	2003	B	99874	Budi	FR	65	515	3	Fall	2001	C	898	D
99874	515	2	Spring	2003	B	99874	Budi	FR	65	515	4	Fall	2001	C	898	O
99874	515	2	Spring	2003	B	99874	Budi	FR	65	515	5	Fall	2008	B	711	F

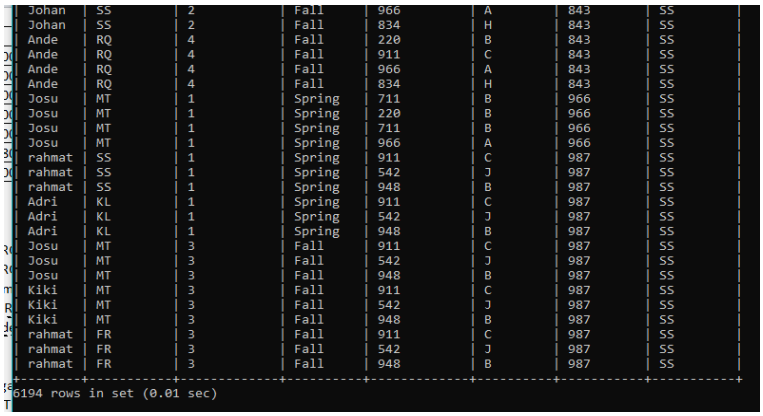
5194 rows in set (0.05 sec)

5. Dengan menggunakan 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)

   Sebelum menggunakan tuning :



   Sesudah menggunakan tuning :



2.2 Tunning : Setting Configuration DBMS

Untuk membatasi penguraian pembahasan pada , kami dari tim pengembang aplikasi membuat beberapa rumusan masalah dalam bentuk pertanyaan yaitu:

BAB III

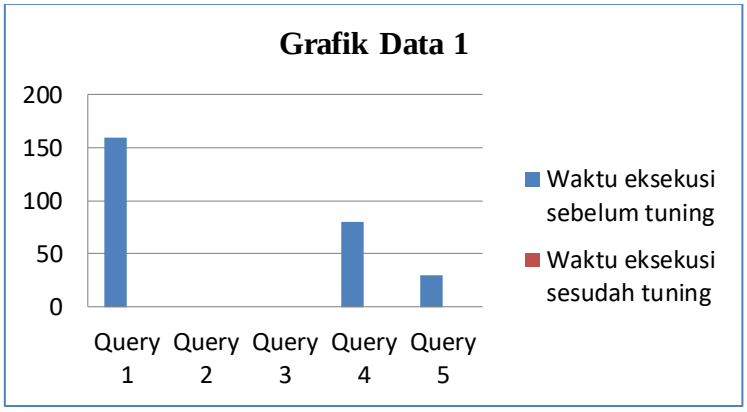
HASIL DAN PEMBAHASAN

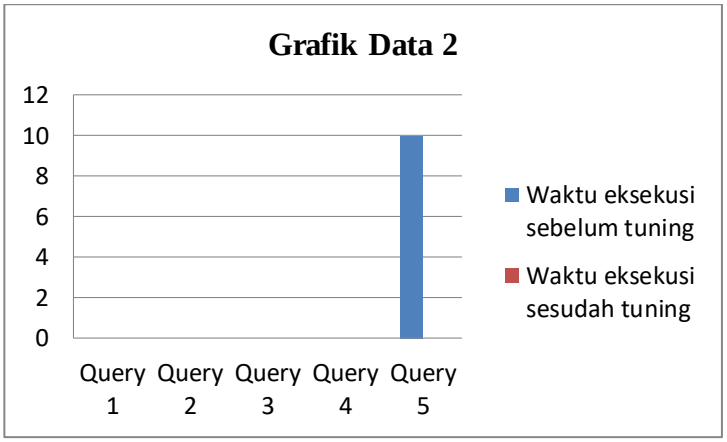
3.1 Hasil

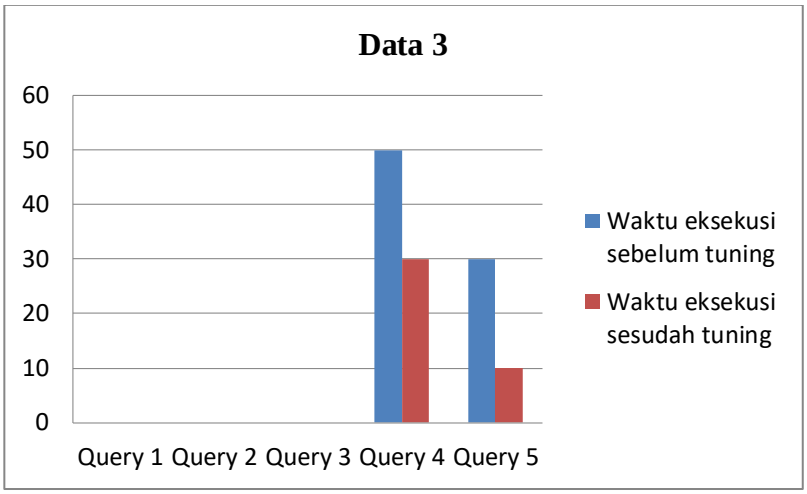
Tabel hasil :

Data ke-	Waktu eksekusi sebelum tuning (ms)					Waktu eksekusi susudah tuning (ms)				
	Q1	Q2	Q3	Q4	Q5	Q1	Q2	Q3	Q4	Q5
1	160	0	0	80	30	0	0	0	0	0
2	0	0	0	0	10	0	0	0	0	0
3	0	0	0	50	30	0	0	0	30	10

Grafik hasil :







Keterangan :

Data 1	advisor = 100, student = 100, section = 200,takes = 200
Data 2	advisor = 200, student = 200, section = 400,takes = 400
Data 3	advisor = 500, student = 500, section = 1000,takes = 1000

3.2 Pembahasan

Berdasarkan data pada tabel hasil dan grafik hasil diatas, dapat dilihat bahwa pada Data 1 dengan advisor = 100, student = 100, section = 200, takes = 200 menghasilkan waktu eksekusi yang ragam berdasarkan kelima query yang telah ditentukan. Data yang dieksekusi menggunakan query 1, 4 dan 5 mengunakan waktu eksekusi program yang lebih lama dibanding dengan menggunakan query 2 dan 3, hal ini dapat terjadi tergantung oleh cara mengecek setiap query yang berbeda-beda. Untuk itu, maka diperlukan tuning agar dapat mempercepat waktu eksekusi data, dalam hal ini saya menggunakan teknik B-Tree untuk tuning index yang sangat berguna pada saat memilih row yang sesuai dengan kriteria tertentu. Index jenis B-Tree dapat dibuat dengan perintah CREATE INDEX. Setelah menggunkan teknik di atas, maka didapatkan waktu eksekusi sesudah dituning menjadi jauh lebih kecil, bahkan mencapai 0 ms.

Dari pembahasan di atas, dapat disimpulkan bahwa tuning dapat digunakan untuk meningkatkan kecepatan dalam eksekusi data sehingga pada akhirnya data dapat dihasilkan secara lebih cepat dan sesuai dengan kebutuhan pengguna.

Daftar Pustaka

Cecilia, C., Mihai, G. (2011). Increasing Database Performance using Indexes, Database Systems Journal.