

**TUGAS BESAR**  
**MANAJEMEN BASIS DATA**  
**Tuning Indexing Dan Setting Cofiguration DBMS**



Disusun Oleh :  
M Junio Azhari (14117177)

**Program Studi Teknik Informatika Jurusan Teknologi  
Produksi, Industri dan Informasi Institut Teknologi  
Sumatera 2019**

## Daftar Isi

Daftar Isi .....	2
BAB 1 Studi Literatur .....	3
1.1 Tunning Indexing .....	3
1.2 Tunning Setting Configuration DBMS .....	3
BAB 2 Deskripsi Percobaan .....	4
2.1 Tuning Indexing .....	4
2.1.1 Sebelum Tuning .....	4
2.1.2 Sesudah Tuning .....	14
BAB 3 Hasil Dan Pembahasan .....	19
3.1 Hasil .....	19
3.2 Grafik Hasil Percobaan .....	19
3.2.1 Percobaan 1 .....	19
3.2.2 Percobaan 2 .....	20
3.2.3 Percobaan 3 .....	20
3.2.4 Percobaan 4 .....	20
3.3 Pembahasan .....	20
Daftar Pustaka .....	22

# **BAB 1**

## **Studi Literatur**

### **1.1 Tunning Indexing**

Tunning indexing adalah bagian penyetelan basis data untuk memilih dan membuat indeks. Tujuan Tunning indexing adalah untuk mengurangi waktu pemrosesan kueri. Potensi penggunaan indeks di lingkungan dinamis dengan beberapa permintaan ad-hoc sebelumnya adalah tugas yang sulit. Penyetelan indeks melibatkan kueri berdasarkan indeks dan indeks dibuat secara otomatis saat itu juga. Tidak diperlukan tindakan eksplisit oleh pengguna basis data untuk penyetelan indeks. Meningkatkan kinerja kueri dan basis data, dapat menggunakan langkah-langkah sebagai berikut:

1. Menggunakan pengoptimal kueri untuk melakukan analisis kueri sehubungan dengan beban kerja dan berdasarkan pengetahuan ini, ia merekomendasikan penggunaan indeks terbaik.
2. Perubahan dalam penggunaan indeks, distribusi permintaan dan kinerjanya dianalisis untuk memeriksa efeknya.
3. menggunakan Wisaya Penyetelan Indeks. SQL profiler digunakan untuk menangkap jejak aktivitas, untuk mengoptimalkan kinerja. Jejak dapat diperpanjang untuk jangka waktu tertentu dengan tujuan menangkap berbagai aktivitas.

### **1.2 Tunning 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 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 2

### Deskripsi Percobaan

#### 2.1 Tuning Indexing

##### 2.1.1 Sebelum Tuning

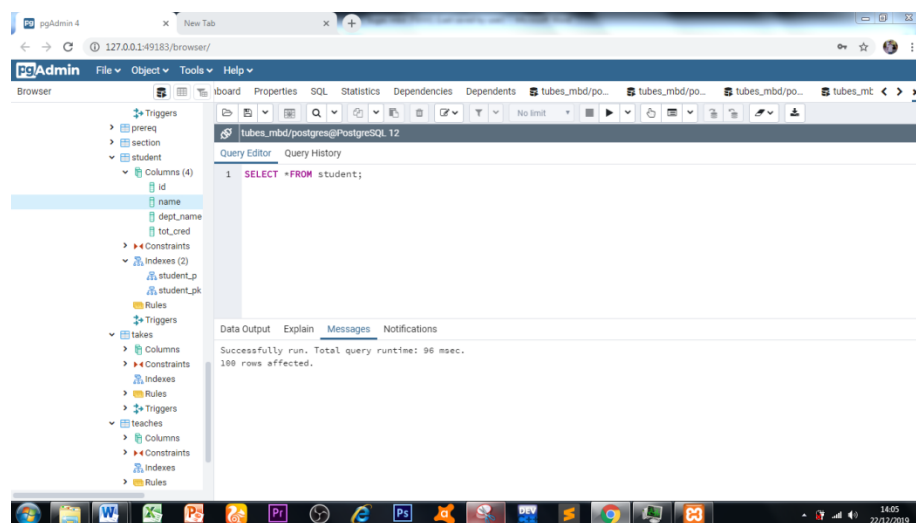
###### 1. Percobaan 1

Pada percobaan pertama menggunakan advisor = 100, student = 100, section = 200, takes = 200 mendapatkan proses waktu yang berbeda. Dapat dilihat pada tabel berikut :

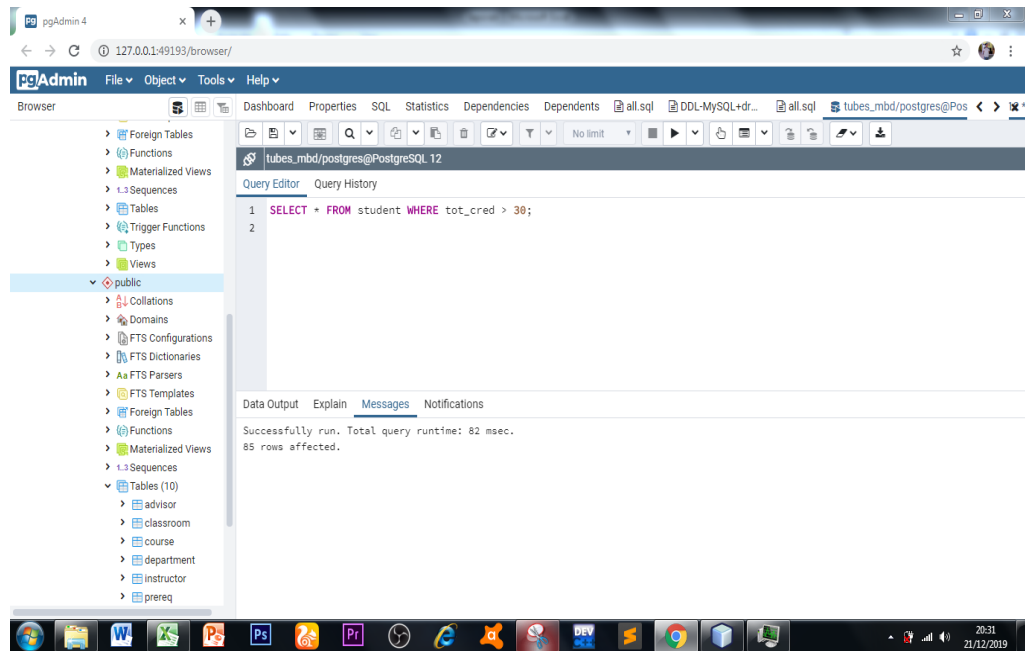
Query :

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;

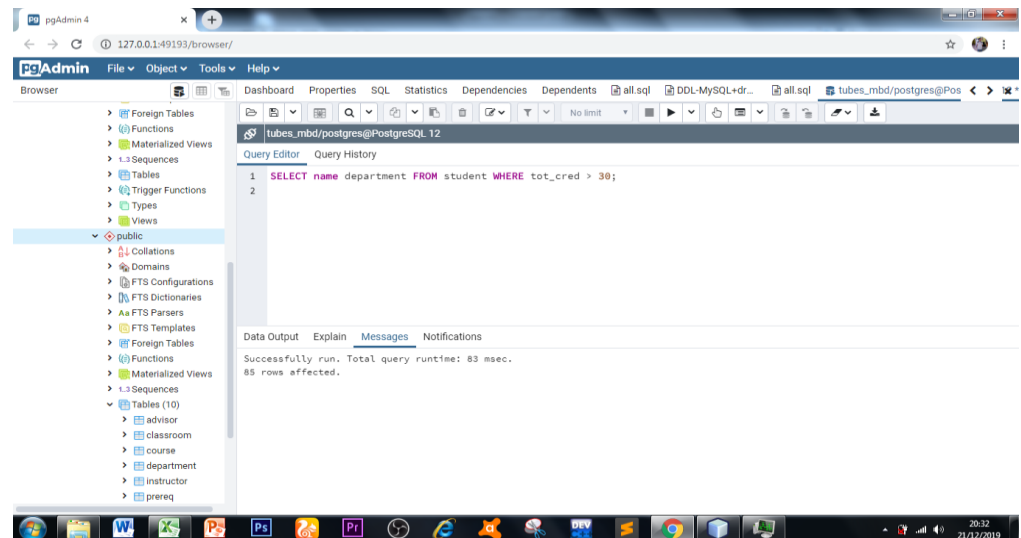
Query	Waktu Sebelum Tuning (msec)
1	96
2	82
3	83
4	89
5	75



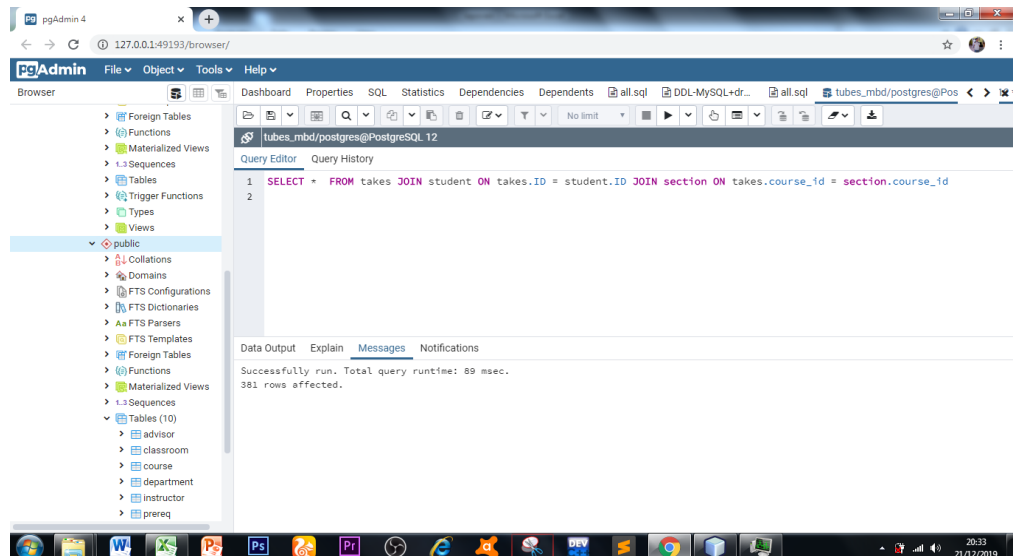
Gambar 1 Query 1



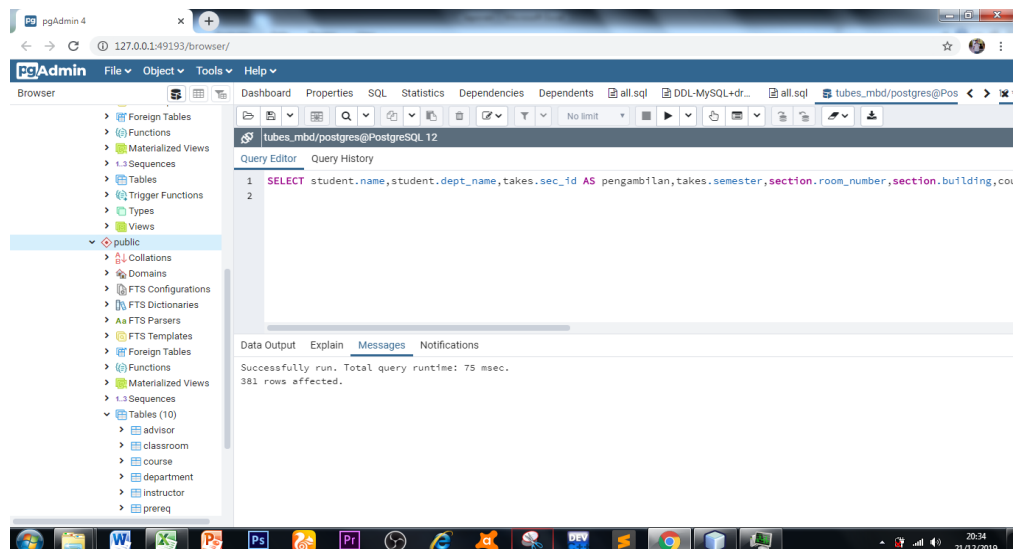
**Gambar 2 Query 2**



**Gambar 3 Query 3**



Gambar 4 Query 4



Gambar 5 Query 5

## 2. Percobaan 2

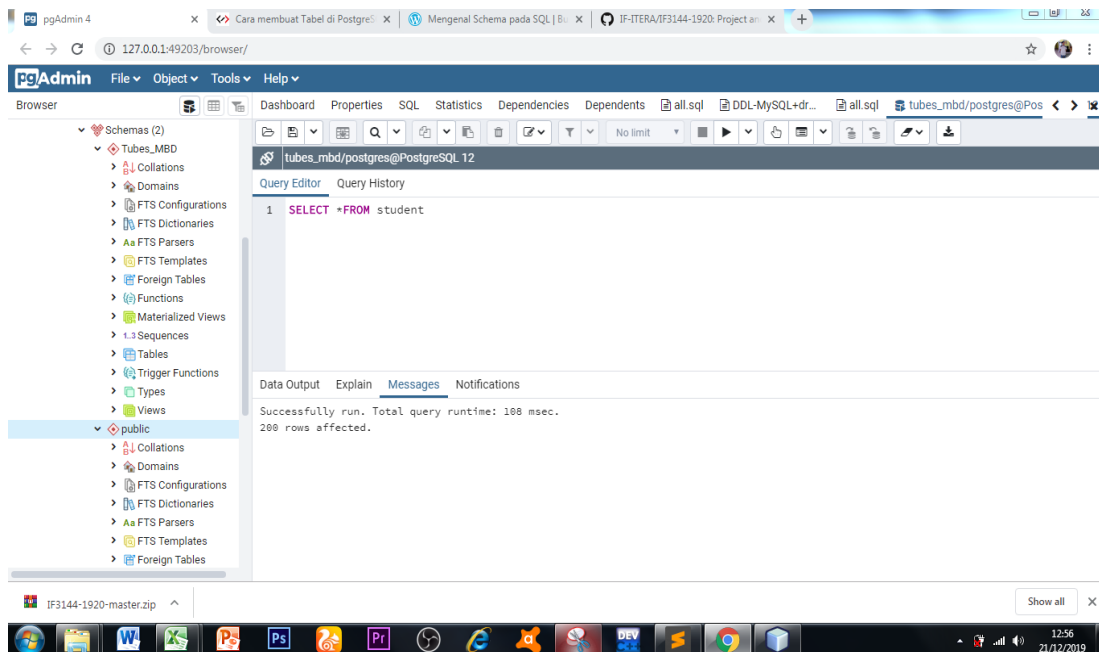
Pada percobaan kedua menggunakan advisor = 200, student = 200, section = 400, takes = 400 mendapatkan proses waktu yang berbeda. Dapat dilihat pada tabel berikut :

Query :

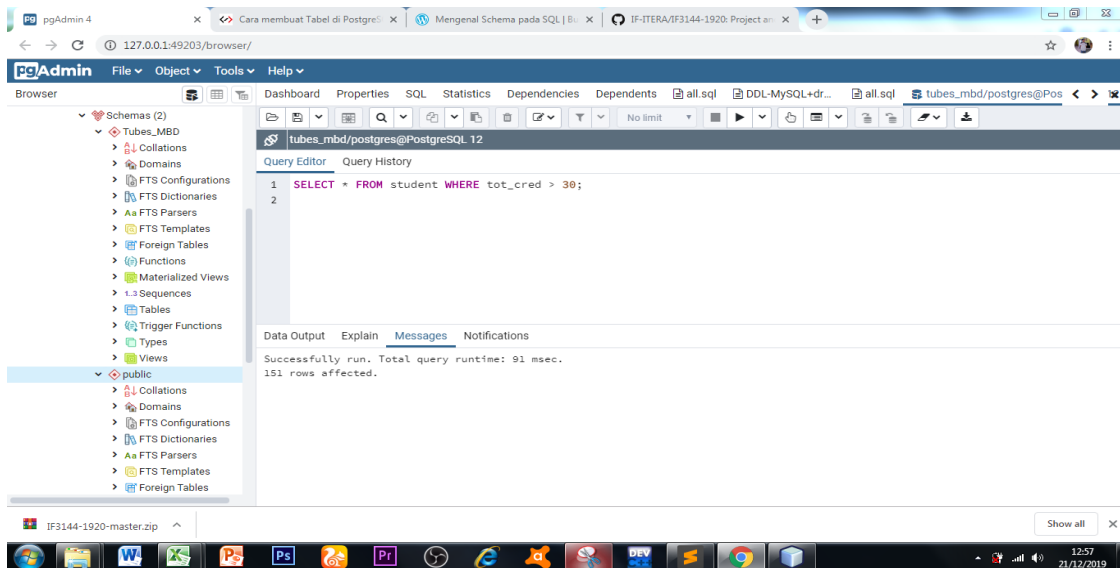
1. SELECT \* FROM student
2. SELECT \* FROM student WHERE tot\_cred > 30;
3. SELECT `name`, department FROM student WHERE tot\_cred > 30;
4. SELECT \* FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course\_id = section.course\_id;

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;

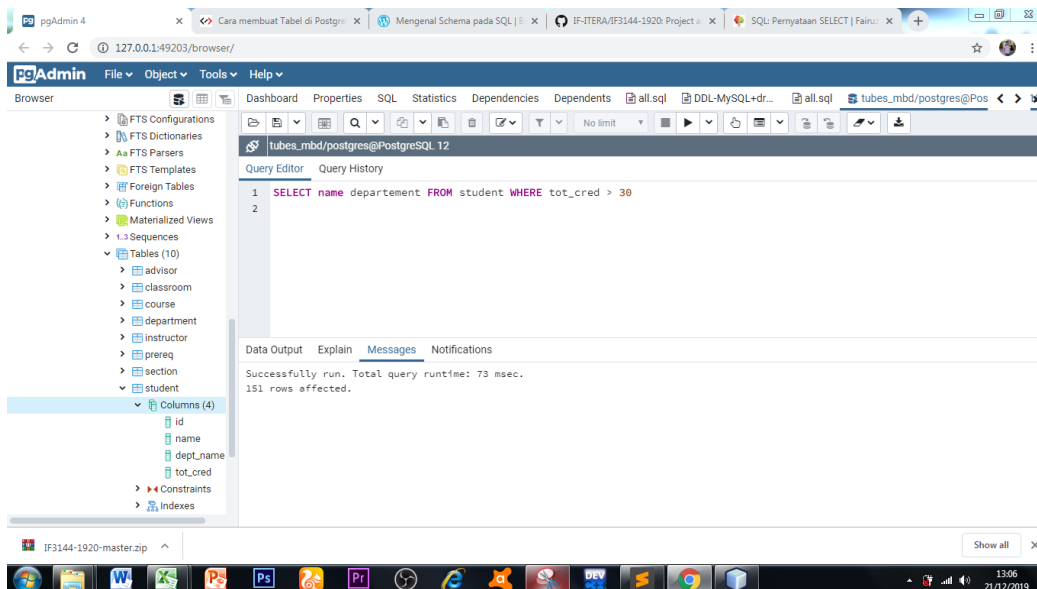
Query	Waktu Sebelum Tuning (msec)
1	108
2	91
3	73
4	148
5	126



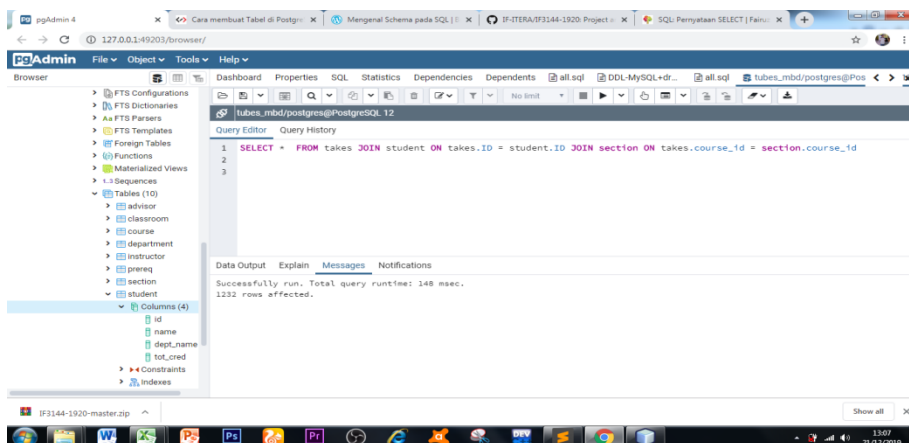
Gambar 6 Query 1



Gambar 7 Query 2

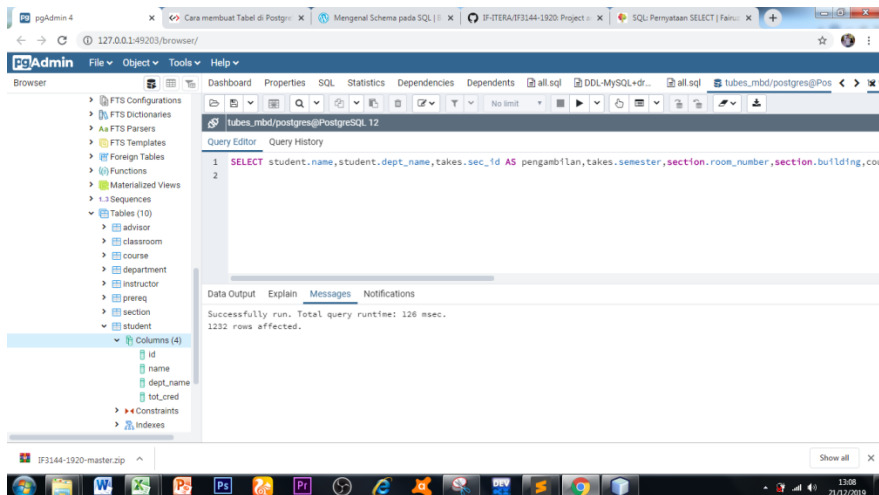


Gambar 8 Query 3



Gambar 9 Query 4





Gambar 10 Query 5

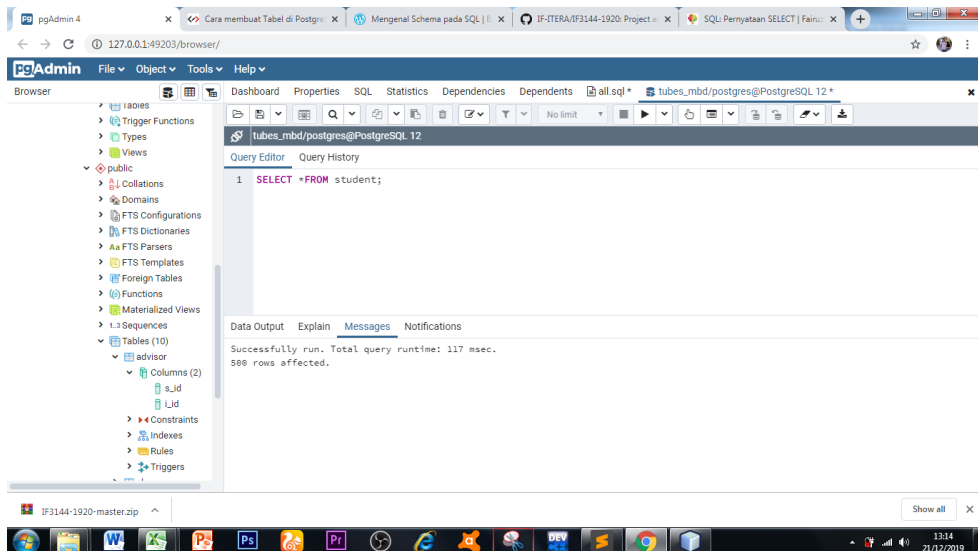
### 3. Percobaan 3

Pada percobaan ketiga advisor = 500, student = 500, section = 1000, takes = 1000 mendapatkan proses waktu yang berbeda. Dapat dilihat pada tabel berikut :

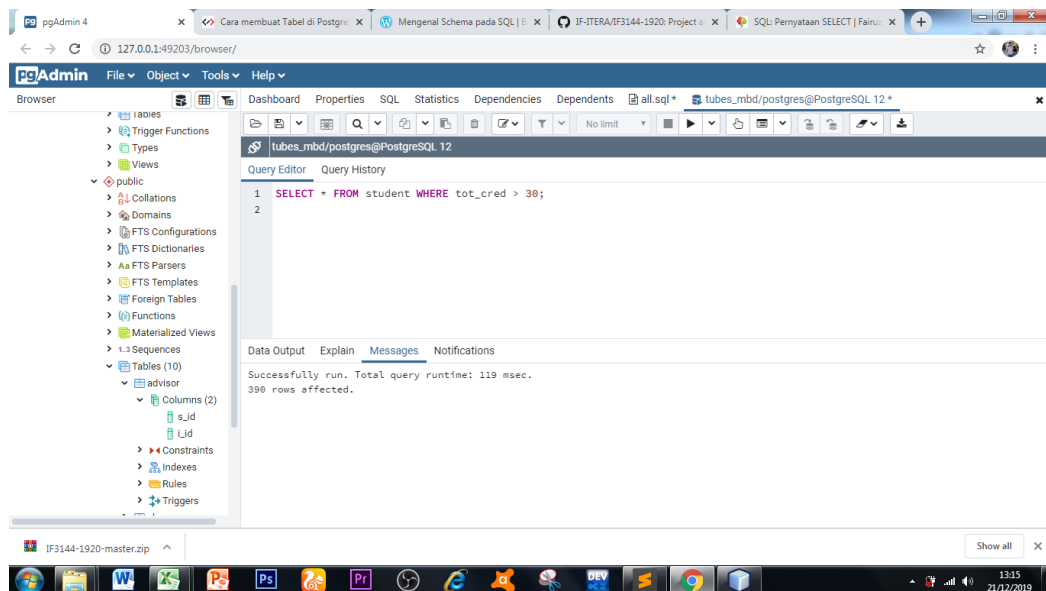
Query :

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;

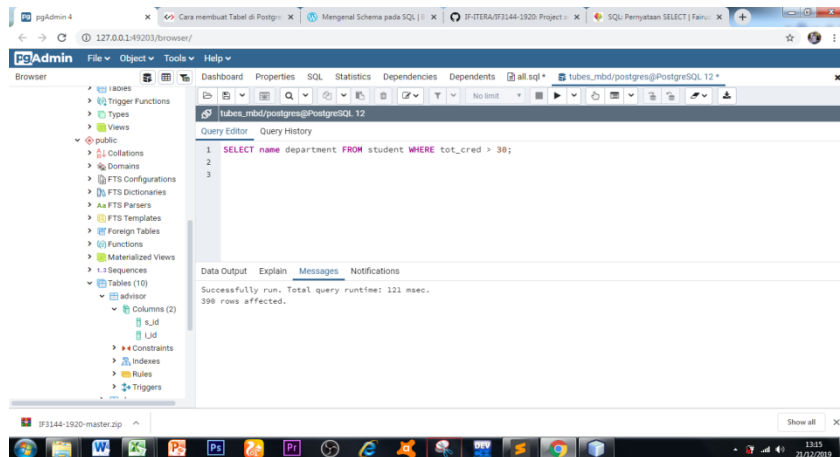
Query	Waktu Sebelum Tuning (msec)
1	117
2	119
3	121
4	210
5	119



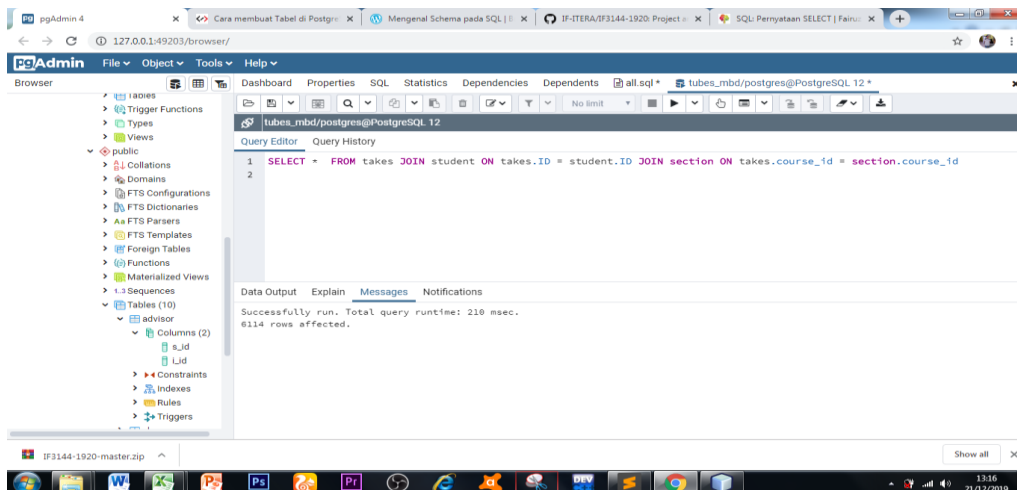
Gambar 11 Query 1



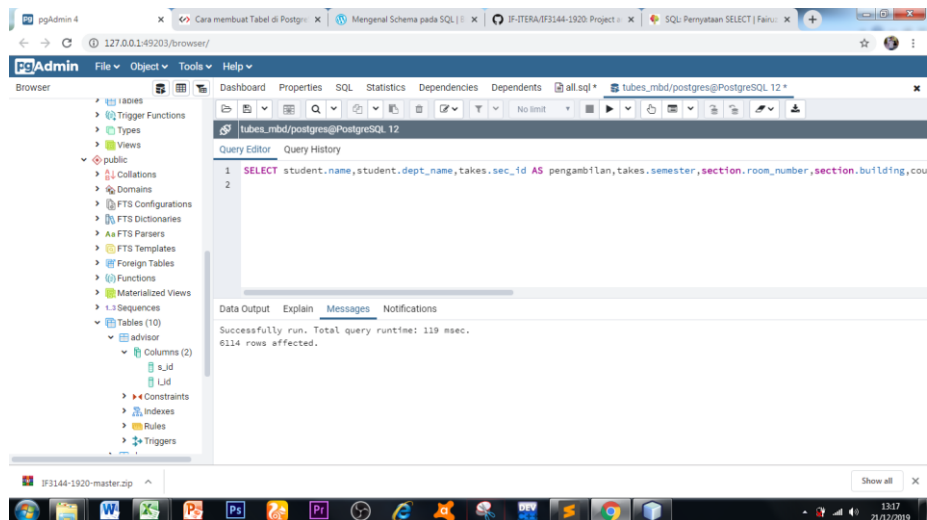
Gambar 12 Query 2



Gambar 13 Query 3



Gambar 14 Query 4



Gambar 15 Query 5

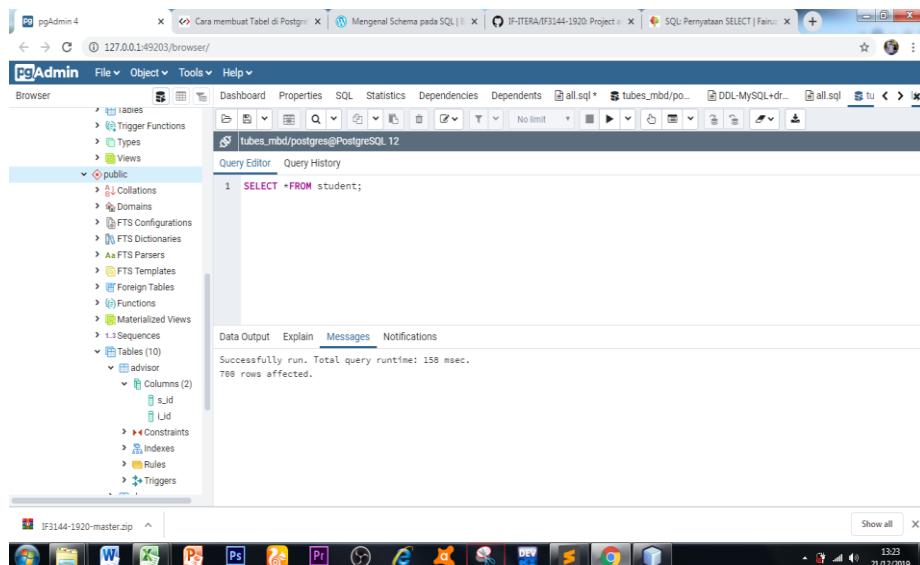
#### 4. Percobaan 4

Pada percobaan ke-4 advisor = 700, student = 700, section = 20000, takes = 20000 mendapatkan proses waktu yang berbeda. Dapat dilihat pada tabel berikut :

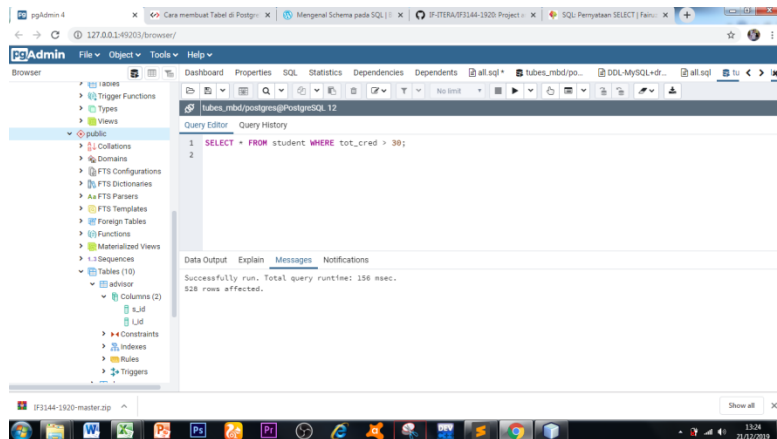
Query :

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;

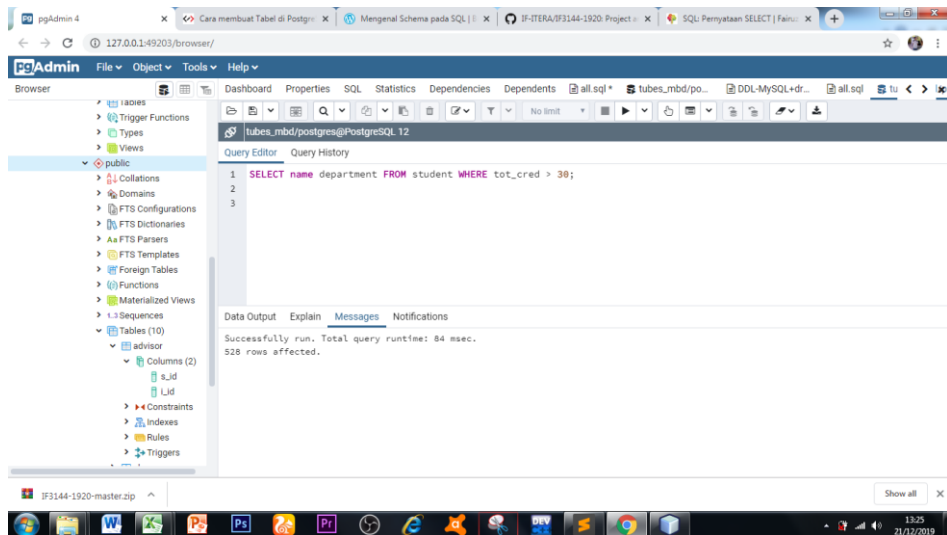
Query	Waktu Sebelum Tuning (msec)
1	158
2	156
3	84
4	11375
5	5558



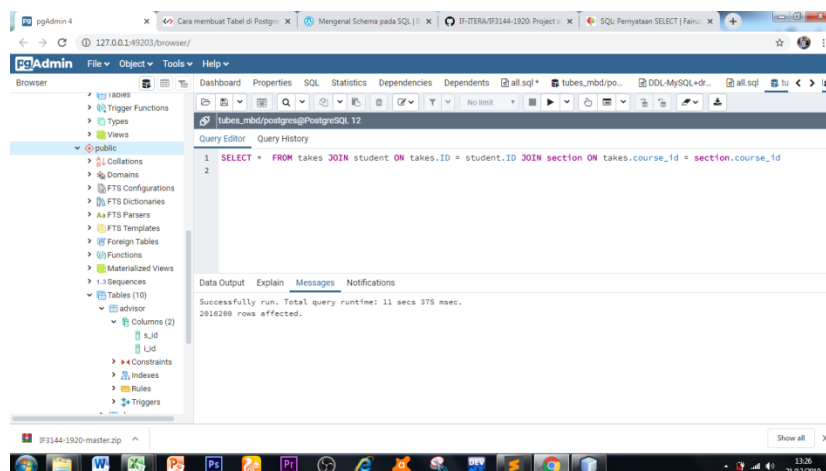
Gambar 16 Query 1



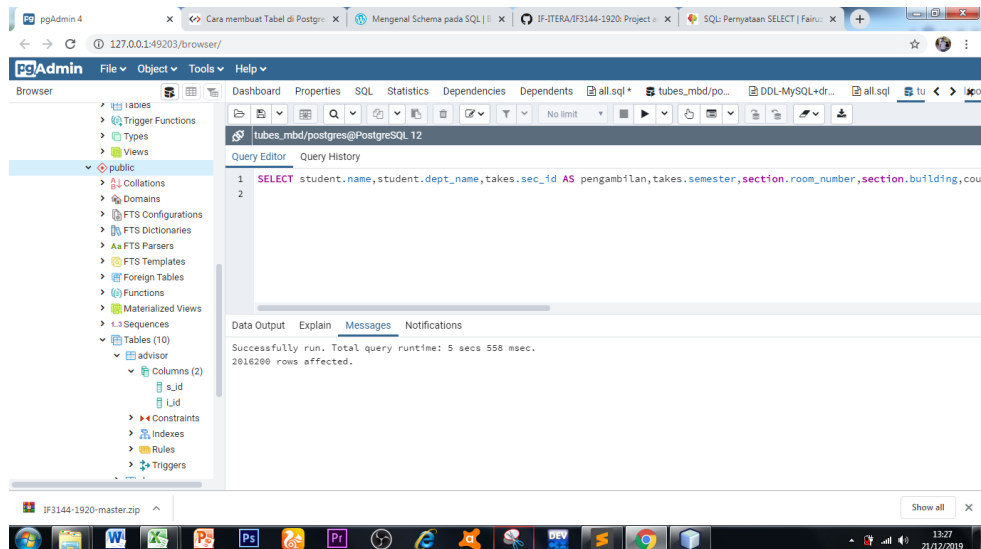
Gambar 17 Query 2



Gambar 18 Query 3



Gambar 19 Query 4



**Gambar 20 Query 5**

## 2.1.2 Sesudah Tuning

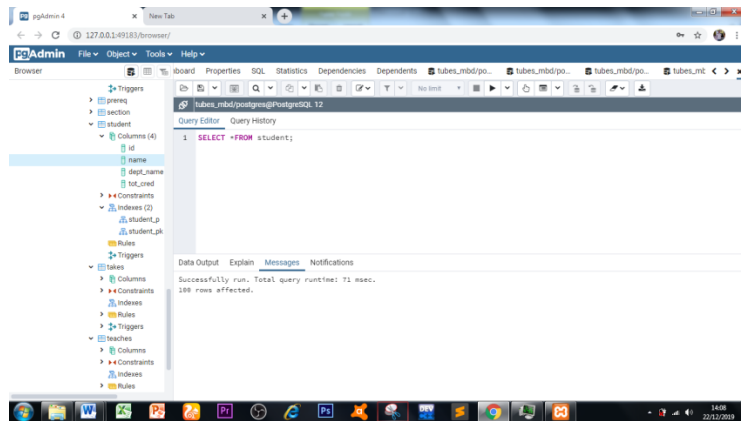
### 1. Percobaan 1

Pada percobaan pertama menggunakan advisor = 100, student = 100, section = 200, takes = 200 mendapatkan proses waktu yang berbeda. Dapat dilihat pada tabel berikut :

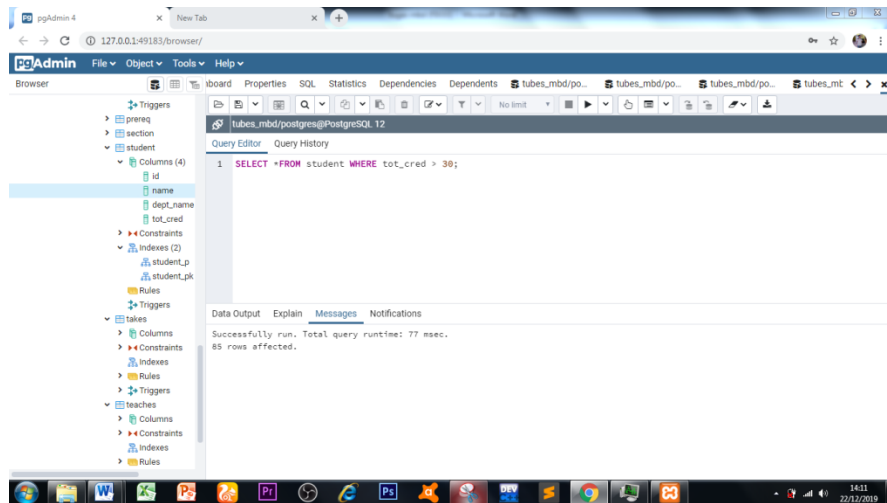
Query :

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;

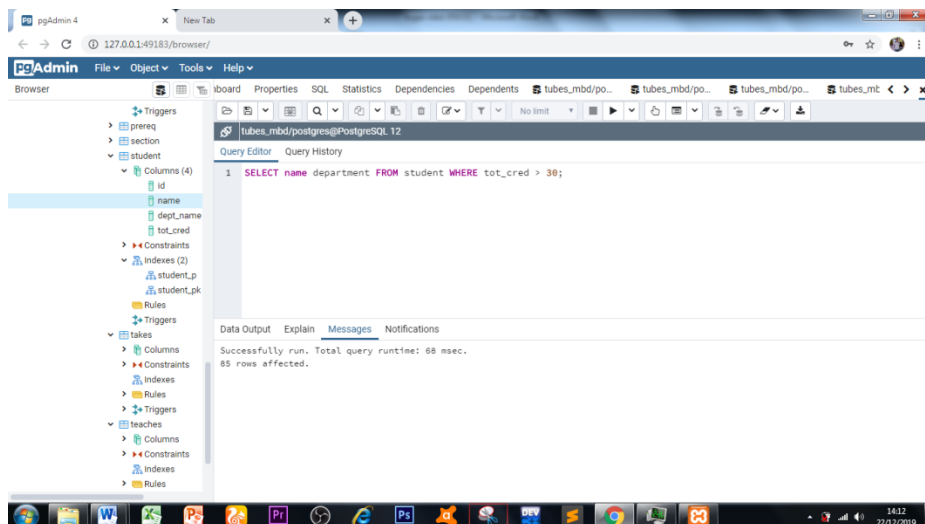
Query	Waktu Sesudah Tuning (msec)
1	71
2	77
3	68
4	104
5	96



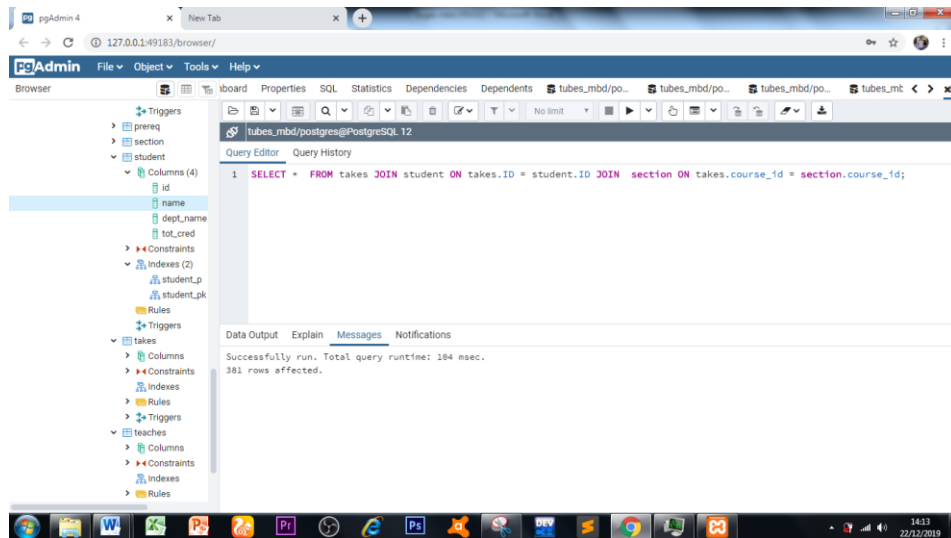
Gambar 21 Query 1



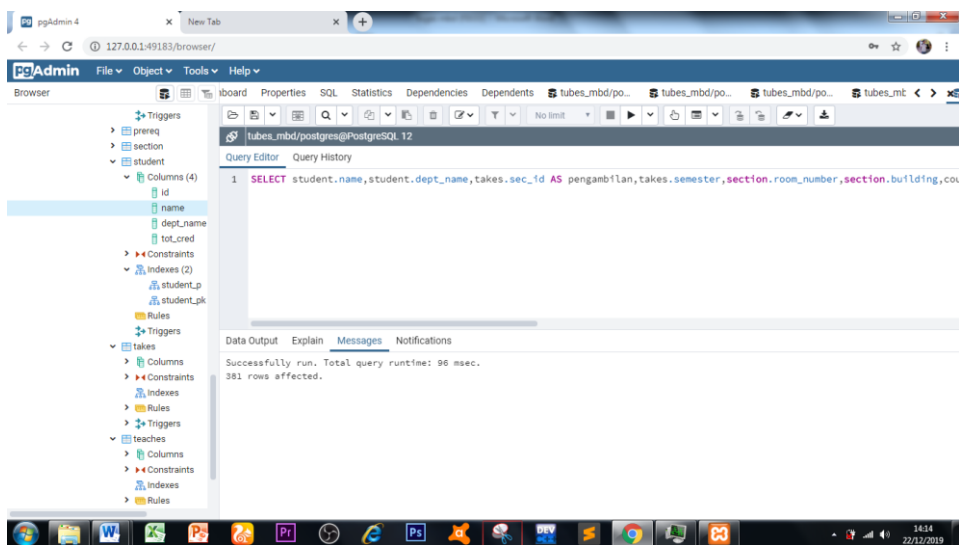
Gambar 22 Query 2



Gambar 23 Query 3



**Gambar 24 Query 4**



**Gambar 25 Query 5**

## 2. Percobaan 2

Pada percobaan kedua menggunakan advisor = 200, student = 200, section = 400, takes = 400 mendapatkan proses waktu yang berbeda. Dapat dilihat pada tabel berikut :

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;

Query	Waktu Sesudah Tuning (msec)
1	76
2	88
3	70
4	148
5	110

### 3. Percobaan 3

Pada percobaan ketiga advisor = 500, student = 500, section = 1000,takes = 1000 mendapatkan proses waktu yang berbeda. Dapat dilihat pada tabel berikut :

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;

Query	Waktu Sesudah Tuning (msec)
1	113
2	106
3	99
4	158
5	117

### 4. Percobaan 4

Pada percobaan ke-4 advisor = 700, student = 700, section = 20000,takes = 20000 mendapatkan proses waktu yang berbeda. Dapat dilihat pada tabel berikut :

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;

Query	Waktu Sebelum Tuning (msec)
1	120
2	132
3	77
4	10093
5	6636

## BAB 3

### Hasil Dan Pembahasan

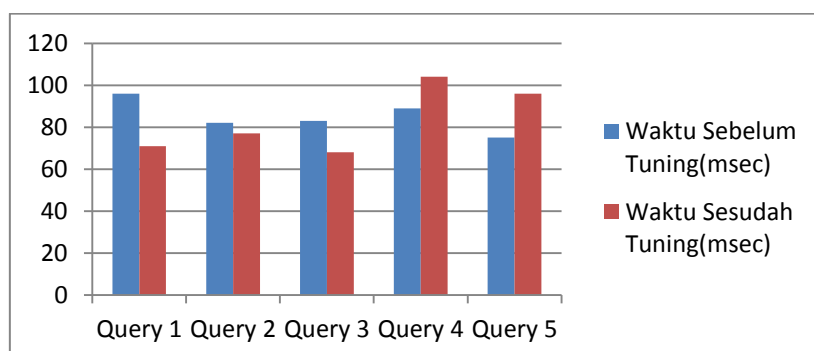
#### 3.1 Hasil

Pada bagian ini, hasil seluruh percobaan tuning dan sebelum tuning dibuat dalam satu tabel agar dapat mempermudah dalam membandingkan waktu sebelum tuning dan waktu sesudah tuning.

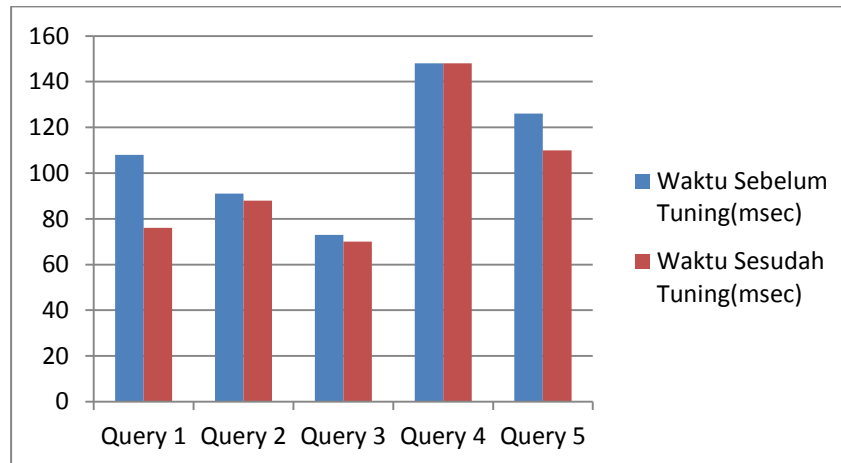
Percobaan	Query	Waktu Sebelum Tuning(msec)	Waktu Sesudah Tuning(msec)
1	1	96	71
	2	82	77
	3	83	68
	4	89	104
	5	75	96
2	1	108	76
	2	91	88
	3	73	70
	4	148	148
	5	126	110
3	1	117	113
	2	119	106
	3	121	99
	4	210	158
	5	119	117
4	1	158	120
	2	156	132
	3	84	77
	4	11375	10093
	5	5558	6636

#### 3.2 Grafik Hasil Percobaan

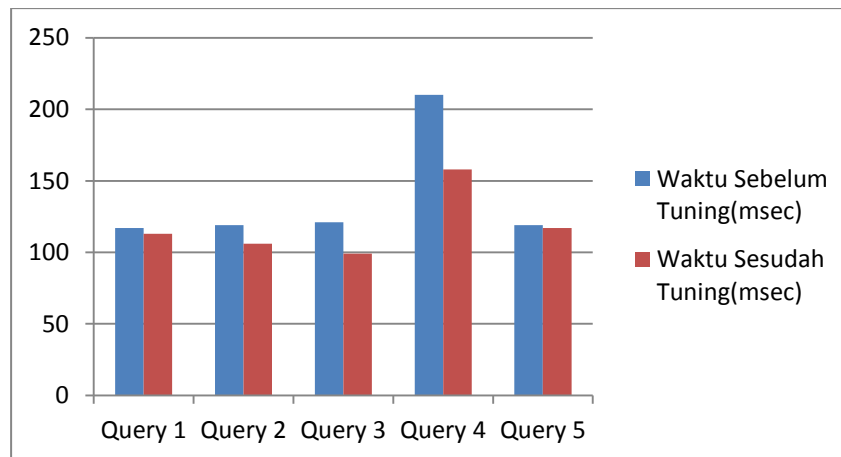
##### 3.2.1 Percobaan 1



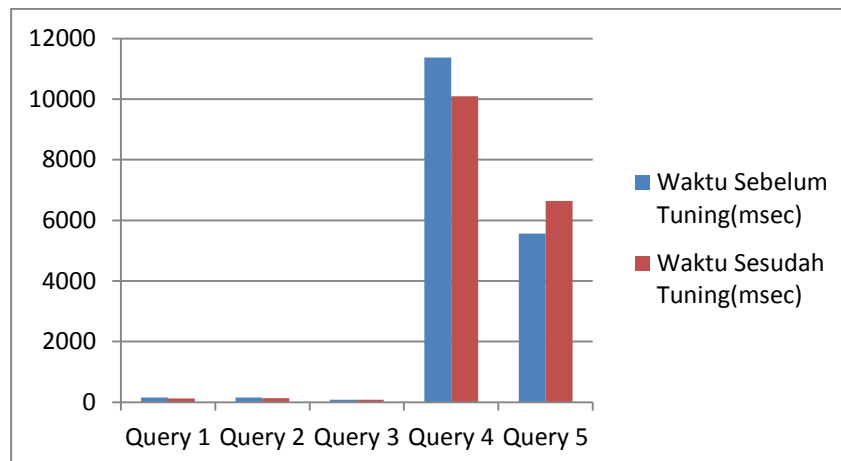
### 3.2.2 Percobaan 2



### 3.2.3 Percobaan 3



### 3.2.4 Percobaan 4



## 3.3 Pembahasan

Berdasarkan data pada tabel dan grafik hasil diatas, dapat dilihat bahwa pada Percobaan 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 hal ini terjadi bergantung pada cara pengecekan setiap query yang berbeda-beda. Untuk itu diperlukan tuning agar dapat mempercepat waktu pengekseskusan data. Tuning yang saya lakukan yaitu menggunakan teknik B-Tree. Setelah menggunakan teknik tersebut waktu yang didapatkan untuk melakukan ekseskusi 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

<https://odetocode.com/articles/237.aspx>

<https://www.careerride.com/DB-efficient-transactions.aspx>

<https://aws.amazon.com/blogs/machine-learning/tuning-your-dbms-automatically-with-machine-learning/>

<https://kelasprogrammer.com/cara-membuat-tabel-di-postgresql/>