

Basis data NoSQL dan MongoDB

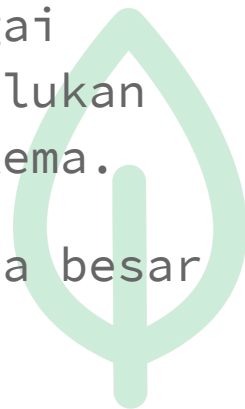
Ahmad Rio Adriansyah S.Si. M.Si.

Apa itu NoSQL

NoSQL adalah sebuah basis data yang dirancang untuk dapat mengakomodasi berbagai macam model data, termasuk key-value, dokumen, kolom, dan graph format.

NoSQL kepanjangan dari “Not Only SQL” digunakan sebagai alternatif terhadap basis data tradisional yang memerlukan relasi dimana data terletak dalam sebuah tabel dan skema.

NoSQL sangat baik digunakan untuk pengerjaan data-data besar yang terdistribusi.



Key-Value

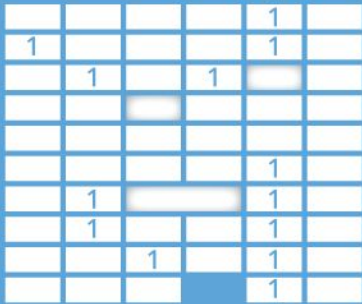


Graph DB

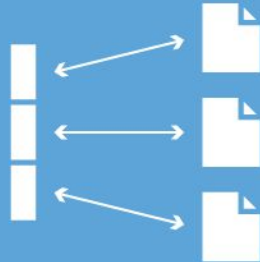


Gambaran ruang dari NoSQL. Kuadran berwarna biru merupakan kumpulan data tersimpan (aggregate store)

Column Family



Document



Mengapa NoSQL?

NoSQL digunakan untuk menyelesaikan tiga masalah ini :

- DATA VOLUME
- DATA VELOCITY
- DATA VARIETY



DATA VOLUME

Mengetahui data yang perlu ditampung semakin besar. Dengan demikian data menjadi sulit untuk disimpan menggunakan basis data yang terelasikan. Waktu eksekusi *query* meningkat dengan bertambahnya ukuran tabel dan jumlah JOIN.

NoSQL memiliki beberapa alternatif dalam menyelesaikan permasalahan model relasi.



DATA VELOCITY



Tidak hanya ukuran data yang besar, data - data tersebut juga berubah dengan cepat.

Beban dalam penulisan data dan proses yang cukup memakan waktu dengan adanya model relasi pada tiap data menyebabkan biaya operasional yang tinggi.

NoSQL mencoba menangani permasalahan ini dengan model data yang lebih fleksibel.



DATA VARIETY

Saat ini data memiliki keragaman format yang tidak mampu ditangani oleh relasi basis data.

NoSQL dirancang untuk mengatasi keragaman data dan secara fleksibel mencoba memenuhi kebutuhan data di masa mendatang.



MongoDB

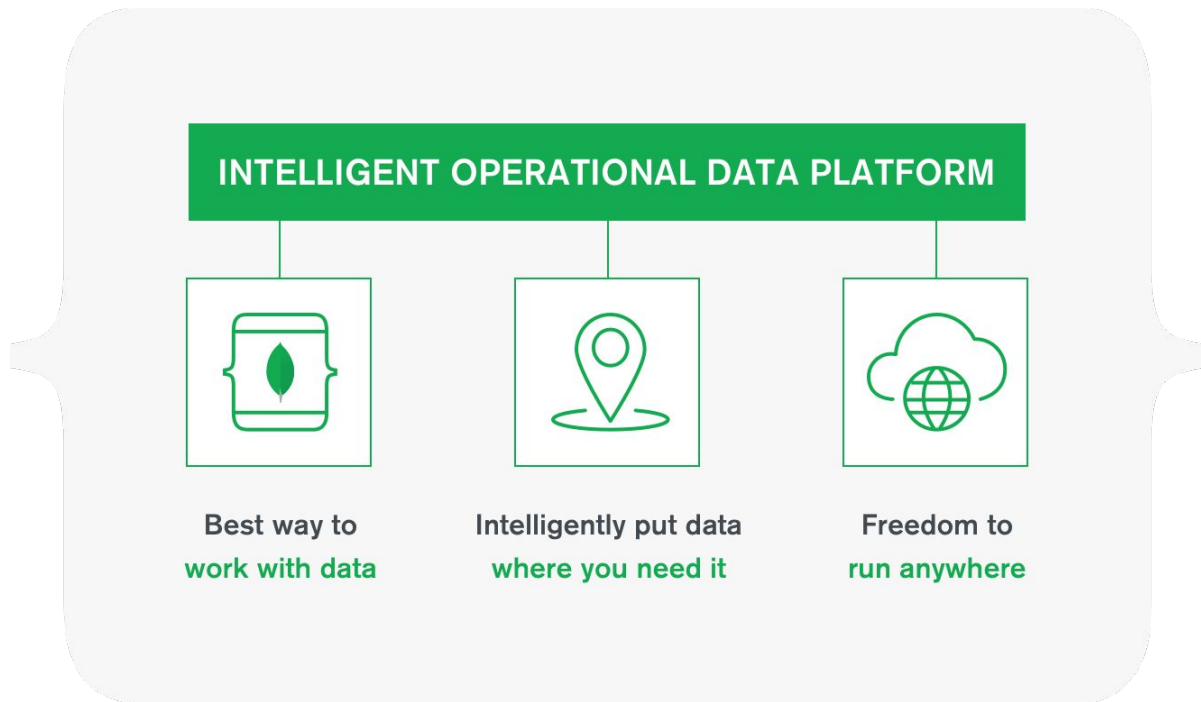
Apa itu MongoDB

- MongoDB adalah sebuah dokumen basis data dengan skalabilitas dan fleksibilitas seperti yang diharapkan dengan kueri dan pengindeksan sesuai kebutuhan.
- MongoDB menyimpan data secara fleksibel, dokumen JSON-like.
- Inti dari MongoDB adalah sebuah basis data yang terdistribusi.
- Model terpetakan dengan sebuah objek dalam kode aplikasi yang anda buat.



Arsitektur MongoDB

— — —



Instalasi

— — —

Dalam terminal, ikuti perintah berikut untuk menginstall MongoDB

```
sudo apt install mongodb
```



Menjalankan MongoDB

— — —

Jalankan service mongod lebih dahulu :

```
sudo service mongod start
```

Pastikan MongoDB berjalan, bukalah file log dan pastikan sistem membaca [initandlisten] waiting for connections on port 27017

```
sudo gedit /var/log/mongodb/mongod.log
```



Menjalankan MongoDB

Untuk mematikan proses service MongoDB gunakan perintah :

```
sudo service mongod stop
```

Untuk menjalankan ulang proses service :

```
sudo service mongod restart
```

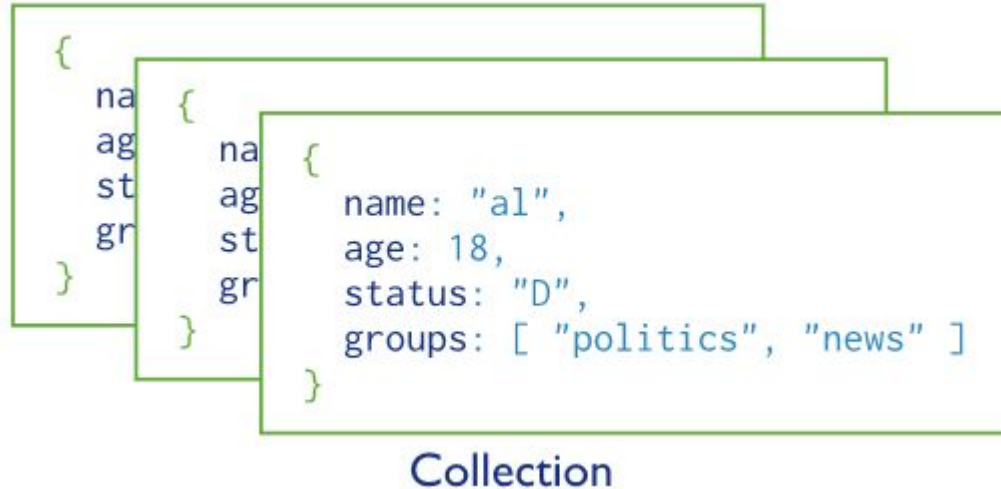
Mulai menggunakan aplikasi MongoDB :

```
mongo
```



Struktur MongoDB

Mongodb menyimpan dokumen dalam collection (analog dengan tabel pada RDBMS) dan meletakkan collection tersebut dalam database.



Mengoperasikan MongoDB

Setelah masuk dalam shell mongo, lakukan perintah berikut untuk melihat daftar seluruh database yang ada

```
show dbs
```

Gunakan perintah berikut untuk melihat daftar collection pada database yang aktif :

```
show collections
```



Mengoperasikan MongoDB

Setelah masuk dalam shell mongo, lakukan perintah berikut untuk mengetahui basis data yang digunakan saat ini :

```
use db
```

Gunakan perintah berikut untuk pindah ke basis data lainnya :

```
use nama_basis_data
```

Gunakan database “toko” untuk saat ini. Tidak seperti program sql lainnya, anda tidak perlu membuat basis data baru sebelum berpindah, perintah tersebut secara otomatis akan mempersiapkan basis data yang anda inginkan.



Mengoperasikan MongoDB

Menyiapkan *collection* untuk menyimpan dokumen :

```
db.nama_collection.insertOne()
```

Gunakan perintah berikut untuk menginput data lebih dari satu :

```
db.nama_collection.insertMany()
```

Contoh :

```
db.inventaris.insertMany([  
    { produk : "pensil", qty : 25, harga : 1500, status: "tersedia" },  
    { produk : "buku", qty : 30, harga : 2000, status: "tersedia" }  
]);
```



Mengoperasikan MongoDB

Setelah menginput data, gunakan perintah berikut untuk melihat isi *collection*:

```
db.inventaris.find({})
```

Tambahkan fungsi `pretty()` agar terlihat lebih rapi.

```
db.inventaris.find({}).pretty()
```

berikut perintah untuk mencari data secara spesifik mengandung sesuatu :

```
db.inventaris.find({qty : 25})
```



Mengoperasikan MongoDB

Melakukan update pada sebuah nilai pada data.

```
db.inventaris.updateOne({ produk : "pensil" }, { $set: { qty : 0 } })
```

Untuk mengupdate semua data dengan ketentuan qty = 0 :

```
db.inventaris.updateMany({ qty : 0 }, { $set: { status: "habis" } })
```

Menghapus data :

```
db.inventaris.deleteOne({ produk : "pensil" })
```

Manghapus banyak data :

```
db.inventaris.deleteMany({ qty: { $lt: 10 } })
```



Query dalam mongoDB

Memilih seluruh dokumen :

```
db.inventaris.find({})
```

Memanggil secara spesifik, contoh produk dengan harga Rp. 1.500 :

```
db.invetaris.find({harga: 1500})
```

Memanggil dengan sebuah kondisi :

```
db.invetaris.find({harga: {$in: [1500,200] }})
```

Spesifik dengan kondisi AND :

```
db.inventaris.find({status: "tersedia", qty: {$lt: 30}})
```



Query dalam mongoDB

Spesifik dengan pengkondisian OR :

```
db.inventaris.find({ $or: [ { status: "tersedia" }, { qty: { $lt: 30 } } ] })
```

Spesifik, OR, dan AND , barang diawali huruf p lebih dari 20, status habis :

```
db.invetaris.find({status: "habis", $or: [ { qty: { $gt: 20 }}, { produk: /^p/ } ] })
```



Import Data from JSON

gunakan perintah mongoimport untuk memulai mengambil data dari JSON :

```
mongoimport -d <nama basis data> -c <nama collection>  
--type json --file <nama file>
```



Memasukkan Data accounts.json

Pada folder Dataset, terdapat file accounts.json yang berisi 2000 records

Masukkan data tersebut dalam mongo db dengan

```
mongoimport -d bank -c account --type json --file  
accounts.json
```



MongoDB dengan python

Pertama, install pymongo menggunakan pip

```
pip install pymongo
```

Setelah selesai, kemudian pindah shell python :

```
import pymongo
```

Cek versi pymongo yang telah terinstal :

```
pymongo.__version__
```



MongoDB dengan python : Database dan Collection

Buatla sebuah file dengan nama mongo_test.py kemudian koneksikan dengan mongoDB :

```
from pymongo import MongoClient  
  
client = MongoClient("mongodb://localhost:27017/")  
  
db = client['toko']  
  
collection = db['inventaris']
```



MongoDB dengan python : Insert One

Menambah satu data dalam collection.

```
detail_barang = {  
    'produk' : 'pensil',  
    'qty' : 25,  
    'harga' : 1500,  
    'status' : 'tersedia'  
}  
  
hasil = collection.insert_one(detail_barang)
```



MongoDB dengan python : Insert multiple

```
detail_barang_1 = {  
    'produk' : 'penghapus',  
    'qty' : 35,  
    'harga' : 500,  
    'status' : "tersedia"  
}
```

```
detail_barang_2 = {  
    'produk' : 'spidol',  
    'qty' : 10,  
    'harga' : 1000,  
    'status' : "tersedia"  
}
```

```
detail_barang_3 = {  
    'produk' : 'rautan',  
    'qty' : 20,  
    'harga' : 500,  
    'status' : "tersedia"  
}
```

```
hasil_banyak = collection.insert_many[detail_barang_1, detail_barang_2, detail_barang_3]
```



MongoDB dengan python : Memanggil data

Untuk memanggil data gunakan method **find_one()** untuk satu data dan method **find()** untuk seluruh data, disesuaikan dengan parameter yang diminta.

```
collection.find_one({'produk' : 'pensil'})
```

atau

```
collection.find_one({'qty' : { $gt : 20 }})
```



MongoDB dengan python : Memanggil data

Untuk memanggil data dengan bantuan method **sort()** akan mempermudah tampilan data dengan menyesuaikan urutan.

```
collection.find().sort('produk')
```

atau gunakan nilai -1 untuk mengurutkan dari bawah

```
collection.find().sort('produk',-1)
```



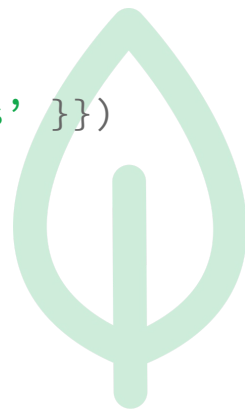
MongoDB dengan python : Update data

Dalam melakukan perubahan data terdapat dua method yang bisa digunakan yaitu `update_one()` untuk satuan dan `update_many()` lebih dari satu data.

```
collection.update_one({ 'produk' : 'pensil' },{ $set: {'qty': 0 }})
```

atau

```
collection.update_many({ 'qty' : 0 },{ $set: { 'status' : 'habis' } })
```



MongoDB dengan python : Menghapus data

Untuk menghapus data terdapat dua method yaitu menghapus satuan **delete_one()** dan sekaligus **delete_many()**.

```
collection.delete_one({'produk' : 'pensil'})
```

atau

```
collection.delete_many({'qty': {'$lt' : 20}})
```

