LAPORAN TUGAS MANAJEMEN BASIS DATA

TUNING DATABASE

DISUSUN OLEH:

BAGAS PANGESTU (14117026)

PROGRAM STUDI : TEKNIK INFORMATIKA

KELAS : MBD - RB

LAMPUNG SELATAN, 22 DESEMBER 2019

INSTITUT TEKNOLOGI SUMATERA

2019

DAFTAR ISI

STUDI LITERATUR 2

1.1. Tuning Indexing 2

1.2. Tuning Setting Configuration DBMS 2

DESKRIPSI PERCOBAAN 3

2.1. Tuning Indexing 3

2.2. Tuning Setting Configuration DBMS 4

HASIL DAN PEMBAHASAN 5

3.1 Percobaan sebelum tuning Indexing 5

3.1.1. percobaan pada data nomor 1 5

3.1.2. percobaan pada data nomor 2 6

3.1.3. percobaan pada data nomor 3 7

3.2 Percobaan setelah tuning Indexing 7

3.2.1. percobaan pada data nomor 1 8

3.2.2. percobaan pada data nomor 2 9

3.2.3. percobaan pada data nomor 3 9

3.2 Data Hasil 10

3.4 Pembahasan 10

DAFTAR PUSTAKA 12

STUDI LITERATUR

1.1. Tuning Indexing

Indeks mengaitkan informasi pencarian dengan entri Server Direktori. Indeks mengambil bentuk file yang disimpan dengan database Server Direktori. Database dalam konteks ini adalah representasi fisik akhiran. Untuk sebagian besar penggunaan, satu sufiks sesuai dengan satu basis data.

Indeks mempercepat pencarian. Indeks berisi daftar nilai, masing-masing terkait dengan daftar pengidentifikasi entri yang sesuai dengan nilai. Server Direktori dapat mencari entri dengan cepat menggunakan daftar pengidentifikasi entri dalam indeks. Tanpa indeks untuk mengelola daftar entri, Direktori Server mungkin harus memeriksa setiap entri dalam akhiran untuk menemukan kecocokan untuk pencarian.

Alasan pencarian yang diindeks mungkin memerlukan pemrosesan yang jauh lebih sedikit daripada pencarian yang tidak terindeksasi menjadi jelas ketika pemrosesan permintaan pencarian dijelaskan.

1.2. Tuning Setting Configuration DBMS

Tuning DBMS terdiri dari sekelompok kegiatan yang digunakan untuk mengoptimalkan dan mengatur kinerja suatu basis data. Ini merujuk pada konfigurasi file database, sistem manajemen basis data (DBMS), serta perangkat keras dan sistem operasi tempat database di-host. Tujuan dari penyetelan basis data adalah untuk memaksimalkan penerapan sumber daya sistem dalam upaya untuk melakukan transaksi seefisien dan secepat mungkin. Sebagian besar DBMS dirancang dengan mempertimbangkan efisiensi; namun, dimungkinkan untuk meningkatkan kinerja basis data melalui pengaturan dan konfigurasi khusus.

Penyetelan sistem manajemen basis data berpusat di sekitar konfigurasi memori dan sumber daya pemrosesan komputer yang menjalankan DBMS. Ini dapat melibatkan pengaturan interval pemulihan DMBS, menetapkan tingkat kontrol konkurensi, dan menetapkan protokol jaringan mana yang digunakan untuk berkomunikasi di seluruh database. Memori yang digunakan oleh DBMS dialokasikan untuk data, prosedur pelaksanaan, cache prosedur, dan ruang kerja. Karena lebih cepat untuk

secara langsung mengakses data dalam memori daripada data pada penyimpanan, dimungkinkan untuk mengurangi waktu akses rata-rata dari transaksi basis data dengan mempertahankan cache data yang berukuran layak. Kinerja basis data juga dapat ditingkatkan dengan menggunakan cache untuk menyimpan prosedur pelaksanaan karena mereka tidak perlu dikompilasi ulang dengan setiap transaksi. Dengan menetapkan sumber daya pemrosesan ke fungsi dan aktivitas tertentu, juga dimungkinkan untuk meningkatkan konkurensi sistem.

DESKRIPSI PERCOBAAN

2.1. Tuning Indexing

Pada tuning indexing saya melakukan beberapa percobaan terhadap data yang diberikan, disini terdapat tabel berupa banyanya data yang haus dieksekusi dan di bandingkan sebelum dan sesudah di tuning.

Tabel 1. Terdapat tujuh data yang masing – masing memiliki kapasistas tertentu

Data
1. advisor = 100, student = 100, section = 200,takes = 200
2. advisor = 200, student = 200, section = 400,takes = 400
3. advisor = 500, student = 500, section = 1000,takes = 1000
4. advisor = 700, student = 700, section = 20000,takes = 20000
5. advisor = 1000, student = 1000, section = 100000,takes = 1000000
6. advisor = 1800, student = 1800, section = 180000,takes = 1800000
7. advisor = 10000, student = 10000, section = 30000000,takes = 30000000

Query

1. SELECT * FROM student
2. SELECT * FROM student WHERE tot_cred > 30;
3. SELECT `name`, department FROM student WHERE tot_cred > 30;
4. SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id;
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

2.2. Tuning Setting Configuration DBMS

Pada tuning DBMS cukup sulit dipahami mengenai sintaks nya akan tetapi saya telah menemukan formula agar response time dapat menjadi optimal yaitu dengan menerapkan sintaks dibawah ini untuk konfigurasi DBMS :

```
#bdb_max_lock =10000


# Comment if you using InnoDB tables
#skip-innodb
innodb_data_home_dir = "C:/xampp/mysql/data"
innodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = "C:xampp/mysql/data"
#innodb_log_arch_dir = C:/xampp/mysql/data"


innodb_buffer_pool_size = 2000M
```

```
innodb_additional_mem_pool_size = 20M


innodb_log_file_size = 20M

innodb_log_buffer_size = 8M

#innodb_flush_log_at_tex_commit = 1

#innodb_lock_wait_timeout = 50


##UTF 8 Setting

#init-connect=\'SET NAMES utf8\'

#collation_server=utf8_unicode_ci

#character_set_server=utf8
```

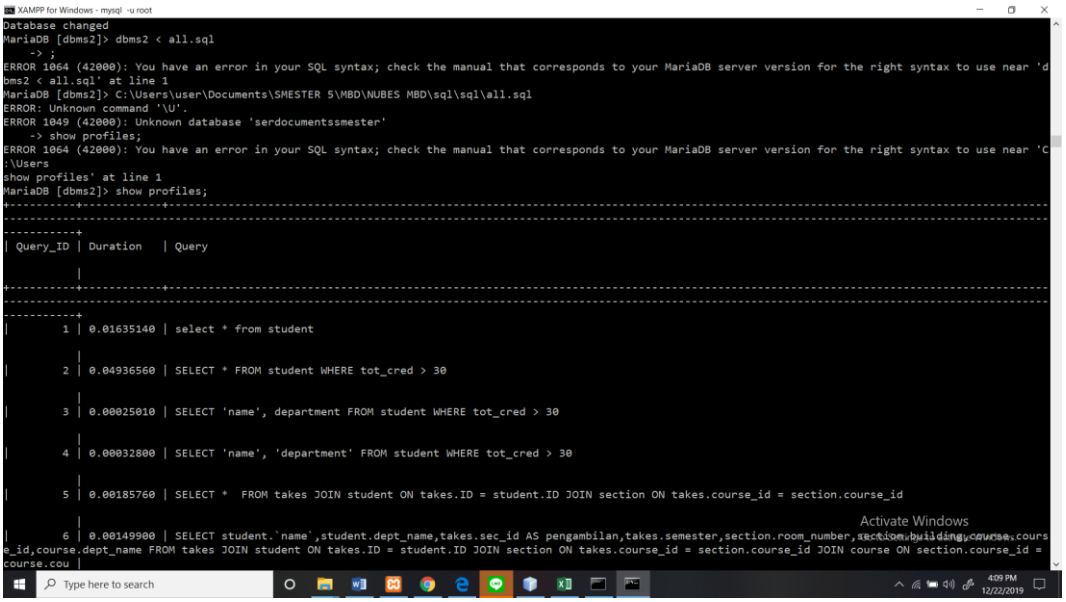
HASIL DAN PEMBAHASAN

3.1 Percobaan sebelum tuning Indexing

3.1.1. percobaan pada data nomor 1

advisor = 100, student = 100, section = 200,takes = 200.

Saya memasukkan percobaan nomor 1 pada database bernama dbms1 dan total terdapat 1070 data. Berikut ini hasil waktu dari proses sebelum di tuning indexing.

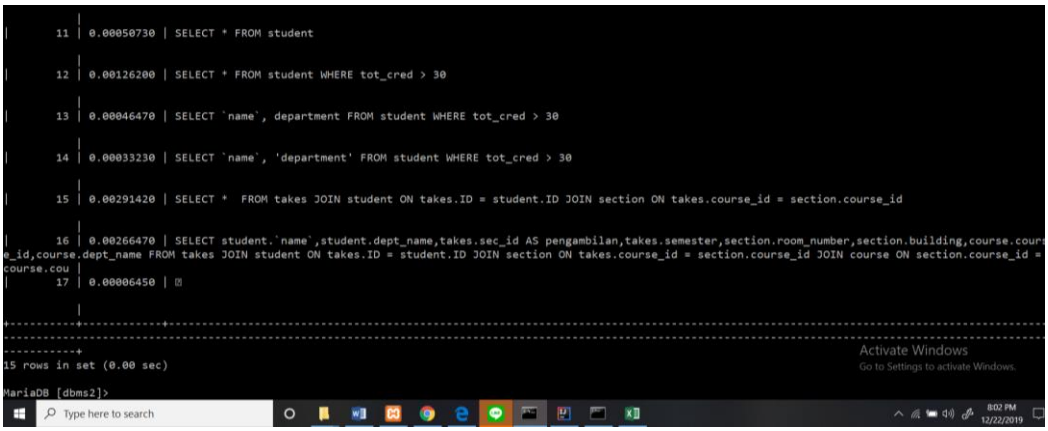


Gambar 1. percobaan menggunakan data no 1 sebelum di tuning.

3.1.2. percobaan pada data nomor 2

advisor = 200, student = 200, section = 400,takes = 400.

Saya memasukkan percobaan nomor 2 pada database bernama dbms2 dan total terdapat 1670 data. Berikut ini hasil waktu dari proses sebelum di tuning indexing.

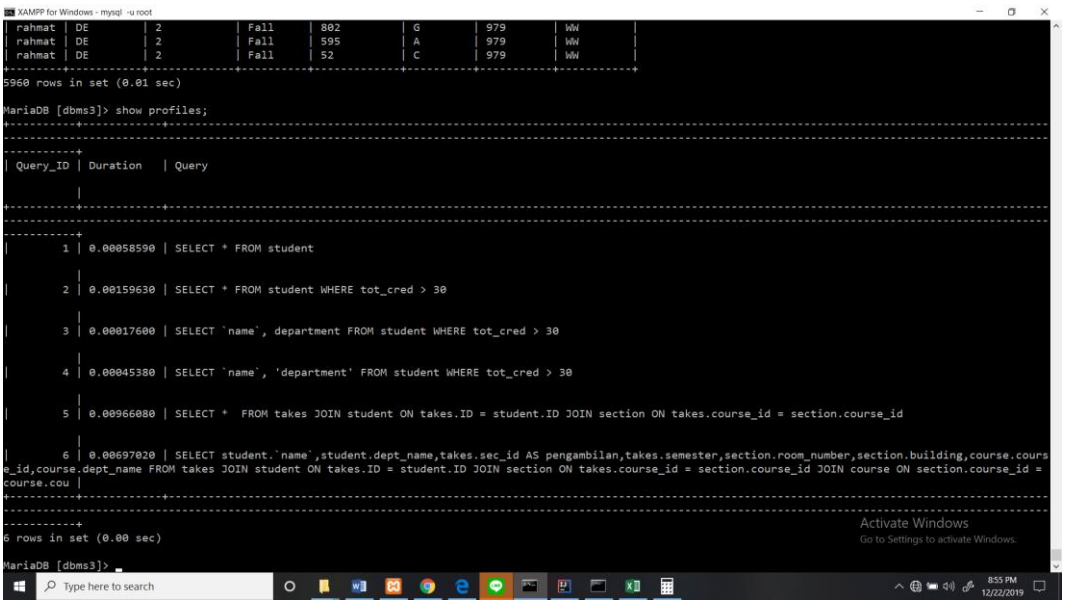


Gambar 2. percobaan menggunakan data no 2 sebelum di tuning.

3.1.3. percobaan pada data nomor 3

advisor = 500, student = 500, section = 1000,takes = 1000

Saya memasukkan percobaan nomor 3 pada database bernama dbms3 dan total terdapat 3470 data. Berikut ini hasil waktu dari proses sebelum di tuning indexing.



Gambar 3. percobaan menggunakan data no 3 sebelum di tuning.

3.2 Percobaan setelah tuning Indexing

Langkah-langkah tambahan yang dilakukan dalam indexing adalah membuat index pada field yang dipilih. Field dipilih berdasarkan frekuensi kemunculan paling besar atau field yang sering digunakan pada query untuk menguji data. Pada percobaan, field yang akan dibuat index yaitu:

1. Field ID pada tabel student.

2. Field sec_id pada tabel section.

3. Field course_id pada tabel course.

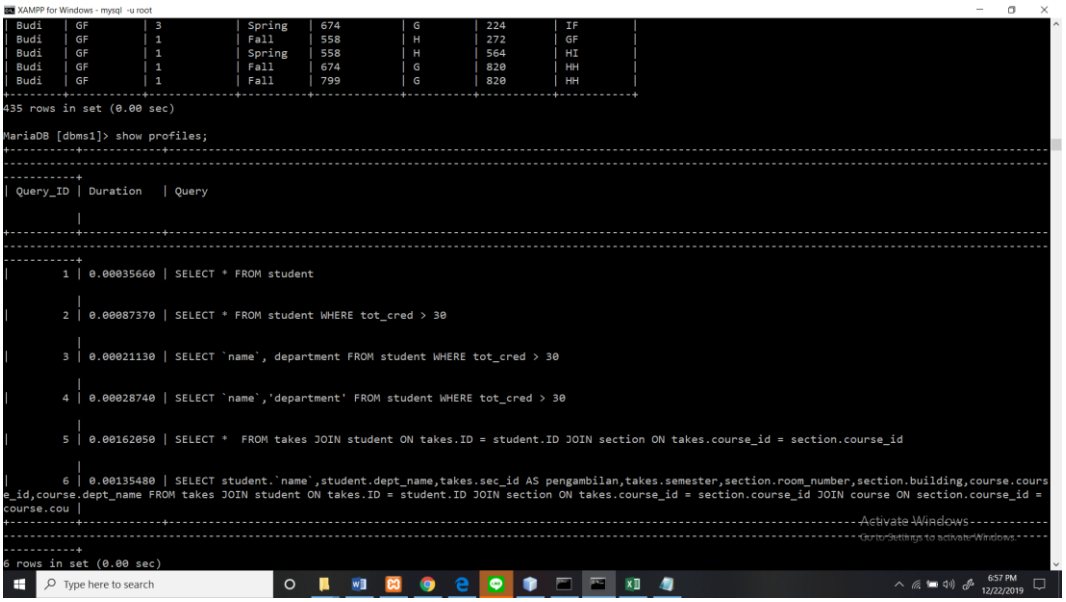
Berikut ini merupakan sintaks dari tuning Indexing :

```
CREATE INDEX student_pk ON student(ID);

CREATE INDEX section_pk ON section(sec_id);

CREATE INDEX course_pk ON course(course_id);
```

3.2.1. percobaan pada data nomor 1

advisor = 100, student = 100, section = 200,takes = 200.

Saya memasukkan percobaan nomor 1 pada database bernama dbms1 dan total terdapat 1070 data. Berikut ini hasil waktu dari proses setelah di tuning indexing.

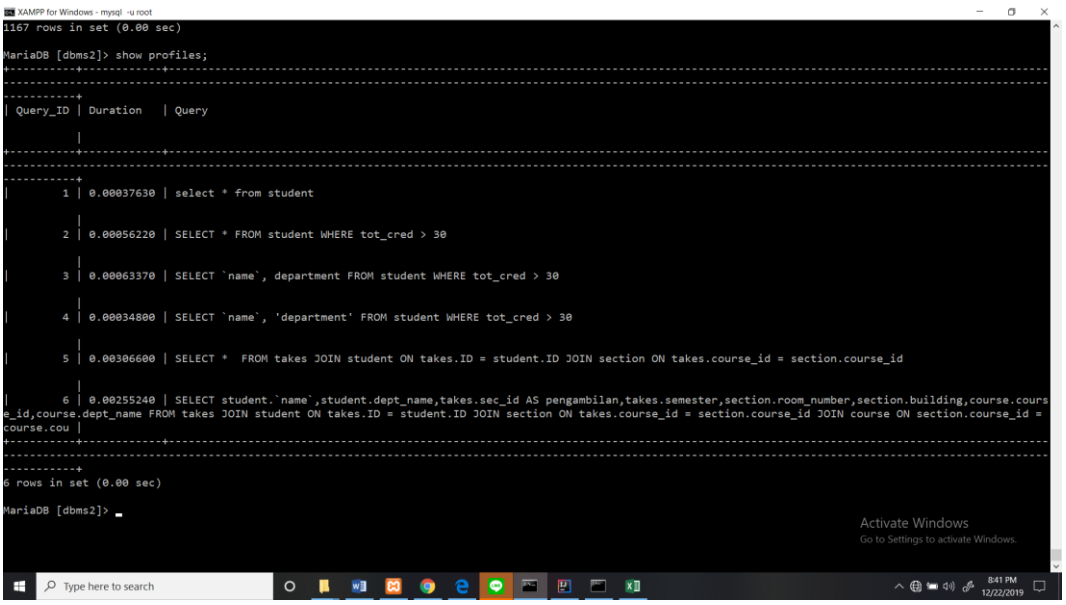


Gambar 4. Percobaan menggunakan data nomor 1 setelah di tuning indexing.

3.2.2. percobaan pada data nomor 2

advisor = 200, student = 200, section = 400,takes = 400.

Saya memasukkan percobaan nomor 2 pada database bernama dbms2 dan total terdapat 1670 data. Berikut ini hasil waktu dari proses setelah di tuning indexing.

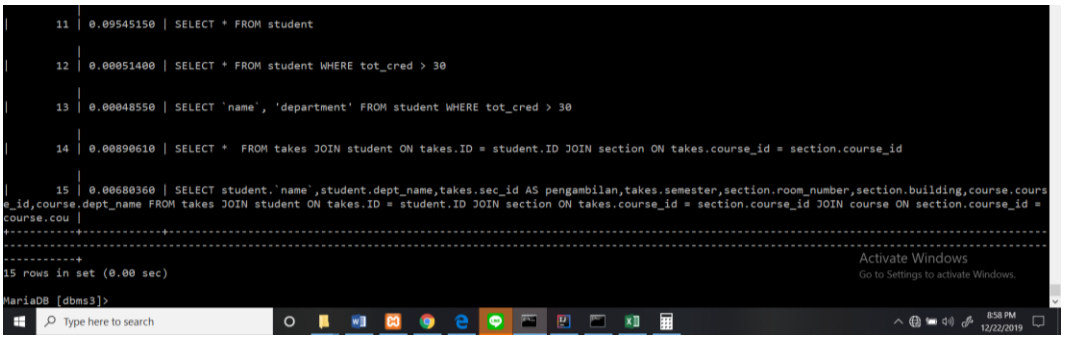


Gambar 5. Percobaan menggunakan data nomor 2 setelah di tuning indexing.

3.2.3. percobaan pada data nomor 3

advisor = 500, student = 500, section = 1000,takes = 1000.

Saya memasukkan percobaan nomor 3 pada database bernama dbms3 dan total terdapat 3470 data. Berikut ini hasil waktu dari proses setelah di tuning indexing.



Gambar 6. Percobaan menggunakan data nomor 3 setelah di tuning indexing.

3.2 Data Hasil

Data hasil dari masing - masing percobaan telah saya masukkan kedalam laporan.xlsx mudai dari data 1, data 2, dan data 3.

3.4 Pembahasan

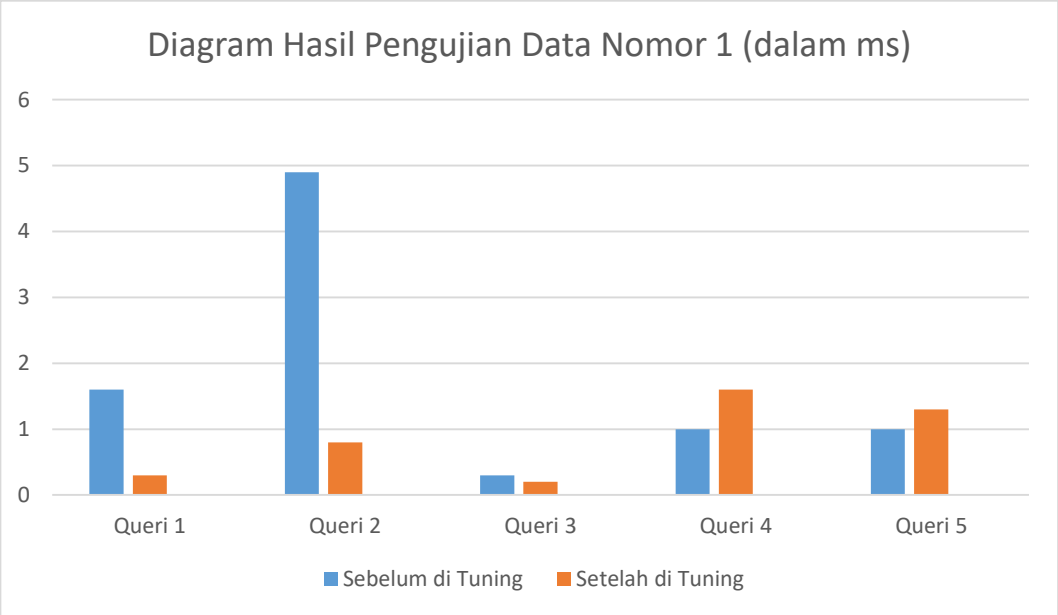


Diagram 1. Pengujian Data nomor 1.

Berdasarkan hasil dari diagram 1, kita telah dapatkan fakta bahwasannya pada Queri 2 terdapat penurunan response time secara signifikan yaitu 4.1 ms, walaupun pada Queri 4 dan 5 ini tidak berjalan baik.

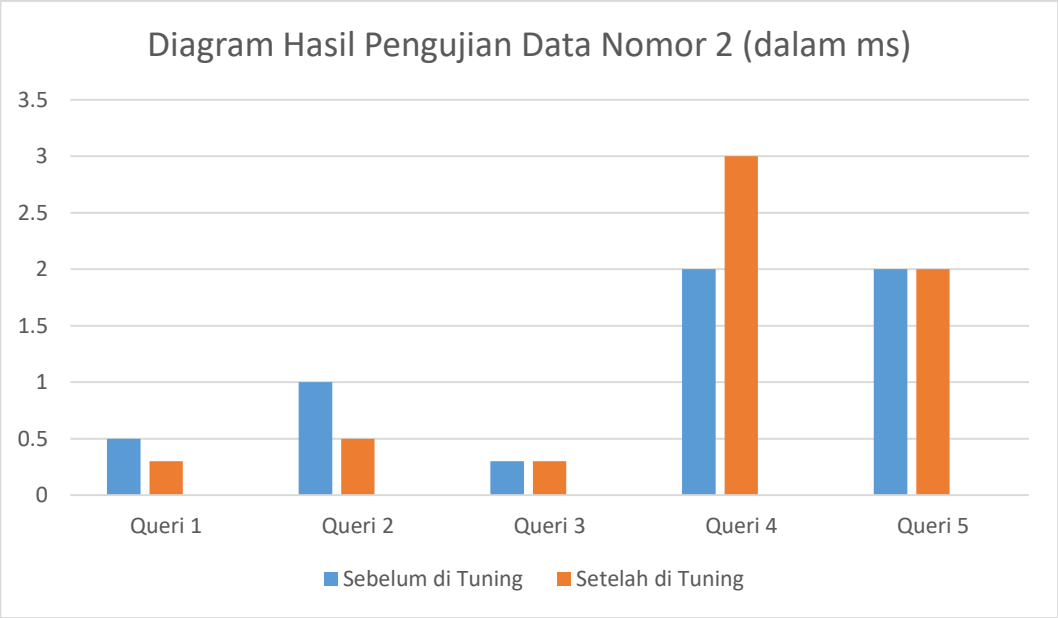


Diagram 2. Pengujian Data nomor 2.

Berdasarkan hasil dari diagram 2, kita telah dapatkan fakta bahwasannya pada Queri 1, 2, dan 3 menurunkan response time, akan tetapi masih tidak terhadap Queri 4, 5.

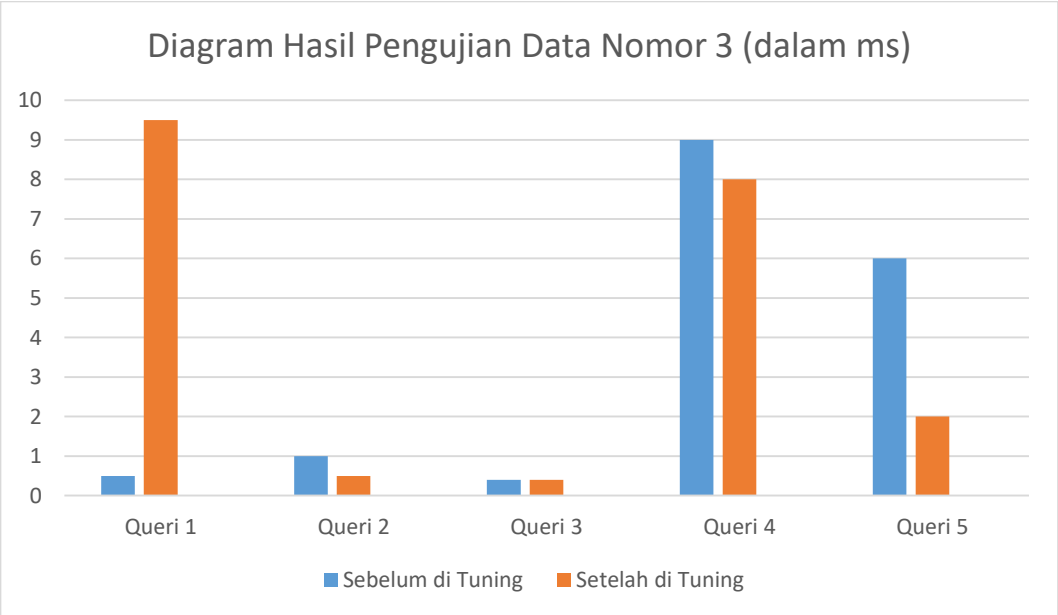


Diagram 3. Pengujian Data nomor 3.

Berdasarkan hasil dari diagram 3, kita telah mendapatkan fakta bahwasannya pada Queri 1 terdapat hasil yang tidak baik, sepertinya tuning indexing kurang cocok untuk data ini.

DAFTAR PUSTAKA

[1] Database Tuning. https://databasemanagement.fandom.com/wiki/Database_Tuning

[2] Tubes MBD. https://www.youtube.com/watch?v=93HKbyJ7y-k&t=1868s

[3] Performance Tuning Sederhana di mysql menggunakan Index. https://pojokprogrammer.net/content/performance-tuning-sederhana-di-mysql-menggunakan-index