

Table of Contents

1. [Pengantar](#)
2. [NoSQL dan MongoDB](#)
3. [Instalasi](#)
 - i. [OS X](#)
 - ii. [Linux](#)
 - iii. [Windows](#)
4. [Menjalankan MongoDB](#)
5. [Menggunakan MongoDB](#)
6. Query
7. Collection
8. Cursor
9. Replication
10. Sharding
11. GridFS
12. Optimasi
13. Keamanan
14. Tips & Trik

Pengantar

Buku ini akan mengajarkan Anda dasar-dasar dari MongoDB. Apakah Anda seorang yang berpengalaman atau tidak, buku ini ditujukan untuk semua orang yang ingin belajar tentang MongoDB document database, keluarga NoSQL.

Butuh bantuan?

Anda dapat mengajukan pertanyaan atau hal lainnya (yang berhubungan dengan buku ini) ke w.kristories@gmail.com.

Untuk informasi tentang NoSQL, Anda dapat mengikuti akun Twitter dibawah ini.



Lisensi

MongoDB Untuk Indonesia by [Wahyu Kristianto](#) is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).



NoSQL dan MongoDB

- [NoSQL](#)
- [MongoDB](#)

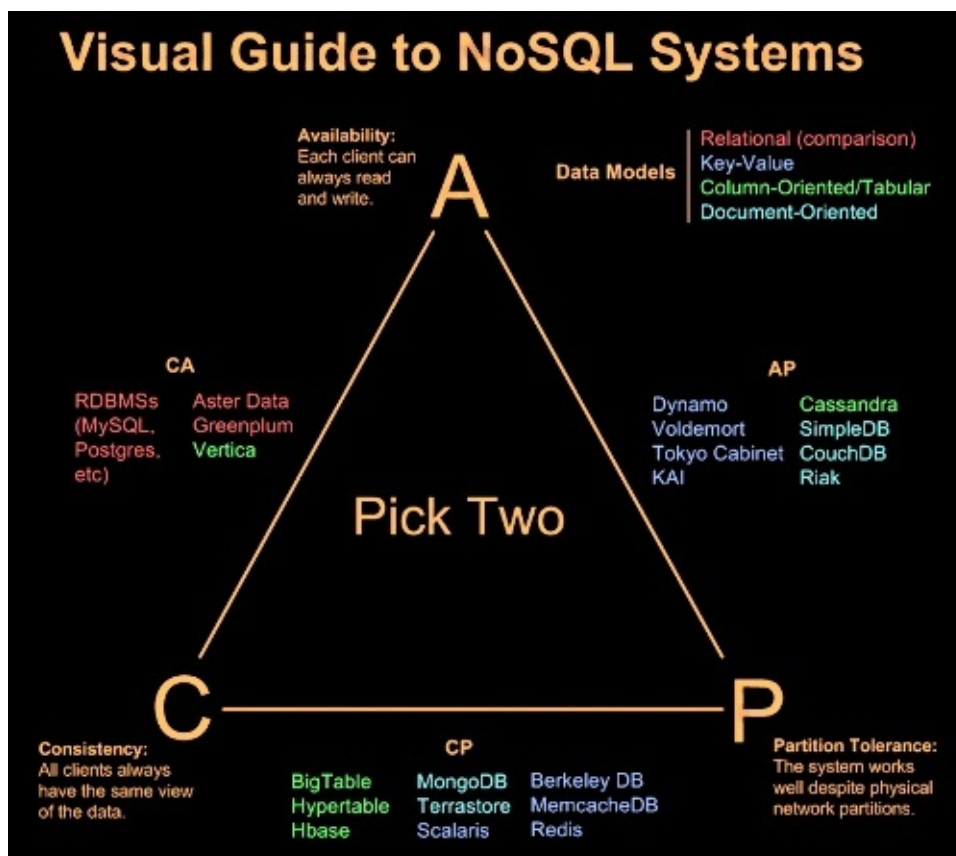
NoSQL

NoSQL (sering ditafsirkan sebagai "tidak hanya SQL") adalah kelas yang luas dari sistem manajemen basis data yang diidentifikasi dengan tidak mematuhi aturan pada model sistem manajemen basis data relasional yang banyak digunakan. NoSQL tidak dibangun terutama dengan table dan umumnya tidak menggunakan SQL untuk memanipulasi data.

Teorema CAP

Teorema CAP menyatakan bahwa, tidak mungkin untuk sebuah sistem komputer terdistribusi secara bersamaan, memberikan ketiga jaminan sebagai berikut :

- **Consistency** - Semua node melihat data yang sama dalam waktu yang bersamaan.
- **Availability** - Jaminan pada setiap permintaan, menerima tanggapan tentang apakah itu berhasil atau tidak.
- **Partition tolerance** - Sistem terus beroperasi meski sistem yang lain gagal.



MongoDB

MongoDB merupakan open-source document database dari keluarga NoSQL, yang memberikan kinerja tinggi, high availability, dan automatic scaling.

Document

Data pada MongoDB dikenal sebagai document. MongoDB menggunakan [BSON](#) serialization sebagai format penyimpanan data untuk document, mirip dengan JSON.

Contoh document

```
{
  "_id": ObjectId("549081be0dbcf82140041a7"),
  "nama": "Wahyu Kristianto",
  "nomor": NumberLong(12345),
  "bahasa": [
    "bahasa",
    "english"
  ]
}
```

Terminologi dan Konsep

Tabel berikut menyajikan terminologi berbagai SQL dan MongoDB.

SQL	MongoDB
database	database
table	collection
row	document
column	field
index	index
table joins	embedded documents dan linking
primary key	primary key

Contoh Penggunaan

SQL

```
MySQL
SELECT *
FROM pengguna
WHERE status = "aktif"
OR nomor = 50
```

MongoDB

```
db.pengguna.find(
{
  $or: [
    { status: "aktif" },
    { nomor: 50 }
  ]
})
```

Instalasi

Untuk proses instalasi, pilih sistem operasi yang digunakan.

- [OS X](#)
- [Linux](#)
- [Windows](#)

Instalasi - OS X

Ada dua cara instalasi pada OS X, yaitu :

- Via [Homerebrew](#)
- Manual

Via Homebrew

Instalasi MongoDB production release

```
brew install mongodb
```

Instalasi dengan dukungan SSL, dapat menambahkan `--with-openssl`

```
brew install mongodb --with-openssl
```

Instalasi MongoDB development release, dapat menambahkan `--devel`

```
brew install mongodb --devel
```

Manual

Unduh file dari <https://www.mongodb.org/downloads> . Contoh yang digunakan saat ini adalah MongoDB versi 2.6.6 / 64-bit.

```
curl -O http://downloads.mongodb.org/osx/mongodb-osx-x86_64-2.6.6.tgz
```

Ekstrak file

```
tar -zxvf mongodb-osx-x86_64-2.6.6.tgz
```

Ubah nama folder `mongodb-osx-x86_64-2.6.6` menjadi `mongodb` .

```
mv mongodb-osx-x86_64-2.6.6 mongodb
```

Tambahkan MongoDB pada [PATH environment variable](#)

```
export PATH=<FOLDER MONGODB>/bin:$PATH
```

Instalasi - Linux

Pada instalasi linux ini, menggunakan distro turunan dari Debian, yaitu Ubuntu. Ada dua cara dalam proses instalasi, yaitu :

- Via package manager (apt-get)
- Manual

Via Package Manager (apt-get)

Impor publik key yang digunakan oleh package management system.

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
```

Buat list file untuk MongoDB dengan nama `mongodb.list` pada direktori `/etc/apt/sources.list.d` dengan menggunakan perintah berikut :

```
echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | sudo tee /etc/apt/sources.list.d/mongo
```

Reload lokal package dan Install MongoDB

```
sudo apt-get update
sudo apt-get install mongodb-org
```

Via Manual

Unduh file dari <https://www.mongodb.org/downloads>. Contoh yang digunakan saat ini adalah MongoDB versi 2.6.6 / 64-bit.

```
wget https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-2.6.6.tgz
```

Ekstrak file.

```
tar -zxvf mongodb-linux-x86_64-2.6.6.tgz
```

Ubah nama folder `mongodb-linux-x86_64-2.6.6` menjadi `mongodb` .

```
mv mongodb-linux-x86_64-2.6.6 mongodb
```

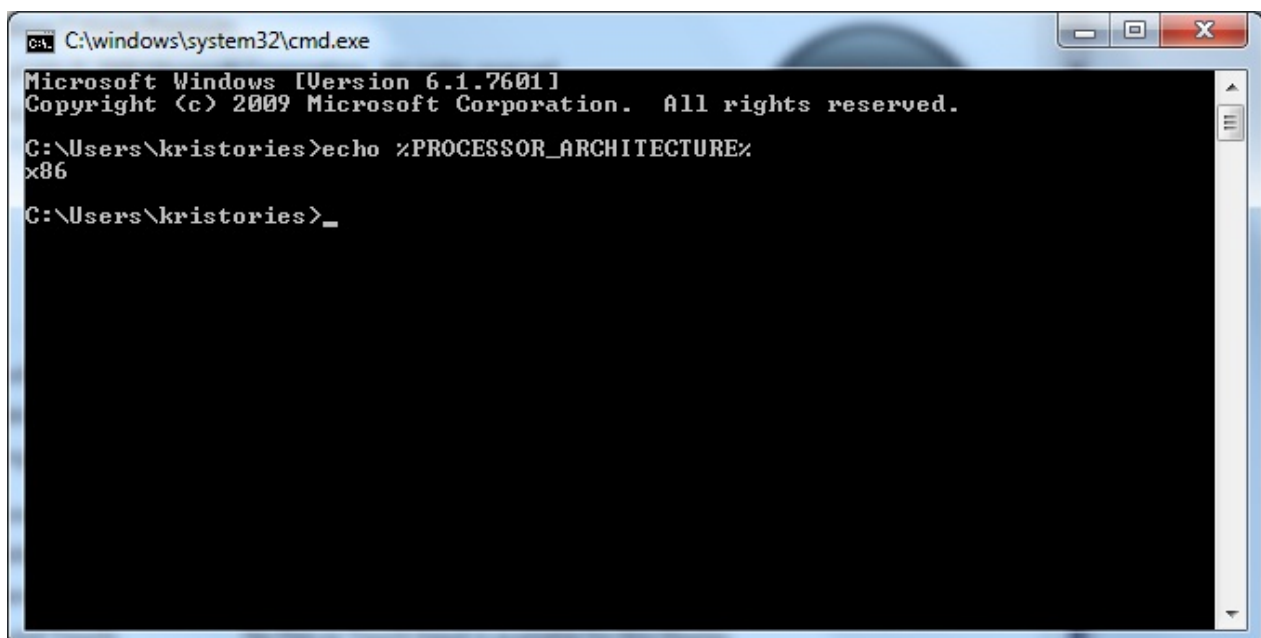
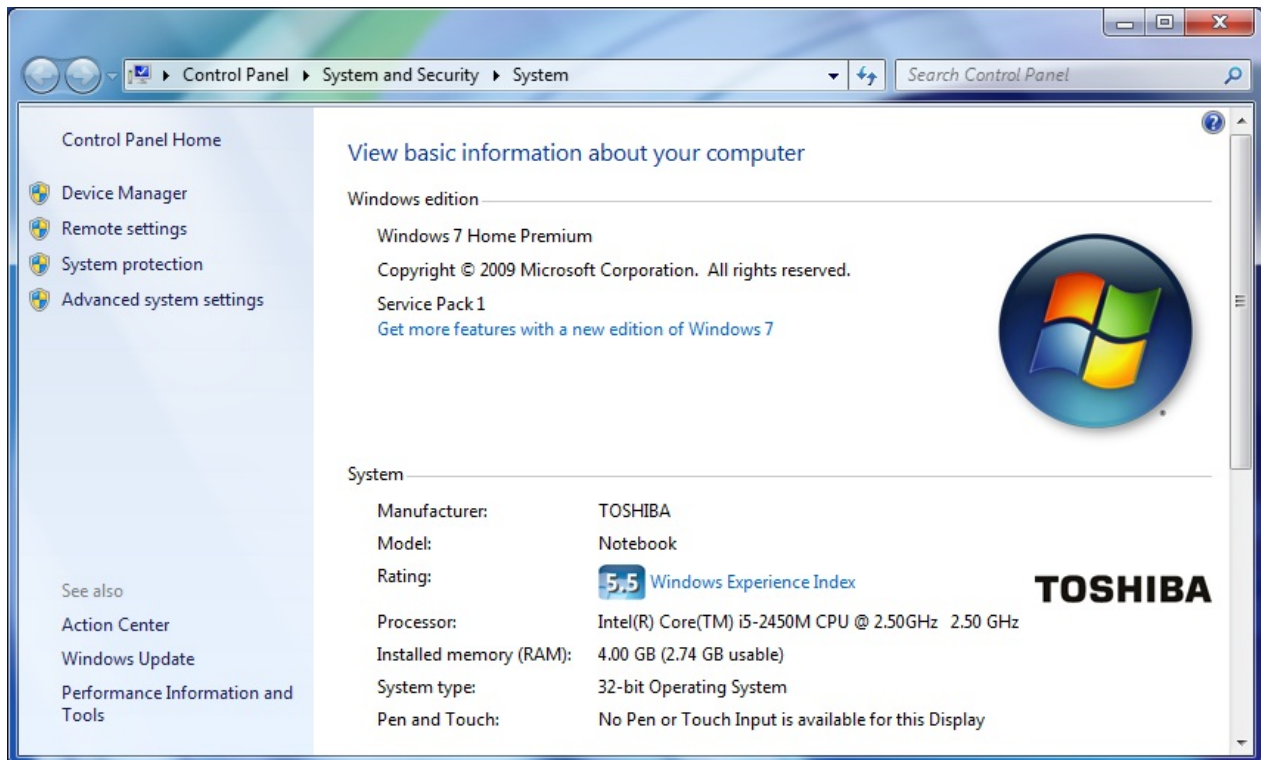
Tambahkan MongoDB pada [PATH environment variable](#)

```
export PATH=<FOLDER_MONGODB>/bin:$PATH
```


Lebih baik menggunakan package manager, karena dengan cara ini, kita sudah termasuk mendapatkan control script, yaitu sebuah script untuk mengatur MongoDB agar dapat berjalan sebagai sebuah daemon, termasuk pengontrolan untuk start & stop.

Instalasi - Windows

Pastikan Anda mengetahui versi Windows yang dipakai. Anda dapat mengetahuinya melalui **Control Panel / System and Security / System** atau melalui Command Prompt `%PROCESSOR_ARCHITECTURE%`.



- Unduh MongoDB dari <https://www.mongodb.org/downloads> sesuai dengan versi Windows yang digunakan.
- Klik dua kali pada file yang sudah diunduh, lalu ikuti panduan instalasinya.
- Ubah nama folder MongoDB.

```
cd \  
move C:\mongodb-win32-* C:\mongodb
```


Menjalankan MongoDB

`mongod` adalah daemon utama untuk sistem MongoDB. Tugas `mongod` adalah menangani semua permintaan dan pengelolaan data.

`mongod` membutuhkan direktori khusus untuk menyimpan data file. Secara default, direktori untuk `mongod` berada di `/data/db`, tetapi Anda dapat menentukan direktori sendiri untuk `mongod`.

Setelah menentukan direktori `mongod`, ubah permission direktori tersebut agar dapat membaca dan menulis.

Menjalankan `mongod`.

```
mongod
```

Jika menginginkan menjalankan MongoDB pada direktori yang berbeda, dapat menambahkan `--dbpath`.

```
mongod --dbpath /folder/yang/akan/dipakai
```

Menggunakan MongoDB

Pada bab ini akan diulas tentang menggunakan MongoDB dasar seperti :

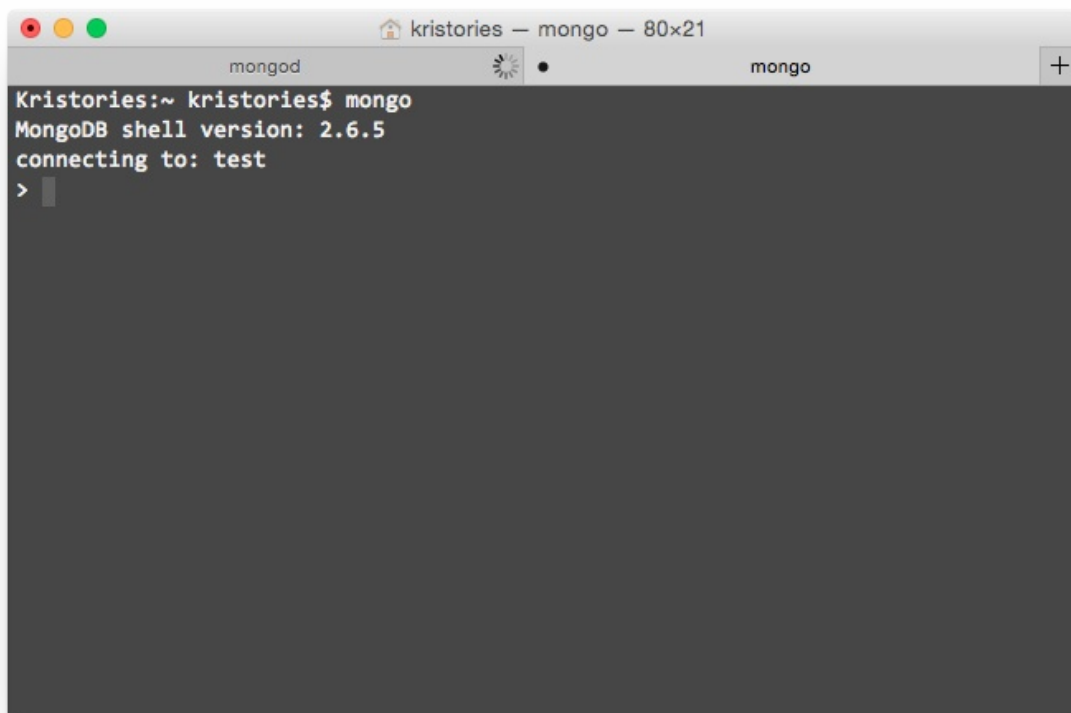
- Menjalankan `mongo` ,
- Database
- Collection
- Document
 - Menambahkan Document
 - Mencari Document
 - Mengubah Document
 - Menghapus Document

Menjalankan `mongo`

Anda dapat menggunakan MongoDB dengan menjalankan `mongo` .

```
mongo
```

Secara otomatis, `mongo` akan menggunakan database **test**.



Database

Dengan MongoDB, Anda tidak perlu repot dengan membuat database terlebih dahulu. Anda dapat langsung menggunakan database yang Anda inginkan, sekalipun database tersebut belum dibuat.

Contoh berikut adalah menggunakan database **indonesia**.

```
> use indonesia
```

Collection

Pengelompokan dokumen di MongoDB disebut collection. Jika Anda pernah menggunakan SQL sebelumnya, collection sama halnya dengan table. Anda dapat melihat Terminologi dan Konsep pada [halaman MongoDB](#).

MongoDB adalah schemaless!

Artinya, MongoDB memiliki skema yang fleksibel. Layaknya folder, Anda dapat menempatkan apapun pada folder tersebut, seperti dokumen, gambar, lagu, atau apapun yang Anda inginkan. Anda tidak dituntut untuk menentukan struktur data pada saat pembuatan collection, dan dapat menempatkan document dengan struktur yang berbeda-beda.

Contoh membuat collection.

```
db.createCollection('provinsi')
```

Document

Document akan sering Anda jumpai. Disinilah data yang akan Anda kelola.

Menambahkan Document

Dibawah ini adalah contoh untuk menambahkan document baru pada collection "**provinsi**".

```
db.provinsi.insert({ nama: "DKI Jakarta" })
```

Anda dapat menambahkan document lainnya dengan struktur yang berbeda.

```
db.provinsi.insert({ nama: "Aceh", ibukota: "Banda Aceh" })
```

```

Kristories:~ kristories$ mongo
MongoDB shell version: 2.6.5
connecting to: test
> use indonesia
switched to db indonesia
> db.createCollection('provinsi')
{ "ok" : 1 }
> db.provinsi.insert({ nama: "DKI Jakarta" })
WriteResult({ "nInserted" : 1 })
> db.provinsi.insert({ nama: "Aceh", ibukota: "Banda Aceh" })
WriteResult({ "nInserted" : 1 })
> db.provinsi.find()
{ "_id" : ObjectId("549450b6911932b76c02dc0c"), "nama" : "DKI Jakarta" }
{ "_id" : ObjectId("549450be911932b76c02dc0d"), "nama" : "Aceh", "ibukota" : "Banda Aceh" }
>

```

Mencari Document

Setelah Anda menambahkan document pada collection, kali ini Anda akan dipandu untuk mencari document yang telah Anda buat sebelumnya.

Mencari seluruh document.

```
db.provinsi.find()
```

Jika ingin mencari document berdasarkan kriteria tertentu, dapat ditambahkan kriteria pada parameter pertama.

```
db.provinsi.find({ "nama": "DKI Jakarta" })
```

Mengubah Document

Data yang sudah dibuat, dapat diubah kembali. Ada dua jenis pengubahan data, yaitu :

- Mengubah spesifik field pada document,
- Mengganti document.

Mengubah spesifik field pada document

`$set` dapat digunakan untuk mengubah hanya pada spesifik field.

Contoh berikut adalah untuk mengubah field "**nama**" dari "**DKI Jakarta**" ke "**Jakarta**".

```

db.provinsi.update(
  { nama: "DKI Jakarta" },
  {
    $set: {

```



```

        nama: "Jakarta"
    }
}
)

```

Mengganti document

Anda juga dapat mengganti document yang sudah ada dengan document baru. Perbedaanannya adalah, seluruh field yang ada pada document yang sudah ada, akan diganti dengan seluruh field yang baru.

```

db.provinsi.update(
  { nama: "Jakarta" },
  { ibukota: "Jawa Barat" }
)

```

Menghapus Document

Berikut adalah contoh untuk menghapus seluruh document yang mempunyai field "**nama**" bernilai "**Aceh**".

```

db.provinsi.remove({ nama: "Aceh" })

```

Jika menghapus hanya satu dokumen saja, dapat menambahkan parameter kedua dengan nilai `boolean` .

```

db.provinsi.remove({ nama: "Aceh" }, 1)

```

Untuk menghapus seluruh document, dapat menghilangkan kriteria.

```

db.provinsi.remove({})

```