



MODUL-2

NOSQL : Database, documents, collections

BASIS DATA LANJUT

Disusun oleh :

AGIT AMRULLAH, M.KOM.

190302356

UNIVERSITAS AMIKOM YOGYAKARTA

Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta 55283

MODUL-01 :

DATABASE, DOCUMENTS, COLLECTIONS

A. Database MongoDB

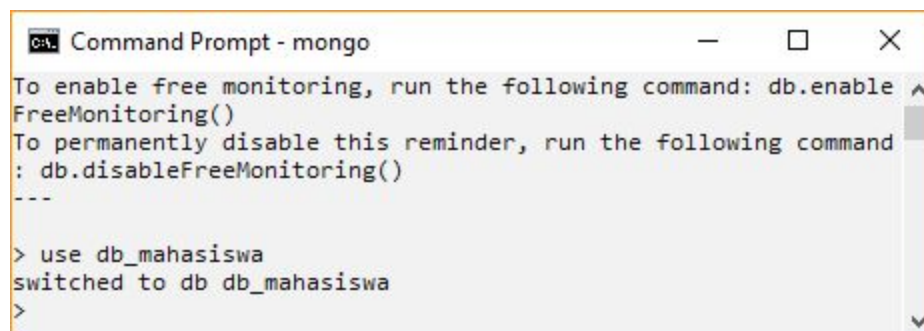
Infrastruktur basis data pada MongoDB berbeda dengan SQL, berikut ini adalah perbedaan MongoDB dengan SQL :

SQL	MongoDB
Table	Collection
Row	Document
Column	Field
Primary key	ObjectId
Index	Index
View	View
Nested table or object	Embedded document
Array	Array

Pada modul ini, kita akan membuat database dengan nama db_mahasiswa. Perintah pada MongoDB dalam membuat database adalah :

> use nama_database

Apabila sukses maka akan terdapat notifikasi pada gambar dibawah ini.



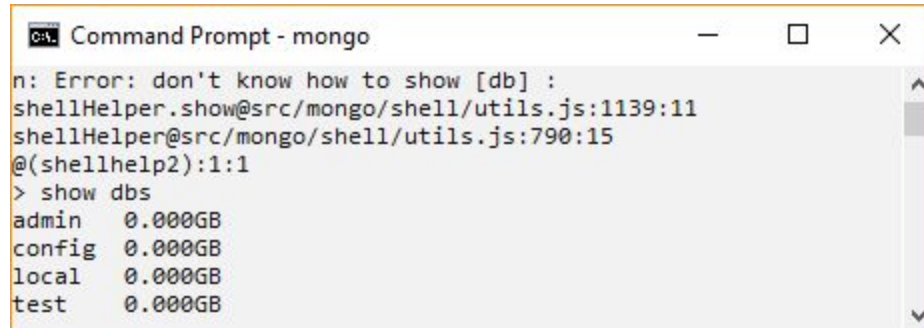
```
Command Prompt - mongo
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> use db_mahasiswa
switched to db db_mahasiswa
>
```

Gambar 1. Database telah sukses dibuat

Kita dapat melihat daftar database yang dibuat pada database MongoDB dengan menggunakan shell method :

> show dbs

Berikut adalah tampilan daftar database yang ada pada MongoDB :



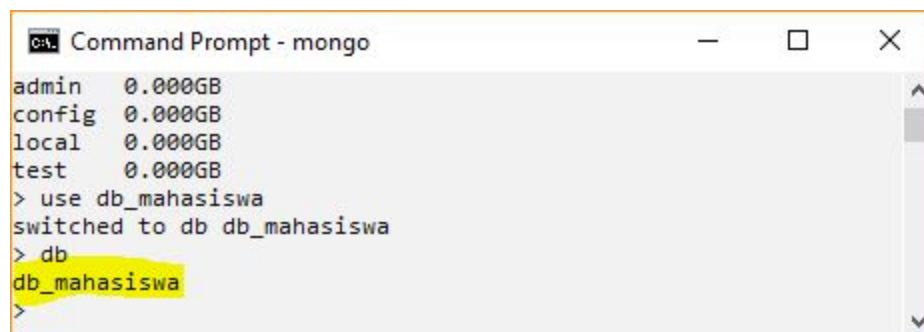
```
Command Prompt - mongo
n: Error: don't know how to show [db] :
shellHelper.show@src/mongo/shell/utils.js:1139:11
shellHelper@src/mongo/shell/utils.js:790:15
@(shellhelp2):1:1
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
test     0.000GB
```

Gambar 2. Daftar database pada MongoDB

Secara default apabila database belum terdapat document pada collection, maka shell method diatas tidak akan memunculkan nama database kita. Untuk itu, kita pastikan kita pada database aktif yang saat ini kita gunakan dengan mengetikan perintah :

> db

Berikut ini adalah tampilan database aktif saat ini (lihat penanda warna kuning) :



```
Command Prompt - mongo
admin    0.000GB
config  0.000GB
local    0.000GB
test     0.000GB
> use db_mahasiswa
switched to db db_mahasiswa
> db
db_mahasiswa
>
```

Gambar 3. Tampilan database aktif

Shell Method : insert()

Penambahan data dilakukan pada dokumen (SQL bernama row) dengan menggunakan beberapa shell method alternatif, diantaranya dapat dilihat pada dibawah ini. Secara otomatis setiap penambahan data akan menambahkan inisialisasi pada field **_id** sebagai inisial unik dari document yang ditambahkan, namun apabila kita melakukan penambahan document dengan menspesifikkan field **_id** maka document yang ditambahkan tidak akan me-generate otomatis pada field **_id**.

> db.collection.insert()

Menambahkan *document* pada *collection*, dapat berupa array/multi dokumen

> db.collection.insertOne()

Menambahkan sebuah *document* pada *collection*

> db.collection.insertMany()

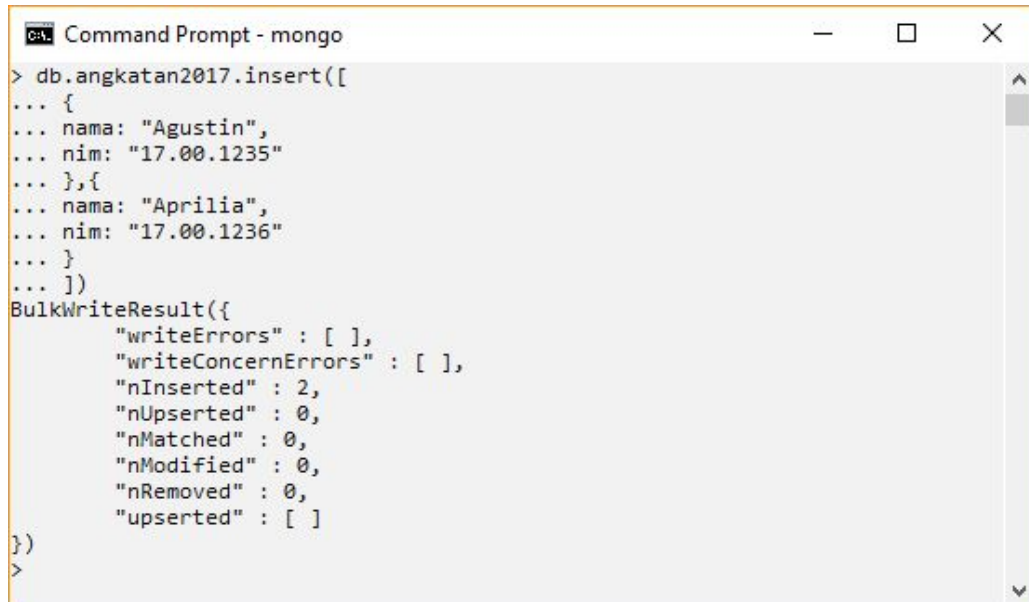
Menambahkan multi *document* pada *collection*

Berikut ini adalah contoh penambahan single dan multiple document pada collection, perbedaannya adalah apabila document yang di-input bertipe multi maka wajib menambahkan kurung siku “[]” untuk menunjukkan dokumen tersebut lebih dari satu.



```
Command Prompt - mongo
> db
db_mahasiswa
> db.angkatan2017.insert({
...  nama: "Leonard",
...  nim: "17.00.1234"
... })
WriteResult({ "nInserted" : 1 })
> db.angkatan2017.find()
{ "_id" : ObjectId("5e62045f286d5089b1a96882"), "nama" : "Leonard", "nim" :
"17.00.1234" }
>
```

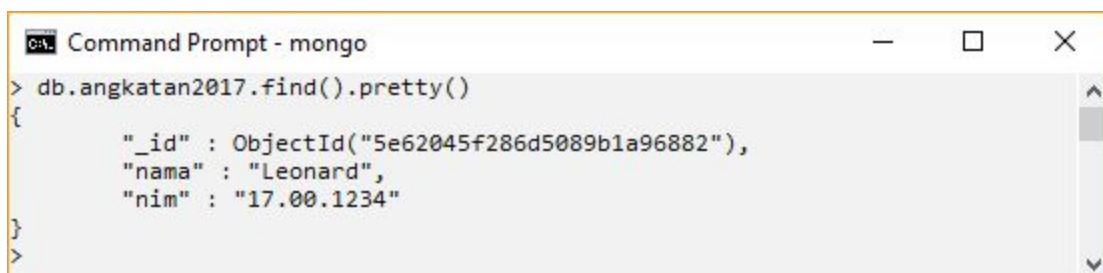
Gambar 4. Penambahan single document



```
Command Prompt - mongo
> db.angkatan2017.insert([
... {
...   nama: "Agustin",
...   nim: "17.00.1235"
... },{
...   nama: "Aprilia",
...   nim: "17.00.1236"
... }
... ])
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 2,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
>
```

Gambar 5. Penambahan document pada collection dengan metode single document dan multiple document

Apabila document sudah berhasil ditambahkan, maka akan terdapat status **WriteResult({ "nInserted": n })** yang berarti adalah document sejumlah n telah berhasil ditambahkan pada collection. Document yang berhasil ditambahkan dapat kita cek datanya pada collection dengan menggunakan shell method **db.collection.find()**, shell method ini dipergunakan untuk menampilkan document pada collection (lihat gambar 4). Permasalahannya adalah data tersebut tidak dapat ditampilkan dengan pembacaan yang mudah. Format BSON (Binary JSON) akan dapat mudah dibaca dengan menambahkan shell method **pretty()** setelah shell method **find()** Tampilan data setelah menggunakan shell method **pretty()** dapat dilihat pada gambar di bawah ini :



```
Command Prompt - mongo
> db.angkatan2017.find().pretty()
{
  "_id" : ObjectId("5e62045f286d5089b1a96882"),
  "nama" : "Leonard",
  "nim" : "17.00.1234"
}
```

Gambar 6. Penggunaan shell method pretty() untuk menampilkan data BSON

Shell Method : remove()

Dalam melakukan penghapusan data, pada shell dapat menggunakan method `remove()`. Metode ini digunakan dengan cara menghapus salah satu document (spesifik) atau multi document. Teknik ini didasari pada referensi data yang akan ditargetkan, sebagai contoh apabila kita memiliki document sapi, kambing, ayam, nanas, rambutan, dan mangga. Kemudian kita menghapus document dengan kategori binatang, maka sapi, kambing, dan ayam akan secara otomatis terhapuskan. Berbeda dengan kita menghapus secara spesifik, sebagai contoh menghapus document dengan pemilik bernama Wahid, maka document dengan data yang dimiliki Wahid akan terhapus secara spesifik. Shell method ini dapat dieksekusi dengan menggunakan method :

```
> db.collection.remove()
```

Pada gambar dibawah ini sebagai contoh kita akan menghapus document secara spesifik pada ObjectId yang ditunjukkan pada tanda warna kuning dibawah ini.

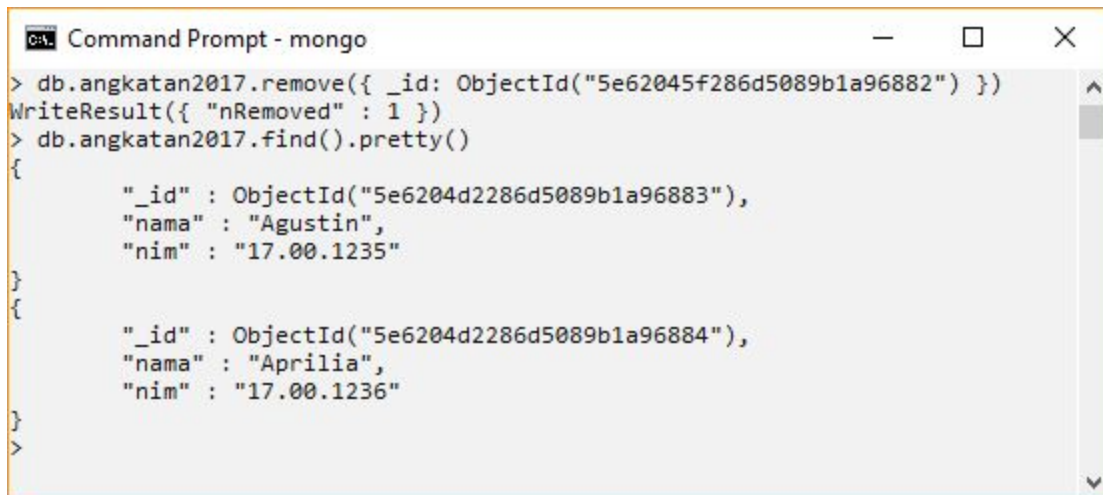


```
Command Prompt - mongo
> db.angkatan2017.find().pretty()
{
  "_id" : ObjectId("5e62045f286d5089b1a96882"),
  "nama" : "Leonard",
  "nim" : "17.00.1234"
}
{
  "_id" : ObjectId("5e6204d2286d5089b1a96883"),
  "nama" : "Agustin",
  "nim" : "17.00.1235"
}
{
  "_id" : ObjectId("5e6204d2286d5089b1a96884"),
  "nama" : "Aprilia",
  "nim" : "17.00.1236"
}
>
```

Gambar 7. Target document yang akan dihapuskan

Method `remove()` yang akan kita eksekusi kita tambahkan detail referensi yaitu pada field `_id` dengan informasi ObjectId detail yang kita targetkan. Detail dapat dilihat pada

gambar 7. Apabila sukses maka respond jumlah data yang dihapus akan ditampilkan pada object **nRemoved**.



```
Command Prompt - mongo
> db.angkatan2017.remove({ _id: ObjectId("5e62045f286d5089b1a96882") })
WriteResult({ "nRemoved" : 1 })
> db.angkatan2017.find().pretty()
{
  "_id" : ObjectId("5e6204d2286d5089b1a96883"),
  "nama" : "Agustin",
  "nim" : "17.00.1235"
}
{
  "_id" : ObjectId("5e6204d2286d5089b1a96884"),
  "nama" : "Aprilia",
  "nim" : "17.00.1236"
}
>
```

Gambar 8. Penghapusan document pada collection

Shell Method : update()

Dalam melakukan perubahan data pada document (update), shell method yang dipergunakan adalah update(). Sebagai contoh kita akan merubah data pada document dari nama *Agustin* menjadi *Wahid* seperti pada gambar 9 dibawah ini. Secara spesifik, identifier unik yang dapat kita pergunakan adalah pada field **_id** sebagai acuan utama kita dalam merubah data. Apabila telah sukses maka akan terdapat informasi jumlah data yang diubah pada object **nModified** yaitu 1 data. Secara umum shell method dalam melakukan update dapat dilihat pada dibawah ini. Object **\$inc** dan **\$set** pada contoh metode update dibawah ini, dinamakan *Update Operator*, object \$inc dipergunakan untuk menambahkan data stok sejumlah n dan object \$set dipergunakan untuk merubah data pada field yang akan dirubah.

```
db.collection.update(  
  { field: n },  
  {  
    $inc: { stock: n },  
    $set: {  
      field: "new string",  
      "field.1": { fieldA: "xyz", fieldB: 3 }  
    }  
  }  
)
```



```
Command Prompt - mongo  
> db.angkatan2017.find({ nim: "17.00.1235" }).pretty()  
{  
  "_id" : ObjectId("5e6204d2286d5089b1a96883"),  
  "nama" : "Agustin",  
  "nim" : "17.00.1235"  
}  
> db.angkatan2017.update({ nim: "17.00.1235" },{$set:{nama: "Wahid"}})  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })  
> db.angkatan2017.find({ nim: "17.00.1235" }).pretty()  
{  
  "_id" : ObjectId("5e6204d2286d5089b1a96883"),  
  "nama" : "Wahid",  
  "nim" : "17.00.1235"  
}  
>
```

Gambar 9. Update document Agustin menjadi Wahid

Shell Method : find()


Dalam melakukan pencarian data, shell method yang dapat digunakan salah satunya adalah find(). Pencarian data pada document dengan cara melakukan spesifikasi data yang akan dicari, pencarian data pada document dengan menunjukkan kemiripan data, dapat menggunakan metode *regular expression*, sebagai contoh pencarian data pada document dengan nama awal agus dapat dilakukan dengan cara :

```
db.collection.find({
  nama: { $regex: /^agus/ }
})
```

Atau pencarian username dengan string agus dan diikuti dengan integer (contoh agus343534) :

```
db.collection.find({
  nama: { $regex: /^agus/i }
})
```

Shell method ini dalam penggunaannya dalam terminal dapat dilihat pada gambar dibawah ini :

A screenshot of a Windows Command Prompt window titled "Select Command Prompt - mongo". The prompt shows a MongoDB query: > db.angkatan2017.find({ nim: "17.00.1235" }).pretty(). The output is a JSON document: { "_id" : ObjectId("5e6204d2286d5089b1a96883"), "nama" : "Agustin", "nim" : "17.00.1235" }. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
CA: Select Command Prompt - mongo
> db.angkatan2017.find({ nim: "17.00.1235" }).pretty()
{
  "_id" : ObjectId("5e6204d2286d5089b1a96883"),
  "nama" : "Agustin",
  "nim" : "17.00.1235"
}
>
```

Gambar 10. Pencarian data pada document dengan nim

Shell Method : dropDatabase()

Dalam melakukan penghapusan database hal pertama yang dilakukan adalah memastikan bahwa kita benar telah aktif pada database yang akan kita hapus. *Pertama* pastikan database aktif yang akan kita hapus telah berada pada shell dengan menggunakan perintah `db`, *kedua* lakukan penghapusan database dengan menggunakan shell method `dropDatabase()`, *ketiga* pastikan database telah terhapus dengan menampilkan daftar database dengan menggunakan shell method **show dbs**. Secara langkah-langkahnya dapat dilihat pada gambar di bawah ini :



```
Select Command Prompt - mongo
> db
db_mahasiswa
> db.dropDatabase()
{ "dropped" : "db_mahasiswa", "ok" : 1 }
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
test     0.000GB
>
```

B. Tugas

Silahkan buka link form yang tersedia untuk mengerjakan tugas laporan.