

MODUL 3

NOSQL & MONGODB

3.1. Tujuan Praktikum

1. Mengetahui perbedaan SQL dan NoSQL
2. Mampu menginstall MongoDB
3. Memahami struktur NoSQL

3.2. Dasar Teori

3.2.1. Mengenal NoSQL

NoSQL memiliki arti Not Only SQL. RDBMS (SQL) pada umumnya menggunakan query sql untuk menyimpan dan melakukan pengambilan data. Akan tetapi, NoSQL dapat menyimpan data dalam berbagai bentuk, baik terstruktur, semi terstruktur, maupun yang tidak terstruktur. Basis data NoSQL termasuk pada Non-Relational Data Management System, tidak adanya relasi langsung antara datanya, tidak memerlukan schema yang tetap, dan lebih mudah untuk melakukan *scaling* pada sistem.

Beberapa perbedaan SQL dan NoSQL diantaranya:

- SQL adalah basis data relasional sedangkan NoSQL tidak relational.
- SQL menggunakan query yang terstruktur dan memiliki schema yang disiapkan terlebih dahulu sedangkan NoSQL tidak memiliki schema tertentu sehingga lebih dinamis dalam bentuk data.
- Pengembangan SQL biasanya secara vertikal, sedangkan NoSQL lebih ke arah horizontal.
- SQL berbasis pada tabel, sedangkan NoSQL bisa memiliki banyak bentuk misalnya pasangan key-value, dokumen, ataupun graph.
- SQL lebih baik untuk bertransaksi dengan banyak baris, sedangkan NoSQL lebih baik untuk transaksi tak berstruktur seperti JSON.

Salah satu alasan kenapa NoSQL menarik ada pada poin 3, yaitu *scalability* atau kemampuan untuk dikembangkan.

Bila dilakukan pemisalan sebagai sebuah bangunan hotel, pengembangan vertikal artinya pengembangan dengan melakukan penambahan tinggi dan besar dari bangunan hotel tersebut. Lebih mudah dari sisi pengurusan dan manajemen akan tetapi membutuhkan biaya dan teknik yang tinggi semakin besar dan tinggi pembangunannya.



Gambar 12. Vertical dan Horizontal Scaling

Pengembangan Horizontal merupakan pengembangan dengan menambah jumlah Hotel. Lebih mudah dari sisi pengembangan akan tetapi terkait sisi manajemen hotel akan lebih sulit semakin banyak hotelnya.

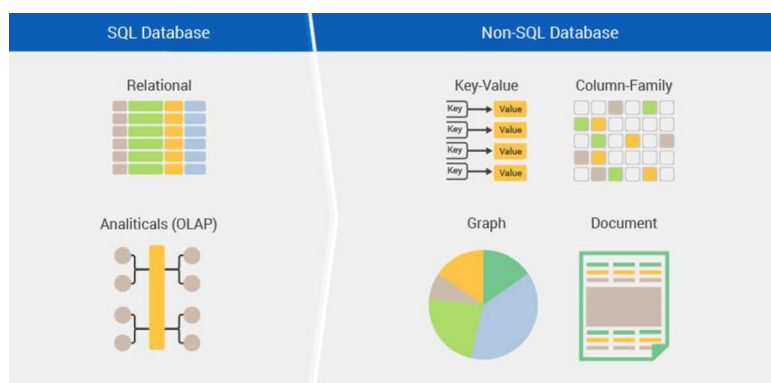
Pengembangan satu server SQL dilakukan dengan menambah kemampuan dasar servernya (dalam hal hotel adalah pondasinya), seperti RAM, CPU, dan SSD. Dan seperti yang kita ketahui, adakan ada batasnya terkait hal tersebut.

Pengembangan NoSQL sederhana dengan menambah jumlah dari servernya, Karena data yang tidak terstruktur dari NoSQL lebih mudah dipecah daripada data berschema dan berelasi seperti SQL.

3.2.2. Tipe-tipe NoSQL

Secara garis besar, basisdata NoSQL dibagi kedalam 4 kategori utama.

- Pasangan Key - Value
- Berbasis Dokumen
- Berbasis Graph
- Berbasis Tabular



Gambar 13. Tipe SQL dan NoSQL database

3.2.3. MongoDB

MongoDB adalah basisdata NoSQL yang berbasis pada dokumen. MongoDB menyimpan datanya dalam bentuk dokumen yang mirip dengan JSON dan tentu saja tidak perlu schema. Jadi satu dokumen dan dokumen yang lain bisa memiliki struktur yang berbeda.

3.2.3. Terminologi

Struktur data pada mongoDB terdiri collection dan document. Untuk lebih mengerti struktur data ini kita akan membandingkannya dengan MySQL pada gambar di bawah ini.

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by MongoDB itself)

Gambar 14. Terminologi

Jadi setiap kolom dijelaskan oleh field, lalu dokumen merepresentasikan baris, dan collection merupakan tabel dimana dokumen dikumpulkan.

Misalkan kita coba mengacu pada saat kita belajar mysql maka database kita akan menjadi,

```
database : rentalfilm
collection : customers
field : id, username(namaid), namalengkap, email
```

collection bisa diibaratkan sebagai tabel dan field bisa diibaratkan sebagai kolom. Untuk membuat collection kita hanya perlu menggunakan query:

```
db.createCollection(<namacollection>, parameter(optional))
```

- capped - Ketika mencapai ukuran maksimum maka data lama akan dihapus agar data baru bisa masuk
- autoIndexId - membuat index otomatis pada kolom _id
- size - menentukan ukuran maksimum dalam byte pada saat parameter capped aktif
- max - menentukan jumlah maksimal dokumen pada saat parameter capped aktif.

Seperti yang dilihat, kita tidak perlu menentukan struktur kolom kita di awal pembuatan seperti mysql, sehingga NoSQL lebih dinamis terkait hal tersebut.

3.2.4. Perintah dasar

Berikut ini adalah perintah-perintah dasar dari MongoDB

Perintah Dasar

mongo	masuk ke dalam terminal mongodb
show dbs	menampilkan semua basisdata pada server
show collections	menampilkan semua collection pada basisdata
use <namabasisdata>	memilih basisdata yang ingin digunakan
db	menampilkan basisdata yang sedang digunakan
db.dropDatabase()	menghapus basisdata yang sedang digunakan
db.createCollection(<namacollection>)	membuat collection pada basisdata yang digunakan

Gambar 15. Perintah dasar MongoDB

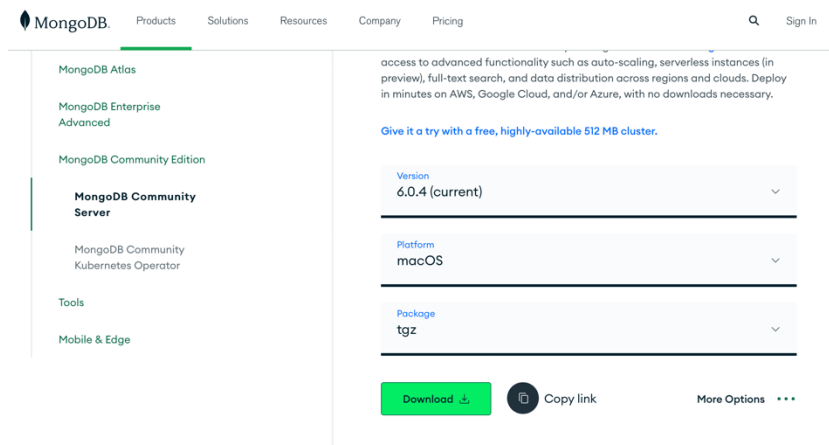
3.3. Software

1. MongoDB
2. Compass
3. MongoDB Shell

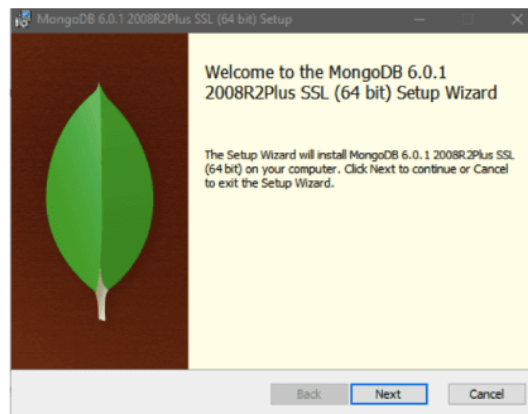
3.4. Tahapan Kerja

3.4.1. Instalasi MongoDB

1. Download MongoDB dari link berikut, download versi stable terakhir 6.0.1(current)
<https://www.mongodb.com/try/download/community>

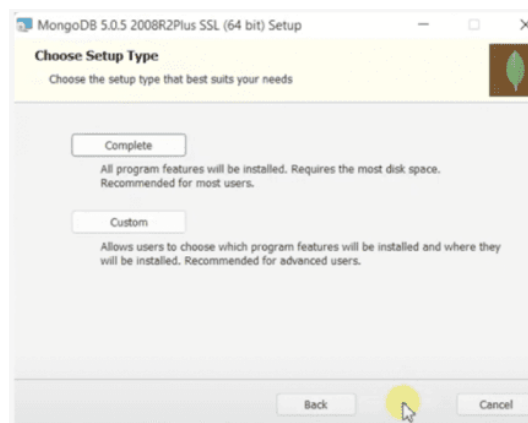


- Setelah selesai download silahkan untuk menjalankan file instalasi.

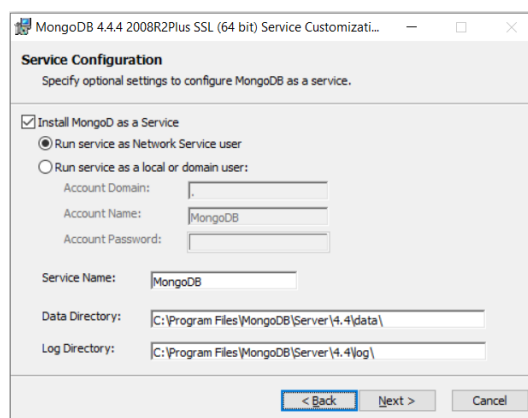


Gambar 16. Instalasi MongoDB

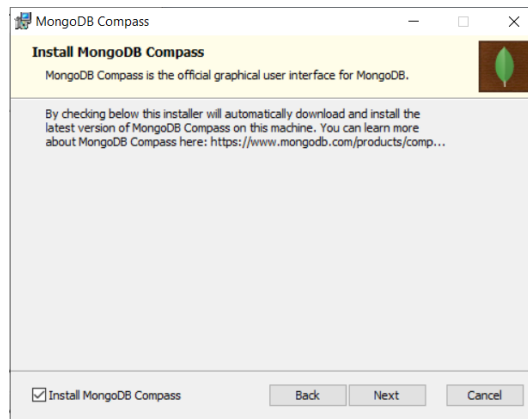
- Lanjutkan instalasi, Pilihlah menu complete untuk saat ini.



- Pada menu konfigurasi silahkan pilih *install mongod as a service* agar kita lebih mudah untuk mengelola server kita. Pilih *run service as Network Service User*.

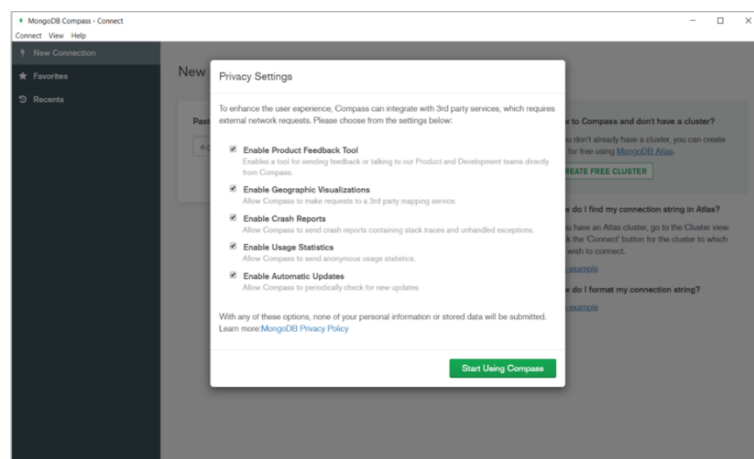


- Pada menu mongoDB Compass, lakukan instalasi. MongoDB Compass merupakan GUI untuk mongoDB, sama seperti MySQL Workbench untuk MySQL.



Gambar 17. Opsi instalasi MongoDB Compass

6. Setelah selesai instalasi mongodb, silahkan tutup jendela instalasi. Untuk mongodb compass akan terlihat tampilan seperti di bawah ini.



Gambar 18. Aplikasi MongoDB

7. Tekan *Start using compass* lalu tutup jendela mongodb compass untuk saat ini.

3.4.2. Set up Environment Variable

1. Download “MongoDB Shell” melalui halaman download di <https://www.mongodb.com/try/download/shell>. Kemudian ekstrak zip file yang sudah didownload ke folder yang diinginkan.

TOOLS

MongoDB Shell Download

MongoDB Shell is the quickest way to connect to (and work with) MongoDB. Easily query data, configure settings, and execute other actions with this modern, extensible command-line interface – replete with syntax highlighting, intelligent autocomplete, contextual help, and error messages.

Note: MongoDB Shell is an open source (Apache 2.0), standalone product developed separately from the MongoDB Server.

Learn more

Version

1.8.0

Platform

Windows 64-bit (8.1+)

Package

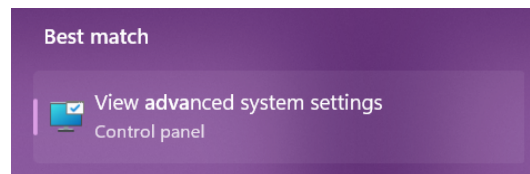
zip

Download

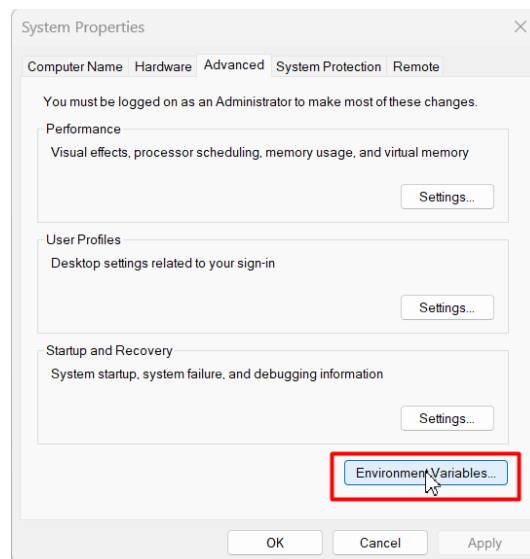
Copy link

More Options

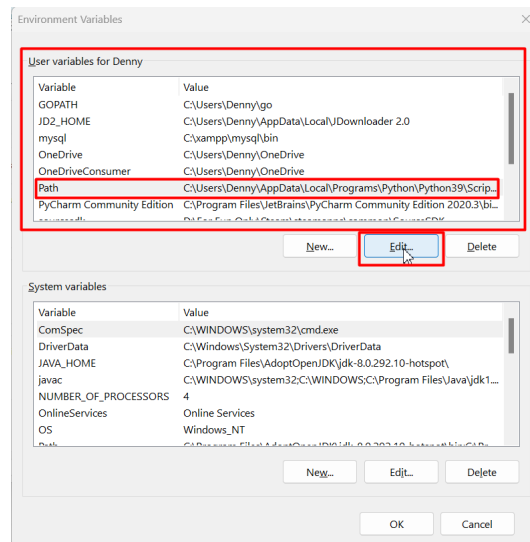
2. Buka search dengan menekan tombol start dan S di keyboard lalu ketik “Advanced System Settings” lalu pilih menu “View Advanced System Setting”.



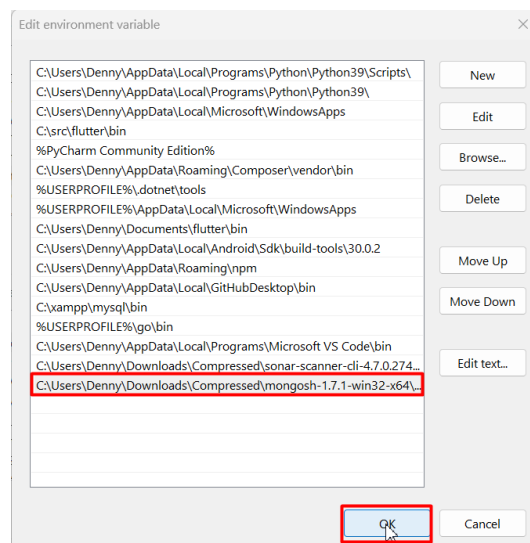
3. Dari jendela baru yang terbuka tekan tombol “Environment Variables”.



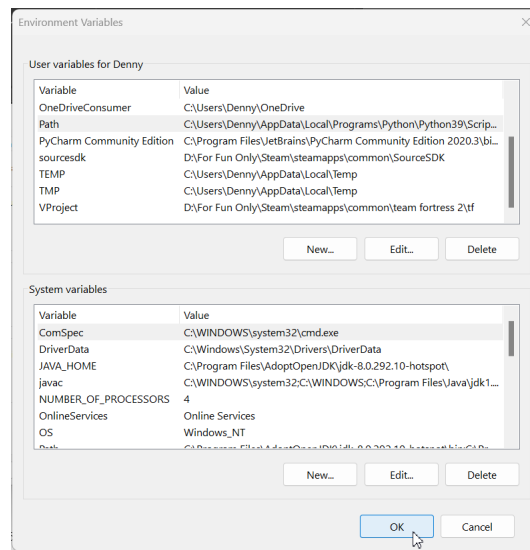
4. Di menu berikutnya pada bagian “User Variables” Cari dan pilih variable “Path” kemudian tekan tombol “Edit”.



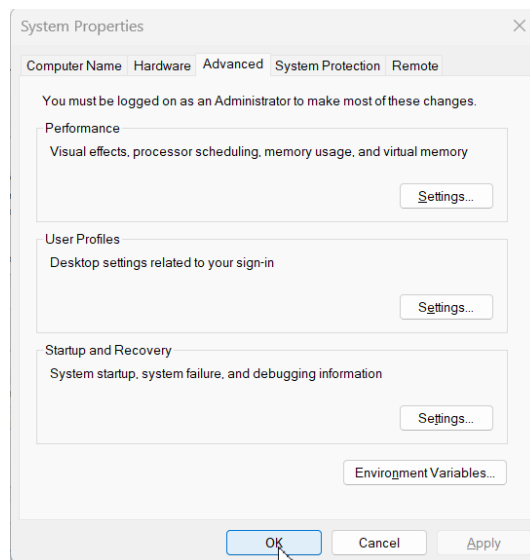
5. Cari dan salin lokasi folder MongoDB Shell yang sudah diekstrak tadi, contohnya “C:\Users\Denny\Downloads\Compressed\mongosh-1.7.1-win32-x64\bin”. Kemudian di menu Edit Environment Variable tadi tambahkan lokasi folder MongoDB bin yang sudah di salin kedalam baris yang kosong dengan mengklik dua kali. Jika sudah tekan tombol OK.



6. Kembali pada menu “Environment Table” Tekan Tombol OK.



7. Lakukan hal yang sama pada “Advanced System Settings”.



3.4.3. Persiapan Basisdata

1. Pada tahap persiapan database kita dapat menggunakan cmd untuk bekerja. untuk menggunakan cmd kita hanya perlu mengetikan pada cmd.

```
mongosh
```

```
mongosh mongodb://127.0.0.1:27017/localhost?directConnection=true&serverSelectionTimeoutMS=2000
C:\Users\Denny>mongosh localhost
Current Mongosh Log ID: 640ed862f99ea9a079fa7f48
Connecting to:  mongodb://127.0.0.1:27017/localhost?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.7.1
Using MongoDB: 6.0.4
Using Mongosh: 1.7.1

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

-----
The server generated these startup warnings when booting
2023-03-13T06:47:26.572+07:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-----
localhost>
```

2. Untuk menunjuk pada suatu server khusus kita bisa menggunakan alamat server kita setelah mongo. pada contoh ini kita akan menggunakan localhost

```
mongosh localhost
```

3. Pastikan path menuju mongod server sudah tersimpan pada windows. Untuk membuat/berpindah basisdata, kita dapat menggunakan query.

```
use <namadatabase>
```

4. Maka mongo akan mengarahkan kita pada basisdata yang kita maksud jika ada, jika belum ada maka akan otomatis untuk dibuat. Kita akan membuat sama seperti pada saat mempelajari mysql.

```
use rentalfilm
```

```
Command Prompt - mongo
C:\Users\thiops>mongo
MongoDB shell version v4.4.4
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("4b3f3756-1d79-4152-9f63-ea0f1cac027") }
MongoDB server version: 4.4.4

The server generated these startup warnings when booting:
2021-04-01T23:43:15.160+07:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-----
> use rentalfilm
switched to db rentalfilm
>
```

5. Untuk melihat semua basisdata yang kita miliki pada server mongodb, kita dapat menggunakan query

```
show dbs
```

```
Command Prompt - mongo
C:\Users\thiop>mongo
MongoDB shell version v4.4.4
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("4b3f3756-1d79-4152-9f63-ea0f1cac027") }
MongoDB server version: 4.4.4

The server generated these startup warnings when booting:
  2021-04-01T23:43:15.160+07:00: Access control is not enabled for the database. Read and write access to data and
configuration is unrestricted
  ---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
  ---
> use rentalfilm
switched to db rentalfilm
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
>
```

Basis data rentalfilm belum muncul karena kita belum memasukan data. sehingga belum tersimpan sepenuhnya

- Setelah berhasil membuat basisdata, sekarang saatnya kita membuat collection. Seperti yang dibahas sebelumnya,
- Sekarang kita akan membuat collection *customers*

```
db.createCollection("customers")
```

```
Command Prompt - mongo
C:\Users\thiop>mongo
MongoDB shell version v4.4.4
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("4b3f3756-1d79-4152-9f63-ea0f1cac027") }
MongoDB server version: 4.4.4

The server generated these startup warnings when booting:
  2021-04-01T23:43:15.160+07:00: Access control is not enabled for the database. Read and write access to data and
configuration is unrestricted
  ---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
  ---
> use rentalfilm
switched to db rentalfilm
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
> db.createCollection("customers")
{ "ok" : 1 }
>
```

- Jika kita cek kembali daftar basis data kita, maka *rentalfilm* sudah terdaftar karena sudah memiliki isi di dalamnya.

```
Command Prompt - mongo
The server generated these startup warnings when booting:
  2021-04-01T23:43:15.160+07:00: Access control is not enabled for the database. Read and write access to data and
configuration is unrestricted
  ---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
  ---
> use rentalfilm
switched to db rentalfilm
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
rentalfilm 0.000GB
>
```

3.4.4. Manipulasi Data

- Insert Data. Memasukan data dapat kita lakukan pada mongo shell dengan menggunakan query,

```
db.<namacollection>.insert(<document>)
```

Misalkan kita ingin memasukkan data,

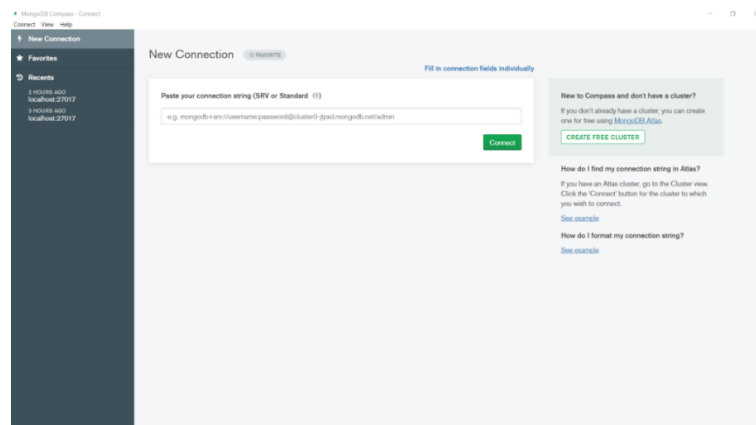
```
id = Astrid,  
fullname = Astrid Tiar,  
email = astrid.tiar@ibik.ac.id
```

maka query kita menjadi,

```
db.customers.insert({  
  id : "Astrid",  
  fullname : "Astrid Tiar",  
  email : "astrid.tiar@ibik.ac.id"  
})
```

```
> db.customers.insert({id : "Astrid", fullname : "Astrid Gruber", email : "astrid.gruber@apple.at"})  
WriteResult({ "nInserted" : 1 })  
>
```

2. Untuk melihat datanya kita bisa menggunakan MongoDB Compass. Jadi pertama bukalah MongoDB Compass kita dan pada menu koneksi kita tekan *connect*, kita akan tersambung dengan localhost pada port 27017 (default port mongodb). Jika ingin mengisi secara manual server kita, bisa dengan memilih menu *fill in connection fields manually*.



3. Setelah itu kita pilih Basis data dan collection yang ingin manipulasi datanya.

The screenshot shows the MongoDB Atlas interface. The 'Databases' tab is active, displaying a table of databases. Below it, the 'Collections' tab is active, showing a table of collections. The 'customers' collection is highlighted, and its details are shown in a modal window.

Database Name	Storage Size	Collections	Indexes
admin	20.0KB	0	1
config	36.0KB	0	2
local	20.0KB	1	1
rentalfilm	4.0KB	1	1

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size
customers	0	-	0.0 B	1	4.0 KB

rentalfilm.customers Documents

Documents Aggregations Schema Explain Plan Indexes Validation

0 FILTER { field: 'value' } OPTIONS

ADD DATA VIEW

Displaying documents 0 - 0 of 0

This collection has no data

It only takes a few seconds to import data from a JSON or CSV file

Import Data

4. Tekan tombol *find* maka datanya akan muncul

The screenshot shows the MongoDB Atlas interface for the 'customers' collection. The 'Documents' tab is active, and a single document is displayed. The document contains fields: _id, id, fullname, and email.

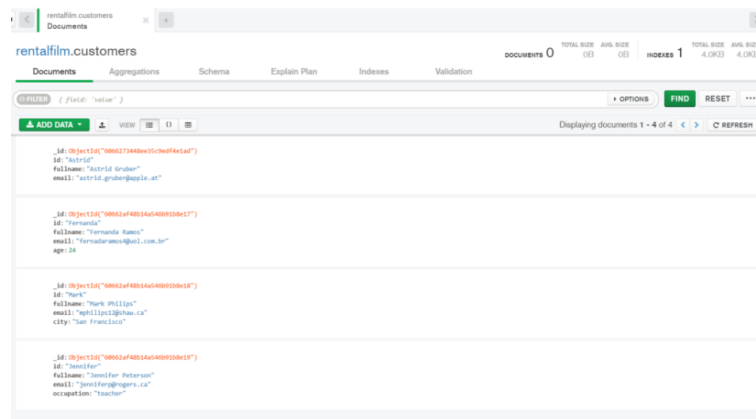
```
{
  "_id": "6066273448e31c9d0f4e1ad",
  "id": "astrid",
  "fullname": "astrid gruben",
  "email": "astrid.gruben@apple.at"
}
```

5. Kita juga bisa memasukkan data menggunakan tombol *add data* --> *Insert document*. Untuk memasukkan data kita hanya perlu memasukkan data tersebut ke dalam list, mirip seperti saat kita bekerja dengan dictionary.

```

db.customers.insert([
  {
    id : "Fernanda",
    fullname : "Fernanda Ramos",
    email : "fernadaramos4@uol.com.br",
    age : 24
  },
  {
    id : "Mark",
    fullname : "Mark Philips",
    email : "mphilips12@shaw.ca",
    city : "San Francisco"
  },
  {
    id : "Jennifer",
    fullname : "Jennifer Peterson",
    email : "jenniferp@rogers.ca",
    occupation : "teacher"
  }
])

```



6. Berbeda dengan sql yang memiliki schema yang tetap, pada NoSQL kita bisa memasukkan field yang berbeda seperti age, city, dan occupation.

Insert Data

insertOne	Memasukan satu dokumen ke dalam collection
<code>db.customers.insertOne({id : "Astrid"})</code>	memasukkan data dengan id Astrid
insertMany	Memasukan banyak dokumen ke dalam collection
<code>db.customers.insertMany([{id : "Astrid"}, {id : "Mark"}])</code>	Memasukkan data dengan id Astrid dan Mark
insert	Memasukan satu/banyak dokumen ke dalam collection
<code>db.customers.insert([{id : "Fernanda"}, {id : "Mark"}])</code>	Memasukkan data dengan id Fernanda dan Mark

7. Update Data. Untuk mengubah data kita kita bisa menggunakan query.

```

db.<namaCollection>.update({<kriteria>}, {$set:{<nilai baru>}})

```

8. Kita ingin mengubah nama dengan id "Jennifer" dari Jennifer Peterson menjadi Jennifer Aniston.

```
db.customers.update(
  {id:"Jennifer"},
  {$set:{fullname:'Jennifer Aniston'}}
)
```

```
_id: ObjectId("60662af48b14a546b91b8e18")
id: "Mark"
fullname: "Mark Phillips"
email: "mphilips12@shaw.ca"
city: "San Francisco"
```

```
_id: ObjectId("60662af48b14a546b91b8e19")
id: "Jennifer"
fullname: "Jennifer Aniston"
email: "jennifery@rogers.ca"
occupation: "teacher"
```

Update Data

updateOne	Mengubah nilai pertama yang ditemukan oleh filter dengan nilai pada set
db.customers.updateOne({id:"Astrid"}, {\$set: {age:24}})	mengubah data age menjadi 24 pada id Astrid
updateMany	Mengubah semua nilai yang ditemukan oleh filter dengan nilai pada set
db.customers.updateMany({id:"Astrid"}, {\$set: {age:24}})	mengubah data age menjadi 24 pada semua id Astrid
update	Mengubah nilai pertama yang ditemukan oleh filter dengan nilai pada set. (mongodb < version 5)
db.customers.update({id:"Astrid"}, {\$set: {age:24}})	Memasukkan data dengan id Fernanda dan Mark
replaceOne	Menghapus dan mengganti data sesuai filter dengan data kedua (delete lalu update)
db.customers.replaceOne({id:"Astrid"}, {id:"Anata"})	Menghapus data Astrid lalu menggantinya dengan data Anata

9. Delete Data. Untuk menghapus data dari tabel dapat menggunakan query

```
db.<nama collection>.remove(<query>)
```

misalkan kita ingin menghapus data dengan id = Jennifer

```
db.customers.remove(
  {id:"Jennifer"}
)
```

```
_id: ObjectId("6066273448ee35c9edf4e1ad")
id: "Astrid"
fullname: "Astrid Gruber"
email: "astrid.gruber@apple.at"
```

```
_id: ObjectId("60662af48b14a546b91b8e17")
id: "Fernanda"
fullname: "Fernanda Ramos"
email: "fernadaramos4@uol.com.br"
age: 24
```

```
> _id: ObjectId("60662af48b14a546b91b8e18")
id: "Mark"
fullname: "Mark Phillips"
email: "mphilips12@shaw.ca"
city: "San Francisco"
```

Delete Data

<i>deleteOne</i>	menghapus nilai pertama yang ditemukan oleh filter
<code>db.customers.deleteOne({id: "Astrid"})</code>	menghapus data dengan id Astrid
<i>deleteMany</i>	menghapus semua nilai yang ditemukan oleh filter
<code>db.customers.deleteMany({id: "Astrid"})</code>	menghapus semua data dengan id Astrid
<i>remove</i>	menghapus nilai pertama yang ditemukan oleh filter
<code>db.customers.remove({id: "Astrid"})</code>	menghapus data dengan id Astrid

3.5. Tugas dan Latihan

1. Buatlah database rentalfilm di mongoDB seperti yang telah dilakukan di pertemuan pertama.
2. Masukkan tiga buah data di setiap collections pada database rentalfilm menggunakan query.
3. Lakukan minimal dua buah query update data dan delete data.