

# Database MongoDB

Workshop Pemrograman WEB

*Yunia Ikawati*

Teknik Informatika-PENS



# Overview



MongoDB



Komponen MongoDB



Install MongoDB



Membuat Database dengan MongoDB



# Tujuan

1. Memahami konsep database NoSQL menggunakan MongoDB
2. Membuat database dan collection
3. Menggunakan query untuk pencarian data
4. Menggunakan agregasi sederhana
5. Menggunakan **MongoDB Compass** untuk mengelola database
6. Memahami struktur data **BSON (Binary JSON)**

# MONGO DB

---

MongoDB merupakan salah satu jenis database yang berbasis **dokumen**, dirancang untuk menyimpan, memproses, dan mengelola data yang tidak terstruktur dan digunakan untuk mengembangkan website berjenis **NoSQL**.

---

MongoDB menggunakan dokumen dengan format **JSON(javascript object notation)** sedangkan SQL menggunakan relasi tabel.

---

MongoDB dapat memahami bahasa pemrograman yang populer, seperti **C, C++, C#, .Net, Go, Java, Node.js, Perl, PHP, Python, Motor, Ruby, Scala, Swift, dan Mongoid**.

# KOMPONEN DALAM SISTEM DATABASE MONGODB

## 1. Database

Merupakan kumpulan koleksi (collections). Setiap database terpisah dan digunakan untuk mengatur data berdasarkan aplikasi atau tujuan tertentu.

## 2. Collection

Seperti tabel dalam basis data relasional, tetapi lebih fleksibel karena tidak memiliki skema tetap. Koleksi menyimpan kumpulan dokumen.

## 3. Document

Unit data terkecil dalam MongoDB yang berisi pasangan key-value. Dokumen disimpan dalam format JSON atau BSON.

## 4. Fields

Elemen individu dari pasangan key-value dalam dokumen. Fields adalah elemen fleksibel yang bisa berisi tipe data apa pun.

Contoh Dokumen:

Json

```
{
  "_id": "BK001",
  "title": "Belajar MongoDB",
  "author": "John Doe",
  "year": 2021,
  "available": true
}
```

# Ciri Utama MongoDB

## 1. Schema-less

Tidak harus memiliki struktur yang sama antar data.

## 2. Berbasis Dokumen

Data disimpan seperti format JSON sehingga mudah dibaca.

## 3. Cepat dan Skalabel

Cocok untuk aplikasi besar dengan banyak pengguna.

## 4. Open Source

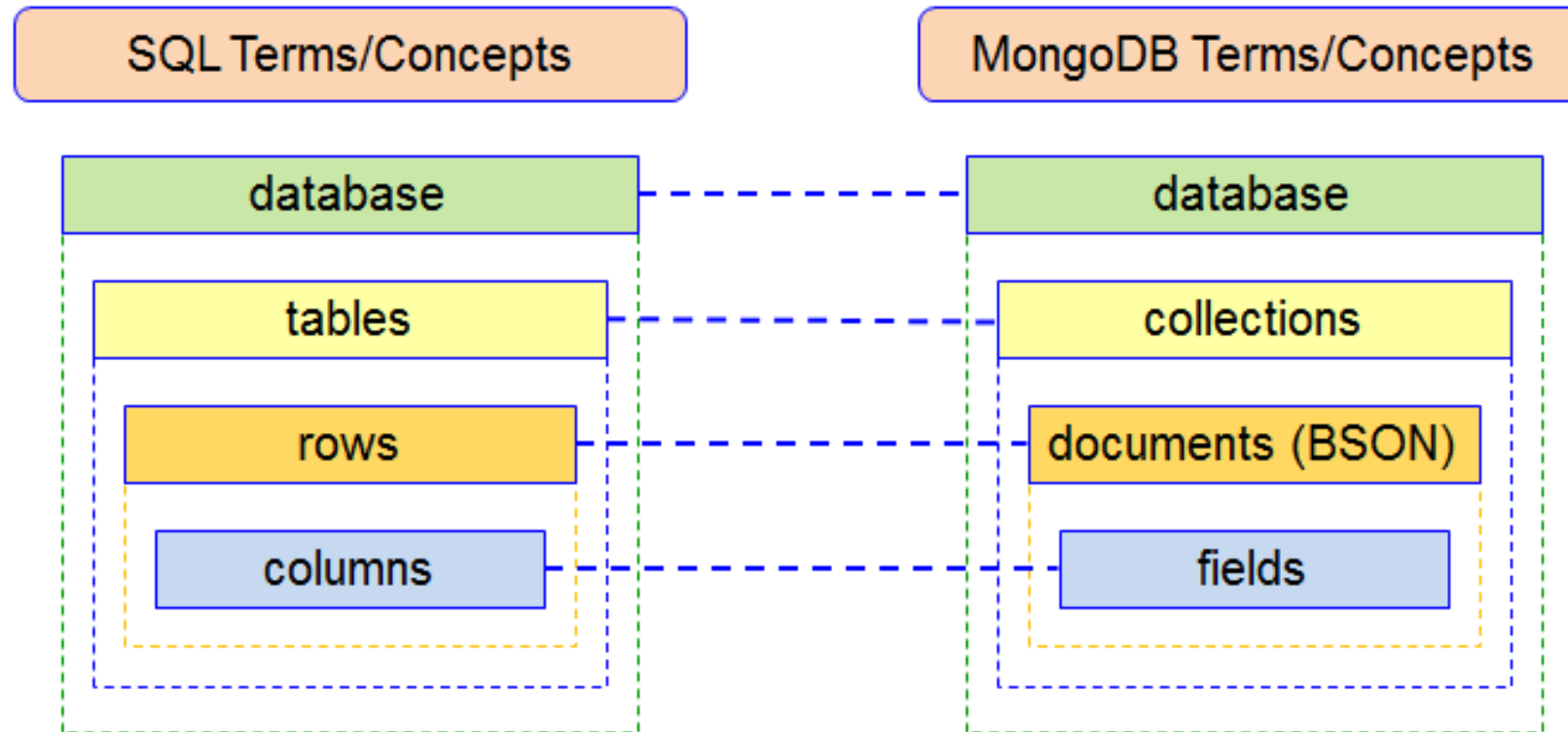
Gratis dan didukung komunitas besar.

## 5. Mudah diintegrasikan

Cocok digunakan dengan Node.js, Python, Java, dan bahasa lain.



# SQL vs mongoDB



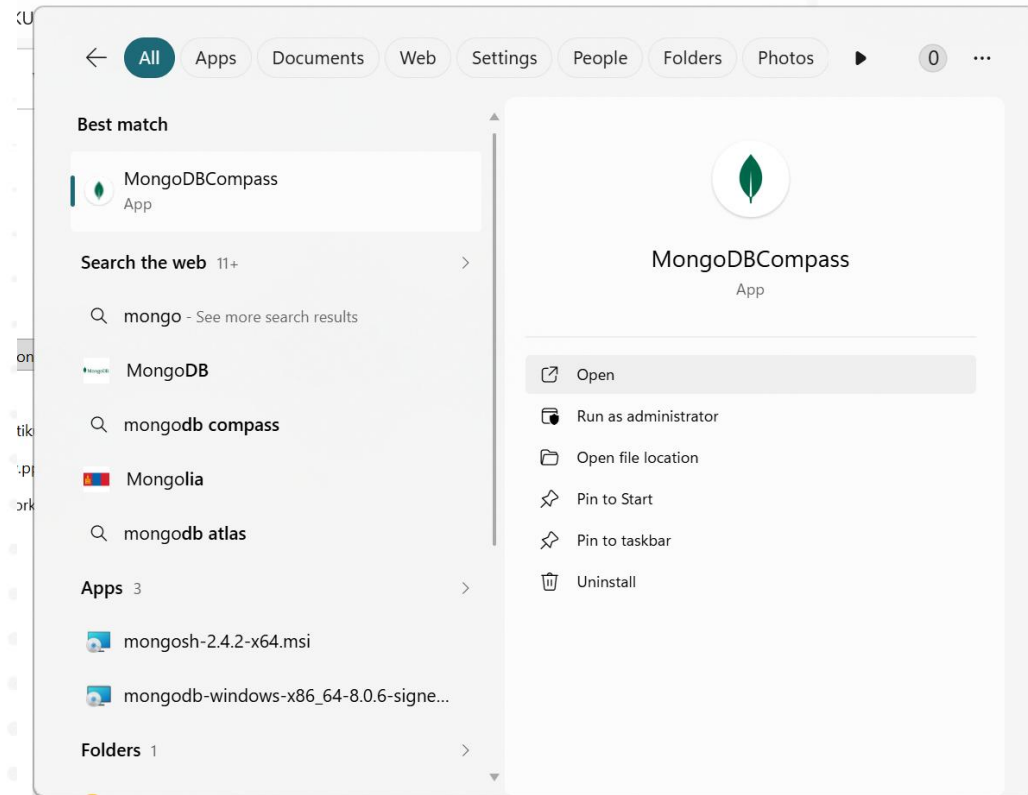


# Perbedaan SQL dan MongoDB

MongoDB (NoSQL)	Database SQL
Data berbentuk dokumen	Data berbentuk tabel
Tidak perlu skema tetap	Skema harus ditentukan
Fleksibel	Lebih kaku
Cocok untuk data besar & dinamis	Cocok untuk data terstruktur

# MongoDB Compass

- MongoDB Compass adalah aplikasi GUI untuk mengelola database MongoDB tanpa harus menggunakan command line.



# Install mongo db

## ➤ Unduh Installer MongoDB

Kunjungi situs resmi MongoDB di **mongodb.com** dan unduh installer versi terbaru.

## ➤ Referensi cara install MongoDB

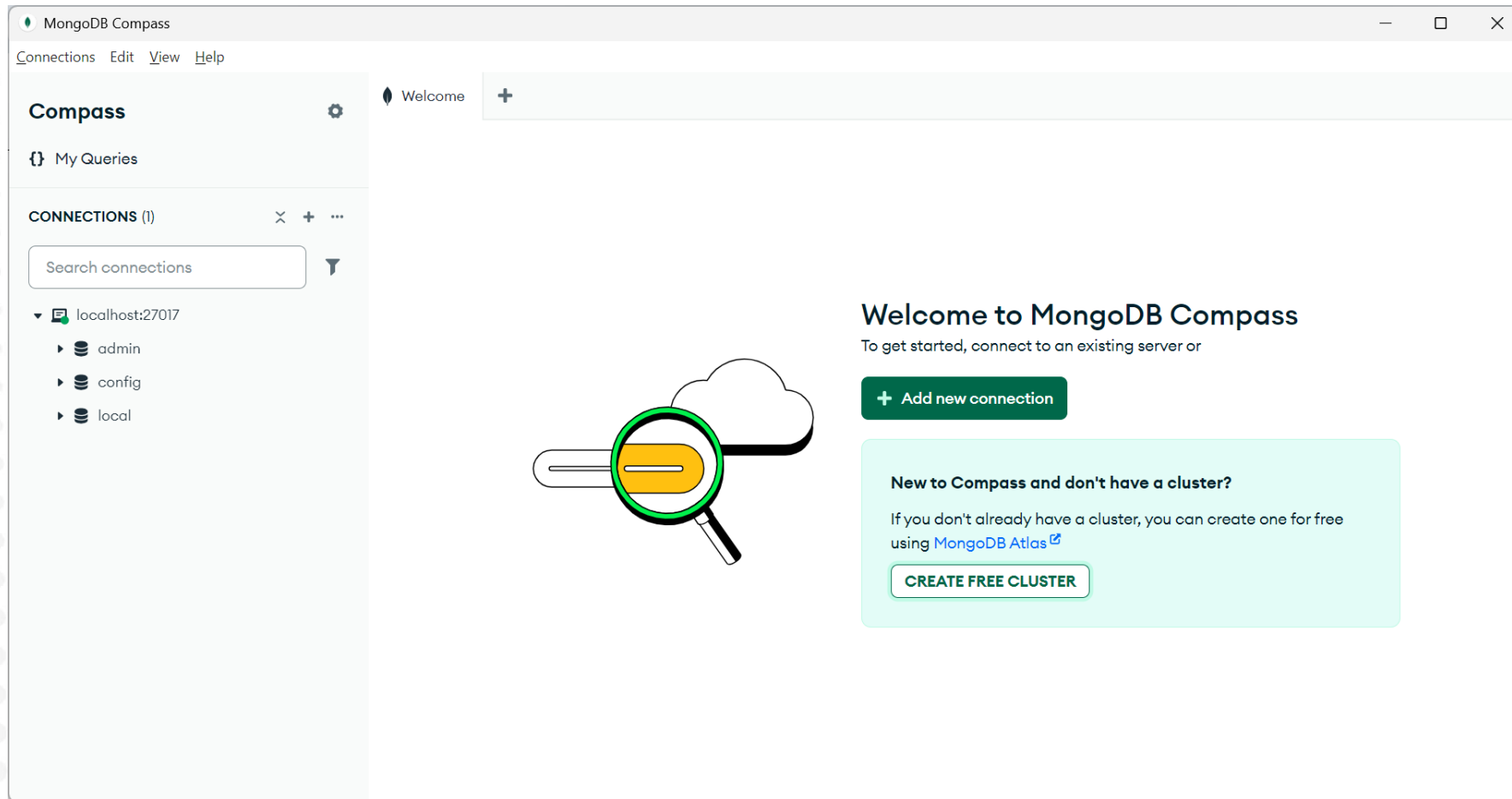
<https://www.youtube.com/watch?v=KYIOJrE3zjk>

➤ Jika sudah berhasil install sesuai Langkah pada link diatas maka nanti akan muncul MongoDB Compass

➤ Install juga MongoDB Shell (Langkah ada di link diatas)



# Tampilan MongoDB compass



# Mongo DB shell

- Buka terminal ketikkan **mongosh** => untuk masuk dan berinteraksi dengan MongoDB menggunakan MongoDB Shell.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.4.2
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\yunia> mongosh
Current Mongosh Log ID: 69d23ce830e54274f5b71235
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true
&serverSelectionTimeoutMS=2000&appName=mongosh+2.4.2
Using MongoDB:      8.0.6
Using Mongosh:      2.4.2

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
  The server generated these startup warnings when booting
  2026-03-11T12:22:58.570+07:00: Access control is not enabled for the
  database. Read and write access to data and configuration is unrestricte
  d
  -----

test> |
```



# KELEBIHAN MONGODB

1. MongoDB menggunakan format dokumen berbasis JSON (BSON) yang tidak memerlukan skema tetap. Ini memungkinkan data yang tidak terstruktur atau semi-terstruktur untuk disimpan dengan mudah.
2. MongoDB mendukung sharding, yang memecah data menjadi beberapa bagian dan menyebarkannya ke berbagai server.
3. Dengan struktur skema yang fleksibel, pengembang dapat lebih cepat menambahkan atau mengubah data tanpa perlu memodifikasi struktur tabel seperti pada database SQL.
4. MongoDB lebih efektif untuk menyimpan data yang kompleks seperti array atau data bersarang.
5. MongoDB dirancang untuk integrasi yang mudah dengan teknologi big data dan penyimpanan berbasis cloud.

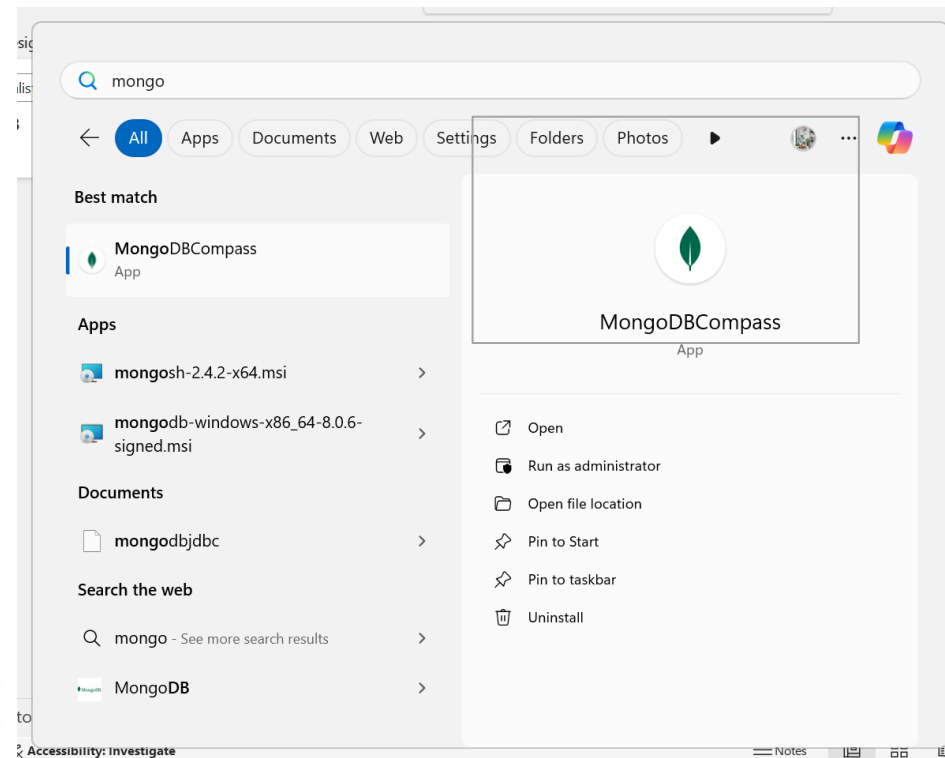
# PERCOBAAN MEMBUAT DATABASE BARU DI MONGODB



# Membuat database baru dengan mongodbc compass

## 1. Buka MongoDB Compass

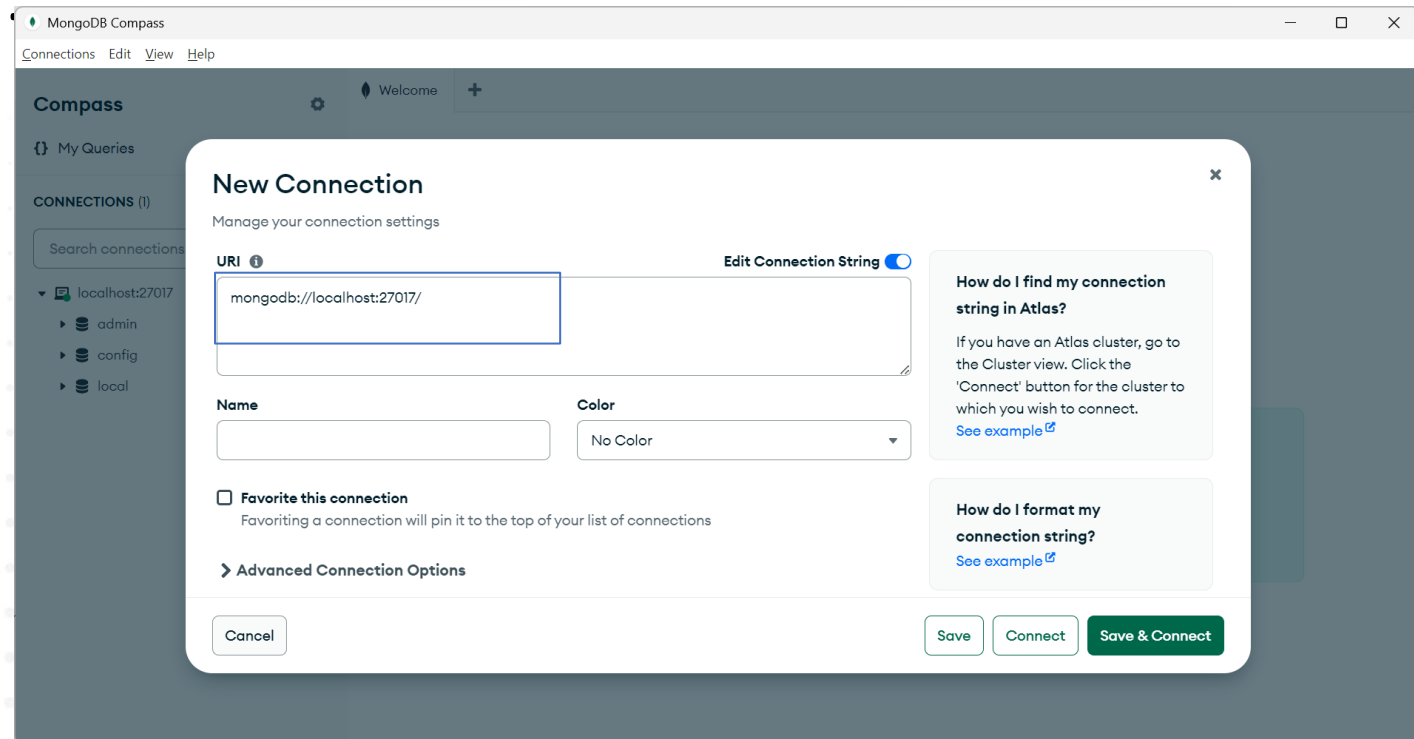
- Pastikan MongoDB Compass telah terinstal di komputer Anda.
- Jalankan aplikasi MongoDB Compass.



# Membuat database baru dengan mongodb compass

## 2. Hubungkan ke Server MongoDB

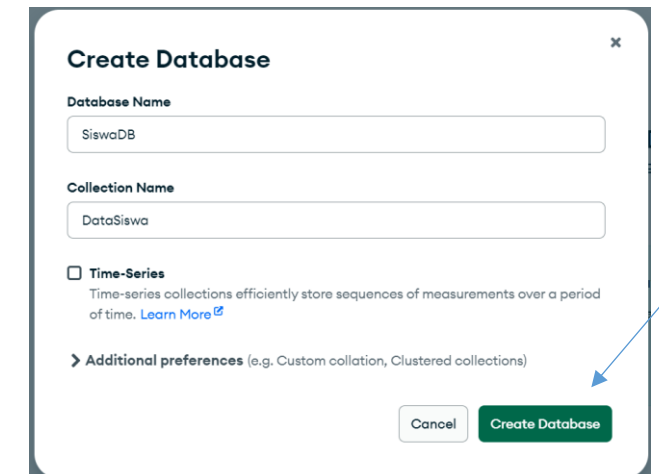
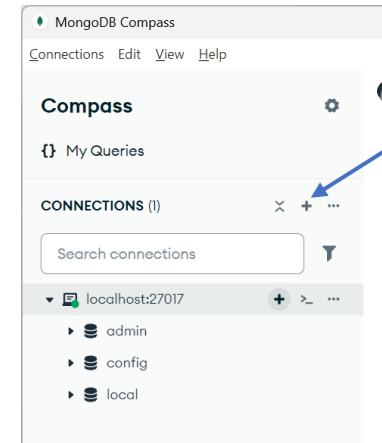
- Di halaman awal MongoDB Compass, masukkan URL koneksi ke server MongoDB Anda (biasanya untuk server lokal).
- Klik tombol "Connect"



# Membuat database baru dengan mongodb compass

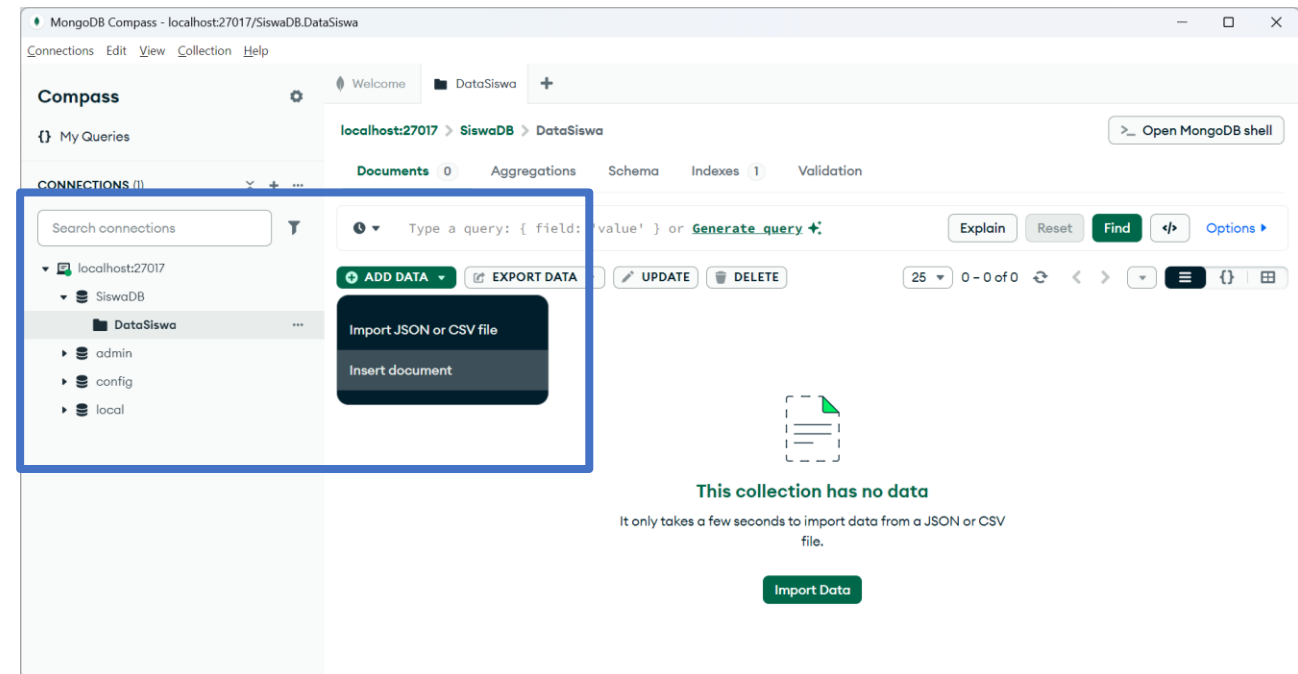
## 3. Buat Database Baru

- ✓ Setelah terhubung ke server MongoDB, klik tombol "**Create Database**" di pojok kanan atas.
- ✓ Masukkan nama database, misalnya : **siswaDB**
- ✓ Masukkan nama koleksi pertama, misalnya : **DataSiswa**
- ✓ Klik tombol "Create Database".



# Membuat database baru dengan mongodb compas

4. Tambahkan Dokumen ke Koleksi
  - a. Buka database SiswaDB yang baru saja dibuat.
  - b. Pilih koleksi DataSiswa
  - c. Klik tombol "Insert Document" untuk menambahkan data baru ke dalam koleksi.
  - d. Masukkan data siswa dalam format JSON, contohnya:





# Membuat database baru dengan mongodb compass

- Data siswa dalam format JSON

## Insert Document



To collection SiswaDB.DataSiswa

VIEW  

```
1 /**
2 * Paste one or more documents here
3 */
4 {
5   "_id": {
6     "$oid": "67e1fccbc6c6c793c73ecdd9"
7   },
8   "nama": "Riyuka",
9   "kelas": "2",
10  "umur": 8
11 }
```

Cancel

Insert

 Type a query: { field: 'value' } or [Generate query](#) 

 ADD DATA

 EXPORT DATA

 UPDATE

 DELETE

```
_id: ObjectId('67e1fccbc6c6c793c73ecdd9')
nama : "Riyuka"
kelas : "2"
umur : 8
```



# Mengelola database mongodb menggunakan mongodb shell

```
Wel... x DataSiswa >_ mongosh: localhost:27017 +
>_MONGOSH
> db["DataSiswa"].find()
<
> use SiswaDB
< switched to db SiswaDB
> db["DataSiswa"].find()
< {
  _id: ObjectId('67e1fccbc6c6c793c73ecdd9'),
  nama: 'Riyuka',
  kelas: '2',
  umur: 8
}
> show dbs
< SiswaDB 40.00 KiB
  admin 40.00 KiB
  config 72.00 KiB
  local 40.00 KiB
> show collections
< DataSiswa
SiswaDB> |
```

# LATIHAN

## 1. Membuat Database dan Collection

Langkah di MongoDB Compass:

Klik Create Database

Isi:

Database Name: kampusDB

Collection Name: mahasiswa

Klik Create Database



# LATIHAN

## 2. Insert Data (Create) menggunakan BSON

Klik **Insert Document**, lalu masukkan data berikut:

```
{  
  "nama": "Andi",  
  "umur": 20,  
  "jurusan": "Informatika",  
  "ipk": 3.5,  
  "aktif": true,  
  "tanggal_masuk": { "$date": "2022-08-01T00:00:00Z" }  
}
```



# LATIHAN

Tambahkan beberapa data lagi:

```
{  
  "nama": "Budi",  
  "umur": 22,  
  "jurusan": "Sistem Informasi",  
  "ipk": 3.2,  
  "aktif": true,  
  "tanggal_masuk": { "$date": "2021-08-01T00:00:00Z" }  
}  
  
{  
  "nama": "Citra",  
  "umur": 21,  
  "jurusan": "Informatika",  
  "ipk": 3.8,  
  "aktif": false,  
  "tanggal_masuk": { "$date": "2020-08-01T00:00:00Z" }  
}
```



# LATIHAN

## 3. Read Data (Menampilkan Data)

Klik tab Documents, semua data akan tampil otomatis.

## 4. Query / Filter Data

Gunakan kolom Filter di Compass:

- ◆ Mahasiswa jurusan Informatika:

```
{ "jurusan": "Informatika" }
```

- ◆ IPK lebih dari 3.3:

```
{ "ipk": { "$gt": 3.3 } }
```

- ◆ Mahasiswa aktif:

```
{ "aktif": true }
```



# LATIHAN

## 5. Update Data

- a. Klik salah satu document
- b. Klik tombol **Edit**
- c. Ubah data, misalnya:  
"umur": 21

## 6. Klik **Update**

# LATIHAN

## 6. Delete Data

- a. Pilih document
- b. Klik **Delete**
- c. Konfirmasi penghapusan

## 7. Sorting Data

Gunakan bagian **Sort**:

- ◆ Urutkan IPK tertinggi:

```
{ "ipk": -1 }
```



# LATIHAN

## 8. Agregasi Sederhana

Masuk ke tab **Aggregations**

Contoh pipeline:

- ◆ Rata-rata IPK:

```
[
  {
    "$group": {
      "_id": null,
      "rata_ipk": { "$avg": "$ipk" }
    }
  }
]
```