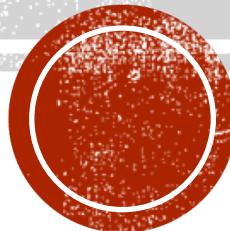


# **NOSQL DAN MONGODB**

Sandi Andrian, S. Pd, MBA



# TENTANG SAYA

## Sandi Andrian, S. Pd., MBA.

- *Studio Head Bandung di Mirum Jakarta*
- *Business Mentor di SBM ITB*
- *Founder eduxa.id*

*Reach me at:*

- [andrian.sandi@gmail.com](mailto:andrian.sandi@gmail.com)
- <https://github.com/andriansandi>
- <https://facebook.com/andriansandi>
- <https://linkedin.com/in/andriansandi>



# APA ITU NOSQL

Database yang tidak menggunakan relasi antar tabel dan tidak menyimpan data dalam format tabel kaku (kolom yang fix) seperti layaknya Relasional Database.



# TIPE NOSQL

## Key-Value

Setiap record data yang disimpan hanya berupa **key** dan **value**.

Tipe ini biasanya digunakan untuk *caching*.

Kelebihannya adalah **read** dan **write** sangat cepat.



## Document Oriented

Setiap record data yang disimpan berupa dokumen-dokumen.

Setiap dokumen satu dengan dokumen lainnya tidak harus mempunyai struktur yang sama walaupun itu dalam satu kumpulan data (dalam terminologi SQL disebut **table**)



## Graph Database

Database ini didesain untuk relasi yang direpresentasikan seperti *graph*.

Terdiri dari **node** dan **edge**.



## Column Family

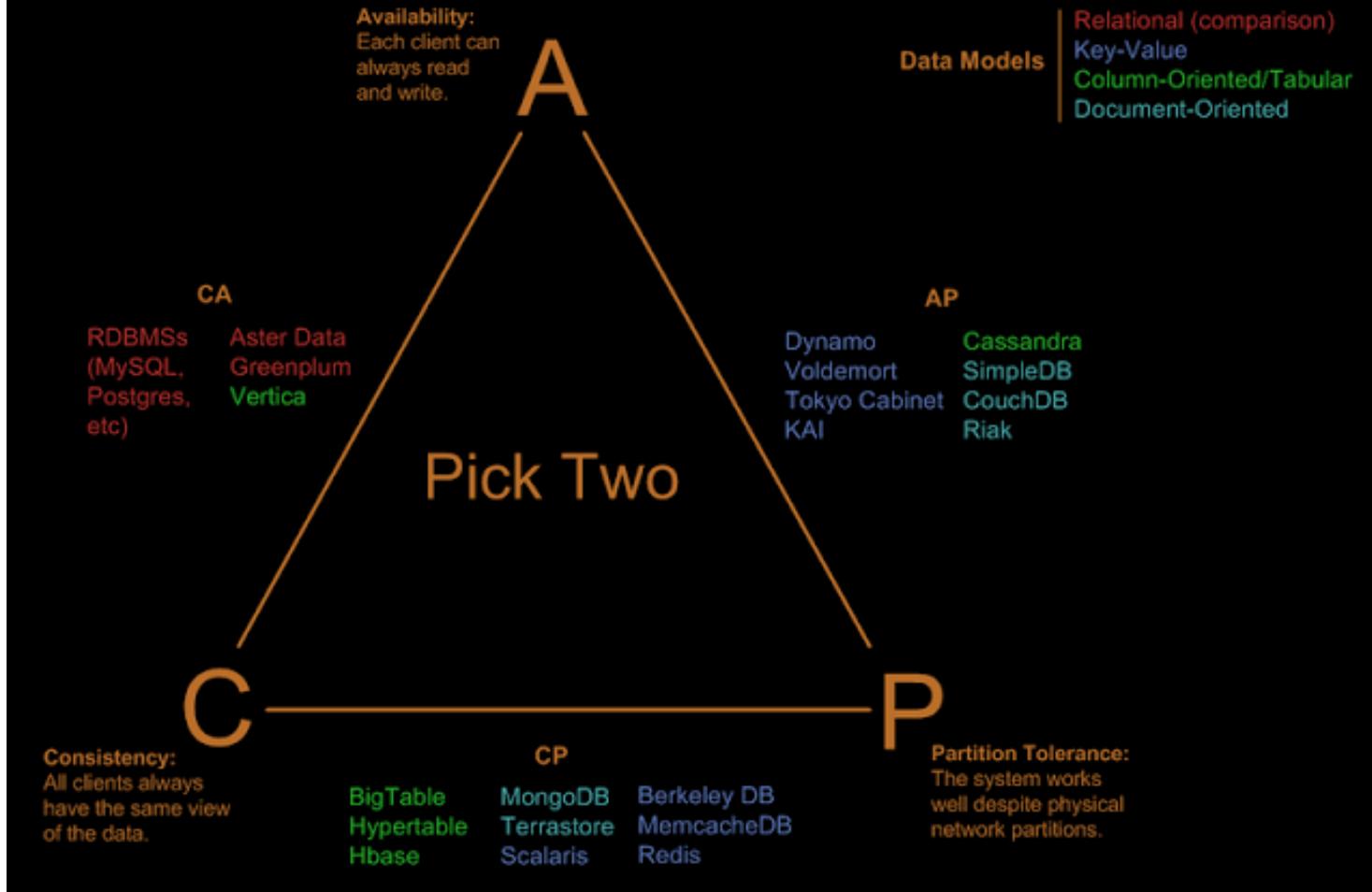
Column database adalah penyimpanan data dengan model seperti **column**. Database ini menggunakan konsep dengan istilah **keyspace**.

Sebuah **keyspace** seperti **schema** dalam SQL. Keyspace berisi **column families** (seperti **tabel** dalam SQL) terdiri dari **rows**, dan **column**.

Kelebihannya adalah **compression data**, **aggregation queries** (SUM, COUNT, AVG, dll), **scalability**, **read** dan **write** sangat cepat.



# Visual Guide to NoSQL Systems



# KEUNTUNGAN NOSQL

- Elastic Scaling
  - RDBMS, semakin tinggi load maka kapasitas server harus ditingkatkan
  - NoSQL relatif membutuhkan *resources* yang lebih kecil
- Flexible Data Models
  - NoSQL tidak membutuhkan struktur seperti RDBMS sehingga lebih fleksibel
- Big Data
  - NoSQL didesain untuk kebutuhan Big Data
- Economics
  - Biaya lebih rendah daripada RDBMS

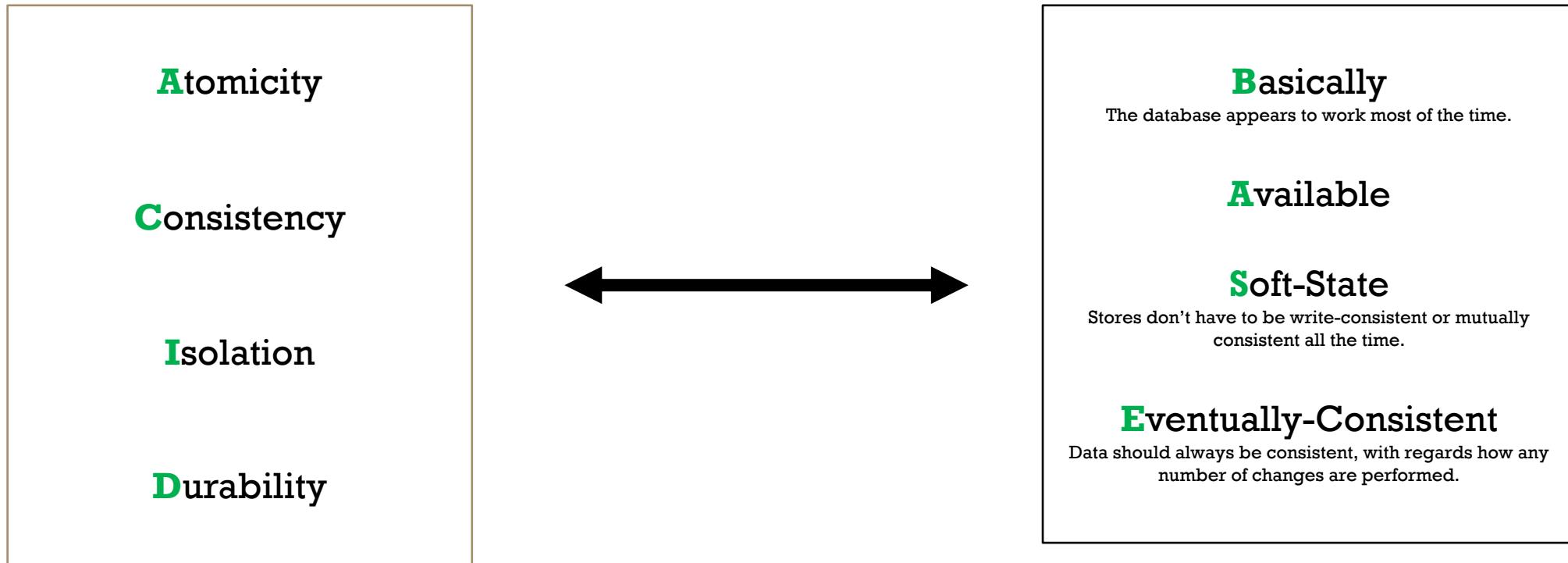


# KEKURANGAN NOSQL

- Support
  - Beberapa NoSQL yang berbasis *open source* masih mengandalkan komunitas
- Lack of Expertise
  - NoSQL merupakan teknologi baru di ranah *database*, sehingga *expert* masih terbatas



# RDB ACID TO NOSQL BASE



# MONGODB



MongoDB adalah salah satu jenis database NoSQL yang berbasis dokumen dengan format JSON.



# FAKTA MONGODB

- Didirikan oleh Dwight Merriman, Eliot Horowitz dan Kevin Ryan pada tahun 2007 – tim DoubleClick.
- Ditulis dengan bahasa C++
- Mendukung banyak bahasa pemrograman, seperti: **JavaScript, Python, Ruby, Perl, Java, Java Scala, C#, C++, Haskell, Erlang.**

```
{  
    "_id" : ObjectId("54c955492b7c8eb21818bd09"),  
    "alamat" : {  
        "street" : "2 Avenue",  
        "zipcode" : "10075",  
        "building" : "1480",  
        "coord" : [ -73.9557413, 40.7720266 ]  
    },  
    "borough" : "Manhattan",  
    "cuisine" : "Italian",  
    "grades" : [  
        {  
            "date" : ISODate("2014-10-01T00:00:00Z"),  
            "grade" : "A",  
            "score" : 11  
        },  
        {  
            "date" : ISODate("2014-01-16T00:00:00Z"),  
            "grade" : "B",  
            "score" : 17  
        }  
    ],  
    "name" : "Vella",  
    "restaurant_id" : "41704620"  
}
```



# TERMINOLOGI MONGODB

- **Database**

Merupakan kumpulan dari *collections*. Setiap database mempunya satu kumpulan file yang disimpan dalam sistem computer. Satu server MongoDB dapat memiliki banyak database.

- **Collection**

Merupakan kumpulan dari dokumen di MongoDB, serupa dengan table di RDBMS.

Collection tidak membutuhkan schema sehingga memungkinkan document untuk memiliki collection dengan field yang berbeda.

- **Document**

Sebuah document di MongoDB terdiri dari satu set key-value dan mempunya schema yang dinamis sehingga strukturnya akan berbeda satu sama lain tergantung kebutuhan.



# TERMINOLOGI RDBMS VS MONGODB

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by mongodb itself)
Database Server and Client	
Mysqld/Oracle	mongod
mysql/sqlplus	mongo



# TERMINOLOGI MONGODB

- Mongod, merupakan sebuah *instance* dari MongoDB
- Mongos, *sharding process*
  - Dianalogikan sebagai router ke database / koneksi seperti **mysql\_connect** pada PHP & MySQL.
  - Melakukan pemrosesan pada semua request yang diterima.
  - Menentukan berapa banyak dan *instance* mana yang akan menerima *query*.
  - *Mongos* melakukan populasi hasil query dan mengirimkannya kembali kepada *client*.
- Mongo, sebuah *interactive shell*. Biasa juga disebut *client* MongoDB
  - Berbasis JavaScript



# KEUNTUNGAN MONGODB

- Schema-less
- Document-based Database
- No complex joins
- Query language via API
- Mudah untuk di-scaling
- Master-slave replication with automated failover (replica sets)



# KENAPA MENGGUNAKAN MONGODB

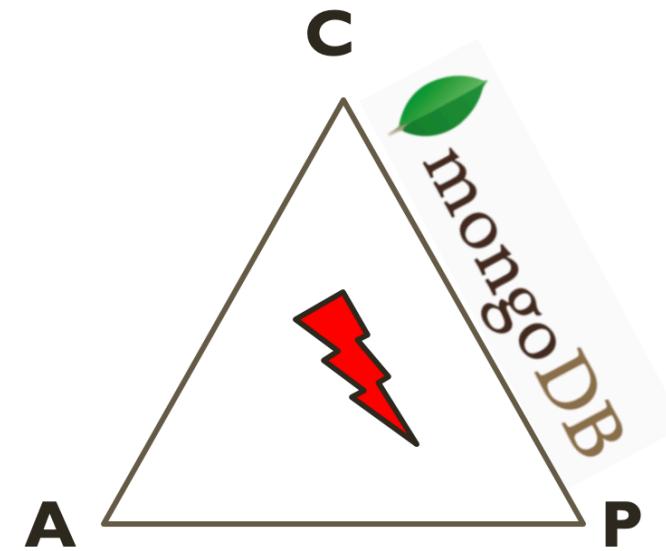
- Query-nya sederhana.
- Fungsionalitas yang ditawarkan memenuhi kebutuhan untuk sebagian besar aplikasi berbasis web.
- Integrasi data mudah dan cepat.
- Tidak membutuhkan ERD.
- Tidak cocok untuk system transaksi yang kompleks.



# PENDEKATAN CAP MONGODB

MongoDB focus pada **Consistency** dan **Partition Tolerance**

- Consistency
  - Semua replica akan mempunya data yang identik
- Availability
  - Sistem akan tetap beroperasi pada *failing nodes*.
- Partition tolerance
  - multiple entry points
  - system remains operational on system split



# SCHEMA-LESS

- MongoDB tidak perlu mendefinisikan *schema* terlebih dahulu
- Setiap document pada collection dapat memiliki data yang berbeda

```
{  
  "_id": 1,  
  "name": { "first": "John", "last": "Backus" },  
  "contribs": [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ],  
  "awards": [  
    {  
      "award": "W.W. McDowell Award",  
      "year": 1967,  
      "by": "IEEE Computer Society"  
    }, {  
      "award": "Draper Prize",  
      "year": 1993,  
      "by": "National Academy of Engineering"  
    }  
  ]  
}
```

```
{  
  "_id": 1,  
  "name": { "first": "John", "last": "Backus" },  
  "contribs": [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ],  
}
```



# JSON FORMAT

- Data is in name / value pairs
- A name/value pair consists of a field name followed by a colon, followed by a value:
  - Example: "name": "R2-D2"
- Data is separated by commas
  - Example: "name": "R2-D2", race : "Droid"
- Curly braces hold objects
  - Example: {"name": "R2-D2", race : "Droid", affiliation: "rebels"}
- An array is stored in brackets []
  - Example [ {"name": "R2-D2", race : "Droid", affiliation: "rebels"}, {"name": "Yoda", affiliation: "rebels"} ]



# TIPE DATA MONGODB

- **String**

- **Integer**

Dapat menyimpan 32bit atau 64bit tergantung dari arsitektur server.

- **Boolean**

- **Double**

- **Min/Max keys**

- This type is used to compare a value against the lowest and highest BSON elements.

- **Arrays**

- **Timestamp**

- **Object**

- Tipe data yang dapat menyimpan document dari collection MongoDB.

- **Null**

- **Symbol**

- Sering digunakan untuk Bahasa non-latin seperti Tagalog atau Kanji misalnya.

- **Date**

- **Object ID**

- This datatype is used to store the document's ID.

- **Binary data**

- **Code**

- This datatype is used to store javascript code into document.

- **Regular expression**



# INSTALASI MONGODB DI MACOS

1. Pastikan di mac kamu mempunyai Homebrew **brew**, jika belum silahkan untuk melakukan instalasi [Homebrew terlebih dahulu](#).
2. Instalasi Xcode
3. Buka terminal di MacOS
4. Ketikkan command sebagai berikut **brew tap mongodb/brew**
5. Lalu instalasi MongoDB di MacOS dengan perintah **brew install [mongodb-community@4.2](#)**

Walkthrough lebih lanjut bisa dilihat di

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-os-x/>



# INSTALASI MONGODB DI UBUNTU

- Import public key yang akan digunakan oleh *packet management system ubuntu*  
`wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key add -`
- Tambahkan sumber packet dari MongoDB di *sources.list* Ubuntu  
`echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.2.list`
- Reload database packet management system Ubuntu  
`sudo apt-get update`
- Install MongoDB  
`sudo apt-get install -y mongodb-org`



# INSTALASI MONGODB DI WINDOWS

1. Unduh **MongoDB Community Edition**
2. Jalankan installer MongoDB
3. Ikuti langkah instalasi MongoDB

Walkthrough lebih lanjut bisa dilihat di <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/>



# CRUD OPERATIONS

- **Create**

- `db.collection.insert()`
- `db.collection.save()`
- `db.collection.update(,,{ upsert: true })`

- **Read**

- `db.collection.find(,)`
- `db.collection.findOne(,)`

- **Update**

- `db.collection.update(,,)`

- **Delete**

- `db.collection.remove(,)`



# CREATE OPERATIONS

`Db.collection` specifies the collection or the ‘table’ to store the document

- `db.collection_name.insert()`
  - Omit the `_id` field to have MongoDB generate a unique key
  - Example `db.parts.insert( {{type: “screwdriver”, quantity: 15 } })`
  - `db.parts.insert({_id: 10, type: “hammer”, quantity: 1 })`
- `db.collection_name.update( , , { upsert: true } )`
  - Will update 1 or more records in a collection satisfying query
- `db.collection_name.save()`
  - Updates an existing record or creates a new record



# READ OPERATIONS

- `db.collection.find( , ).cursor modified`
  - Provides functionality similar to the SELECT command
  - `where condition , fields in result set`
  - Example: `var PartsCursor = db.parts.find({parts: "hammer"}).limit(5)`
  - Has cursors to handle a result set
  - Can modify the query to impose limits, skips, and sort orders.
  - Can specify to return the 'top' number of records from the result set
- `db.collection.findOne( , )`



# UPDATE OPERATIONS

- `db.collection_name.insert()`
  - Omit the `_id` field to have MongoDB generate a unique key
  - Example `db.parts.insert( {type: "screwdriver", quantity: 15} )`
  - `db.parts.insert({_id: 10, type: "hammer", quantity: 1})`
- `db.collection_name.save()`
  - Updates an existing record or creates a new record
- `db.collection_name.update( , , { upsert: true } )`
  - Will update 1 or more records in a collection satisfying query
- `db.collection_name.findAndModify( , , , , )`
  - Modify existing record(s) – retrieve old or new version of the record



# DELETE OPERATIONS

- `db.collection_name.remove(, )`
  - Delete all records from a collection or matching a criterion
  - specifies to delete only 1 record matching the criterion
  - Example: `db.parts.remove(type: /h/ } )` - remove all parts starting with h
- `Db.parts.remove()` – delete all documents in the parts collections



# CRUD EXAMPLE

```
> db.user.insert({  
    first: "John",  
    last : "Doe",  
    age: 39  
})
```

```
> db.user.update({  
    "_id" : ObjectId("51"),  
    {  
        $set: {  
            age: 40,  
            salary: 7000  
        }  
    })
```

```
> db.user.find () {  
    "_id" : ObjectId("51"),  
    "first" : "John",  
    "last" : "Doe",  
    "age" : 39
```

```
}
```

```
> db.user.remove({ "first": /J/ })
```



# OPERATOR QUERY

## Comparison

Name	Description
<code>\$eq</code>	Matches values that are equal to a specified value.
<code>\$gt</code>	Matches values that are greater than a specified value.
<code>\$gte</code>	Matches values that are greater than or equal to a specified value.
<code>\$in</code>	Matches any of the values specified in an array.
<code>\$lt</code>	Matches values that are less than a specified value.
<code>\$lte</code>	Matches values that are less than or equal to a specified value.
<code>\$ne</code>	Matches all values that are not equal to a specified value.
<code>\$nin</code>	Matches none of the values specified in an array.

## Logical

Name	Description
<code>\$and</code>	Joins query clauses with a logical <b>AND</b> returns all documents that match the conditions of both clauses.
<code>\$not</code>	Inverts the effect of a query expression and returns documents that do <i>not</i> match the query expression.
<code>\$nor</code>	Joins query clauses with a logical <b>NOR</b> returns all documents that fail to match both clauses.
<code>\$or</code>	Joins query clauses with a logical <b>OR</b> returns all documents that match the conditions of either clause.



# OPERATOR QUERY

## Element

Name	Description
<code>\$exists</code>	Matches documents that have the specified field.
<code>\$type</code>	Selects documents if a field is of the specified type.

## Array

Name	Description
<code>\$all</code>	Matches arrays that contain all elements specified in the query.
<code>\$elemMatch</code>	Selects documents if element in the array field matches all the specified <code>\$elemMatch</code> conditions.
<code>\$size</code>	Selects documents if the array field is a specified size.



# MON

## Cloud Provider & Region

Choose your preferred cloud provider and the region nearest to clients

AWS, N. Virginia (us-east-1) >

Select a cloud provider to see its region availability.



Configure a **free tier cluster** by first selecting a region labeled with **FREE TIER AVAILABLE** then choose the M0 option in the Cluster Tier below.

★ recommended region

### NORTH AMERICA

N. Virginia (us-east-1) ★

**FREE TIER AVAILABLE**

Ohio (us-west-1) ★

N. California (us-west-1)

Oregon (us-west-2) ★

Montreal (ca-central-1)

### EUROPE

Ireland (eu-west-1) ★

London (eu-west-2)

Frankfurt (eu-central-1) ★

**FREE TIER AVAILABLE**

São Paulo (sa-east-1)

### ASIA

Tokyo (ap-northeast-1)

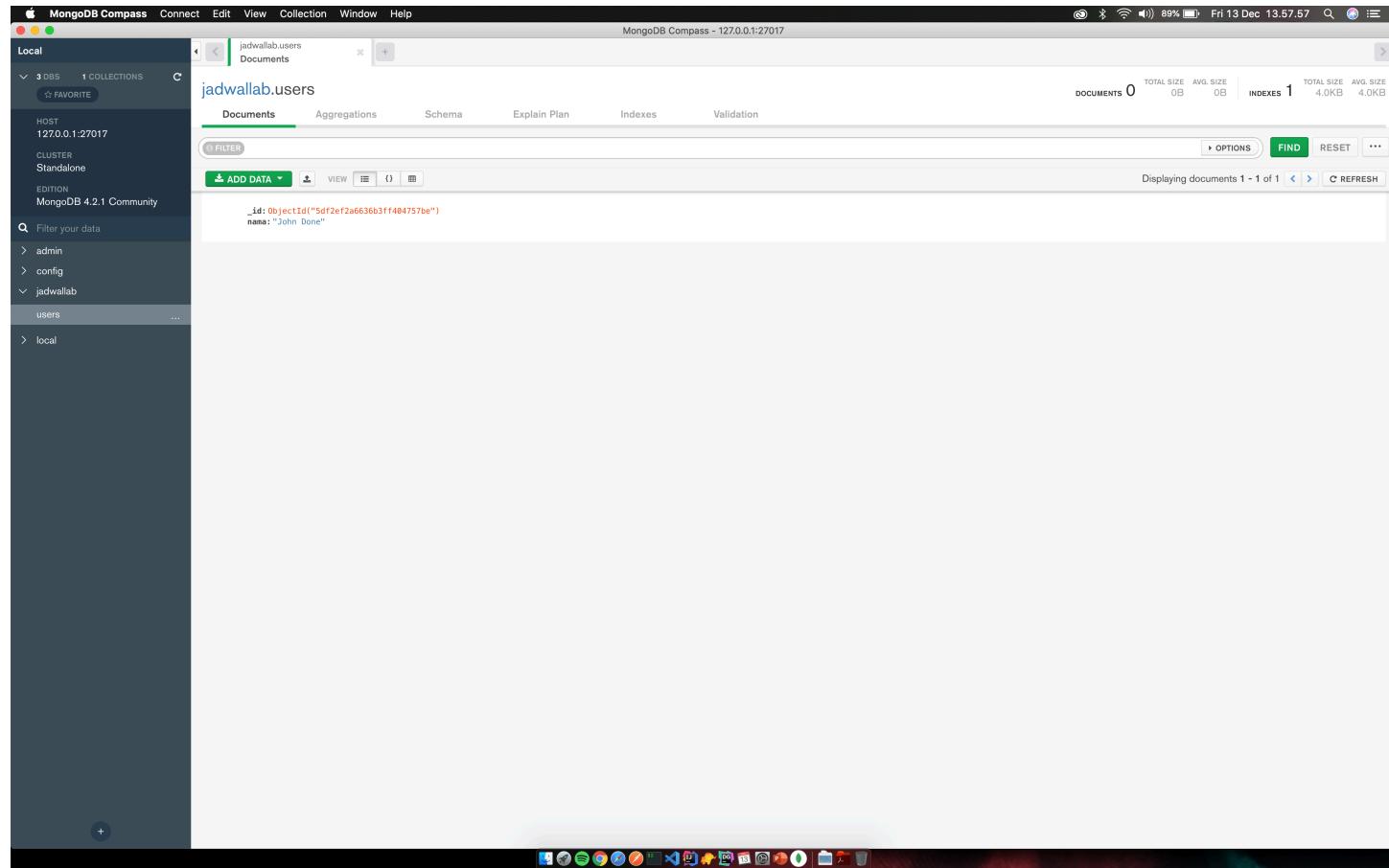
Seoul (ap-northeast-2)

Singapore (ap-southeast-1)

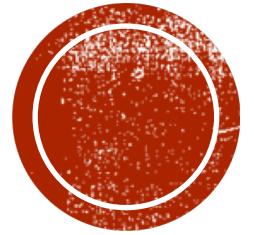
Mumbai (ap-south-1)

### SOUTH AMERICA

# MONGODB COMPASS



- Sebuah GUI layaknya PHPMyAdmin untuk melakukan administrasi data di MongoDB
- Gratis
- Unduh di <https://www.mongodb.com/download-center/compass>



# Q & A