

LAPORAN PRAKTIKUM
TEKNOLOGI CLOUD
PERTEMUAN KE – 6



Disusun Oleh :

NAMA	: TARISA DWI SEPTIA
NIM	: 205410126
JURUSAN	: TEKNIK INFORMATIKA
JENJANG	: S1

Sekolah Tinggi Management Informatika Komputer

AKAKOM

YOGYAKARTA

2020

MODUL 6

DaaS – ElephantSQL

A. Tujuan

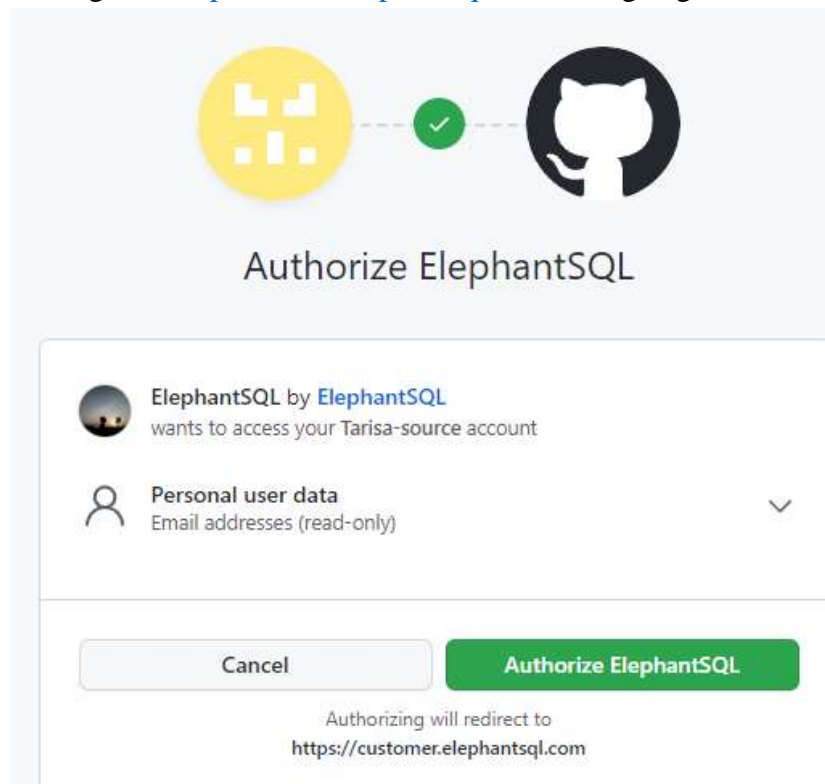
- Mahasiswa dapat melakukan proses otomatisasi pada setiap bagian dari pengaturan dan menjalankan cluster PostgreSQL.

B. Dasar Teori

ElephantSQL melakukan proses otomatisasi pada setiap bagian dari pengaturan dan menjalankan cluster PostgreSQL. Tersedia di semua platform cloud dan aplikasi di seluruh dunia. ElephantSQL ditawarkan oleh tim kecil namun stabil, berkomitmen, dan pekerja keras dari Swedia. Dengan 20.000+ database yang berjalan dan beberapa tahun pengalaman DBA PostgreSQL.

C. Praktik

- a. Membuat instance PostgreSQL
 1. Login ke <https://www.elephantsql.com/> dengan github



2. Membuat instance Posgresql

Select a plan and name - Step 1 of 4


Name:

Plan:

Tags:

Tags are used to separate your instances between projects. This is primarily used in the project listing view for easier navigation and access control.

Tags allow admins to manage team members' access to different groups of instances.

Plan: 
Tiny Turtle

See the plan page to learn about the different plans.


3. Lalu klik select region - pilih provider yang anda inginkan - lalu klik Confirm


Select a region and data center - Step 2 of 4

Progress: Plan **Region** Configure Launch

(Dedicated plans only)

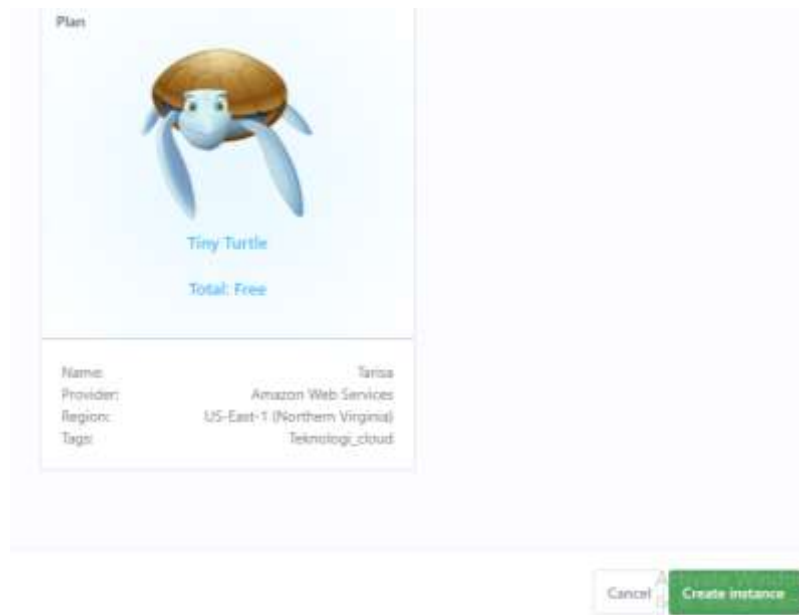
Data center:



Plan: 
Tiny Turtle

See the plan page to learn about the different plans.

4. Setelah langkah diatas, seluruh detail akan ditampilkan, dan kemudian klik

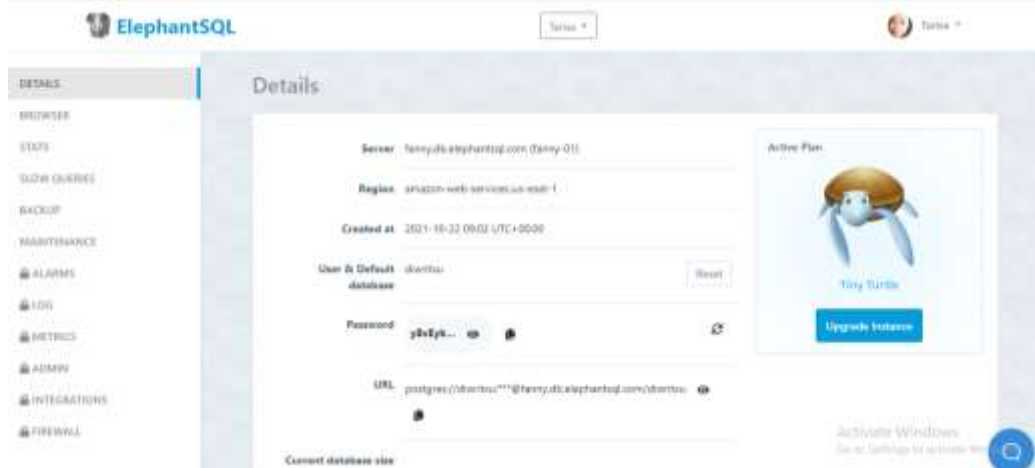


create instance.

5. Setelah di create akan muncul tampilan seperti ini



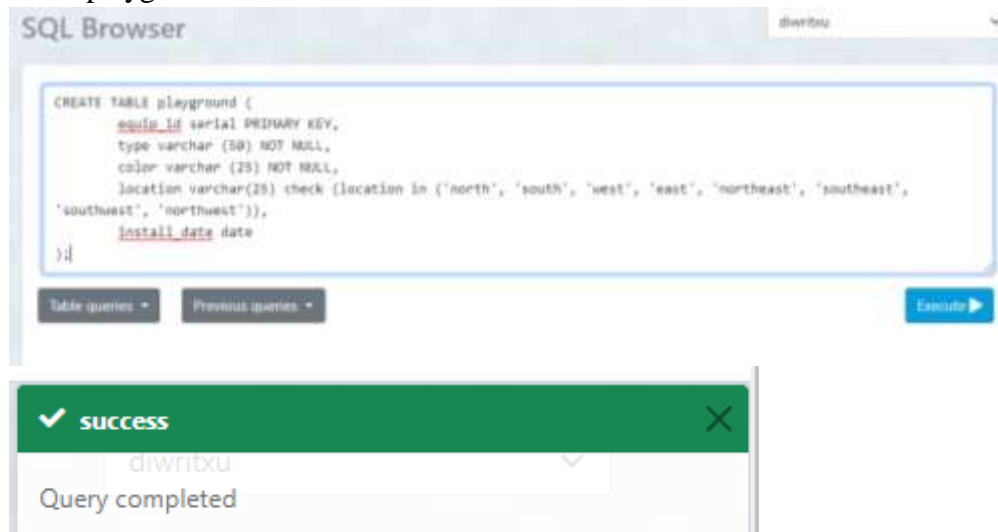
6. Untuk melihat detail, klik link nama instances, dan akan mendapatkan tampilan seperti gambar dibawah. Untuk masuk ke administrasi query, klik menu browser.



b. Melakukan proses uji coba

1. Membuat dan menghapus table

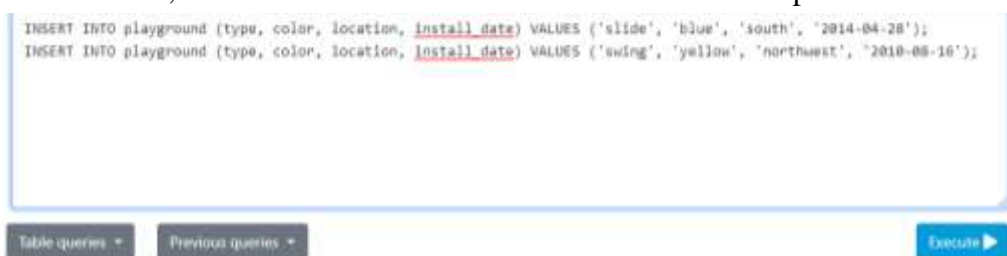
Setelah dapat terhubung ke sistem database PostgreSQL, kita dapat melakukan beberapa tugas sederhana. Pertama, kita akan membuat sebuah tabel untuk menyimpan beberapa data. Mari kita buat sebuah tabel yang mendeskripsikan daftar playground.



Kita telah membuat tabel playground untuk menyimpan data peralatan yang dimiliki. Tabel ini dimulai dengan sebuah ID yang bertipe serial. Tipe ada ini adalah sebuah auto-incrementing integer. Kita dapat memberikan konstrain primary key untuk kolom ini yang artinya isi kolom harus unik dan tidak null. Untuk dua kolom equip_id dan install_date kita tidak memberikan panjangnya. Hal ini karena beberapa tipe kolom tidak memerlukan panjang maksimum. Kita lalu membuat agar kolom type dan color tidak dapat dikosongkan. Kita juga membuat sebuah kolom location dan sebuah konstrain untuk membatasi nilainya dari 8 opsi. Kolom terakhir adalah kolom install_date untuk memberitahu kapan kita memasangnya.

2. Add, Query, dan Delete Data di Table

Setelah memiliki sebuah tabel, kita dapat memasukkan beberapa data ke dalamnya. Kita memasukkan data dengan memanggil tabel yang ingin ditambahkan, lalu memberikan nama kolom dan data untuk tiap kolom tersebut.



Ada beberapa hal yang perlu diperhatikan saat memasukkan data untuk meminimalkan kesalahan. Pertama, nama kolom tidak boleh diberi tanda "", namun isi kolom tersebut perlu diberikan. Hal lain yang perlu dilakukan adalah memastikan untuk tidak memberikan data kolom equip_id secara manual. Hal ini karena isinya akan dibuat secara otomatis. Kita dapat melihat data dengan memasukan *SELECT * FROM playground;*

Maka outputnya akan muncul seperti ini :

equip_id	type	color	location	install_date
1	slide	blue	south	2014-04-28
2	swing	yellow	northwest	2010-08-16

Disini, kita dapat melihat bahwa kolom equip_id secara otomatis sudah ada dan data yang lain sukses ditambahkan.

Jika ingin menghapus data maka digunakan perintah *DELETE FROM playground WHERE type = 'slide'*;

Table queries ▾

Previous queries ▾

Execute ▶

No rows returned

Setelah itu kita melihat lagi table dengan perintah *SELECT * FROM playground;*

Untuk memastikan data sudah terhapus.

Table queries ▾

Previous queries ▾

Execute ▶

equip_id	type	color	location	install_date
2	swing	yellow	northwest	2010-08-16

3. Menambah dan Menghapus Kolom dari Tabel

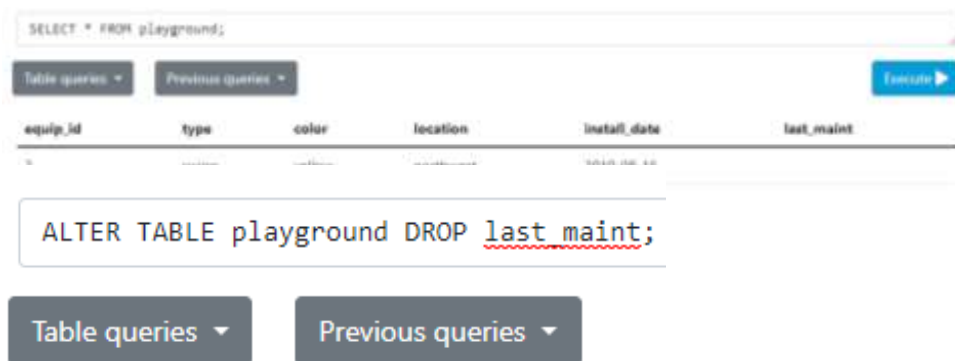
Jika ingin memodifikasi sebuah tabel setelah dibuat untuk menambah kolom



baru, kita dapat melakukannya dengan mudah.

Kemudian melihat informasi table kembali dengan perintah `SELECT * FROM playground;`

Kita juga dapat menghapus sebuah kolom dengan mudah. Menghapusnya dengan cara dengan menggunakan perintah `ALTER TABLE playground DROP`

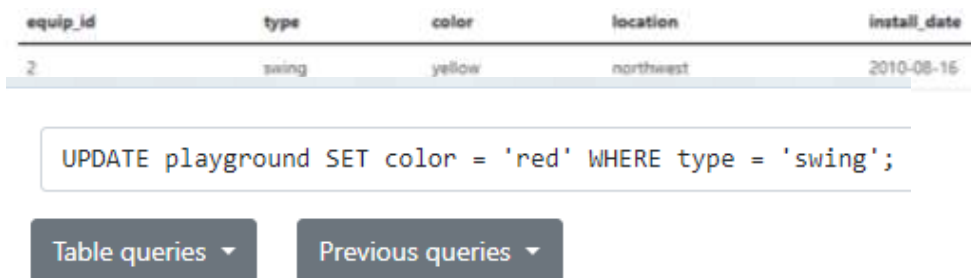


`last_maint;`

Setelah dihapus 1 table maka tampilan akan menjadi seperti semula

4. Mengupdate Data di Tabel

Kita tahu bagaimana menambah data ke dalam sebuah tabel dan bagaimana menghapusnya, tapi kita belum membahas bagaimana untuk memodifikasi data yang sudah ada. Misalnya, kita ingin mengambil data dengan tipe "swing" dan



mengganti warnanya menjadi merah (red) (ini akan mengganti seluruh data yang memiliki tipe "swing"):

Memeriksa apakah perintah tersebut berhasil dengan `SELECT * FROM playground;`

equip_id	type	color	location	install_date
2	swing	red	northwest	2010-08-16

D. Kesimpulan

Setelah melakukan praktik di atas dapat disimpulkan bahwa mahasiswa mampu melakukan proses otomasi pada setiap bagian dari pengaturan dan menjalankan cluster PostgreSQL.