



## MODUL PRAKTIKUM

### SD3203-Teknologi Basis Data

Program Studi Sains Data  
Fakultas Sains  
Institut Teknologi Sumatera

2025

# **MODUL 9**

## **Document Database**

## Document Database

### 1. Tujuan Praktikum

- a. Memahami konsep *document database* dan NoSQL
- b. Menguasai dan menerapkan konsep NoSQL menggunakan Mongodb

### 2. Konsep Dasar

#### 2.1. Document Database

*Document database* adalah basis data yang tidak menggunakan SQL (NoSQL) dalam melakukan transaksi atau pengelolaan data. Dalam NoSQL database, data-data tersimpan dalam bentuk dokumen seperti XML atau JSON. Tidak seperti database relasional dimana data tersimpan secara terstruktur dan memiliki skema tetap, NoSQL tidak memiliki struktur tetap sehingga lebih fleksibel saat menangani data besar. Dalam *document database* atau No SQL, data disimpan dalam dokumen-dokumen yang berdiri sendiri yang memiliki struktur berbeda-beda sehingga cocok untuk data dengan bentuk semi terstruktur atau tidak tetap.

Basis data No SQL memiliki beberapa istilah yang digunakan untuk menyebutkan table dan baris dalam database, yaitu *collection* yang merupakan tabel - tabel yang ada pada suatu database, serta *document* yang merupakan baris - baris atau data yang dimiliki suatu *collection*. Pada implementasinya, No SQL memiliki beberapa tipe, diantaranya:

- Dokumen (contoh: MongoDB, CouchDB)
- Key-Value (contoh: Redis, DynamoDB)
- Kolom Lebar (contoh: Cassandra, HBase)
- Graf (contoh: Neo4j, ArangoDB)

#### 2.2. Mongodb



Gambar 1. No SQL database Mongodb

MongoDB adalah basis data NoSQL berjenis dokumen (*document-oriented database*) yang bersifat *open-source* dan *cross-platform*. MongoDB menyimpan data dalam format BSON (Binary JSON), yang memungkinkan struktur data fleksibel dan efisien. Dalam MongoDB tidak terdapat table atau baris, tetapi terdapat koleksi (*collection*) yang dalam SQL disebut table, serta terdapat *document* atau dokumen yang biasa

diangap baris. MongoDB menyimpan data sebagai rekaman yang terdiri dari koleksi dan dokumen. Penyimpanan data dalam MongoDB terdiri dari pasangan *key* dan *value*. Berikut adalah table berisi perbedaan istilah pada RDBMS dengan MongoDB.

Tabel 1 Perbedaan RDBMS dengan MongoDB

RDBMS	MongoDB
Database	Database
Table, View	Collection
Row	Document (JSON, BSON)
Column	Field
Index	Index
Join	Embedded Document
Foreign Key	Reference
Partition	Shard

### 2.3. CRUD dalam MongoDB

Basis data dokumen seperti MongoDB juga menyediakan operasi CRUD (*create*, *read*, *update*, dan *delete*) untuk membuat dan mengelola seperti halnya SQL dengan menggunakan perintah atau kueri yang berbeda. Berikut adalah table berisi perintah CRUD MongoDB.

Perintah	MongoDB	Kueri
CREATE	insert	db.collection.insert (<document>)
	save	db.collection.save (<document>)
	update	db.collection.update (<query>, <update>, { upsert: true })
READ	find	db.collection.find (<query>, <projection>)
	findOne	db.collection.findOne (<query>, <projection>)
UPDATE	update	db.collection.update(<query>, <update>, <options>)
DELETE	delete	db.collection.deleteOne( <query>, <justOne> )

### 2.4. Operator dalam MongoDB

Pada MongoDB terdapat beberapa operator perbandingan seperti halnya pada SQL. Berikut adalah beberapa operator dalam MongoDB.

Operator	Fungsi	Contoh
\$eq	Sama dengan (equal)	{harga: {\$eq: 50000}}
\$ne	Tidak sama dengan (not equal)	{stok: {\$ne: 0}}
\$gt	Lebih besar dari	{harga: {\$gt: 70000}}
\$gte	Lebih besar atau sama	{stok: {\$gte:10}}
\$lt	Kurang dari	{tahun_terbit: \$lt:2022}}
\$lte	Kurang atau sama	{harga: {\$lte: 100000}}
\$in	Termasuk dalam daftar nilai	{kategori: {\$in: ["Teknologi", "Desain"]}}
\$nin	Tidak termasuk dalam daftar nilai	{penulis: {\$nin: ["Andi", "Budi"]}}

### 3. Langkah Praktikum

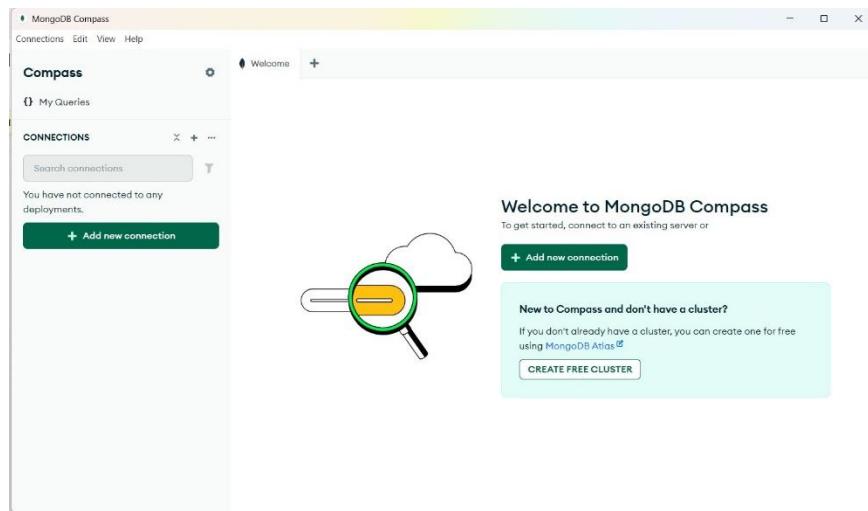
#### 3.1. Instalasi Mongodb

Pada praktikum ini, *document database* yang digunakan adalah MongoDB dengan menggunakan MongoDB Compass sebagai GUI-nya. Pastikan MongoDB Compass sudah terinstal pada computer kemudian buka aplikasinya. Jika MongoDB belum terinstal, ikuti langkah-langkah instalasinya sebagai berikut.

- Unduh MongoDB pada tautan berikut ini <https://www.mongodb.com/try/download/community> pilih MongoDB Community Server Download dengan platform Windows atau disesuaikan perangkat masing-masing kemudian pilih *package msi*.
- Instal aplikasi yang dijalankan
- Ceklis pada bagian MongoDB Compass lalu ikuti tahapan instalasinya dengan klik **Next** pada tiap langkahnya hingga selesai.
- Untuk memeriksa apakah MongoDB sudah terkoneksi atau belum, ketik **Services** pada **Search** kemudian periksa apakah koneksi MongoDB sudah berjalan atau belum. Jika berhasil koneksi MongoDB memiliki status *Running*.
- Buka aplikasi MongoDB Compass untuk memulai praktikum.

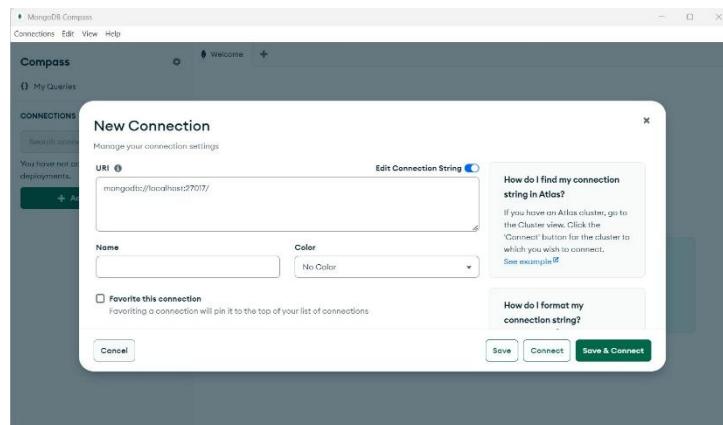
#### 3.2. Koneksi MongoDB

Untuk melakukan transaksi CRUD pada database dokumen menggunakan MongoDB, dibutuhkan GUI untuk mempermudah interaksi dengan pengguna. Pada praktikum ini digunakan MongoDB Compass sebagai GUI-nya. Berikut adalah tampilan MongoDB Compass saat dijalankan pertama kali.

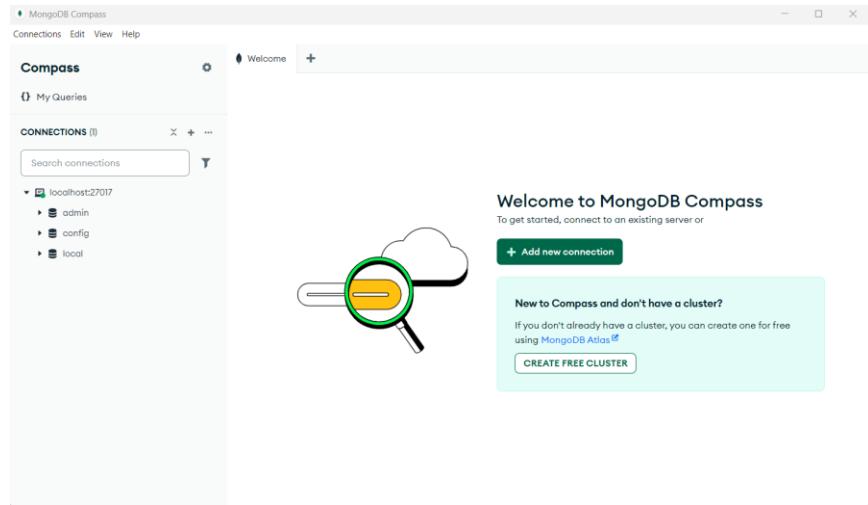


Gambar 2. Tampilan MongoDB

Sebelum membuat database, koneksi pada MongoDB Compass harus dibuat terlebih dahulu dengan klik pada **Add new connection** kemudian klik **Save & Connect**. Koneksi yang berhasil dibuat akan berada pada sisi kiri di bagian **Connections** seperti pada gambar berikut.



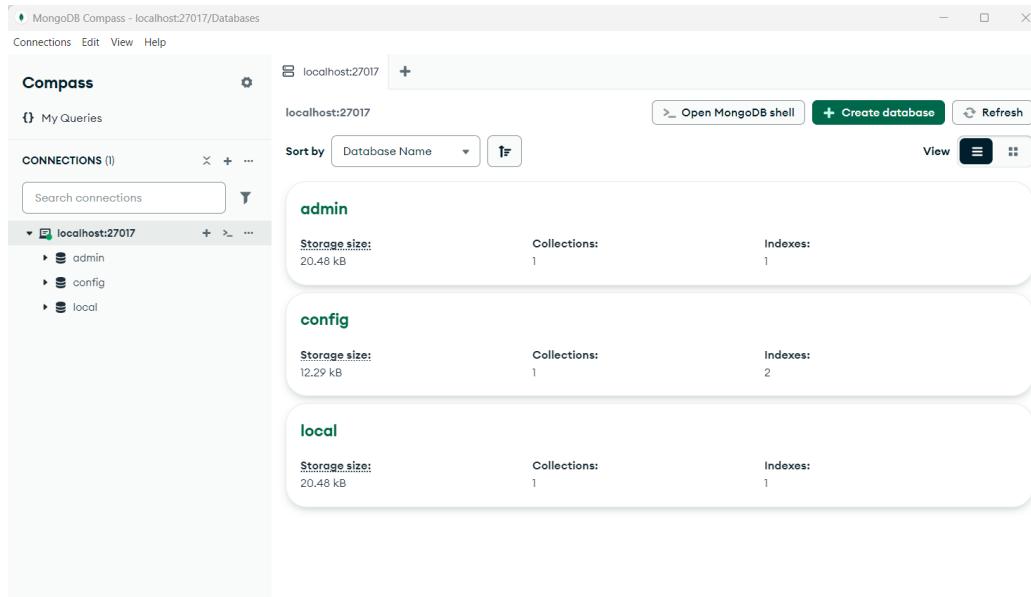
Gambar 3 Langkah membuat koneksi pada MongoDB Compass



Gambar 4 Tampilan koneksi MongoDB yang berhasil dibuat

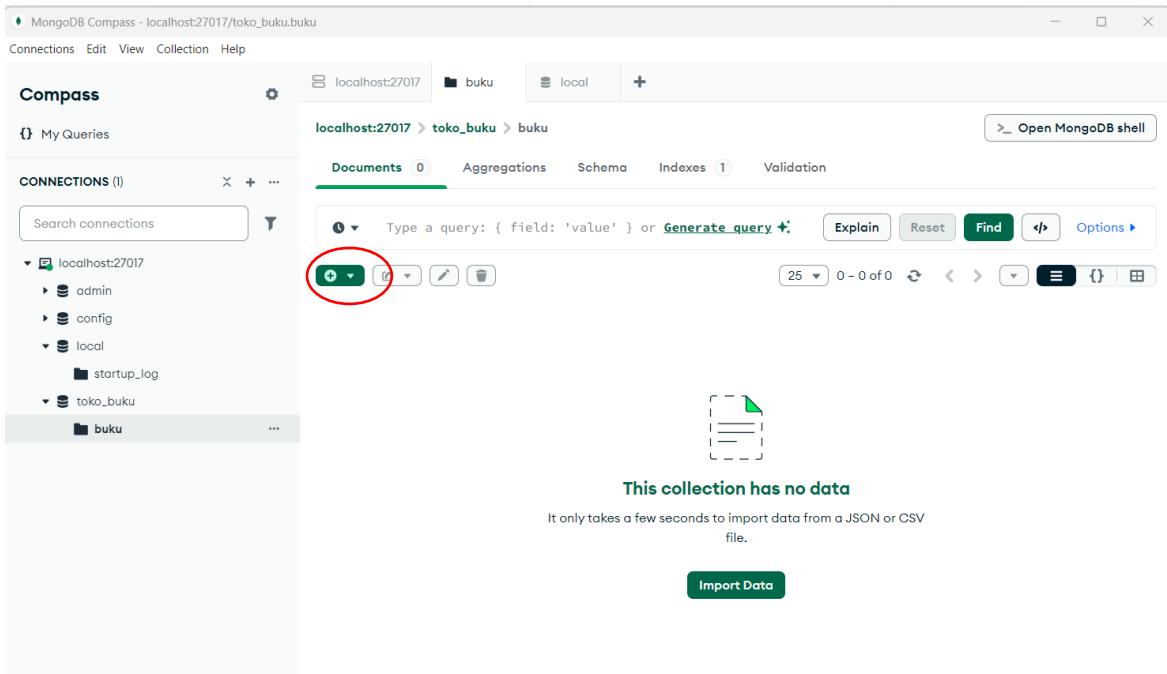
### 3.3.Membuat Document dan Collection (CREATE)

Untuk membuat database baru, klik pada koneksi yang dipilih atau pada **localhost:27017** kemudian klik pada bagian **Create database** seperti pada gambar berikut.



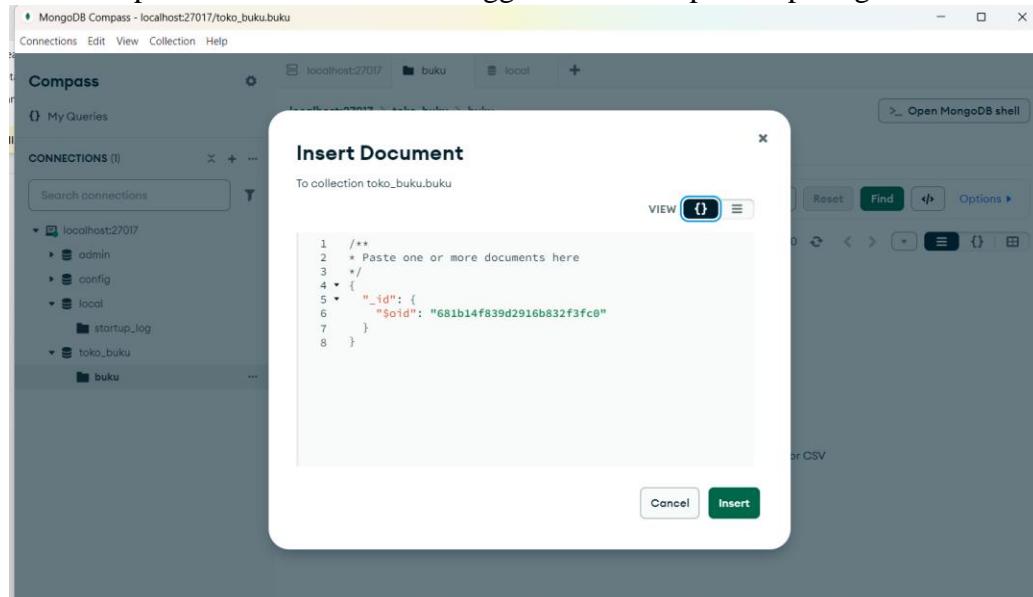
Gambar 5 Tampilan database pada koneksi localhost:27017

Setelah klik Create database, tuliskan **toko\_buku** pada bagian **Database Name** lalu tuliskan **buku** pada bagian **Collection Name** karena database yang akan dibuat adalah database toko buku yang di dalamnya berisi koleksi buku. Berikut adalah tampilan saat berhasil membuat database **toko\_buku**.



Gambar 6 Tampilan database setelah database toko\_buku dibuat

Pada Gambar 6, untuk menambah dokumen klik bagian pada lingkaran merah atau **Add** kemudian pilih **Insert Dokument** hingga muncul tampilan seperti gambar berikut.



Gambar 7 Tampilan saat menambahkan dokumen pada MongoDB

Dari Gambar 7, pada baris pertama hingga seterusnya ketikkan kueri untuk menambahkan sebuah data buku seperti pada potongan kode berikut ini.

```
{
  "judul": "Belajar MongoDB untuk Pemula",
  "penulis": "Andi Setiawan",
```

```
"tahun_terbit": 2022,  
"kategori": "Teknologi",  
"stok": 10,  
"harga": 75000  
}
```

Potongan kode di atas merupakan perintah untuk menambahkan koleksi buku pada document toko buku. Ketika data berhasil dimasukkan, data akan memiliki sebuah id unik otomatis dari MongoDB untuk mengidentifikasi setiap dokumen dan dapat diisi manual jika diperlukan. Dengan menggunakan langkah yang sama, data juga dapat dimasukkan secara bersamaan dengan format mirip array seperti pada potongan kode berikut ini.

```
[  
  {  
    "judul": "Pemrograman Web Dasar",  
    "penulis": "Dewi Lestari",  
    "tahun_terbit": 2020,  
    "kategori": "Pemrograman",  
    "stok": 12,  
    "harga": 65000  
  },  
  {  
    "judul": "Data Science untuk Pemula",  
    "penulis": "Rudi Hartanto",  
    "tahun_terbit": 2023,  
    "kategori": "Data Science",  
    "stok": 7,  
    "harga": 120000  
  },  
  {  
    "judul": "Desain UI/UX Modern",  
    "penulis": "Lisa Anggraeni",  
    "tahun_terbit": 2021,  
    "kategori": "Desain",  
    "stok": 9,  
    "harga": 95000  
  }  
]
```

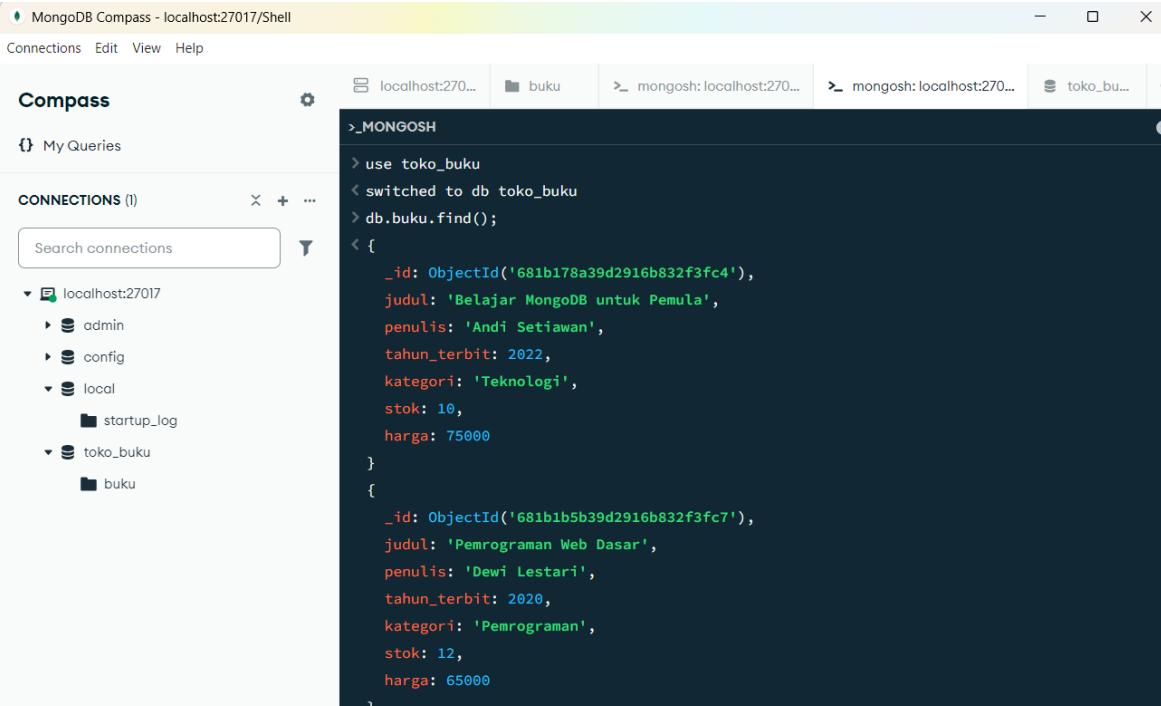
Insert data juga dapat dilakukan menggunakan kueri yang dituliskan dalam MongoDB shell. Berikut adalah contoh kueri untuk menambahkan data buku Pengolahan Citra Digital menggunakan kueri.

```
db.buku.insertOne({  
  judul: "Pengolahan Citra Digital",  
  penulis: "Putra Andi",  
  tahun_terbit: 2014,  
  kategori: "Teknologi",  
  stok: 5,  
  harga: 78000  
});
```

### 3.4.Menampilkan Data (READ)

Untuk menampilkan data yang telah disimpan dapat menggunakan fungsi `find`. Klik database yang akan ditampilkan, pada praktikum ini adalah **toko\_buku** kemudian pada sisi kanan klik **Open MongoDB shell**. Tuliskan kueri berikut pada jendela **Open MongoDB shell** untuk menampilkan seluruh data atau koleksi pada dokumen **toko\_buku**.

```
db.buku.find();
```



The screenshot shows the MongoDB Compass interface. On the left, the connections sidebar lists 'localhost:27017' with its databases: admin, config, local, and toko\_buku. The 'toko\_buku' database is selected, and its collection 'buku' is shown. The main pane displays the results of the `db.buku.find()` command. The output shows two documents from the 'buku' collection:

```
_id: ObjectId('681b178a39d2916b832f3fc4'),
judul: 'Belajar MongoDB untuk Pemula',
penulis: 'Andi Setiawan',
tahun_terbit: 2022,
kategori: 'Teknologi',
stok: 10,
harga: 75000

}
{
_id: ObjectId('681b1b5b39d2916b832f3fc7'),
judul: 'Pemrograman Web Dasar',
penulis: 'Dewi Lestari',
tahun_terbit: 2020,
kategori: 'Pemrograman',
stok: 12,
harga: 65000
```

Gambar 8 Tampilan hasil dari perintah `find()`

Sama halnya dengan `SELECT`, pada MongoDB juga dapat melihat atau menampilkan data berdasarkan kondisi tertentu. Berikut adalah contoh kueri untuk menampilkan buku di atas tahun 2022.

```
db.buku.find({ "tahun_terbit": { $gte: 2022 } });
```

### 3.5.Memperbarui Data (UPDATE)

Dalam MongoDB, update memiliki fungsi yang sama dengan SQL yaitu untuk mengubah data atau memperbarui data. Berikut adalah perintah untuk mengubah stok buku pada buku berjudul **Belajar MongoDB untuk Pemula** menjadi tujuh.

```
db.buku.updateOne(
  {judul: "Belajar MongoDB untuk Pemula"},
  {$set: {stok: 7}}
);
```

Fungsi update juga bisa dilakukan sekaligus dengan menggunakan updateMany() seperti pada potongan program berikut ini.

```
db.buku.updateMany  
  {kategori: "Teknologi"},  
  {$inc: {harga: 5000}}  
);
```

Fungsi **\$inc** digunakan untuk menambah nilai numerik pada harga sebesar 5000 sehingga semua buku dengan kategori **Teknologi** akan bertambah harganya sebanyak 5000.

### 3.6.Menghapus Data (DELETE)

Untuk menghapus data dalam dokumen pada MongoDB dapat menggunakan fungsi deleteOne atau deleteMany. Berikut adalah contoh untuk menghapus satu dan beberapa data pada MongoDB.

```
db.buku.deleteOne({judul: "Node.js Dasar"});
```

```
db.buku.deleteMany({kategori: "Teknologi"});
```