

LAPORAN TUGAS BESAR
MANAGEMENT DATABASE RA
Tuning Indexing

Untuk Memenuhi tugas besar matakuliah
manajemen basis data IF3144



Disusun Oleh :

Nur Ali Majid / 14115015

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

Daftar Isi

BAB 1 STUDI LITERATUR.....	3
1.1 Tuning : Indexing	3
1.2 Tuning : Setting Configuration DBMS.....	4
BAB 2 DESKRIPSI PERCOBAAN	5
2.1 Tuning : Indexing	5
BAB III HASIL DAN PEMBAHASAN	22
3.1 Hasil	22
3.2 Pembahasan.....	23
Daftar Pustaka.....	25

BAB 1 STUDI LITERATUR

1.1 Tuning : Indexing

Indeks yang efektif adalah salah satu cara terbaik untuk meningkatkan kinerja dalam aplikasi database. Tanpa indeks, mesin SQL Server seperti pembaca yang mencoba menemukan kata dalam buku dengan memeriksa setiap halaman. Dalam istilah basis data, pemindaian tabel terjadi ketika tidak ada indeks yang tersedia untuk membantu kueri. Salah satu pekerjaan paling penting untuk database adalah menemukan indeks terbaik untuk digunakan saat membuat rencana eksekusi.

Indeks Tuning adalah bagian penyetelan basis data untuk memilih dan membuat indeks. Tujuan indeks tuning adalah untuk mengurangi waktu pemrosesan kueri. Potensi penggunaan indeks di lingkungan dinamis dengan beberapa permintaan ad-hoc sebelumnya adalah tugas yang sulit. Indeks tuning melibatkan kueri berdasarkan indeks dan indeks dibuat secara otomatis saat itu juga. Tidak diperlukan tindakan eksplisit oleh pengguna basis data untuk indeks tuning.[1]

Index tuning dapat digunakan untuk meningkatkan kinerja kueri dan basis data. Adapun langkah-langkah yang dapat dilakukan :

- Menggunakan pengoptimal kueri dengan penggunaan indeks terbaik untuk melakukan analisis kueri.
- Perubahan dalam penggunaan indeks, distribusi permintaan dan kinerjanya dianalisis untuk memeriksa efeknya.
- Ini juga merekomendasikan cara menyetel basis data untuk satu set kecil pertanyaan masalah.

Banyak basis data memiliki satu indeks khusus per tabel di mana semua data dari satu baris ada dalam indeks. SQL Server menyebut indeks ini sebagai *clustering index*. *Clustering index* lebih mirip dengan buku telepon karena

setiap entri indeks berisi semua informasi yang dibutuhkan, tidak ada referensi untuk mengikuti untuk mengambil nilai data tambahan.

Sebagai aturan umum, setiap tabel non-sepele harus memiliki *clustering index*. Dalam SQL Server, membuat kunci utama akan secara otomatis membuat *clustering index* (jika tidak ada) menggunakan kolom kunci utama sebagai kunci indeks. Indeks Clustered adalah indeks yang paling efektif (ketika digunakan, mereka selalu mencakup permintaan), dan dalam banyak sistem database akan membantu database mengelola ruang yang dibutuhkan untuk menyimpan tabel secara efisien.

1.2 Tuning : Setting Configuration DBMS

Sistem Manajemen Basis Data (DBMS) adalah komponen paling penting dari aplikasi intensif data apa pun. Mereka dapat menangani sejumlah besar data dan beban kerja yang kompleks. Tetapi mereka sulit untuk dikelola karena mereka memiliki ratusan "tombol" konfigurasi yang mengontrol faktor-faktor seperti jumlah memori yang digunakan untuk cache dan seberapa sering menulis data ke penyimpanan.

Untuk melakukan performance tuning pada MySQL Setelah Anda menginstal MySQL, Anda harus mengkonfigurasinya. Untungnya, Anda tidak perlu mengkonfigurasi MySQL setiap kali Anda me-restart itu. Konfigurasi Anda disimpan dalam file opsi, yang juga disebut sebagai file konfigurasi. Dengan konfigurasi ini, Anda perlu memikirkan kinerja server. Kinerja server sangat penting untuk aplikasi yang stabil. Kinerja mendorong profitabilitas dan produktivitas untuk suatu perusahaan, jadi Anda harus menyesuaikan database MySQL Anda untuk kinerja yang optimal. Artikel ini membahas penyesuaian dan konfigurasi kinerja untuk server MySQL Anda dan databasenya.

BAB 2 DESKRIPSI PERCOBAAN

2.1 Tuning : Indexing

Untuk melakukan sebuah tuning indexing, pada percobaan kali ini dengan menggunakan database dan java yang telah disediakan. Percobaan dilakukan dengan membandingkan waktu sebelum dilakukan tuning dan setelah dilakukan tuning.

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

```
MariaDB [mbd]> create index ind_student ON student(ID,tot_cred) using btree;
Query OK, 0 rows affected (0.777 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [mbd]> create index ind_takes ON takes(ID) using btree;
Query OK, 0 rows affected (0.230 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [mbd]> create index ind_takes2 ON takes(ID,course_id) using btree;
Query OK, 0 rows affected (0.336 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [mbd]> create index ind_course ON section(course_id) using btree;
Query OK, 0 rows affected (0.247 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [mbd]> create index index_source_dept_name on course (dept_name) using btree;
Query OK, 0 rows affected (0.359 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [mbd]> create index index_student_tot_cred on student (tot_cred) using btree;
Query OK, 0 rows affected (0.240 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Gambar 1. Index Tuning Data 1

1. Menggunakan query 1 (SELECT * FROM student) Waktu eksekusi sebelum tuning.

76290	Yohan	GF	28
76512	Ahmad	HH	67
77221	Johan	HH	127
77676	Adri	GF	9
78545	Budi	DK	59
81563	yuyun	WW	13
81884	Ahmad	IF	28
82254	Johan	DF	17
82693	yuyun	DE	24
84130	Ande	HH	14
87493	rahmat	DK	112
87518	Ahmad	HH	75
87902	Kiki	ED	82
91005	Budi	WW	90
91179	Budi	DE	99
91407	yuyun	DF	128
91639	Yohan	WW	79
91653	Josu	MT	106
91679	Adri	IF	114
9241	yuyun	DE	105
94432	Kiki	GO	69
9565	Yohan	IF	39
96546	Kiki	GF	48
98878	Josu	GO	57
99221	Budi	ED	39
99939	Josu	ED	128

100 rows in set (0.001 sec)

Waktu setelah tuning

77676	Adri	GF	9
78545	Budi	DK	59
81563	yuyun	WW	13
81884	Ahmad	IF	28
82254	Johan	DF	17
82693	yuyun	DE	24
84130	Ande	HH	14
87493	rahmat	DK	112
87518	Ahmad	HH	75
87902	Kiki	ED	82
91005	Budi	WW	90
91179	Budi	DE	99
91407	yuyun	DF	128
91639	Yohan	WW	79
91653	Josu	MT	106
91679	Adri	IF	114
9241	yuyun	DE	105
94432	Kiki	GO	69
9565	Yohan	IF	39
96546	Kiki	GF	48
98878	Josu	GO	57
99221	Budi	ED	39
99939	Josu	ED	128

100 rows in set (0.001 sec)

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

Waktu sebelum tuning

70678	Ahmad	WW	40
72305	Kiki	WW	41
72360	yuyun	IF	86
72715	Josu	WW	91
72748	yuyun	GF	62
73116	Budi	ED	91
76512	Ahmad	HH	67
77221	Johan	HH	127
78545	Budi	DK	59
87493	rahmat	DK	112
87518	Ahmad	HH	75
87902	Kiki	ED	82
91005	Budi	WW	90
91179	Budi	DE	99
91407	yuyun	DF	128
91639	Yohan	WW	79
91653	Josu	MT	106
91679	Adri	IF	114
9241	yuyun	DE	105
94432	Kiki	GO	69
9565	Yohan	IF	39
96546	Kiki	GF	48
98878	Josu	GO	57
99221	Budi	ED	39
99939	Josu	ED	128

82 rows in set (0.001 sec)

Waktu setelah tuning

70678	Ahmad	WW	40
72305	Kiki	WW	41
72360	yuyun	IF	86
72715	Josu	WW	91
72748	yuyun	GF	62
73116	Budi	ED	91
76512	Ahmad	HH	67
77221	Johan	HH	127
78545	Budi	DK	59
87493	rahmat	DK	112
87518	Ahmad	HH	75
87902	Kiki	ED	82
91005	Budi	WW	90
91179	Budi	DE	99
91407	yuyun	DF	128
91639	Yohan	WW	79
91653	Josu	MT	106
91679	Adri	IF	114
9241	yuyun	DE	105
94432	Kiki	GO	69
9565	Yohan	IF	39
96546	Kiki	GF	48
98878	Josu	GO	57
99221	Budi	ED	39
99939	Josu	ED	128
+-----+			
82 rows in set (0.001 sec)			

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

Waktu sebelum tuning

rahmat	DK
Ahmad	WW
Kiki	WW
yuyun	IF
Josu	WW
yuyun	GF
Budi	ED
Ahmad	HH
Johan	HH
Budi	DK
rahmat	DK
Ahmad	HH
Kiki	ED
Budi	WW
Budi	DE
yuyun	DF
Yohan	WW
Josu	MT
Adri	IF
yuyun	DE
Kiki	GO
Yohan	IF
Kiki	GF
Josu	GO
Budi	ED
Josu	ED
+-----+	
82 rows in set (0.001 sec)	

Waktu setelah tuning

yuyun	IF
Josu	WW
yuyun	GF
Budi	ED
Ahmad	HH
Johan	HH
Budi	DK
rahmat	DK
Ahmad	HH
Kiki	ED
Budi	WW
Budi	DE
yuyun	DF
Yohan	WW
Josu	MT
Adri	IF
yuyun	DE
Kiki	GO
Yohan	IF
Kiki	GF
Josu	GO
Budi	ED
Josu	ED

82 rows in set (0.001 sec)

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

Waktu sebelum tuning

416	H	4	Spring	2010	C+	96546	Kiki	GF	48	452	4	Spring	2010	I
96546	D	1	Fall	2005	B+	96546	Kiki	GF	48	582	1	Fall	2005	I
20	G	2	Fall	2007	B	96546	Kiki	GF	48	697	1	Spring	2008	G
96546	O	2	Fall	2007	B	96546	Kiki	GF	48	697	2	Fall	2007	I
416	G	1	Fall	2004	B-	96546	Kiki	GF	48	947	1	Fall	2004	G
631	H	1	Fall	2004	B-	96546	Kiki	GF	48	947	2	Fall	2007	G
96546	I	3	Spring	2007	A-	99221	Budi	ED	39	124	1	Spring	2004	I
733	N	3	Spring	2007	A-	99221	Budi	ED	39	124	2	Spring	2009	G
99221	B	3	Spring	2007	A-	99221	Budi	ED	39	124	3	Spring	2007	I
20	I	1	Spring	2003	B-	99221	Budi	ED	39	572	1	Spring	2003	I
99221	O	1	Fall	2007	B	99221	Budi	ED	39	950	1	Fall	2007	I
631	F	1	Fall	2007	B	99221	Budi	ED	39	950	2	Fall	2005	A
99221	J	1	Fall	2007	B	99221	Budi	ED	39	950	3	Fall	2004	G
78	O	1	Spring	2009	A	99221	Budi	ED	39	984	1	Spring	2009	G
99221	P	1	Spring	2009	A	99221	Budi	ED	39	984	2	Fall	2006	G
416	F	1	Spring	2009	A	99221	Budi	ED	39	984	3	Fall	2002	G
631	L	2	Fall	2006	A-	99221	Budi	ED	39	984	1	Spring	2009	G
99221	P	2	Fall	2006	A-	99221	Budi	ED	39	984	2	Fall	2006	G
416	F	2	Fall	2006	A-	99221	Budi	ED	39	984	3	Fall	2002	G
631	L													

110 rows in set (0.041 sec)

Waktu setelah tuning

```

| 99221 | Budi | ED | 39 | 984 | 1
| | Spring | 2009 | G | | 416 | | P |
| 99221 | 984 | 1 | Spring | 2009 | A
| 99221 | Budi | ED | 39 | 984 | 2
| | Fall | 2006 | G | | 631 | | F |
| 99221 | 984 | 1 | Spring | 2009 | A
| 99221 | Budi | ED | 39 | 984 | 3
| | Fall | 2002 | G | | 631 | | L |
| 99221 | 984 | 2 | Fall | 2006 | A-
| 99221 | Budi | ED | 39 | 984 | 1
| | Spring | 2009 | G | | 416 | | P |
| 99221 | 984 | 2 | Fall | 2006 | A-
| 99221 | Budi | ED | 39 | 984 | 2
| | Fall | 2006 | G | | 631 | | F |
| 99221 | 984 | 2 | Fall | 2006 | A-
| 99221 | Budi | ED | 39 | 984 | 3
| | Fall | 2002 | G | | 631 | | L |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+
410 rows in set (0.004 sec)

```

5. Menggunakan query 5 (SELECT
student.`name`,student.dept_name,takes.sec_id AS
pengambilan,takes.semester,section.room_number,section.building,cours
e.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).

Waktu sebelum tuning.

```

| Budi | I | ED | 124 | 3 | DK | Spring | 631
| Budi | G | ED | 124 | 3 | DK | Spring | 20
| Budi | I | ED | 124 | 1 | DK | Spring | 20
| Budi | I | ED | 572 | 1 | IF | Fall | 631
| Budi | I | ED | 950 | 1 | IF | Fall | 78
| Budi | A | ED | 950 | 1 | IF | Fall | 631
| Budi | G | ED | 950 | 1 | IF | Fall | 631
| Budi | G | ED | 984 | 1 | MT | Spring | 416
| Budi | G | ED | 984 | 1 | MT | Spring | 631
| Budi | G | ED | 984 | 1 | MT | Spring | 631
| Budi | G | ED | 984 | 2 | MT | Fall | 416
| Budi | G | ED | 984 | 2 | MT | Fall | 631
| Budi | G | ED | 984 | 2 | MT | Fall | 631
+-----+-----+-----+-----+-----+-----+-----+
410 rows in set (0.004 sec)

```

Waktu setelah tuning

Budget	Category	Subcategory	Amount	Unit	Season	Total
Budget	G	ED	124	3	DK	Spring 631
Budget	I	ED	124	3	DK	Spring 20
Budget	I	ED	124	1	DK	Spring 20
Budget	I	ED	572	1	IF	Fall 631
Budget	I	ED	950	1	IF	Fall 78
Budget	A	ED	950	1	IF	Fall 631
Budget	G	ED	950	1	IF	Spring 416
Budget	G	ED	984	1	MT	Spring 631
Budget	G	ED	984	1	MT	Spring 631
Budget	G	ED	984	2	MT	Fall 416
Budget	G	ED	984	2	MT	Fall 631
Budget	G	ED	984	2	MT	Fall 631
Budget	G	ED	984	2	MT	Fall 631

+-----+-----+-----+-----+-----+-----+

410 rows in set (0.004 sec)

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

```

MariaDB [mbd]> create index index_student_id on student(id) using btree;
Query OK, 0 rows affected (0.373 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [mbd]> create index index_course_dept_name on course(dept_name)
using btree;
Query OK, 0 rows affected, 1 warning (0.318 sec)
Records: 0 Duplicates: 0 Warnings: 1

MariaDB [mbd]> create index index_takes_course_id on takes(course_id)us
ing btree;
Query OK, 0 rows affected (0.276 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

Gambar 2. Index Tuning Data 2

1. Menggunakan query 1 (SELECT * FROM student)

Waktu eksekusi sebelum tuning.

87209	Kiki	DF	11
87387	Kiki	HH	30
87445	Ahmad	MT	73
88596	Johan	HH	63
88798	Johan	HH	88
89091	Ande	DF	87
89828	Kiki	DE	64
89930	Budi	GF	50
90193	Budi	DF	110
90469	Yohan	FR	8
90654	Budi	KL	79
9083	Ahmad	KL	57
91099	Kiki	FR	80
92010	Adri	HH	41
92150	yuyun	KL	24
92559	Yohan	MT	46
92614	rahmat	DE	20
93418	yuyun	FR	32
94238	Ahmad	GT	107
95660	Adri	EL	98
96810	Kiki	EL	47
97649	Adri	GF	93
9810	Josu	GT	80
99	yuyun	HH	9
99394	Ahmad	HH	2
99501	Ahmad	MT	54

200 rows in set (0.036 sec)

Waktu setelah tuning

90469	Yohan	FR	8
90654	Budi	KL	79
9083	Ahmad	KL	57
91099	Kiki	FR	80
92010	Adri	HH	41
92150	yuyun	KL	24
92559	Yohan	MT	46
92614	rahmat	DE	20
93418	yuyun	FR	32
94238	Ahmad	GT	107
95660	Adri	EL	98
96810	Kiki	EL	47
97649	Adri	GF	93
9810	Josu	GT	80
99	yuyun	HH	9
99394	Ahmad	HH	2
99501	Ahmad	MT	54

200 rows in set (0.001 sec)

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

Waktu sebelum tuning

85410	Budi	KL	96
85713	Adri	DF	81
85950	Budi	DE	79
8605	Kiki	FR	120
87036	rahmat	MT	102
87445	Ahmad	MT	73
88596	Johan	HH	63
88798	Johan	HH	88
89091	Ande	DF	87
89828	Kiki	DE	64
89930	Budi	GF	50
90193	Budi	DF	110
90654	Budi	KL	79
9083	Ahmad	KL	57
91099	Kiki	FR	80
92010	Adri	HH	41
92559	Yohan	MT	46
93418	yuyun	FR	32
94238	Ahmad	GT	107
95660	Adri	EL	98
96810	Kiki	EL	47
97649	Adri	GF	93
9810	Josu	GT	80
99501	Ahmad	MT	54

146 rows in set (0.001 sec)

Waktu setelah tuning

88596	Johan	HH	63
88798	Johan	HH	88
89091	Ande	DF	87
89828	Kiki	DE	64
89930	Budi	GF	50
90193	Budi	DF	110
90654	Budi	KL	79
9083	Ahmad	KL	57
91099	Kiki	FR	80
92010	Adri	HH	41
92559	Yohan	MT	46
93418	yuyun	FR	32
94238	Ahmad	GT	107
95660	Adri	EL	98
96810	Kiki	EL	47
97649	Adri	GF	93
9810	Josu	GT	80
99501	Ahmad	MT	54

+-----+
146 rows in set (0.001 sec)

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

Waktu sebelum tuning

Ahmad	MT
Johan	HH
Johan	HH
Ande	DF
Kiki	DE
Budi	GF
Budi	DF
Budi	KL
Ahmad	KL
Kiki	FR
Adri	HH
Yohan	MT
yuyun	FR
Ahmad	GT
Adri	EL
Kiki	EL
Adri	GF
Josu	GT
Ahmad	MT

+-----+
146 rows in set (0.001 sec)

Waktu setelah tuning

Ahmad	MT
Ahmad	MT
Johan	HH
Johan	HH
Ande	DF
Kiki	DE
Budi	GF
Budi	DF
Budi	KL
Ahmad	KL
Kiki	FR
Adri	HH
Yohan	MT
yuyun	FR
Ahmad	GT
Adri	EL
Kiki	EL
Adri	GF
Josu	GT
Ahmad	MT

146 rows in set (0.001 sec)

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

Waktu sebelum tuning

GT	550	B	80	583	3	Fall	2006	E
99	823	5	9	823	1	Spring	2004	A+
HH	743	H	9	823	2	Fall	2002	yuyun
99	823	5	9	823	2	Spring	2004	A+
HH	104	K	9	823	3	Spring	2002	yuyun
99	823	5	9	823	3	Spring	2002	I
HH	490	G	5	823	4	Fall	2010	yuyun
99	823	P	5	823	5	Spring	2004	yuyun
HH	550	9	5	823	5	Spring	2004	D
99	823	5	9	823	5	Spring	2004	D
HH	104	K	2	453	1	Spring	2010	A
99394	453	2	2	453	1	Spring	2005	Ahmad
HH	550	P	2	453	2	Spring	2010	A
99394	453	2	2	453	2	Spring	2010	Ahmad
HH	543	H	2	453	2	Spring	2010	C

1206 rows in set (0.037 sec)

Waktu setelah tuning

543	I	3	Fall	2006	B+	9810	Josu
9810	583	80	3	Fall	2006	E	
GT	550	B	5	Spring	2004	A+	yuyun
99	823	9	823	1	Fall	2002	B
HH	743	H	5	Spring	2004	A+	yuyun
99	823	9	823	2	Fall	2002	D
HH	104	K	5	Spring	2004	A+	yuyun
99	823	9	823	3	Spring	2002	I
HH	490	G	5	Spring	2004	A+	yuyun
99	823	9	823	4	Fall	2010	E
HH	550	P	5	Spring	2004	A+	yuyun
99	823	9	823	5	Spring	2004	D
HH	104	K	2	Spring	2010	A	Ahmad
99394	453	2	453	1	Spring	2005	E
HH	550	P	2	Spring	2010	A	Ahmad
99394	453	2	453	2	Spring	2010	C
HH	543	H					

1206 rows in set (0.009 sec)

5. Menggunakan query 5 (SELECT
 student.`name`,student.dept_name,takes.sec_id AS
 pengambilan,takes.semester,section.room_number,section.building,cours
 e.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).

Waktu sebelum tuning.

Josu	GT	DF	1	Fall	997	J
152	GT	DF	1	Fall	490	I
Josu	GT	DF	3	Fall	104	D
583	GT	DF	3	Fall	543	C
Josu	GT	DF	3	Fall	550	E
583	GT	DF	3	Fall	550	E
yuyun	HH	KL	5	Spring	743	B
823	HH	KL	5	Spring	104	D
yuyun	HH	KL	5	Spring	490	I
823	HH	KL	5	Spring	550	E
yuyun	HH	KL	5	Spring	104	D
823	HH	KL	5	Spring	104	D
Ahmad	HH	KL	2	Spring	550	E
453	HH	KL	2	Spring	543	C
Ahmad	HH	KL	2	Spring	543	C
453	HH	KL	2	Spring	543	C

1206 rows in set (0.007 sec)

Waktu setelah tuning

Josu	GT	1	Fall	997	J
152	DF				
Josu	GT	1	Fall	490	I
152	DF				
Josu	GT	3	Fall	104	D
583	DF				
Josu	GT	3	Fall	543	C
583	DF				
Josu	GT	3	Fall	550	E
583	DF				
yuyun	HH	5	Spring	743	B
823	KL				
yuyun	HH	5	Spring	104	D
823	KL				
yuyun	HH	5	Spring	490	I
823	KL				
yuyun	HH	5	Spring	550	E
823	KL				
yuyun	HH	5	Spring	104	D
823	KL				
Ahmad	HH	2	Spring	550	E
453	KL				
Ahmad	HH	2	Spring	543	C
453	KL				

-----+-----+-----+-----+-----+-----+
1206 rows in set (0.007 sec)

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

```
MariaDB [mbd]> create index ind_takes3 ON takes(ID,course_id) using btree;
Query OK, 0 rows affected, 1 warning (0.532 sec)
Records: 0 Duplicates: 0 Warnings: 1

MariaDB [mbd]> create index ind_student ON student(tot_cred) using btree;
ERROR 1061 (42000): Duplicate key name 'ind_student'
MariaDB [mbd]> create index ind_student2 ON student(tot_cred) using btree;
Query OK, 0 rows affected, 1 warning (0.634 sec)
Records: 0 Duplicates: 0 Warnings: 1

MariaDB [mbd]> create index ind_student3 ON student(ID,dept_name) using btree;
Query OK, 0 rows affected (0.343 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [mbd]> create index ind_section2 ON section(course_id,sec_id) using btree;
Query OK, 0 rows affected (0.434 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Gambar 3. Index Tuning Data 3

1. Menggunakan query 1 (SELECT * FROM student).

Waktu eksekusi sebelum tuning.

96266	Ande	KL	20
96401	Adri	HI	71
96585	Kiki	KL	82
96825	Johan	KL	13
96998	yuyun	HI	114
97084	Adri	FR	107
97435	Kiki	FR	70
98082	Josu	DF	112
99005	Adri	DK	99
99048	Yohan	EL	45
99068	Ahmad	HI	30
99128	rahmat	DK	38
9922	Kiki	FR	10
99259	Ahmad	DF	27
99280	Johan	WW	9
99354	Kiki	ED	38
9947	Josu	DF	19
99489	Josu	IF	116
99731	Kiki	KL	24

+-----+-----+-----+-----+

500 rows in set (0.001 sec)

Waktu setelah tuning

96153	Ahmad	FR	123
96266	Ande	KL	20
96401	Adri	HI	71
96585	Kiki	KL	82
96825	Johan	KL	13
96998	yuyun	HI	114
97084	Adri	FR	107
97435	Kiki	FR	70
98082	Josu	DF	112
99005	Adri	DK	99
99048	Yohan	EL	45
99068	Ahmad	HI	30
99128	rahmat	DK	38
9922	Kiki	FR	10
99259	Ahmad	DF	27
99280	Johan	WW	9
99354	Kiki	ED	38
9947	Josu	DF	19
99489	Josu	IF	116
99731	Kiki	KL	24

+-----+-----+-----+-----+

500 rows in set (0.002 sec)

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

Waktu sebelum tuning

94755	Yohan	KL	69
94928	Josu	EL	54
94961	Yohan	ED	96
95134	Josu	HI	125
95242	Kiki	WW	53
95460	Budi	HI	101
96125	Budi	DK	38
96153	Ahmad	FR	123
96401	Adri	HI	71
96585	Kiki	KL	82
96998	yuyun	HI	114
97084	Adri	FR	107
97435	Kiki	FR	70
98082	Josu	DF	112
99005	Adri	DK	99
99048	Yohan	EL	45
99128	rahmat	DK	38
99354	Kiki	ED	38
99489	Josu	IF	116

381 rows in set (0.002 sec)

Waktu setelah tuning

2	94698	Adri	KL	107
2	94755	Yohan	KL	69
2	94928	Josu	EL	54
2	94961	Yohan	ED	96
2	95134	Josu	HI	125
2	95242	Kiki	WW	53
2	95460	Budi	HI	101
2	96125	Budi	DK	38
2	96153	Ahmad	FR	123
2	96401	Adri	HI	71
2	96585	Kiki	KL	82
2	96998	yuyun	HI	114
2	97084	Adri	FR	107
2	97435	Kiki	FR	70
2	98082	Josu	DF	112
2	99005	Adri	DK	99
2	99048	Yohan	EL	45
2	99128	rahmat	DK	38
2	99354	Kiki	ED	38
2	99489	Josu	IF	116

381 rows in set (0.002 sec)

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

Waktu sebelum tuning

Yohan	KL
Josu	EL
Yohan	ED
Josu	HI
Kiki	WW
Budi	HI
Budi	DK
Ahmad	FR
Adri	HI
Kiki	KL
yuyun	HI
Adri	FR
Kiki	FR
Josu	DF
Adri	DK
Yohan	EL
rahmat	DK
Kiki	ED
Josu	IF
+-----+-----+	
381 rows in set (0.002 sec)	

Waktu setelah tuning

Josu	EL
Yohan	ED
Josu	HI
Kiki	WW
Budi	HI
Budi	DK
Ahmad	FR
Adri	HI
Kiki	KL
yuyun	HI
Adri	FR
Kiki	FR
Josu	DF
Adri	DK
Yohan	EL
rahmat	DK
Kiki	ED
Josu	IF
+-----+-----+	
381 rows in set (0.002 sec)	

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

Waktu sebelum tuning

DF	19	819	3	Fall	2009	F	
9	J	1	4	2004	C+	9947	Josu
9947	819	19	819	Fall	2004	C	C
DF	A	1	5	2004	C+	9947	Josu
653	819	19	819	Fall	2004	A	A
9947	F	1	6	2004	C+	9947	Josu
DF	819	19	819	Fall	Spring	2002	F
611	E	1	1	2006	A+	99489	Josu
9947	819	19	819	Fall	2006	G	G
695	M	1	2	2006	A+	99489	Josu
99489	862	116	862	Fall	2006	C	C
IF	90	1	3	2006	A+	99489	Josu
99489	862	116	862	Fall	2003	C	C
IF	653	1	4	2006	A+	99489	Josu
99489	862	116	862	Fall	Spring	2006	G
653	L	1	5	2006	A+	99489	Josu
IF	862	116	862	Fall	Spring	2009	F
99489	862	116	862	Fall	Spring	2009	F
IF	698	1	5	2006	A+	99489	Josu
99489	862	116	862	Fall	Spring	2009	F
IF	9	B					

+-----+-----+-----+-----+-----+-----+-----+-----+

5814 rows in set (0.149 sec)

+-----+-----+-----+-----+-----+-----+-----+-----+

5814 rows in set (0.032 sec)

Waktu setelah tuning

5. 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).

Waktu sebelum tuning.

```

Johan | ww | 1 | Fall | 306 | E
933 | ww | 1 | Fall | 306 | E
Johan | ww | 1 | Fall | 611 | A
933 | ww | 1 | Fall | 611 | A
Josu | HI | 4 | Spring | 90 | G
933 | HI | 4 | Spring | 90 | G
Josu | HI | 4 | Spring | 695 | F
933 | HI | 4 | Spring | 695 | F
Josu | HI | 4 | Spring | 672 | A
933 | HI | 4 | Spring | 672 | A
Josu | HI | 4 | Spring | 306 | E
933 | HI | 4 | Spring | 306 | E
Josu | HI | 4 | Spring | 306 | E
933 | HI | 4 | Spring | 306 | E
Josu | HI | 4 | Spring | 611 | A
933 | HI | 4 | Spring | 611 | A
rahmat | EL | 1 | Spring | 790 | J
946 | EL | 1 | Spring | 790 | J
rahmat | EL | 1 | Spring | 90 | G
946 | EL | 1 | Spring | 90 | G
-----+-----+-----+-----+-----+-----+
5814 rows in set (0.069 sec)

```

Waktu setelah tuning

```

Johan | ww | 1 | Fall | 611 | A
933 | ww | 1 | Fall | 611 | A
Josu | HI | 4 | Spring | 90 | G
933 | HI | 4 | Spring | 90 | G
Josu | HI | 4 | Spring | 90 | G
933 | HI | 4 | Spring | 90 | G
Josu | HI | 4 | Spring | 695 | F
933 | HI | 4 | Spring | 695 | F
Josu | HI | 4 | Spring | 672 | A
933 | HI | 4 | Spring | 672 | A
Josu | HI | 4 | Spring | 306 | E
933 | HI | 4 | Spring | 306 | E
Josu | HI | 4 | Spring | 306 | E
933 | HI | 4 | Spring | 306 | E
Josu | HI | 4 | Spring | 611 | A
933 | HI | 4 | Spring | 611 | A
rahmat | EL | 1 | Spring | 790 | J
946 | EL | 1 | Spring | 790 | J
rahmat | EL | 1 | Spring | 90 | G
946 | EL | 1 | Spring | 90 | G
-----+-----+-----+-----+-----+-----+
5814 rows in set (0.020 sec)

```

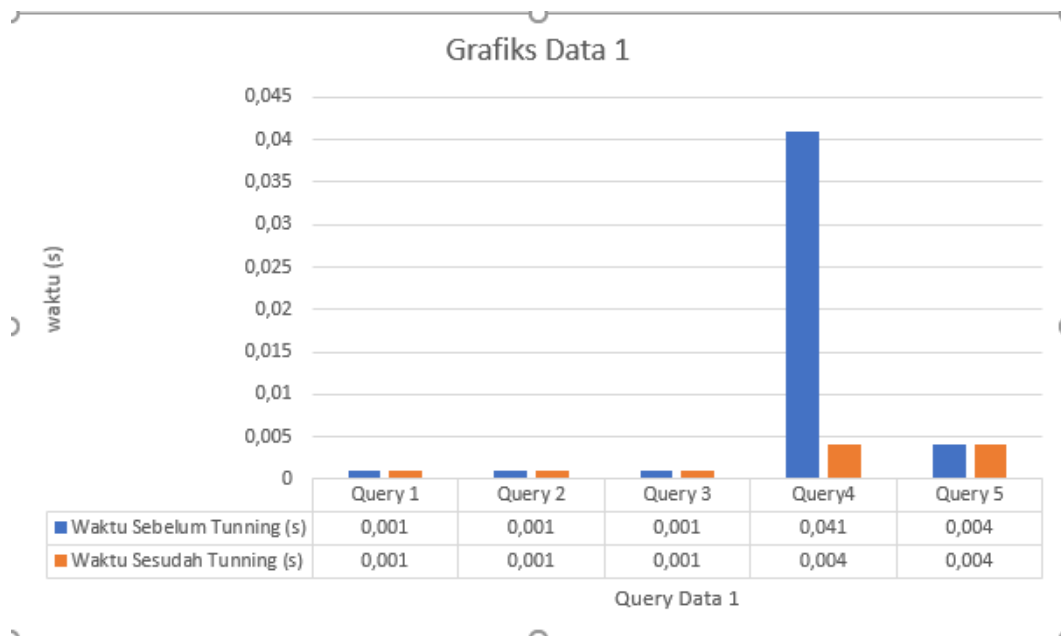
BAB III HASIL DAN PEMBAHASAN

3.1 Hasil

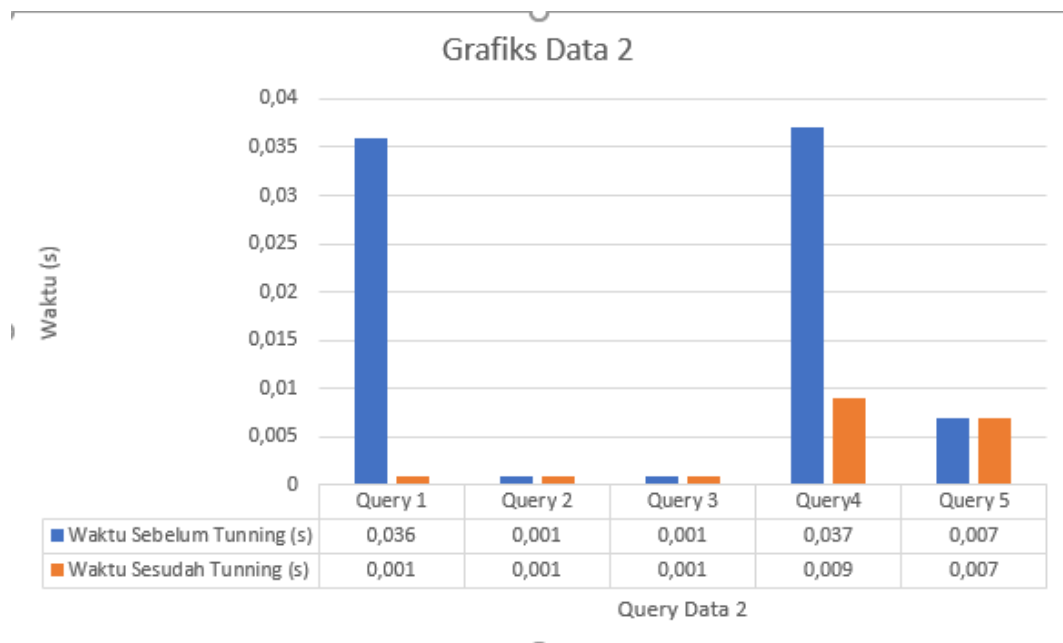
Tabel Hasil :

Data	Waktu Sebelum Tunning (s)					Waktu Sesudah Tunning (s)				
	Query 1	Query 2	Query 3	Query4	Query 5	Query 1	Query 2	Query 3	Query4	Query 5
advisor = 100, student = 100, section = 200,takes = 200	0,001	0,001	0,001	0,041	0,004	0,001	0,001	0,001	0,004	0,004
advisor = 200, student = 200, section = 400,takes = 400	0,036	0,001	0,001	0,037	0,007	0,001	0,001	0,001	0,009	0,007
advisor = 500, student = 500, section = 1000,takes = 1000	0,001	0,002	0,002	0,149	0,069	0,002	0,002	0,002	0,032	0,02

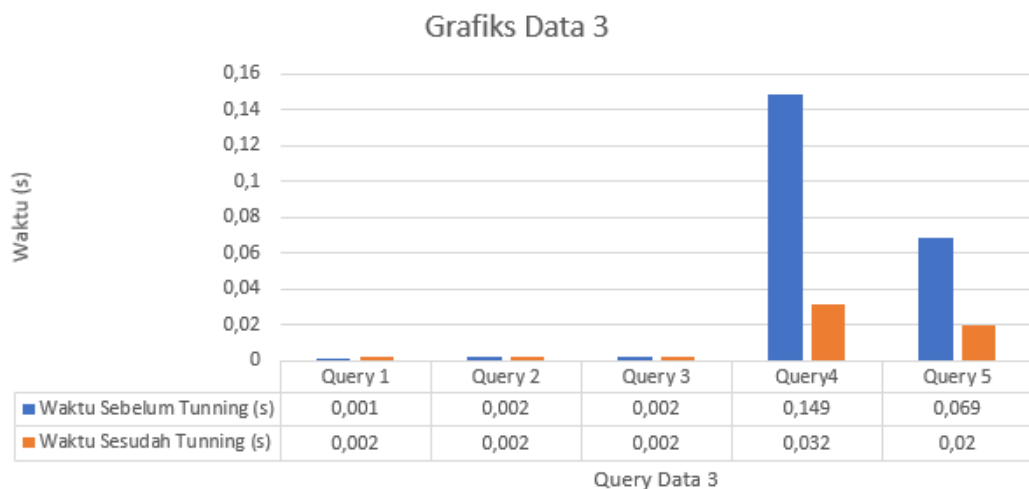
Grafik Hasil : Grafik Data 1.



Grafik Data 2.



Grafik Data 3.



3.2 Pembahasan Hasil.

Berdasarkan data pada tabel dan grafik hasil diatas, dapat dilihat bahwa pada Data 1 (advisor = 100, student = 100, section = 200, takes = 200) menghasilkan waktu eksekusi yang beragam. Data yang dieksekusi menggunakan query 4 dan 5 membutuhkan waktu eksekusi yang lebih lama disbanding query 1,2 dan 3, hal ini terjadi bergantung pada cara pengecekan setiap query yang berbeda-beda. Untuk itu diperlukan tuning agar dapat mempercepat waktu pengeksekusian data. Tuning yang saya lakukan yaitu menggunakan teknik B-Tree. Setelah menggunakan teknik

tersebut waktu yang didapatkan untuk melakukan eksekusi data dapat menjadi lebih kecil dari sebelum dilakukannya tuning.

Dari penjelasan tersebut dapat disimpulkan bahwa melakukan eksekusi data dengan menggunakan teknik tuning dapat memperkecil waktu eksekusi data dan sesuai dengan kebutuhan pengguna.

Daftar Pustaka

- [1] M. Basisdata, “Manajemen Basis Data Semester I – 2019/20,” 2019.

- [2] UniversalClass,
“<https://www.universalclass.com/articles/computers/performance-tuning-and-configurations-for-your-mysql-server.htm>” Diakses pada tanggal 22 Desember 2019