**Belajar MongoDB: Pengenalan Dasar MongoDB untuk Pemula**

**# [MongoDB](https://www.petanikode.com/topik/mongodb)** **# [NoSQL](https://www.petanikode.com/topik/nosql)** **#**[**SQL**](https://www.petanikode.com/topik/sql)

Ada beberapa kekurangan dalam database SQL yang pernah saya temukan, seperti skema database yang kaku *(fixed)*, susah membuat query untuk tabel dengan relasi yang kompleks, susah diperbesar sekalanya, dsb.

Database NoSQL *(Not Only SQL)* hadir untuk menutupi kekurangan-kekurangan tersebut. Selain itu, NoSQL sudah menjadi tuntutan teknologi yang harus dipelajari dalam pengembangan software modern masa kini.

Contohnya, sekarang lagi populer yang namanya MEAN stack (MongoDB, ExpressJs, AngularJS, dan NodeJS).

Pada artikel ini, kita akan belajar MongoDB yang merupakan salah satu database NoSQL paling populer.

**Apa itu MongoDB?**

MongoDB adalah salah satu jenis database NoSQL yang berbasis dokumen dengan fomat JSON.

Pada database SQL, data disimpan dalam bentuk tabel. Sedangkan pada MongoDB data disimpan dalam bentuk dokumen dengan format JSON.

Contoh data di MongoDB:

{

"\_id" : ObjectId("54c955492b7c8eb21818bd09"),

"alamat" : {

"street" : "2 Avenue",

"zipcode" : "10075",

"building" : "1480",

"coord" : [ -73.9557413, 40.7720266 ]

},

"borough" : "Manhattan",

"cuisine" : "Italian",

"grades" : [

{

"date" : ISODate("2014-10-01T00:00:00Z"),

"grade" : "A",

"score" : 11

},

{

"date" : ISODate("2014-01-16T00:00:00Z"),

"grade" : "B",

"score" : 17

}

],

"name" : "Vella",

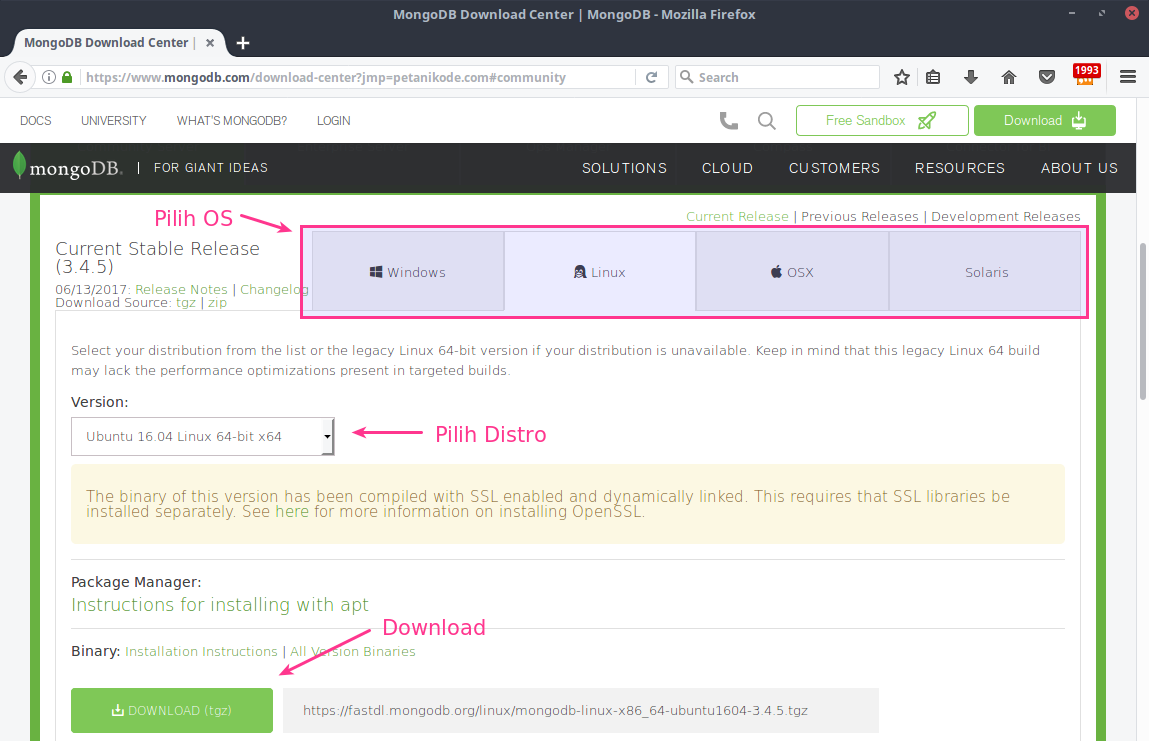
"restaurant\_id" : "41704620"

}

Untuk lebih jelasnya, mari kita langsung praktek…

**Instalasi MongoDB di Linux**

Jika kita ingin menggunakan MongoDB versi terbaru, maka kita harus men-download versi terakhir di website reseminya ([mongodb.com](https://www.mongodb.com/download-center?jmp=petanikode.com)).



Tapi, jika ingin menggunakan versi lama yang ada di repositori, kita bisa menginstalnya dengan perintah:

sudo apt install mongodb

Setelah itu, untuk memastikan sudah terinstal dengan benar, coba ketik perintah mongo --version atau mongod --version untuk memeriksa versinya.



Untuk keperluan belajar dasar, kita bisa pakai yang mana saja. Versi lama atau versi beru tidak masalah.

Pada panduan ini, saya menggunakan versi yang lama.

**Cara Mengoperasikan MongoDB**

Ada beberapa perintah yang disediakan MongoDB untuk mengoperasikannya, seperti perintah mongod *(MongoDB Daemon)* untuk membuat server dan mongo untuk masuk ke *Shell* MongoDB.

**Menghidupkan dan Mematikan Server MongoDB**

Kita bisa menghidupkan dan mematikan server MongoDB dengan perintah service di Ubuntu dan keluarga Debian lainnya.

Menyalakan server MongoDB:

sudo service mongodb start

Mematikan server MongoDB:

sudo service mongodb stop

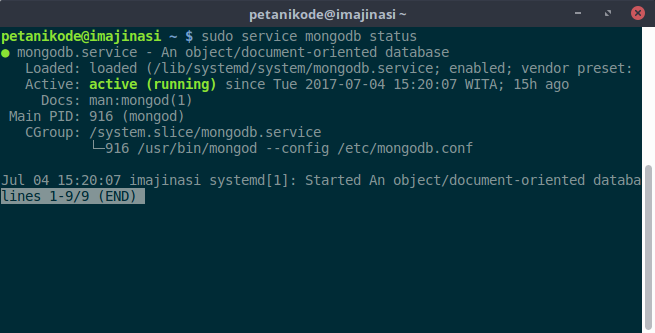
Menyalakan ulang server MongoDB:

sudo service mongodb restart

Melihat status server MongoDB:

sudo service mongodb status

Gunakan tombol q untuk keluar dari status.



Selain itu, kita juga bisa menggunakan perintah mongod dengan syarat harus ditentukan dbpath-nya.

Contoh:

Matikan dulu server yang sudah berjalan:

sudo service mongodb stop

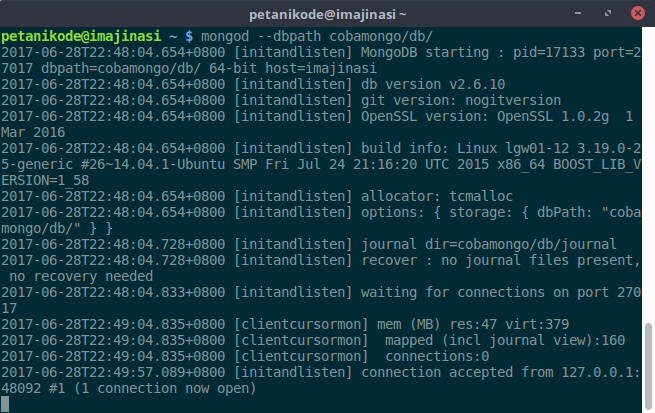
Setelah itu buat direktori baru untuk menyimpan dokumen database-nya.

mkdir -p cobamongo/db

Setelah itu jalankan servernya:

mongod --dbpath cobamongo/db/

Jika muncul seperti ini,



berarti server MongoDB sudah hidup dan siap menerima koneksi dari alamat 127.0.0.1 (localhost).

Untuk mematikannya, tinggal tekan saja tombol Ctrl+c.

Untuk selanjutnya, kita akan menggunakan server yang berjalan secara background. Jadi silahkan nyalakan kembali servernya:

sudo service mongodb start

**Mengakses Server MongoDB**

Kalau tadi kita menggunakan perintah mongod untuk menyalakan server, maka untuk mengakses servernya, kita gunakan perintah mongo.

Perintah mongo adalah program *client* untuk membuka *Shell* MongoDB.

Pastikan servernya hidup, dan cobalah untuk mengetik perintah:

mongo

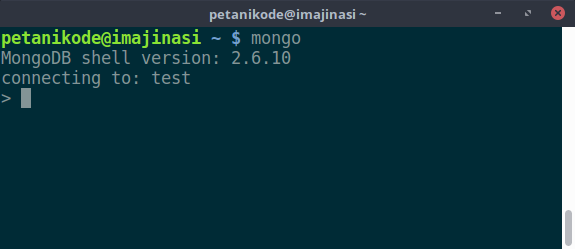
atau

mongo 192.169.1.1

Jika server-nya berada di tempat lain dengan IP 192.169.1.1.

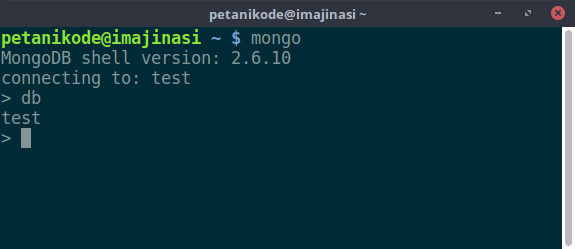
Karena server-nya berada di localhost, jadi kita bisa langsung ketik mongo saja.

Kalau muncul seperti ini, berarti kita berhasil masuk ke *Shell* atau server MongoDB:

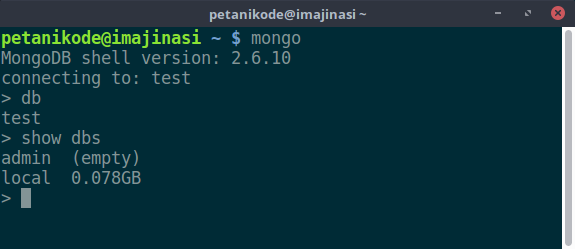


Secara *default*, database yang kita gunakan adalah test.

Untuk melihat database yang sedang aktif, gunakan perintah db pada *Shell* MongoDB.



Sedangkan untuk melihat daftar database, gunakan perintah show dbs.



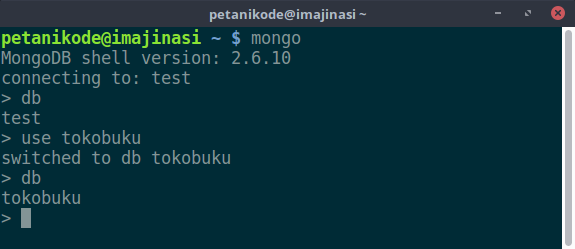
Saat ini, di server hanya ada database admin dan local.

Untuk keluar dari *Shell* MongoDB, tekan Ctrl+c.

**Membuat Database dan Koleksi Baru**

Mari kita buat Database baru bernama tokobuku.

Silahkan ketik perintah use tokobuku untuk membuat dan menggunakan database tokobuku.



Database ini belum tersimpan, karena masih kosong.

Selanjutnya, mari kita coba buat sebuah koleksi *(collection)*. MongoDB menggunakan istilah koleksi untuk menggantikan tabel.

Koleksi berisi kumpulan dokumen atau data dalam format JSON. Kalau di SQL kita menyebutnya dengan record/baris.

Koleksi bisa dibuat dengan perintah:

db.createCollections("nama\_koleksi")

atau bisa juga otomatis dibuat ketika kita melakukan insert data.

**Insert Data**

Insert data dapat kita lakukan dengan perintah berikut:

db.<koleksi>.insert(<data>)

atau

db.<koleksi>.save(<data>)

Keterangan:

* <koleksi> adalah nama koleksi yang akan dibuat;
* <data> adalah data yang akan kita simpan dalam koleksi tersebut dengan format JSON.

Sebagai contoh, kita akan membuat koleksi bernama buku:

db.buku.insert({

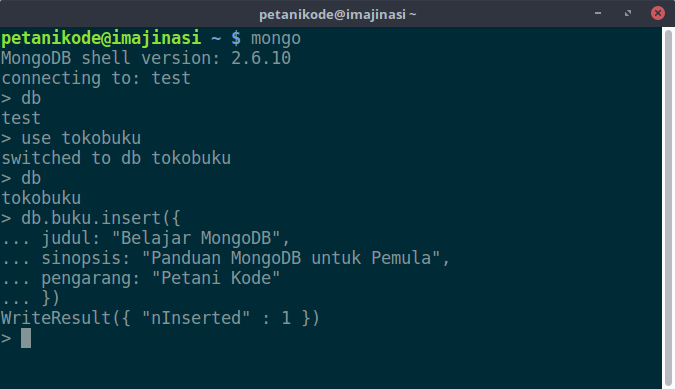
judul: "Belajar MongoDB",

sinopsis: "Panduan MongoDB untuk Pemula",

pengarang: "Petani Kode"

})

Maka kita akan memiliki satu koleksi dan satu data di dalamnya.



Coba tambahkan lagi data baru.

db.buku.insert({

judul: "Pemrograman Javascript dan MongoDB",

sinopsis: "Panduan Pemrograman Js dan MongoDB",

pengarang: "Petani Kode",

harga: 98000

})

Maka sekarang kita punya dua data dalam koleksi buku.

Kita bisa melihatnya dengan fungsi count().

Contoh:

> db.buku.count()

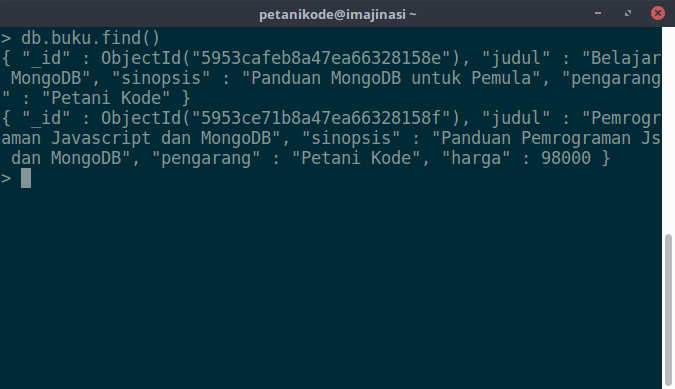
2

**Menampilkan Data**

Untuk menampilkan data yang sudah ditambahkan, kita bisa menggunakan method find():

db.<koleksi>.find()

Contoh:



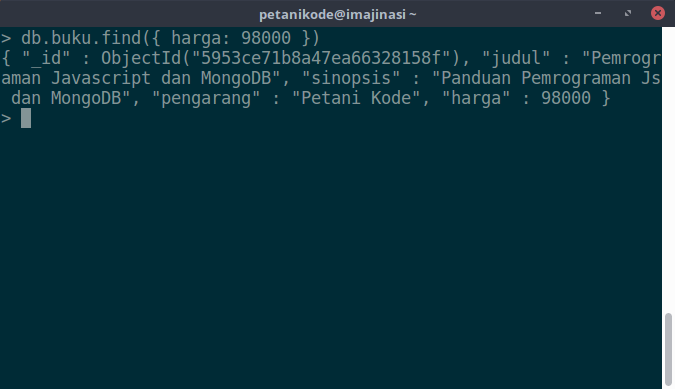
Kita juga bisa menambahkan fungsi pretty() agar ditampilkan dengan rapi.



Coba juga untuk menampilkan buku yang memiliki harga 98000.

db.buku.find({ harga: 98000 })

Maka, buku yang hanya berharga 98000 saja yang akan ditampilkan.



Paham…

Tapi apa itu "\_id" dan ObjectId("5953ce71b8a47ea66328158f"), kenapa selalu ada di setiap data?

Atribut \_id bisa dibilang seperti *Primary Key*. Atribut ini akan otomatis dibuat oleh MongoDB disetiap pengisian data.

**Mengubah Data**

Untuk mengubah data, kita bisa menggunakan fungsi update().

db.<koleksi>.update(<query>, <data baru>)

Keterangan:

* <query> adalah kata kunci data yang akan kita ubah, sama seperti WHERE dalam SQL.
* <data baru> adalah data barunya.

Sekarang, mari kita coba mengubah harga bukunya dari 98000 menjadi 75000.

db.buku.update(

{

judul: "Pemrograman Javascript dan MongoDB"

},

{

judul: "Pemrograman Javascript dan MongoDB",

sinopsis: "Panduan Pemrograman Js dan MongoDB",

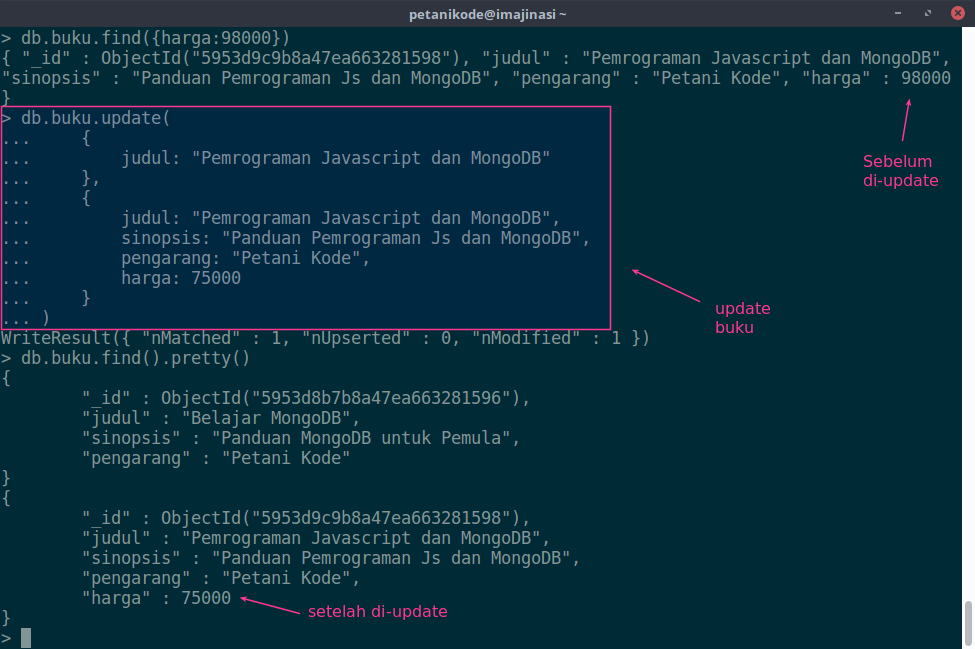
pengarang: "Petani Kode",

harga: 75000

}

)

Maka hasilnya, semua buku yang berjudul "Pemrograman Javascript dan MongoDB" akan diturunkan harganya menjadi 75000.



**Menghapus Data**

Untuk menghapus data, kita bisa menggunakan perintah remove().

db.<koleksi>.remove(<query>)

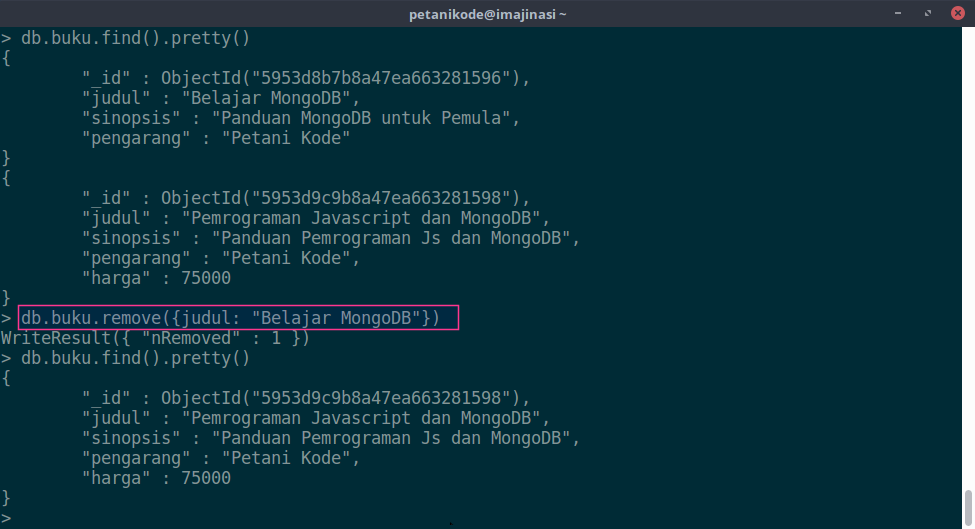
Sama seperti update data, penghapusan data juga memiliki <query> untuk menentukan data yang akan dihapus.

Contoh:

Kita akan menghapus buku yang berjudul "Belajar MongoDB", karena stoknya sudah habis .

db.buku.remove({judul: "Belajar MongoDB"})

Maka data buku yang berjudul "Belajar MongoDB" sudah tiada.



Untuk menghapus semua data dalam koleksi, kita bisa kosongkan nilai query-nya.

db.<koleksi>.remove({})

**Menhapus Dabase dan Koleksi**

Bila kita ingin menghapus database dan koleksi, kita bisa menggunakan method drop() dan dropDatabase().

Menghapus koleksi:

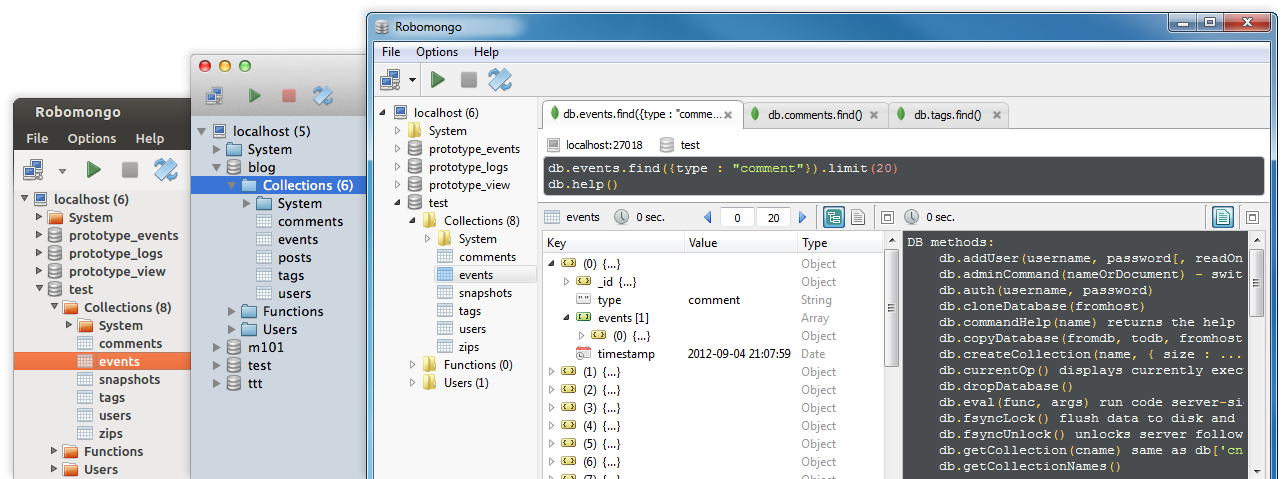
db.<koleksi>.drop();

Menghapus database:

db.dropDatabase();

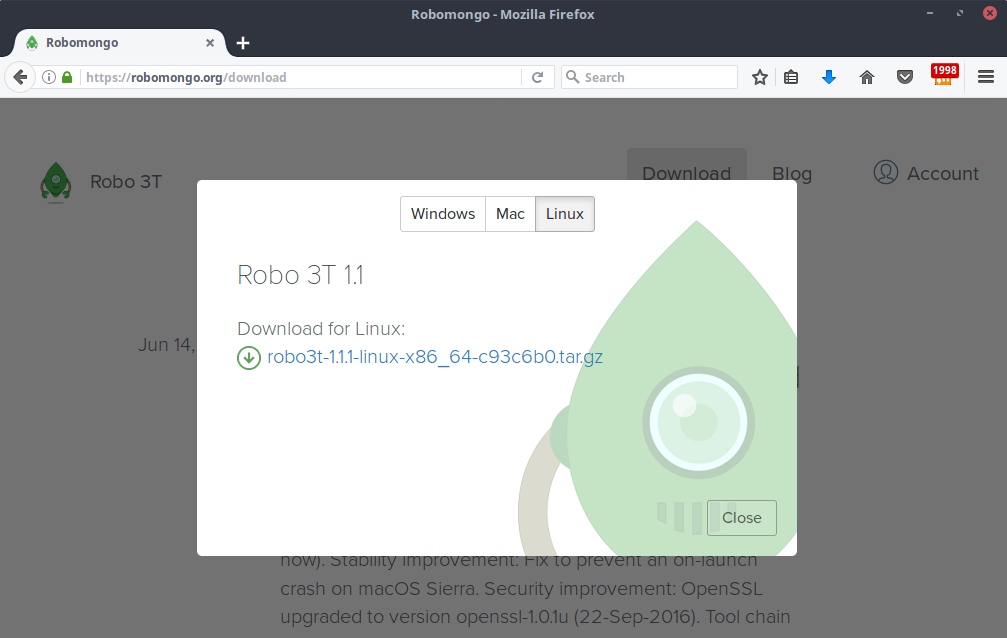
**Menggunakan Robo 3T**

Robo 3T (sebelumnya bernama Robomongo) adalah salah satu *tools* berbasis GUI untuk memanajemen database MongoDB. *Tools* ini *opensource* dengan lisensi GPL-3.0 dan tersedia untuk semua jenis sistem operasi.

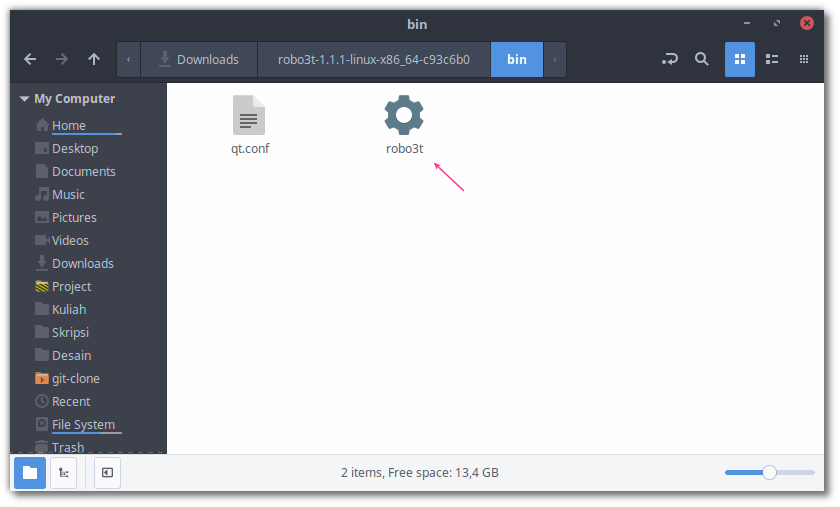


**Instalasi dan Menggunakan Robo 3T**

Pertama silahkan download Robo 3T di website resminya ([robomongo.org](https://robomongo.org/download)).



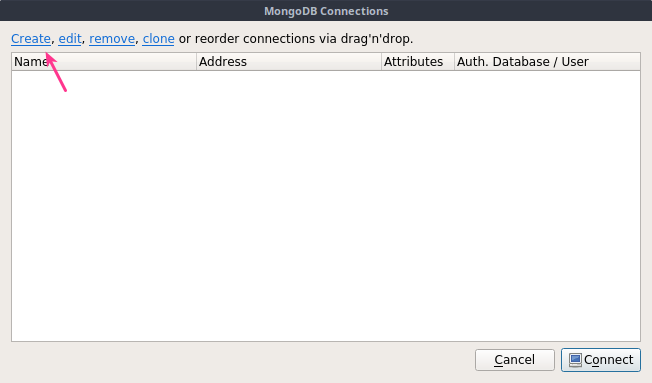
Setelah itu, ekstrak dan coba jalankan file bin/robo3t.



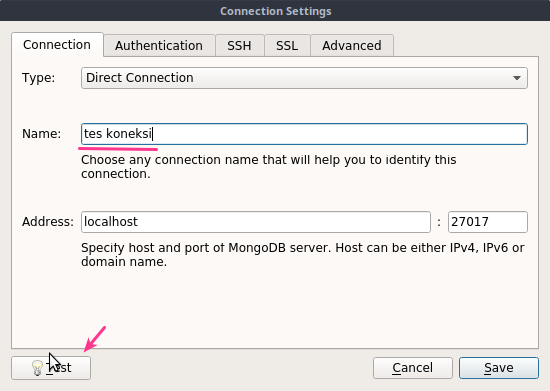
Maka jendela EULA akan muncul, silahkan klik ***“I Agree”*** kemudian klik ***Finish***.



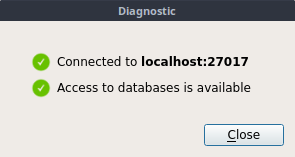
Maka jendela baru *MongoDB Connection* akan terbuka. Silahkan buat koneksi, klik *Create*.



Silahkan isi nama koneksinya dengan apa saja yang kamu sukai, kemudian klik ***Test***.

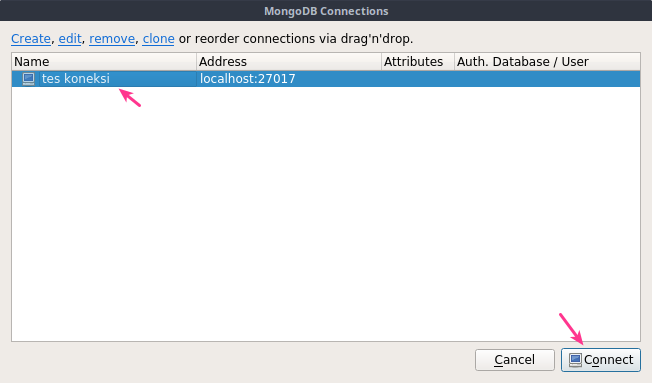


Kalau muncul jendela seperti ini, berarti koneksi berhasil.

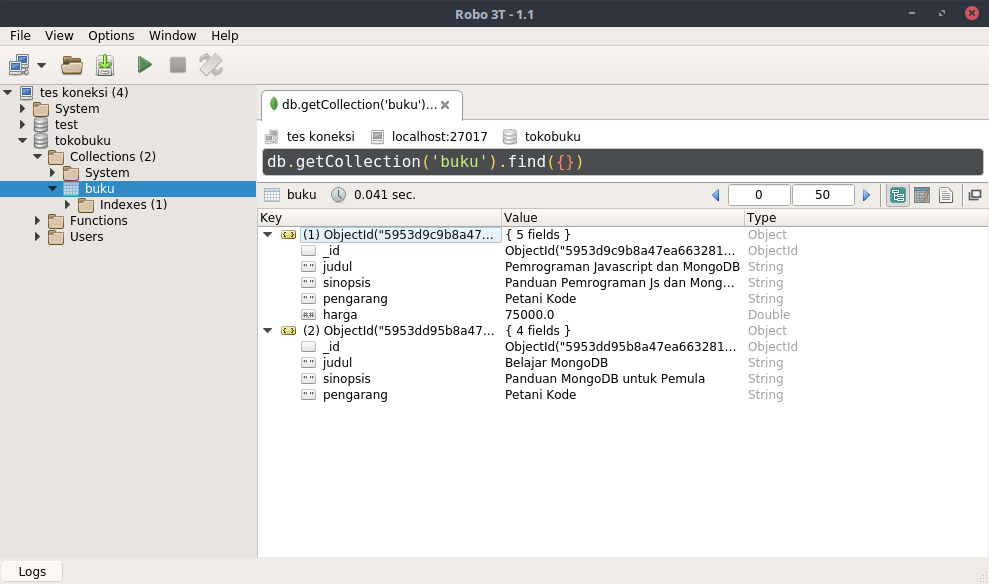


Setelah itu, silahkan klik ***Save*** untuk menyimpan koneksi. Maka sekarang kita punya satu koneksi.

Silahkan klik ***Connect*** untuk mulai membuka database MongoDB dari Robo 3T.



Berhasil…



Sekarang kita bisa mengelola database MongoDB tanpa harus menggunakan *Shell* di Terminal.

**Membuat Menu Robo 3T**

Robo 3T yang kita jalankan tadi adalah *portable*. Untuk mengintegrasikan dengan sistem, silahkan taruh direktori Robo 3T di dalam direktori /opt, lalu buatkan *shortcut* pada menu.

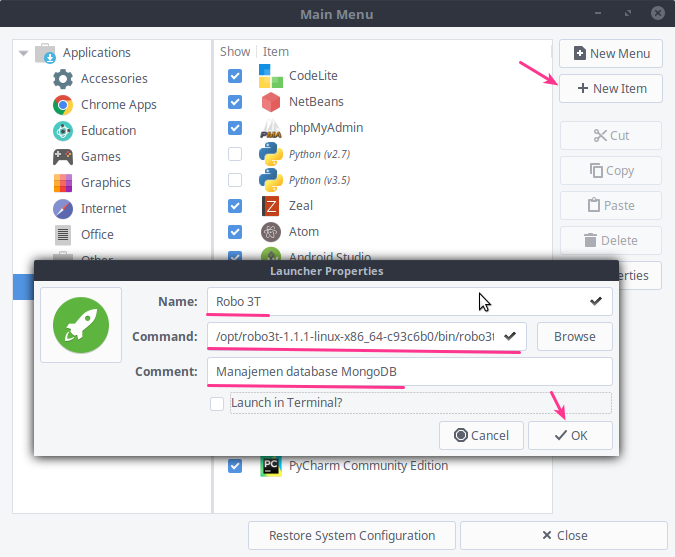
Gunakan perintah ini untuk memindahkannya:

sudo mv ~/Downloads/robo3t-1.1.1-linux-x86\_64-c93c6b0/ /opt

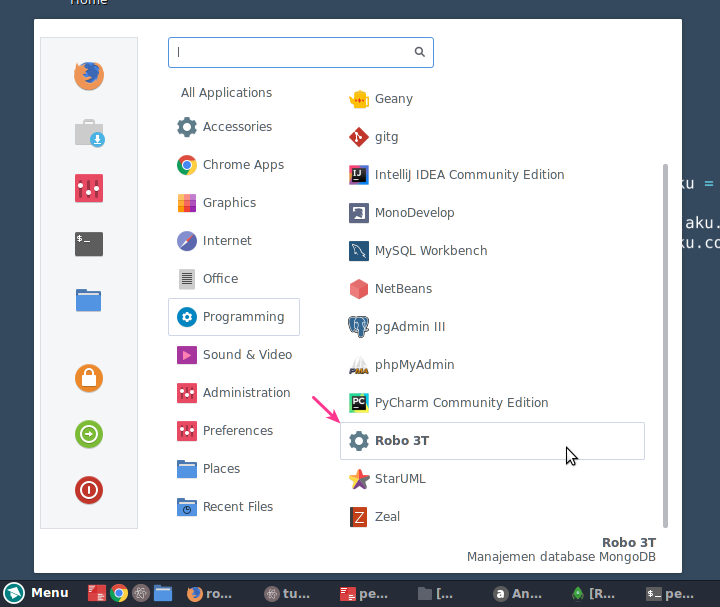
atau

sudo cp ~/Downloads/robo3t-1.1.1-linux-x86\_64-c93c6b0/ /opt

Lalu tambahkan menu launcher di Menu Editor.



Maka sekarang, kita sudah punya menu Robo 3T yang bisa kita buka kapan saja.



**Apa Selanjutnya?**

Kita sudah mengetahui dasar-dasar MongoDB dan sudah mencoba dua cara dalam mengelola database MongoDB, yaitu menggunakan *Shell* di terminal dan Robo 3T.

Menggunakan Robo 3T memang lebih mudah, tinggal klik-klik saja. Tapi pengetahuan tentang *Shell* juga tidak kalah pentingnya, karena akan banyak berguna dalam pemrograman.

Selanjutnya mungkin bisa mencoba ide eksperimen berikut:

* Menghubungkan Python dengan MongoDB dan Membuat aplikasi CRUD;
* Menghubungkan PHP dengan MongoDB dan Membuat aplikasi CRUD;
* Menghubungkan Javascript dengan MongoDB dan Membuat aplikasi CRUD;
* Menghubungkan Ruby dengan MongoDB dan Membuat aplikasi CRUD.

Terima kasih sudah mengikuti panduan ini sampai akhir, saya harap semuanya bisa dipahami.

Ada pertanyaan?

Sampaikan di kolom komentar!

Referensi:

* Dok. MongoDB — [Getting Started With MongoDB](https://docs.mongodb.com/getting-started/shell/)