

NAMA : NAUFAL AL FIKRI  
NIM : 17050623011  
KELAS : D3 MI 2017

## TUGAS BASDAT LANJUT

### ONE to MANY dan MANY to MANY

DATABASE 1 (One to Many) :

Database Inventory dengan tabel “equipment” dan “borrowers”

Tabel Equipment :

- propertyNumber
- itemName
- description
- dateAcquired
- accountableEmployee

Table Borrowers :

- employeeID
- employeeFname
- employeeLname
- employeeMname
- status

## CREATE DATABASE

```
C:\WINDOWS\system32\cmd.exe - mongo.exe
> use inventory
switched to db inventory
> db
{
  "equipment": 1,
  "inventory": 1
}
```

## CREATE dan INSERT TABLE

Insert equipment :

```
> db.equipment.insert({_propertyNumber : 1, itemName : "chain", description: "Big Chain", dateAcquired : "2019-02-27", accountableEmployee : "Naufal"})
WriteResult({ "nInserted" : 1 })
```

Insert borrowers :

```
> db.borrowers.insert([
  {employeeID: "1", employeeFname: "naufal", employeeLname: "fikri", employeeMname: "al", employeeStatus: "Student"},
  {employeeID: "2", employeeFname: "jihad", employeeLname: "utama", employeeMname: "satrio", employeeStatus: "Student"},
  {employeeID: "3", employeeFname: "agus", employeeLname: "santoso", employeeMname: "budi", employeeStatus: "Student"}])
2019-02-27T23:25:00.659+0800 E QUERY [js] SyntaxError: missing ) after element list @(shell):3:0
> db.borrowers.insert([
  {employeeID: "1", employeeFname: "naufal", employeeLname: "fikri", employeeMname: "al", employeeStatus: "Student"},
  {employeeID: "2", employeeFname: "jihad", employeeLname: "utama", employeeMname: "satrio", employeeStatus: "Student"},
  {employeeID: "3", employeeFname: "agus", employeeLname: "santoso", employeeMname: "budi", employeeStatus: "Student"}])
BulkWriteResult({
  "writeErrors": [ ],
  "writeConcernErrors": [ ],
  "nInserted": 3,
  "nUpserted": 0,
  "nMatched": 0,
  "nModified": 0,
  "nRemoved": 0,
  "upserted": [ ]
})
```

NAMA : NAUFAL AL FIKRI

NIM : 17050623011

KELAS : D3 MI 2017

## READ

```
> db.equipment.find().pretty()
{
  "_id" : ObjectId("5c76a78686589d7f3d9c80b2"),
  "propertyNumber" : 1,
  "itemName" : "chair",
  "description" : "Big Chair",
  "dateAcquired" : "2019-02-27",
  "accountableEmployee" : "Naufal"
}

> db.equipment.find().pretty()
{
  "_id" : ObjectId("5c76a8ee86589d7f3d9c80b3"),
  "propertyNumber" : 2,
  "itemName" : "table",
  "description" : "circle table",
  "dateAcquired" : "2019-02-27",
  "accountableEmployee" : "Naufal"
}

> db.equipment.find().pretty()
{
  "_id" : ObjectId("5c76ac3786589d7f3d9c80b7"),
  "propertyNumber" : 3,
  "itemName" : "fan",
  "description" : "big fan",
  "dateAcquired" : "2019-02-27",
  "accountableEmployee" : "Naufal"
}

> db.borrowers.find().pretty()
{
  "_id" : ObjectId("5c76abf286589d7f3d9c80b4"),
  "employeeID" : "1",
  "employeeName" : "naufal",
  "employeeMname" : "fikri",
  "employeeMname" : "al",
  "employeeStatus" : "Student"
}

> db.borrowers.find().pretty()
{
  "_id" : ObjectId("5c76abf286589d7f3d9c80b5"),
  "employeeID" : "2",
  "employeeName" : "jihad",
  "employeeMname" : "utama",
  "employeeMname" : "satrio",
  "employeeStatus" : "Student"
}

> db.borrowers.find().pretty()
{
  "_id" : ObjectId("5c76abf286589d7f3d9c80b6"),
  "employeeID" : "3",
  "employeeName" : "jihad",
  "employeeMname" : "utama",
  "employeeMname" : "satrio",
  "employeeStatus" : "Student"
}

> db.borrowers.find({employeeID:"1"}).pretty()
{
  "_id" : ObjectId("5c76abf286589d7f3d9c80b4"),
  "employeeID" : "1",
  "employeeName" : "naufal",
  "employeeMname" : "fikri",
  "employeeMname" : "al",
  "employeeStatus" : "Student"
}
```

## UPDATE

```
> db.equipment.update({itemName:"table"}, {$set: {dateAcquired : "2019-02-28"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.equipment.find().pretty()
{
  "_id" : ObjectId("5c76a78686589d7f3d9c80b2"),
  "propertyNumber" : 1,
  "itemName" : "chair",
  "description" : "Big Chair",
  "dateAcquired" : "2019-02-27",
  "accountableEmployee" : "Naufal"
}

> db.equipment.find().pretty()
{
  "_id" : ObjectId("5c76a8ee86589d7f3d9c80b3"),
  "propertyNumber" : 2,
  "itemName" : "table",
  "description" : "circle table",
  "dateAcquired" : "2019-02-28",
  "accountableEmployee" : "Naufal"
}

> db.equipment.find().pretty()
{
  "_id" : ObjectId("5c76ac3786589d7f3d9c80b7"),
  "propertyNumber" : 3,
  "itemName" : "fan",
  "description" : "big fan",
  "dateAcquired" : "2019-02-27",
  "accountableEmployee" : "Naufal"
}
```

Untuk menghapus satu document dapat digunakan syntax:

NAMA : NAUFAL AL FIKRI

NIM : 17050623011

KELAS : D3 MI 2017

## DELETE

```
> db.borrowers.remove({employeeFname:'agus'})
WriteResult({ "nRemoved" : 1 })
> db.borrowers.find().pretty()
{
  "_id" : ObjectId("5c76abf286589d7f3d9c80b4"),
  "employeeID" : "1",
  "employeeFname" : "naufal",
  "employeeLname" : "fikri",
  "employeeMname" : "al",
  "employeeStatus" : "Student"
}
{
  "_id" : ObjectId("5c76abf286589d7f3d9c80b5"),
  "employeeID" : "2",
  "employeeFname" : "jihad",
  "employeeLname" : "utama",
  "employeeMname" : "satrio",
  "employeeStatus" : "Student"
}
}
```

Apa yang sudah dijabarkan diatas adalah fungsi-fungsi dasar dalam mongodb. Jika ingin melihat fungsi lainnya dalam mongodb dapat dilihat di <https://www.tutorialspoint.com/mongodb/>

**Geri Muhano**  
Web programmer di [erabelajar.com](http://erabelajar.com). Alumni UIN Jakarta.

Share this post

Satu Komentar Developer Erabelajar

## DATABASE 2 (Many to Many) :

Databse kampus memiliki 2 tabel yaitu tabel “mahasiswa” dan tabel “matkul”.

Tabel Mahasiswa :

- nim
- nama
- alamat
- id\_matkul

Tabel Matkul :

- id\_matkul
- nama\_matkul
- sks
- harga

## CREATE DB Kampus,

## INSERT dan READ Tabel Mahasiswa

```
C:\WINDOWS\system32\cmd.exe - mongo.exe
switched to db kampus
> db.mahasiswa.insert([{"_nim":1,"nama":"naufal",alamat:"surabaya",id_matkul:"01"},
... {"_nim":2,"nama":"budi",alamat:"surabaya",id_matkul:"02"},
... {"_nim":3,"nama":"jaka",alamat:"surabaya",id_matkul:"01"},
... {"_nim":4,"nama":"desi",alamat:"sidoarjo",id_matkul:"02"}]) Many
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 4,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
> db.mahasiswa.find().pretty()
{
  "_id" : ObjectId("5c76b30b86589d7f3d9c80b8"),
  "_nim" : "1",
  "nama" : "naufal",
  "alamat" : "surabaya",
  "id_matkul" : "01"
}
{
  "_id" : ObjectId("5c76b30b86589d7f3d9c80b9"),
  "_nim" : "2",
  "nama" : "budi",
  "alamat" : "surabaya",
  "id_matkul" : "02"
}
{
  "_id" : ObjectId("5c76b30b86589d7f3d9c80ba"),
  "_nim" : "3",
  "nama" : "jaka",
  "alamat" : "surabaya",
  "id_matkul" : "01"
}
{
  "_id" : ObjectId("5c76b30b86589d7f3d9c80bb"),
  "_nim" : "4",
  "nama" : "desi",
  "alamat" : "sidoarjo",
  "id_matkul" : "02"
}
```

Dari 2 tabel tersebut bisa kita lihat bahwa 4 mahasiswa bisa mengambil lebih dari 1 matkul. Sedangkan 1 matkul

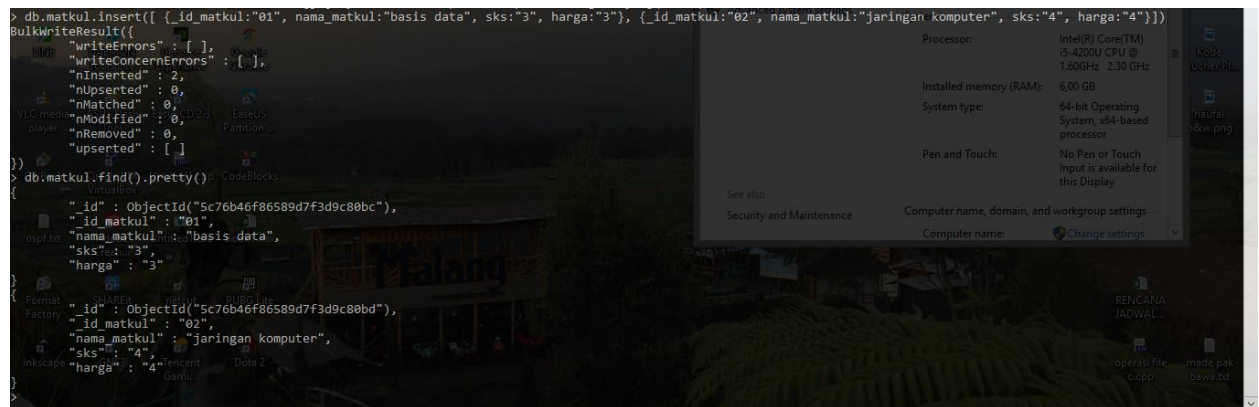
NAMA : NAUFAL AL FIKRI

NIM : 17050623011

KELAS : D3 MI 2017

## CREATE dan READ Tabel Matkul

```
> db.matkul.insert([ {_id_matkul:"01", nama_matkul:"basis data", sks:"3", harga:"3"}, {_id_matkul:"02", nama_matkul:"jaringan komputer", sks:"4", harga:"4"}])
BulkWriteResult({
  "writeErrors": [ ],
  "writeConcernErrors": [ ],
  "nInserted": 2,
  "nUpserted": 0,
  "nMatched": 0,
  "nModified": 0,
  "nRemoved": 0,
  "upserted": [ ]
})
> db.matkul.find().pretty()
{
  "_id" : ObjectId("5c76b46f86589d7f3d9c80bc"),
  "_id_matkul" : "01",
  "nama_matkul" : "basis data",
  "sks" : "3",
  "harga" : "3"
}
{
  "_id" : ObjectId("5c76b46f86589d7f3d9c80bd"),
  "_id_matkul" : "02",
  "nama_matkul" : "jaringan komputer",
  "sks" : "4",
  "harga" : "4"
}
```

The image shows a Windows 10 desktop environment. In the foreground, a MongoDB terminal window is open, displaying the results of a database insertion and a find operation. The insertion command successfully added two documents to the 'matkul' collection. The find command returned two documents: one for 'basis data' (ID: 01, sks: 3, harga: 3) and another for 'jaringan komputer' (ID: 02, sks: 4, harga: 4). In the background, a system information window is open, showing details about the processor (Intel(R) Core(TM) i5-4200U), installed memory (6.00 GB), and system type (64-bit Operating System). The desktop background is a dark, abstract image with some text and graphics. Various icons are visible on the taskbar and desktop, including 'RENCANA JADWAL...', 'operasi File', 'made pak', and '62x9.txt'.