

# Bab 1 : Pengenalan MongoDB

Didalam bab 1 : pengenalan MongoDB anda akan mempelajari :

- -pengertian dari mongodb
- -perbedaan dari non sql dan sql

## Apa itu mongoDB ?

MongoDB adalah database **noSQL** yang menganut sistem **Document Database** dan cara penulisan Seperti **JSON**. MongoDB didirikan pada tahun 2007 yang dirilis ke public pada tahun 2009 dikembangkan oleh perusahaan bernama 10gen yang sekarang berganti nama menjadi MongoDB inc.

MongoDB tidak menggunakan bahasa sql ,namun menggunakan **javascript** sebagai bahasa utama untuk manipulasi document , tetapi jangan khawatir MongoDB bisa dipakai hampir di semua bahasa pemrograman karena ada library masing masing yang bisa mentranslate ke bahasa pemrograman masing masing.

Struktur dari MongoDB mirip Seperti JSON , contoh dari struktur MongoDB :

```
{
  "_id": "5cf0029caff5056591b0ce7d",
  "firstname": "Jane",
  "lastname": "Wu",
  "address": {
    "street": "1 Circle Rd",
    "city": "Los Angeles",
    "state": "CA",
    "zip": "90404"
  },
  "hobbies": ["surfing", "coding"]
}
```

Sebenarnya MongoDb menggunakan struktur data BSON, Apa itu BSON ? **BSON** kependekan dari binary JSON adalah binary encoded serialization document seperti json , sama seperti JSON , DI BSON juga kita bisa menggunakan embedded object , array ,dan lain lain . Keunggulan dari BSON ialah type data yang

didukung lebih banyak , jika kalian masih pusing tentang json tidak apa apa karena kedepanya penulisanya hampir tidak ada bedanya dengan JSON.

## Perbedaan noSQL dan sql **[belum lengkap]**

### Apa itu sql

Sql adalah Structured Query Language yaitu bahasa khusus yang digunakan untuk membuat dan mengolah database,syntax dari sql ini cukup mudah dipahami oleh manusia , contohnya SELECT FROM,WHERE,dll

### Apa itu noSQL

NoSQL bukan merupakan bahasa.NoSQL adalah database yang tidak memiliki perintah SQL dan konsep penyimpananya semistruktural atau tidak stuktural dan tidak harus memiliki relasi layaknya table table Mysql

Empat mekanisme dalam NoSQL

- Table-oriented, contoh: Google dengan Big Table dan juga Facebook dengan Cassandra,
- Graph-oriented,
- Document-oriented database, contoh: MongoDB dan juga CouchDB,
- Key-value store, contoh: Memcache dan Redis

### Perbedaan besar dari sql dan non sql

	SQL	NoSQL
Query	+	-
Transaksi	+	-
Konsistensi	+	-
Scalability	-	+
Management	-	+
Schema	-	+

## **Cara install MongoDB**

**DI WINDOWS :**

## **Bab 2 : Pengenalan & crud mongoDB**

Didalam bab 2: pengenalan MongoDB anda akan mempelajari :

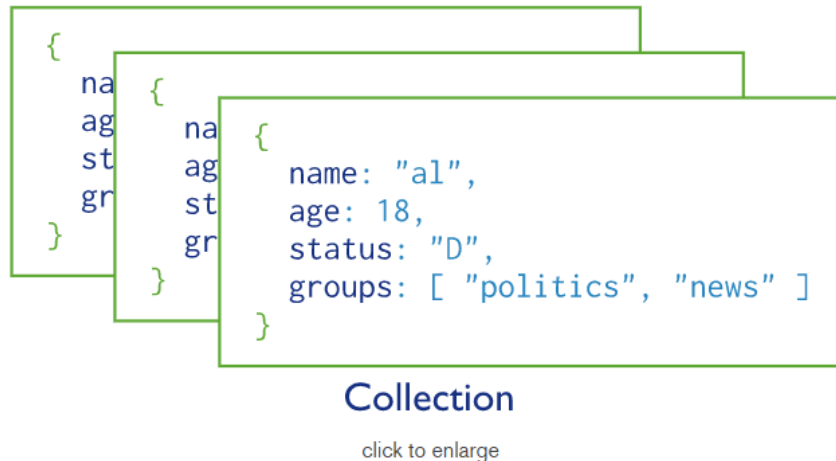
### **Memahami Database ,collections & documents ?**

#### **Database**

Database adalah container / wadah untuk menyimpan collection

#### **Collection**

Collection adalah tempat penyimpanan document jika dianalogikan ke database sql , collection itu seperti table. Coba perhatikan gambar berikut



Gambar itu menjelaskan , Collection itu seperti container dari document /data/

## Document

MongoDB Menyimpan data sebagai bson document ,jika dianalogikan ke database sql , document itu seperti record record dari tabel.

```
{
  name: "sue",
  age: 26,
  status: "A",
  groups: [ "news", "sports" ]
}
```

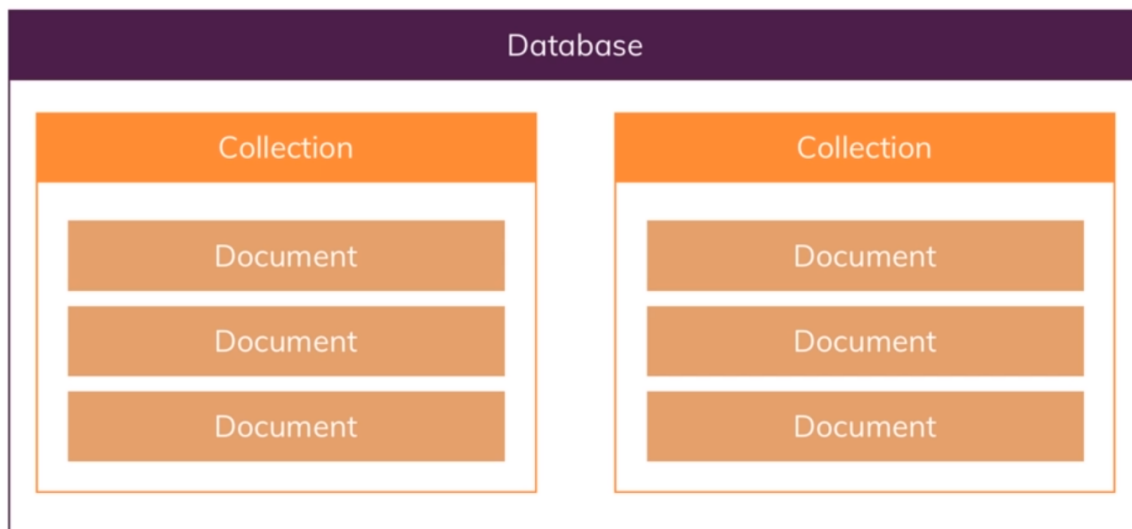
← field: value

← field: value

← field: value

← field: value

Jika belum mengerti bisa lihat gambar ini



Digambar tersebut menjelaskan Database menampung collection dan collection menampung documents

## Membuat Database dan collection

Membuat database dan collection di mongoDB Sangatlah mudah,kita langsung praktek saja.

Saya anggap kalian sudah menginstall Mongo dan setting path nya , jika belum kembali ke bab 1 bagian cara install mongoDB.

## Mengakses Server MongoDB

Silahkan kalian buka terminal atau shell , untuk mengakses server mongoDB Sangat mudah Kalian hanya cukup ketik perintah **Mongo**

Perintah Mongo Berfungsi untuk membuka shell MongoDB.

Silahkan kalian ketik perintah ini di terminal kalian.

```
mongo
```

Kalau berhasil seharusnya muncul seperti ini, jika sudah begini kalian sukses masuk ke mongoDB Shell

```
C:\Users\asus>mongo
MongoDB shell version v4.2.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("2fd4dad4-42dc-429c-aef0-688e6682a505") }
MongoDB server version: 4.2.6
Server has startup warnings:
2020-07-02T09:37:39.568+0700 I CONTROL [initandlisten]
2020-07-02T09:37:39.568+0700 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-07-02T09:37:39.568+0700 I CONTROL [initandlisten] ** Read and write access to data and configuration is
unrestricted.
2020-07-02T09:37:39.568+0700 I CONTROL [initandlisten]
---
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()
---
>
```

Unuk cek nya coba kalian ketik perintah ini

```
show dbs
```

maka akan memunculkan list database , jika kalian belum pernah pakai mongoDB seharusnya secara default ada 3 database bawaan dari mongoDB

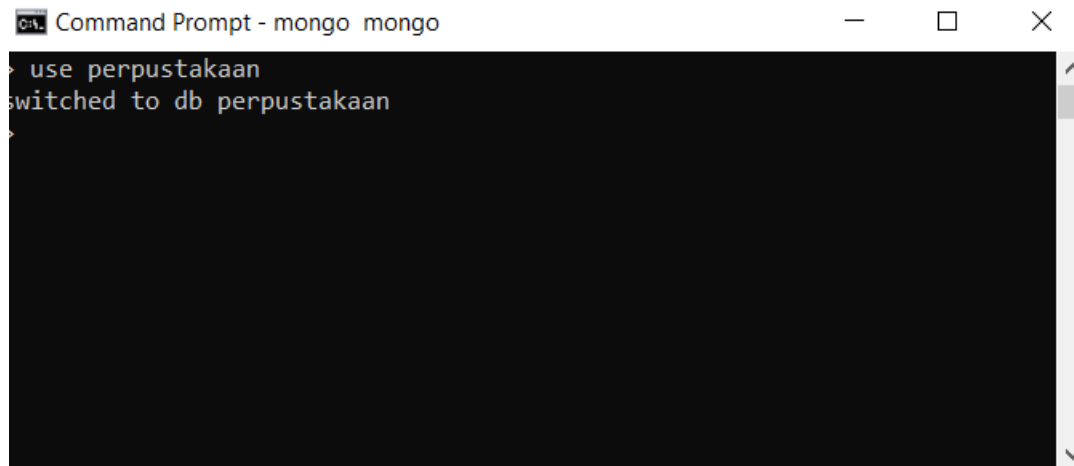
**Untuk keluar dari mongo Server tekan CTRL + C**

## Membuat Database

Untuk membuat database sangatlah simple kalian hanya perlu mengetikkan perintah

```
use namaDatabase
```

disini saya akan mencoba membuat database perpustakaan



```
Command Prompt - mongo mongo
> use perpustakaan
switched to db perpustakaan
>
```

sangatlah Mudah bukan ? tetapi jika kalian ketikkan perintah show dbs ,database yang telah dibuat tidak ada di list atau tidak tersimpan , itu karena di mongoDB kita wajib membuat **collection** terlebih dahulu

## Membuat Collection

Membuat collections bisa dibuat dengan perintah :

```
db.createCollection("nama_collections")
```

disini saya mencoba membuat collection buku

```
Command Prompt - mongo mongo
> db.createCollections("nama_collections")
2020-07-03T19:34:02.491+0700 E QUERY [js] uncaught exception: SyntaxError: illegal character :
@ (shell):1:21
> db.createCollections("nama_koleksi")
2020-07-03T19:34:32.359+0700 E QUERY [js] uncaught exception: TypeError: db.createCollections is not a function :
@ (shell):1:1
> use perpustakaan
switched to db perpustakaan
> db.createCollection("buku")
{ "ok" : 1 }
```

Gimana sangat mudah bukan membuat database dan collection di mongo ? oke selanjut kita akan membahas CRUD ,tapi sebelum itu kita akan membahas lebih detail tentang JSON

## Memahami JSON

Sekarang saya akan membahas singkat tentang **JSON**, pembahasan ini hanya singkat saja yah jika ingin lebih detail tentang json bisa kunjungi link [ini](#)

**JSON(Javascript Object Nation)** adalah format pertukaran data yang ringan , mudah dibaca dan ditulis oleh manusia , serta mudah diterjemahkan dan dibuat(generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa pemrograman javascript, tetapi JSON dapat digunakan di semua bahasa pemrograman seperti C,C++,C#,Java ,Javascript,Perl,Python,dll.Oleh karena sifat tersebut menjadikan JSON ideal sebagai bahasa pertukaran-data.

Sebenarnya JSON bukan hanya untuk pertukaran tetapi bisa untuk :

- JSON sebagai format untuk bertukar data client dan server atau antar aplikasi, Contoh Restful API
- JSON sebagai tempat menyimpan data ,seperti database MongoDB
- JSON digunakan untuk menyimpan konfigurasi project , contoh : file composer.json pada php dan package.json pada Node js
- JSON digunakan untuk menyimpan konfigurasi dan penyimpan data pada hugo
- JSON digunakan untuk menyimpan konfigurasi project pada Node js
- JSON digunakan untuk menyimpan data manifest;
- dan masih banyak lagi

Pada point 2 “JSON sebagai tempat menyimpan data seperti MongoDB” yah struktur data MongoDB itu seperti JSON tetapi lebih tepatnya lagi MongoDB menggunakan BSON.

Sekarang kita lihat contoh struktur dari JSON

```
{
  "name" : "Tegar akmal",
  "age" : 17,
}
```

Itu adalah stuktur sederhana dari JSON

Berikut penjelasan tentang stuktur JSON :

- {} = json selalu dimulai dengan tanda kurung kurawal dan ditutup dengan kurung }.
- Key = key itu ialah kunci dari value , jika di sturuktur diatas key adalah “name”,”age”
- Value = ialah isi dari key ,jika di sturuktur diatas value adalah “tegar akmal”
- JSON selalu menggunakan kata kutip di key dan value
- JSON selalu menggunakan koma untuk memisahkan data yang lebih dari Satu

Berikut contoh lain dari JSON

```
{
  "name" : "Tegar akmal",
  "age" : 17,
  "hobi" : ["coding","tidur","main games"]
}
```

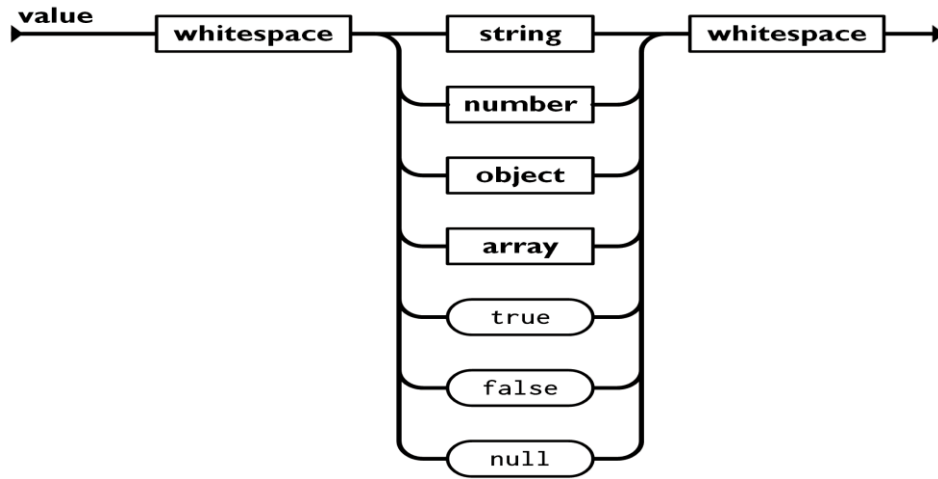
Value dari JSON bisa berupa array juga

```
{
  "name" : "Tegar akmal",
  "age" : 17,
  "hobi" : ["coding","tidur","main games"],
  "socialMedia" : {
    "facebook" : "muhammad Tegar Akmal",
    "github" : "tegar97"
  }
}
```

Valuenya juga bisa bersarang seperti contoh diatas .



Berikut ialah type data yang didukung JSON



## JSON VS BSON

**/- MASIH RISET -/**

## CREATE READ,UPDATE DELETE (CRUD) MONGO DB

Sekarang kita akan membahas tentang CRUD dari MongoDB, Method CRUD di MongoDB sangat simpel disbanding sql.

Pastikan kalian telah terhubung dengan MongoShell dan telah masuk ke database, didalam kasus ini saya akan memakai database perpustakaan

Untuk masuk kedatabase cukup dengan **use namaDatabase**

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

## CREATE

Yang pertama saya akan membahas create terlebih dahulu. Create berfungsi untuk memasukan data ke database.

Di MongoDB Untuk memasukan data ada 3 method

Tips : “Untuk mempermudah menulis codenya kalian bisa pakai code editor seperti visual studio code, sublime ,dll lalu codenya copas ke terminal ”

### A.InsertOne()

Berfungsi untuk memasukan **1 data saja** .

Untuk penggunaan InsertOne seperti ini

```
db.namaCollection.insertOne({key : "value"})
```

disini saya akan membuat collection dengan nama “buku” dan isi dari collection nya adalah nama, pengarang, harga

```
db.buku.insertOne({
  name : "become master php",
  author : "asep",
  price : 100
})
```

Maka jika berhasil akan muncul seperti ini

```
> db.buku.insertOne({
...   name : "become master php",
...   author : "asep",
...   price : 100
... })
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5efff066c6c4eed8bc932218")
}
>
```

## B.InsertMany()

Jika tadi hanya bisa memasukan 1 data saja insertMany bisa memasukan data lebih dari 1.

Untuk penggunaan insertMany hampir sama dengan insertOne yang membedakan hanya ada array sebelum object / ( {} )

```
db.namaCollection.insertMany([ {key : "value"} ])
```

Disini saya akan memasukan 3 buku maka codenya jadi seperti :

```
db.buku.insertMany([
  {
    name : "cara membuat nasi goreng",
    author : "bacdim",
    price : 200
  },
  {
    name : "become master mongoDB",
    author : "tegar",
    price : 0
  },
  {
    name : "become master node js",
    author : "tegar",
    price : 24
  }
])
```

Maka kalau berhasil akan muncul seperti ini

```
db.buku.insertMany([
-   {
-     name : "cara membuat nasi goreng",
-     author : "bacdim",
-     price : 200
-   },
-   {
-     name : "become master mongoDB",
-     author : "tegar",
-     price : 0
-   },
-   {
-     name : "become master node js",
-     author : "tegar",
-     price : 24
-   }
- ])
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("5efff6f3c6c4eed8bc93221f"),
    ObjectId("5efff6f3c6c4eed8bc932220"),
    ObjectId("5efff6f3c6c4eed8bc932221")
  ]
}
```

Peraturan :

1. Karena ini bson maka key tidak perlu menggunakan tanda kutip (" ")
2. Untuk key yang bertype **string wajib menggunakan tanda kutip** sedangkan yang bertype integer , bollean ,dll tidak perlu