WEB PROGRAMMING

MODUL PRAKTIKUM

DAFTAR ISI

[11 Database 2](#_Toc56520056)

[11.1 Introduction 2](#_Toc56520057)

[11.1.1 Instalasi MongoDB 2](#_Toc56520058)

[11.1.2 Database Administration 4](#_Toc56520059)

[11.2 Node.js dan MongoDB 6](#_Toc56520060)

[11.2.1 Find() 7](#_Toc56520061)

[11.2.2 InsertOne() 8](#_Toc56520062)

[11.3 Mongoose Object-Document Mapper (ODM) 9](#_Toc56520063)

[11.3.1 Schema 10](#_Toc56520064)

[11.3.2 Schema & Model Module 11](#_Toc56520065)

[11.3.3 Query 12](#_Toc56520066)

[11.4 CRUD 13](#_Toc56520067)

[11.4.1 Create 13](#_Toc56520068)

[11.4.2 Read 14](#_Toc56520069)

[11.4.3 Update 15](#_Toc56520070)

[11.4.4 Delete 16](#_Toc56520071)

# Database

## Introduction

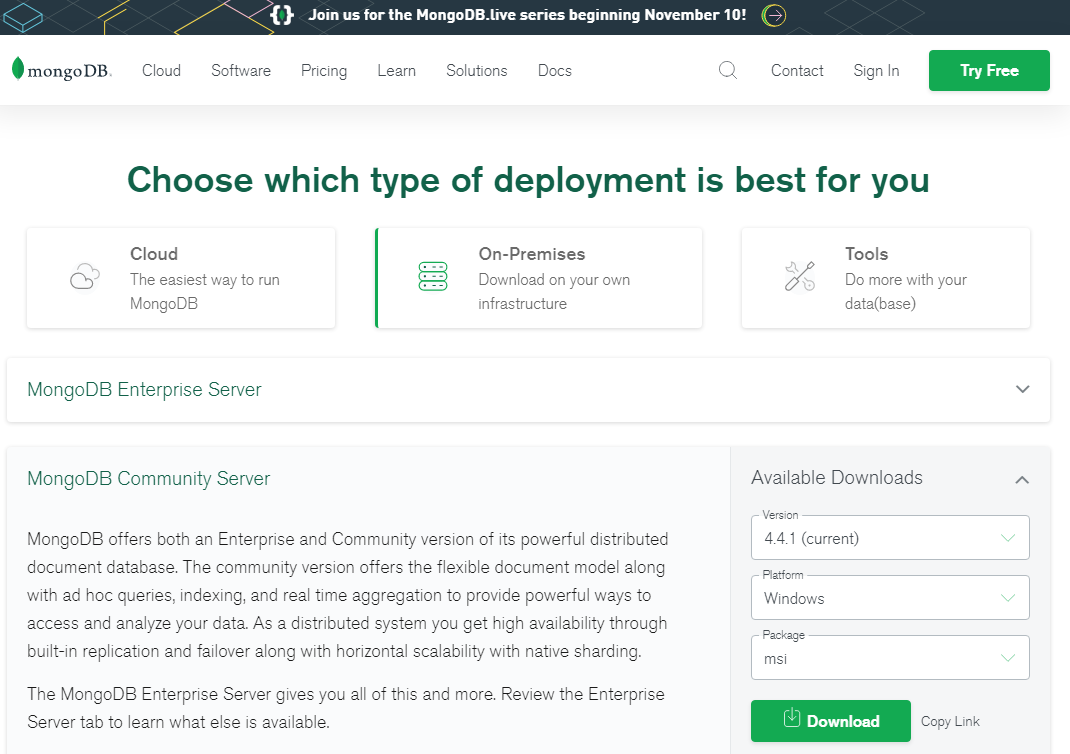
Menyimpan data bisa dibilang bagian terpenting dari pengembangan aplikasi. Tanpa penyimpanan jangka panjang, cara Anda berinteraksi dengan pengguna Anda akan sulit. Data di setiap aplikasi yang Anda buat hingga saat ini menghilang setiap kali Anda memulai ulang aplikasi. Jika data dari jejaring sosial menghilang setiap kali pengguna menutup browsernya atau setiap kali Anda memulai ulang aplikasi itu, pengguna harus membuat akun baru dan memulai dari awal.

Database adalah organisasi data Anda yang dirancang untuk akses mudah dan perubahan efisien yang dibuat oleh aplikasi Anda. Basis data seperti gudang: semakin banyak item yang perlu Anda simpan, semakin senang Anda dengan sistem terorganisir yang membantu Anda menemukan item tersebut. Seperti server web, aplikasi Anda terhubung ke database MongoDB dan meminta data.

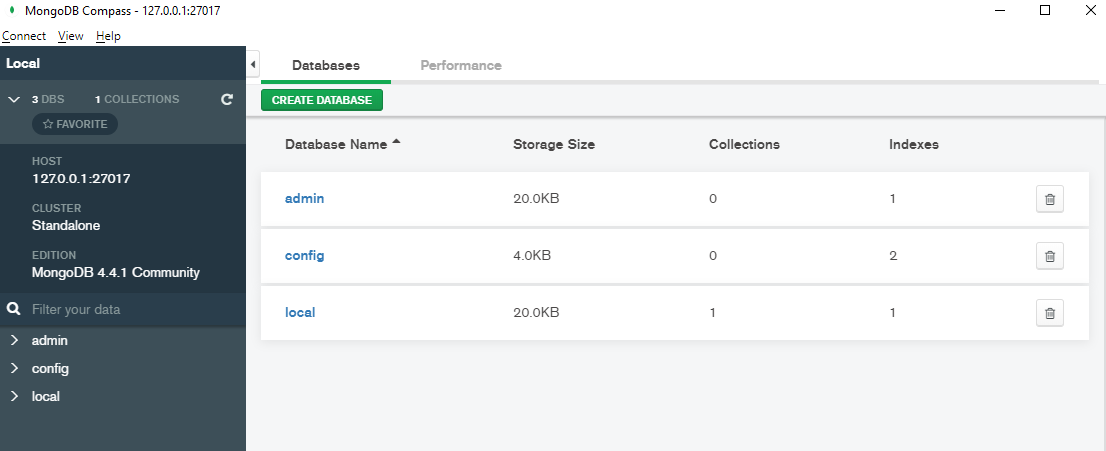
MongoDB adalah program database open-source yang mengatur data dengan menggunakan dokumen. Dokumen MongoDB menyimpan data dalam struktur seperti JSON, memungkinkan Anda menggunakan key-value pairing untuk mengaitkan objek data dengan properti.

### Instalasi MongoDB

Unduh paket instalasi MongoDB dari URL ini: <https://www.mongodb.com/try#community> Pilih On-Premise, dikarenakan kita akan melakukan instalasi di komputer masing-masing.

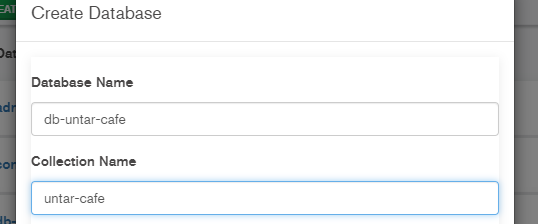


Jalankan instalasi dengan user Administrator, lalu pilihlah fitur Complete. Centang pilihan Install MongoDB as a service. Apabila sebelumnya memilih fitur Complete, maka seharusnya MongoDB Compass akan muncul. Aplikasi tersebut akan meminta Connection String. Masukkan connection string berikut lalu klik Connect : mongodb://127.0.0.1:27017. Tampilan di bawah seharusnya muncul setelah menjalankan instalasi dan memasukkan conenction string.

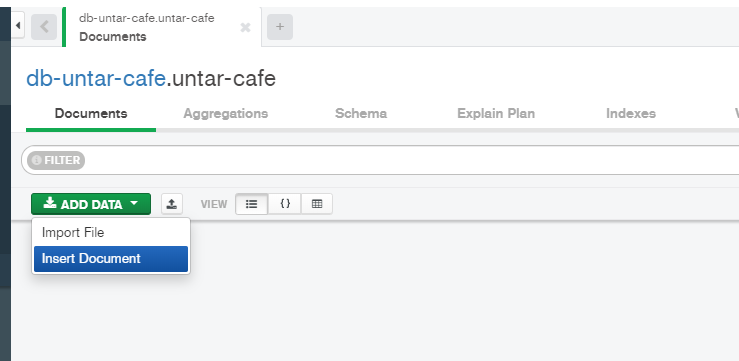


### Database Administration

Untuk memulai menggunakan MongoDB, klik Create Database. Masukkan nama database dan collection, contoh seperti di bawah. Collection digunakan untuk menyimpan data. Dalam sebuah database dapat terdiri dari beberapa Collection. Apabila memiliki beberapa kumpulan data yang dapat dikategorikan dalam beberapa kumpulan, maka buatlah Collection sebanyak kumpulan tersebut.



Untuk mulai memasukkan data menggunakan MongoDB Compass, klik pada salah satu Collection, lalu klik Add Data. Terdapat 2 metode yaitu Import File dan Insert Document. Kali ini kita menggunakan fitur Insert Document.



MongoDB menggunakan JSON untuk melakukan komunikasi data, sehingga saat menambahkan data, gunakankanlah notasi JSON. Lihat contoh untuk menambah data berikut ini:



## Node.js dan MongoDB

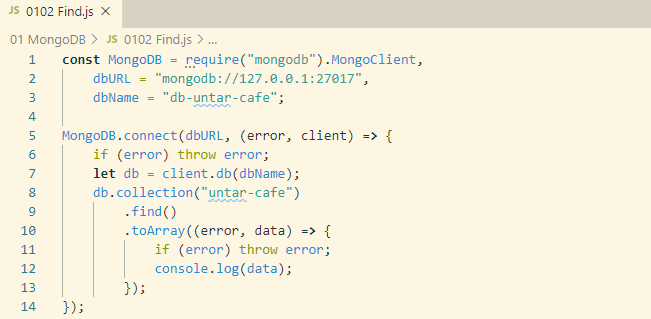
Untuk mulai menggunakan MongoDB sebagai layanan basis data, lakukan instalasi module mongodb di Terminal Visual Studio Code menggunakan script berikut: npm install mongodb. Untuk memastikan modul telah terinstall, cek di folder node\_modules dan package-lock.json.



Apabila pesan Connected! Muncul, maka artinya sudah sukses tersambung ke MongoDB.

### Find()

Gunakan perintah Find() untuk mencari data. Line 8 adalah perintah untuk mengakses Collection, pastikan memasukkan Collection yang terdaftar di database. Apabila dijalankan di terminal, maka akan menghasilkan output yang berisi data yang sudah dimasukkan sebelumnya.



### InsertOne()

Gunakan perintah InsertOne() untuk menambahkan sebuad data ke dalam Collection. Gunakan notasi JSON untuk menyiapkan data, dan pastikan telah menggunakan Collection yang benar.



Untuk melihat apakah data berhasil terseimpan, jalankan kembali fungsi Find sebelumnya, atau klik Refresh di panel sebelah kiri MongoDB Compass.

## Mongoose Object-Document Mapper (ODM)

Setelah mencoba Express.js dan melihat bagaimana module tersebut membantu menangani Request dan Response HTTP, paket lain tersedia untuk membantu komunikasi antara aplikasi Node.js dan database-nya.

Module yang akan Anda gunakan disebut Mongoose. Mongoose adalah Object-Document Mapper (ODM) yang bertugas untuk menjalankan perintah MongoDB dengan mempertahankan struktur berorientasi objek aplikasi Anda.

Apabila hanya menggunakan MongoDB saja akan sulit untuk menjaga konsistensi satu dokumen yang disimpan dengan yang berikutnya. Mongoose mengubah membangun model dengan skema yang menentukan jenis data apa yang dapat disimpan. Model adalah kelas untuk objek JavaScript yang digunakan Mongoose untuk mengatur query database.

Untuk mulai menggunakan Mongoose, lakukan instalasi menggunakan npm di terminal dengan script sebagai berikut: npm install mongoose. Dengan Mongoose, Anda tidak perlu lagi memerlukan mongodb di bagian atas main.js atau menggunakan salah satu kode MongoDB dari bab sebelumnya.

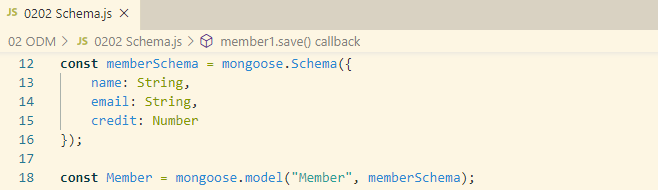
Siapkan connection string ke database MongoDB. (Aturan yang sama berlaku di sini seperti pada koneksi MongoDB seperti sebelumnya.) Kemudian tetapkan koneksi database ke variabel db, yang dapat Anda gunakan nanti di file untuk perubahan data atau perubahan status database.



### Schema

Skema adalah seperti definisi kelas, atau cetak biru tentang bagaimana Anda ingin data diatur untuk objek tertentu dalam aplikasi. Untuk menghindari data yang tidak konsisten, di mana beberapa dokumen memiliki field email dan yang lainnya tidak, misalnya, Anda dapat membuat skema yang menyatakan bahwa semua objek kontak harus memiliki bidang email untuk disimpan ke database.

Karena Anda ingin menyimpan member untuk café, maka perlu dibuat skema untuk pelanggan member. Schema dapat memiliki konstruktor yang digunakan untuk membuat objek skema dengan parameter yang diberikan. Kemudian tambahkan properti objek untuk menyatakan nama field objek dan tipe datanya. Misalnya, nama seseorang tidak boleh angka. Langkah berikutnya adalah membuat konstanta Member dengan memanggil fungsi model() dari Mongoose.



Model yang sudah dibentuk ini dapat digunakan untuk membuat data berdasarkan Schema yang sudah dibuat, kemudian disimpan ke database. Terdapat 2 cara untuk membuat object. Pertama dengan cara membuat object terlebih dahulu, lalu lakukan penyimpanan. Cara kedua, proses penyimpanan dilakukan bersamaan saat membuat object.



### Schema & Model Module

Sekarang setelah memiliki cara untuk menyimpan data dalam bentuk model Mongoose, sebaiknya model dipisahkan sehingga tidak tercampur di file utama. Seperti memisahkan tampilan dan kontrol, buatlah folder model di tingkat root aplikasi. File ini adalah tempat untuk menuliskan source code model. Pindahkan semua skema dan kode definisi model ke masing-masing file.

Perlu ditambahkan kode module.exports = mongoose.model("Member", memberSchema); supaya module dapat diakses oleh file lain. Sedangkan di file penerima gunakan kode require seperti biasa, yaitu const Member = require("./models/Member");.



### Query

Gunakan fungsi findOne() untuk melakukan pencarian terhadap sebuah data. Bisa menggunakan pencarian kata secara penuh, atau dapat juga menggunakan metode RegEx. Buka tautan <https://mongoosejs.com/docs/queries.html> untuk mempelajari fungsi-fungsi pencarian lainnya yang dapat digunakan.



## CRUD

### Create

Untuk memasukkan record, atau dokumen ke dalam koleksi, gunakan metode save(). Metode Save() membutuhkan 2 parameter yaitu Error, dan Product. Error digunakan untuk menerima pesan error dari proses save, sedangkan Product digunakan untuk menerima data hasil proses save.

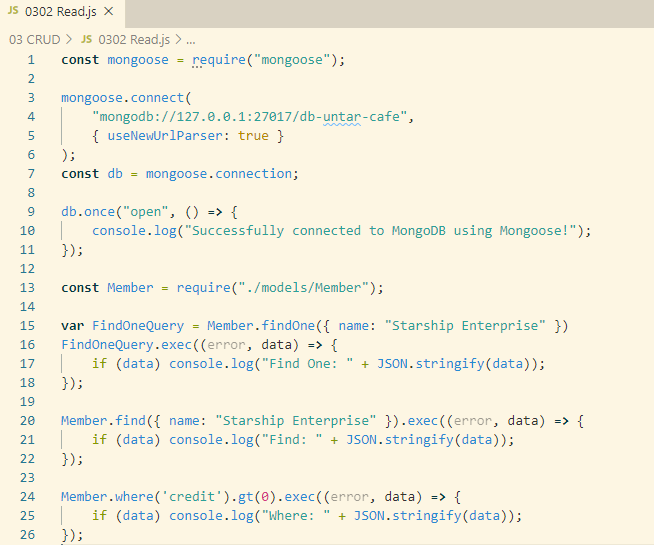


Line 22 menampilkan pesan error apabila terdapat problem saat menyimpan data. Line 23 menampilkan data dalam notasi JSON ke layar terminal.

### Read

Terdapat beberapa metode untuk melakukan pencarian, yaitu: find(), findOne(), where(). Fitur-fitur pencarian lainnya dapat dilihat di <https://docs.mongodb.com/manual/reference/operator/query>.

FindOne() mengembalikan hanya 1 dari hasil pencarian. Find() mengembalikan seluruh hasil pencarian. Where() membutuhkan parameter nama field, lalu diikuti dengan fungsi filter, misalnya gt utk Greater Than.



Line 15 membuat variable query, lalu dijalankan di line 16. Line 20 dan 24 adalah cara lain untuk menjalankan query, tanpa pembuatan variable terlebih dahulu.

### Update

Mongoose memiliki beberapa fungsi untuk melakukan update data, yaitu update(), updateOne(), dan updateMany(). Update menerima 3 parameter, yaitu filter untuk mencari data; field yang diubah beserta valuenya; dan fungsi callback yang dijalankan setelah selesai menjalankan update, seperti pada fungsi save().



Line 15 menjalankan fungsi update. Line 17 adalah action yang dijalankan setelah fungsi update, dalam kasus ini yaitu menampilkan data menggunakan fungsi find.

### Delete

Terdapat 2 fungsi untuk melakukan penghapusan data, yaitu deleteOne() dan deleteMany(). Fungsi ini membutuhkan 2 parameter, yaitu filter untuk pencarian data; dan fungsi callback yang digunakan untuk menjalankan perintah setelah fungsi delete selesai dijalankan.



Line 15 menghapus data menggunakan filter yang diberikan. Line 17 adalah perintah yang dijalankan setelah fungsi delete dijalankan. Pada kasus ini, dilakukan pencarian dengan filter yang sama untuk menghapus. Apabila benar, maka tidak ada data yang ditampilkan.