

# DATA ENGINEER DAY 1: SQL DAN NO SQL

## TUJUAN PRAKTIKUM

1. Melakukan operasi *create*, *read*, *update* dan *delete* pada RDBMS PostgreSQL dan basis data MongoDB
2. Melakukan *query* pada masing-masing basis data menggunakan *query* SQL dan *query* Mongo Shell

## DASAR TEORI

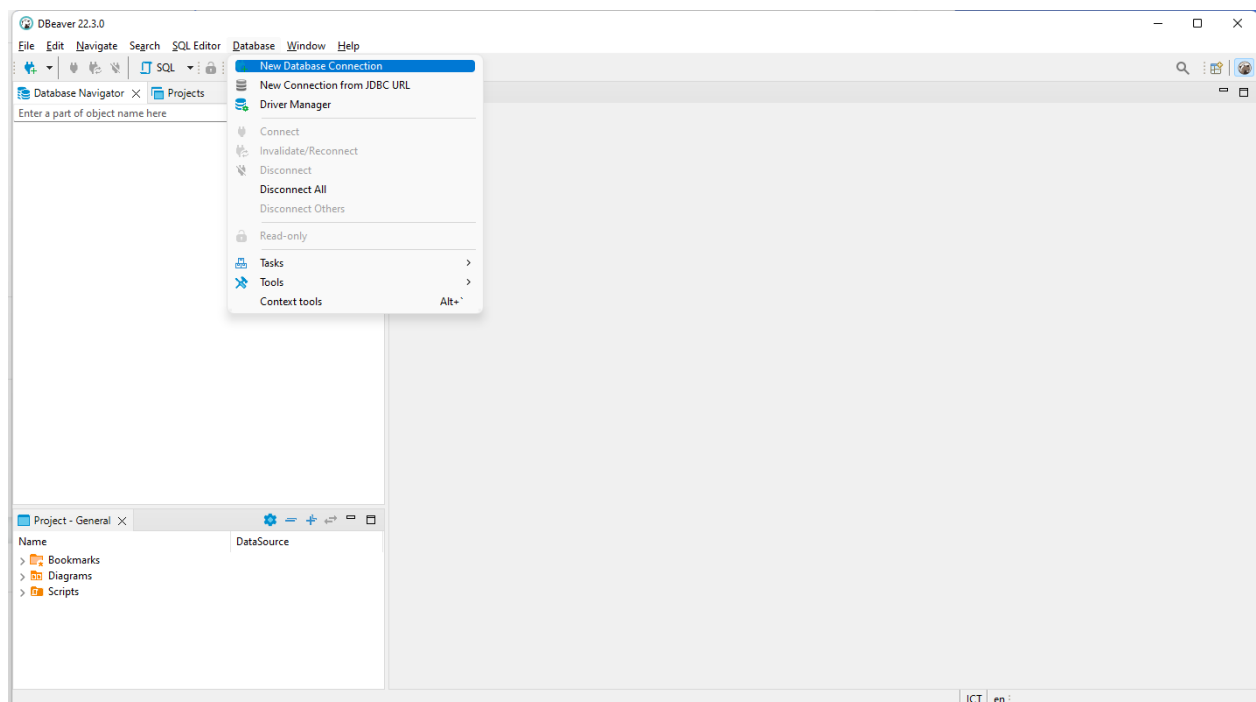
Slide Data Engineer Day 2: 📅 Day1 - Storage: SQL & NoSQL

## ALAT PERCOBAAN

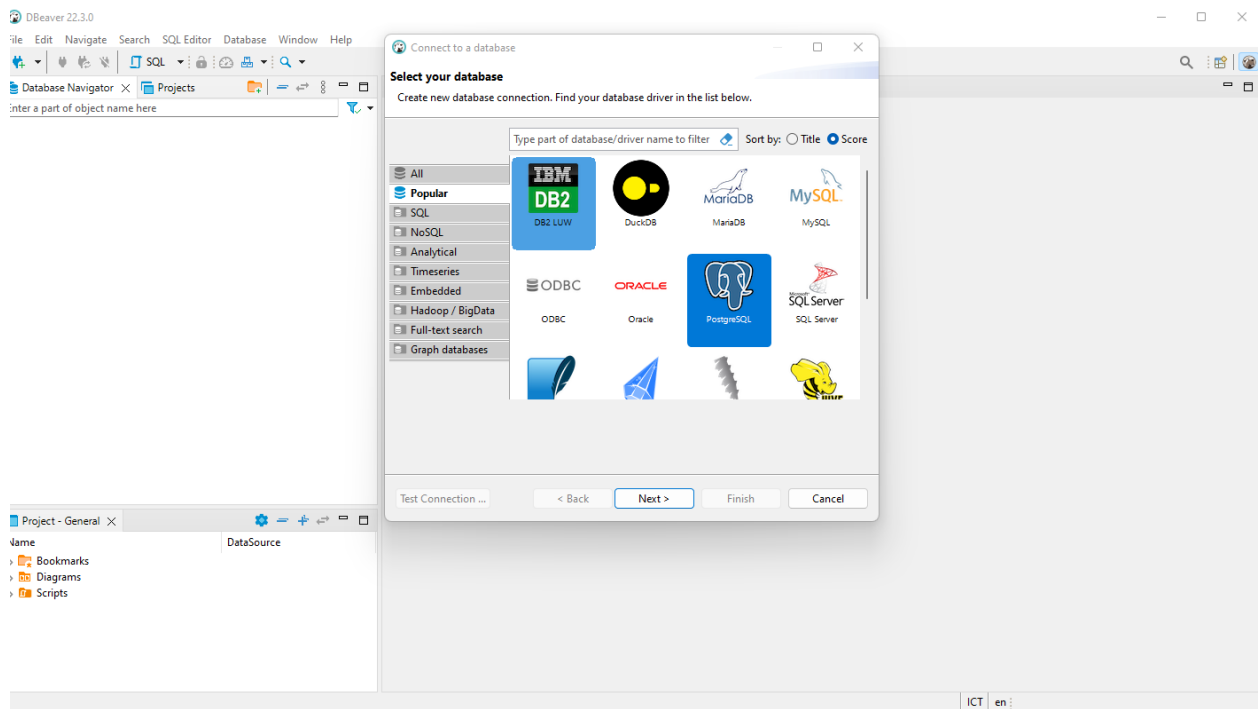
1. MongoDB Compass: download <https://www.mongodb.com/try/download/compass> dan install
2. DBeaver: download <https://dbeaver.io/download/> dan install

## AKTIVITAS 1: QUERY POSTGRESQL MENGGUNAKAN SQL

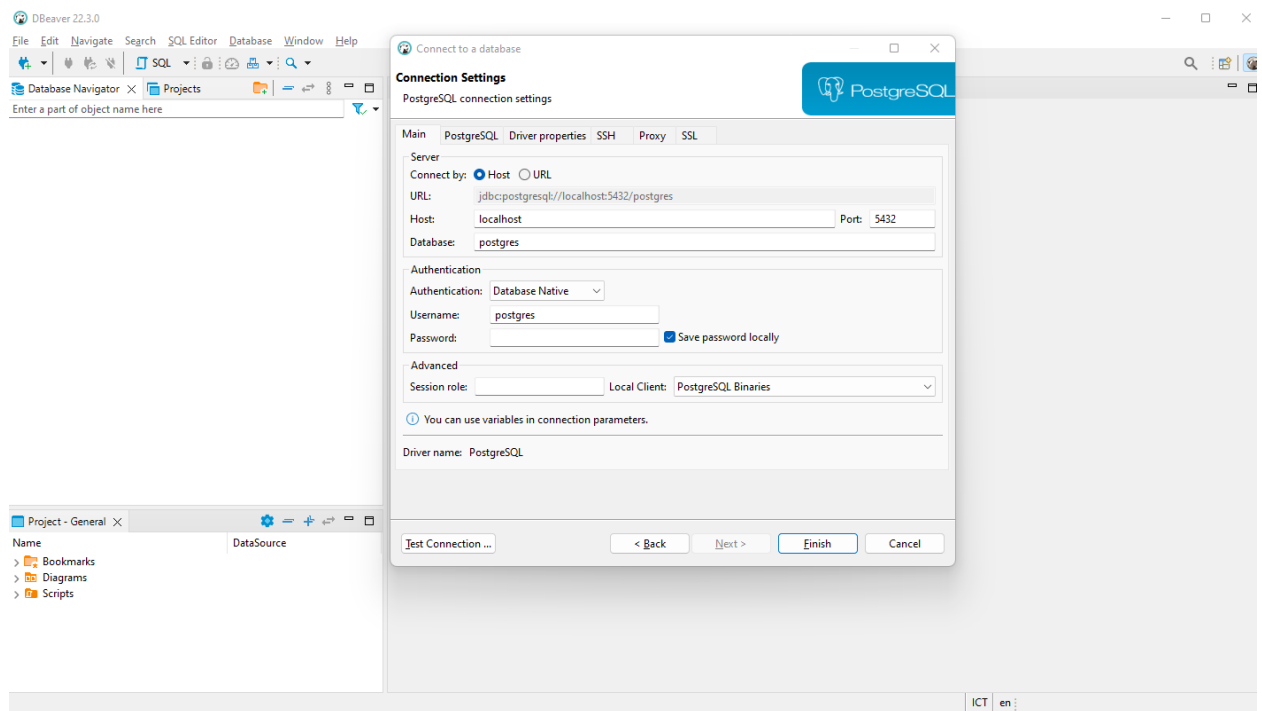
1. Buka aplikasi DBeaver dan buat koneksi baru ke database
2. Koneksi dibuat dengan spesifikasi sebagai berikut:
  - a. Host: 103.23.198.225
  - a. Port: 5432
  - b. Username: csup
  - c. Password: Password1!
  - d. Selain 4 kolom isian di atas, dibiarkan saja dengan isian bawaan



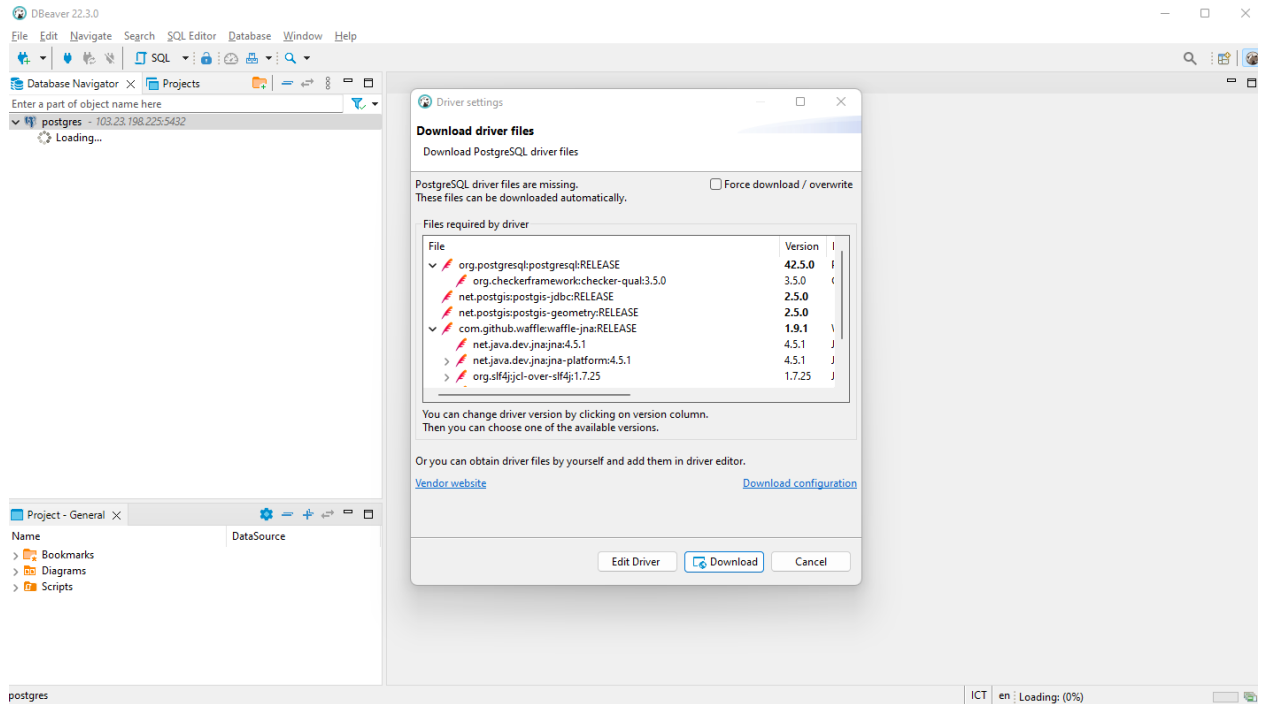
Gambar 1.1: Pilih 'Database' lalu pilih 'New Database Connection'



Gambar 1.2: Pilih database 'PostgreSQL' lalu klik 'Next'

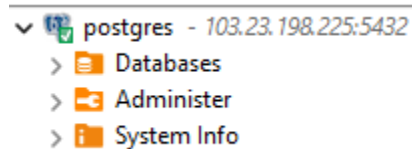


Gambar 1.3: Isi Host, Port, Username dan Password lalu klik 'Finish'



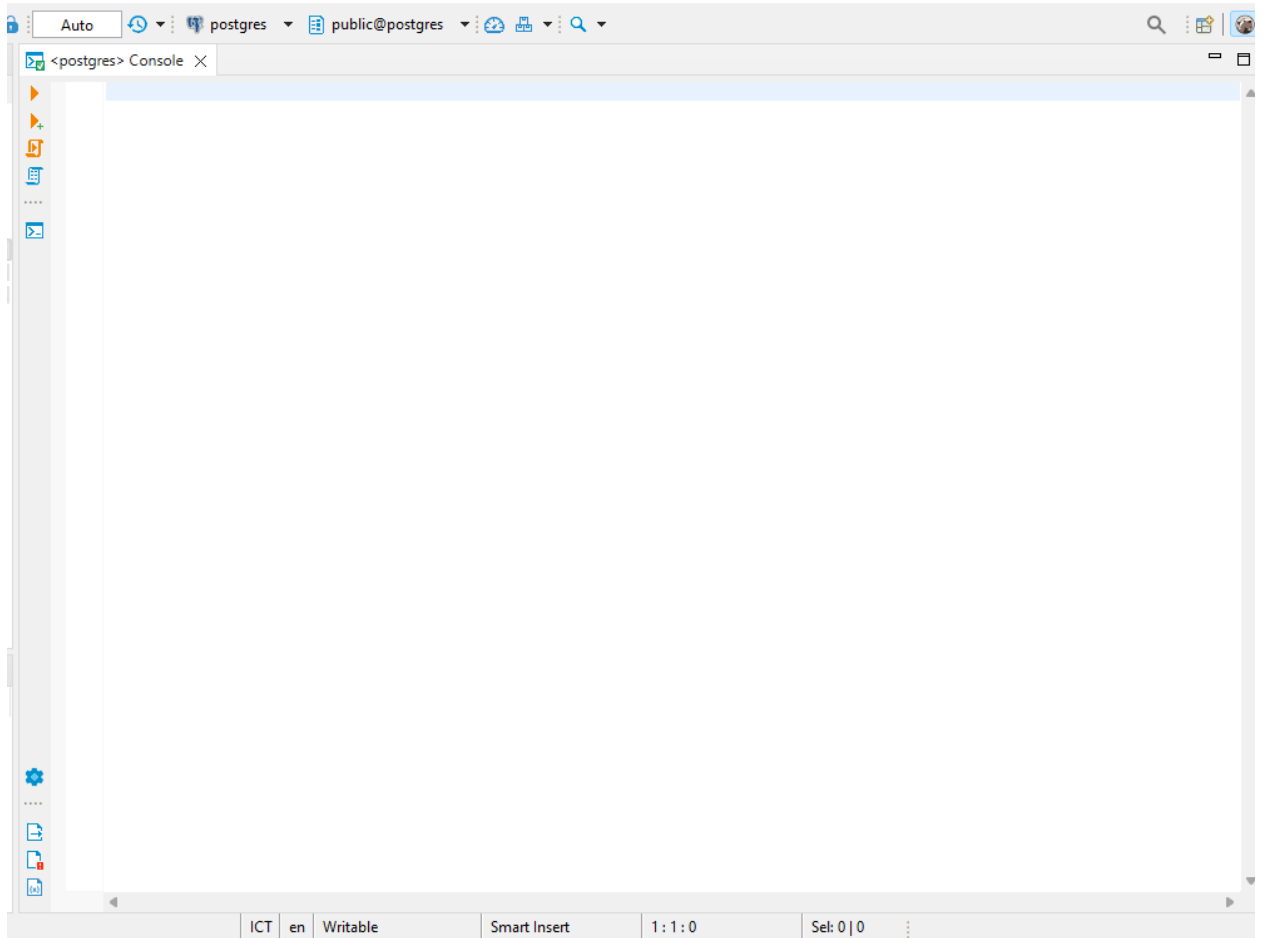
Gambar 1.4: Download driver jika keluar tampilan seperti ini

### 3. Pastikan anda telah terhubung dengan server database



Gambar 1.5: DBeaver telah terhubung dengan server database

### 4. Pada panel atas, klik 'SQL Editor' di menu dropdown klik 'Open SQL console', langkah 5 dan seterusnya akan dilakukan di SQL console yang dibuka di Langkah ini



Gambar 1.6: SQL console

5. Buatlah tabel dengan nama anda, sebagai contoh: akhmal. Tabel dibuat dengan format:
  - a. NIK
  - b. Nama
  - c. Alamat
  - d. No\_telp
  - e. NIK sebagai primary key

```
CREATE TABLE public.akhmal (  
    nik varchar NOT NULL,  
    nama varchar NOT NULL,  
    alamat varchar NOT NULL,  
    no_telp varchar NOT NULL,  
    CONSTRAINT akhmalpk PRIMARY KEY (nik)  
);
```

6. Lakukan operasi 'insert' ke dalam tabel yang dibuat pada Langkah 5

```
INSERT INTO public.akhmal (nik, nama, alamat, no_telp)
VALUES ("1234321", "akhmal", "jalan Sinabung 2",
"08898765432")
```

7. Lakukan 2-3 kali operasi 'insert' dengan perubahan data di atas dan dengan command di atas
8. Lakukan operasi 'update' pada salah satu row di tabel yang dibuat pada Langkah 5

```
UPDATE akhmal SET nama = "akhmalr" WHERE nik = 1234567
```

9. Lakukan operasi 'delete' pada salah satu row di tabel yang dibuat pada Langkah 5

```
DELETE FROM akhmal WHERE nik = "55665432"
```

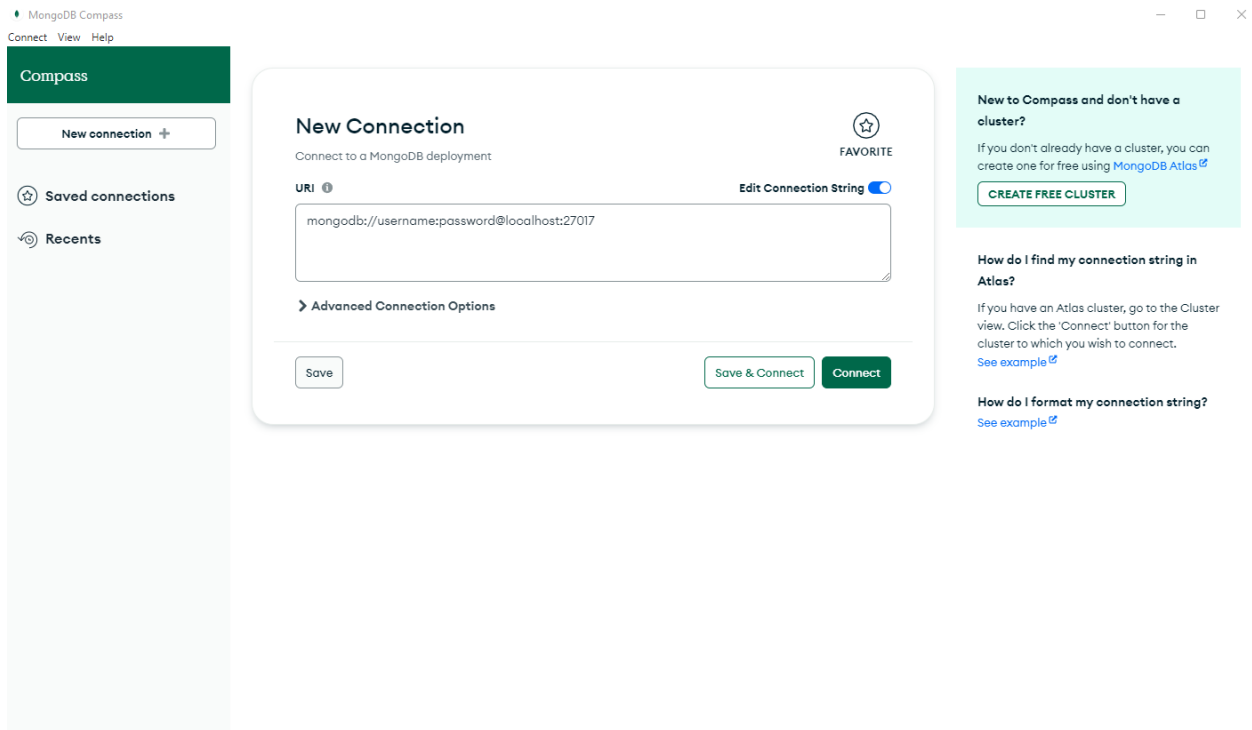
10. Lakukan operasi 'select' pada salah satu row di tabel yang dibuat pada Langkah 5

```
SELECT * FROM akhmal WHERE nik = "123454321"
```

## AKTIVITAS 2: QUERY MONGO SHELL

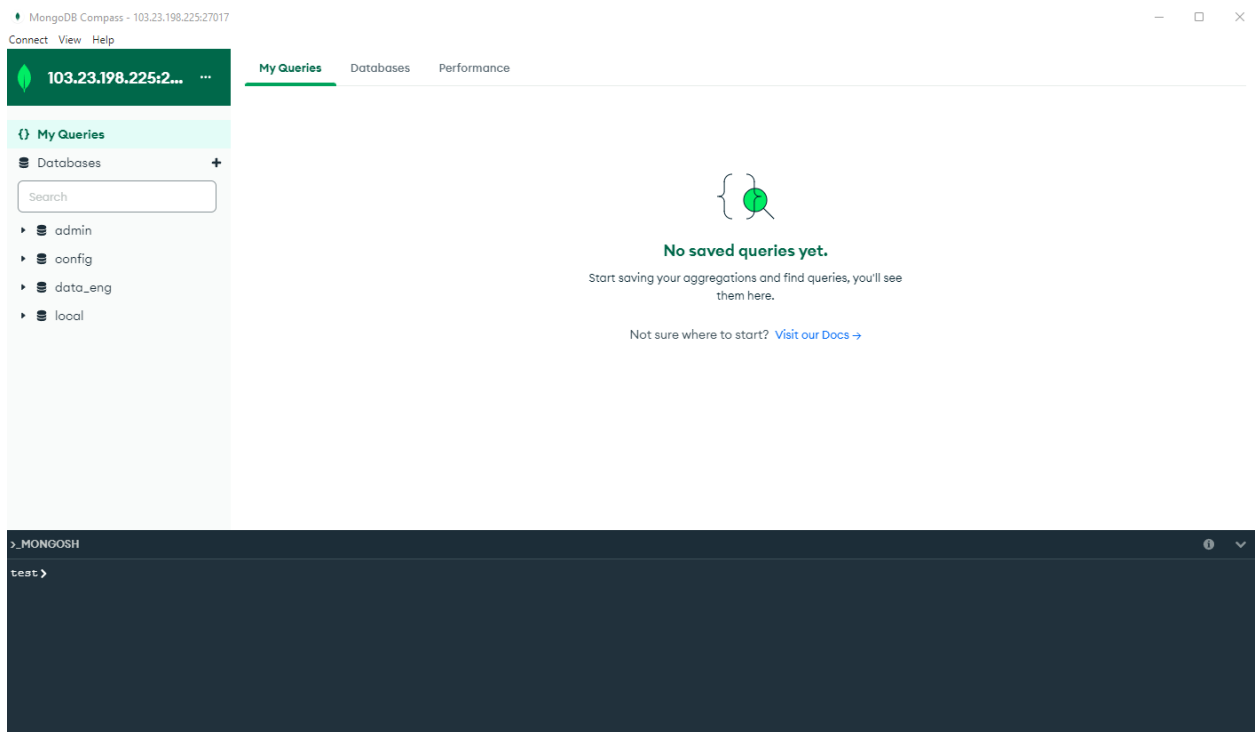
2. Buka aplikasi MongoDB Compass dan hubungkan dengan server menggunakan URI
3. Koneksi dibuat dengan spesifikasi sebagai berikut:
  - a. Host: 103.23.198.225
  - b. Port: 27017
  - c. Username: csup
  - d. Password: Password1!
4. Buatlah URI dari spesifikasi koneksi di nomor 2 dengan pola:

```
mongodb://username:password@host:port
```



Gambar 2.1: klik 'connect' setelah mengisi URI

5. Buka MongoSH di panel paling bawah, langkah 6 dan seterusnya akan dilakukan di MongoSH yang dibuka di Langkah ini



Gambar 2.2: MongoSH

6. Ketik command untuk memakai database 'data\_eng' :

```
use data_eng
```

7. Buatlah collection dengan nama anda, nama sebagai contoh: akhmal

```
db.createCollection('akhmal')
```

8. Lakukan operasi 'insert' ke dalam collection yang telah dibuat dengan format JSON:

```
Data JSON
{
  "nama": "akhmal",
  "kelas": "basis data",
  "alamat": "jalan rawa simprug",
  "no_telp": "088800088808"
}
```

Menggunakan command: (data JSON dapat diubah)

```
db.insertOne({"nama": "akhmal", "kelas": "basis
data", "alamat": "jalan rawa
simprug", "no_telp": "088800088808"})
```

9. Lakukan 2-3 kali insersi dengan mengubah data di atas (misal, alamat diubah)  
10. Lakukan operasi 'update' pada salah satu document di collection yang dibuat pada Langkah 7

```
db.updateOne({"nama": "akhmal"}, {$set: {"alamat": "jalan
Sinabung 2"}})
```

11. Lakukan operasi 'delete' pada salah satu document di collection yang dibuat pada Langkah 7

```
db.deleteOne({"alamat": "jalan tentara pelajar"})
```

12. Lakukan operasi 'find' (sama dengan SELECT di SQL) pada salah satu document di collection yang dibuat pada Langkah 7

```
db.findOne({"no_telp": "08812345678"})
```