

MODUL 7

MAPREDUCE & TRANSACTION

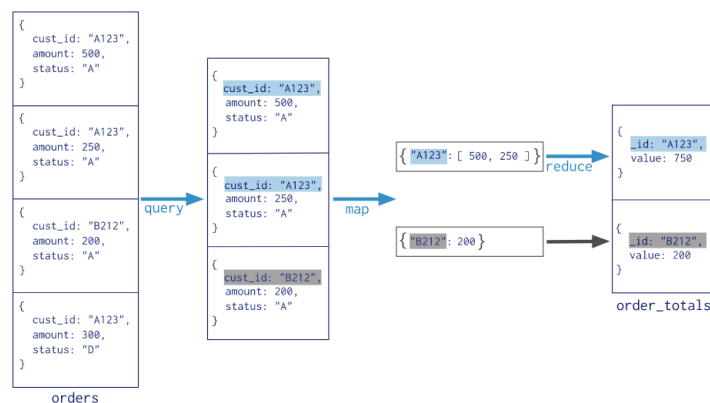
7.1. Tujuan Praktikum

1. Mahasiswa mampu melakukan setup cluster mongodb
2. Mahasiswa mampu melakukan replikasi db
3. Mahasiswa mampu implementasi transaksi
4. Mahasiswa mengenal tools mongodb

7.2. Dasar Teori

7.2.1. Map Reduce

Merupakan data proses paradigma data pada MongoDB yang bertujuan untuk menyatukan atau memadatkan data bervolume banyak menjadi hasil agregat yang berguna.



Gambar 20. Gambar Map Reducing

Contohnya pada gambar diatas, dari data pada koleksi order dilakukan perintah map reducing dimana data tersebut di satukan berdasarkan data “cust_id” sehingga menjadi “order_totals”.

Map Reduce dibagi menjadi 2 function. Function pertama adalah Map dimana semua data akan dikelompokkan berdasarkan key-value dan kemudian Reduce digunakan untuk melakukan operasi pada data yang tadi sudah di Mapping.

Pada versi 6.0 map-reduce sudah tidak ada, fungsi ini digantikan pada aggregate pipeline yaitu gabungin \$group dan \$out.

```
db.orders.aggregate([
  { $group: { _id: "$cust_id", value: { $sum: "$price" } } },
  { $out: "order_totals" }
])
```

7.2.2. Transactions

Transaksi memainkan peran penting dalam menjaga konsistensi dan integritas data dalam database. Transaksi mewakili satu unit kerja yang terdiri dari beberapa operasi yang dijalankan dalam satu urutan. Pada bagian ini, kita akan membahas konsep transaksi di MongoDB, penggunaannya, dan bagaimana transaksi membantu menyelesaikan berbagai operasi.

7.2.3. Gambaran umum transaksi

MongoDB mendukung transaksi multi-dokumen, sehingga Anda dapat melakukan beberapa operasi baca dan tulis di beberapa dokumen dalam satu transaksi atomik. Sebuah transaksi dapat melibatkan beberapa operasi, misalnya:

- Membuat dokumen baru
- Memperbarui dokumen yang sudah ada
- Menghapus dokumen
- Membaca dokumen

Tujuan mendasar dari sebuah transaksi adalah untuk menjalankan semua atau tidak sama sekali dari operasi-operasinya. Ini berarti, jika ada operasi dalam transaksi yang gagal, seluruh transaksi akan dibatalkan, dan basis data akan kembali ke kondisi awal, sehingga memastikan konsistensi data.

Transaksi di MongoDB sangat penting untuk mencapai properti ACID berikut ini:

- **Atomicity:** Memastikan bahwa semua operasi dalam transaksi dieksekusi, atau tidak ada yang dieksekusi.
- **Consistency:** Menjamin bahwa, setelah menyelesaikan transaksi, basis data tetap dalam keadaan konsisten.
- **Isolation:** Menjamin bahwa operasi dalam transaksi terisolasi dari transaksi lain yang sedang dieksekusi secara bersamaan.
- **Durability:** Menjamin bahwa setelah transaksi berhasil diselesaikan, efeknya akan disimpan secara permanen di dalam database.

7.2.4. Penggunaan transaksi

Untuk memulai transaksi di MongoDB, Anda harus mendapatkan sesi dan kemudian memulai transaksi menggunakan metode `startTransaction()`. Setelah melakukan operasi yang diperlukan, Anda dapat meng-commit transaksi untuk menerapkan perubahan pada basis data, atau membatalkannya untuk membuang perubahan.

Berikut ini contoh ilustrasi penggunaan transaksi:

```
// Create New client and connect to server
client = MongoClient(uri, server_api=ServerApi('1'))
wc_majority = WriteConcern("majority", wtimeout =1000)

// Define callback for database operations
def callback(session):
    local = session.client.Mahasiswa.DataDiri

    local.insert_one({"NPM":"192310004", "nama" : "denny dolok 2", "alamat" :
    "bogor"}, session = session)

// start session on this session, executes callback once and then commit
transaction
with client.start_session() as session:
    session.with_transaction(
        callback,
        read_concern=ReadConcern("local"),
        write_concern=wc_majority,
        read_preference=ReadPreference.PRIMARY,
    )
```

7.2.4. Keterbatasan transaksi

Meskipun transaksi memberikan manfaat yang sangat besar dalam hal konsistensi dan integritas data, sangat penting untuk mengetahui beberapa keterbatasannya:

- Transaksi hanya tersedia di MongoDB versi 4.0 ke atas.
- Transaksi dapat menyebabkan overhead kinerja, terutama untuk beban kerja yang banyak menulis.
- Di dalam cluster MongoDB, transaksi hanya mendukung durasi maksimum 60 detik.

Singkatnya, transaksi adalah fitur yang kuat dari MongoDB, memastikan integritas data, dan konsistensi dalam database. Dengan memahami penggunaan dan implikasinya, Anda dapat menggunakannya secara efektif dalam aplikasi Anda sesuai dengan kebutuhan spesifik Anda.

7.3. Software

1. MongoDB
2. Atlas

7.4. Tahapan Kerja

7.4.1. MapReduce

1. Buka Command Prompt dan ketik `mongosh` untuk masuk kedalam terminal monogdb.

```

C:\Users\Denny>mongosh
Current Mongosh Log ID: 64631947e388b9d765d0482a
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.7.1
Using MongoDB:      6.0.4
Using Mongosh:       1.7.1

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

-----
The server generated these startup warnings when booting
2023-05-16T06:57:32.139+07:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-----

test> |

```

2. Ganti ke database rentalfilm dengan menggunakan perintah use.

```

test> use rentalfilm
switched to db rentalfilm
rentalfilm>

```

3. Pada versi 6.0 map reduce sudah tidak bisa digunakan sehingga diganti dengan aggregate gabungan antara \$group dan \$out, dengan contoh sebagai berikut.

```

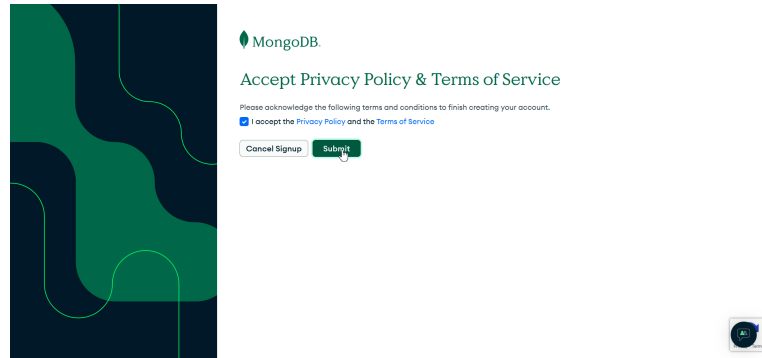
rentalfilm> db.Film.aggregate([
  { $group: {
    _id: "$Length"
    total_rating: {
      $avg : "$Rating"
    }
  }},
  { $out: 'Rating-Based-On-Length'
}
])

```

Pada aggregate diatas pertama dilakukan pengelompokan berdasarkan field Length dari collection Film, yang kemudian dilakukan perhitungan rata-rata Rating berdasarkan kelompok data tersebut. Dan terakhir akan membuat collection baru dengan '\$out' dari data yang telah diproses tadi.

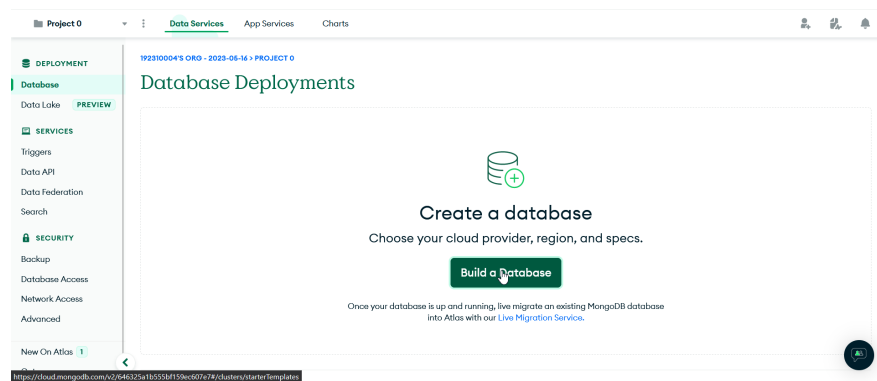
7.4.2. Setup akun MongoDB atlas dan membuat cluster

1. Buka halaman registrasi MongoDB Atlas dengan membuka link <https://account.mongodb.com/account/register>.
2. Buat akun baru dengan menggunakan metode apa saja.
3. Apabila diminta untuk menyetujui Policy dan Terms of Service, centah checkboxnya lalu tekan Submit.

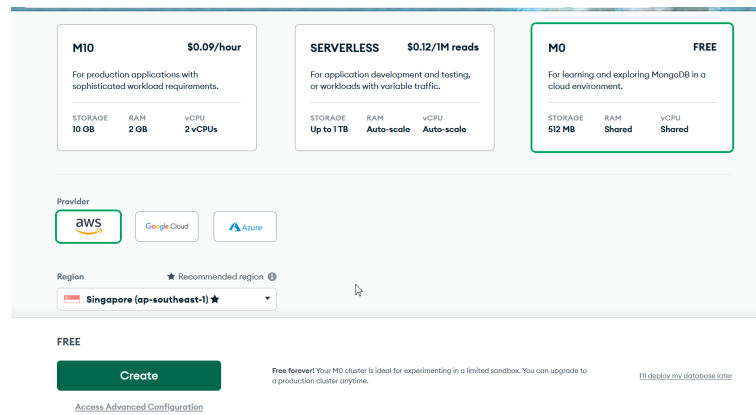


4. Kemudian jika muncul form singkat tentang tujuan menggunakan MongoDB isi seperti ini.

5. Pada halaman utama dashboard MongoDB atlas tekan tombol 'Build a Database'.



6. Pilih tipe yang free dan ganti region menjadi Singapore. Lalu tekan tombol Create.



7. Dihalaman Selanjutnya ketik username dan password yang akan digunakan ketika ingin mengakses database mongoDB, lalu tekan tombol “Create User”

8. Scroll kebawah dan tekan tombol “Add My Current IP Address”. Ini akan memasukan IP yang saat ini digunakan kedalam whitelist sehingga hanya lewat IP yang ditambahkan saja MongoDB bisa diakses.

9. Jika sudah tekan tombol Finish and Close.
10. MongoDB atlas sudah dibuat dengan satu buah Cluster dengan 3 buah node replica set.

7.4.3. Mengenal Python

1. Install Python ke komputer anda dengan mengunduh terlebih dahulu di <https://www.python.org/downloads/>.
2. Download Python Installer dan mulai proses instalasi.

7.4.4. Koneksi Python dengan mongdb

1. Buka Command Prompt dan Jalankan perintah “py -m pip install pymongo”

```
D:\MONGO>py -m pip install pymongo
Collecting pymongo
  Downloading pymongo-4.3.3-cp311-cp311-win_amd64.whl (392 kB)
    382.5/382.5 kB 1.6 MB/s eta 0:00:00
Collecting dnspython<3.0.0,>=1.16.0
  Using cached dnspython-2.3.0-py3-none-any.whl (283 kB)
Installing collected packages: dnspython, pymongo
Successfully installed dnspython-2.3.0 pymongo-4.3.3

[notice] A new release of pip available: 22.3.1 -> 23.1.2
[notice] To update, run: C:\Users\Denny\AppData\Local\Programs\Python\Python311\python.exe -m pip install --upgrade pip
```

2. Buka Visual Studio Code dan buat file baru dengan nama MongoDB.py.
3. Ketik code berikut

```
from pymongo.mongo_client import MongoClient
from pymongo.server_api import ServerApi
uri =
"mongodb+srv://<Username>:<Password>@cluster0.mi4imwt.mongodb.net/?retryWrites
=true&w=majority"
# Create a new client and connect to the server
client = MongoClient(uri, server_api=ServerApi('1'))
# Send a ping to confirm a successful connection

try:
    client.admin.command('ping')
    print("Pinged your deployment. You successfully connected to MongoDB!")
except Exception as e:
    print(e)
```

Ganti <Username> dan <Password> sesuai dengan yang telah dimasukan ketika membuat database di MongoDB Atlas.

4. Jalankan script tersebut dengan melakukan perintah “py MongoDB.py” pada terminal di Visual Studio Code

```
D:\MONGO>py MongoDB.py
```

```
D:\MONGO>py MongoDB.py
Pinged your deployment. You successfully connected to MongoDB!
```

Python sudah berhasil terhubung dengan Database mongoDB yang dibuat jika pesan diatas muncul.

7.4.5. Transaction mongodb

1. Transaction pada MongoDB menggunakan Pymongo bisa dilakukan dengan menggunakan script Berikut.

```

from pymongo.mongo_client import MongoClient
from pymongo.read_concern import ReadConcern
from pymongo.read_preferences import ReadPreference
from pymongo.write_concern import WriteConcern
from pymongo.server_api import ServerApi

uri =
"mongodb+srv://<Username>:<Password>@cluster0.mi4imwt.mongodb.net/?retryWrites
=true&w=majority"
# Create a new client and connect to the server
client = MongoClient(uri, server_api=ServerApi('1'))
wc_majority = WriteConcern("majority", wtimeout = 1000)

def callback(session):
    local = session.client.Mahasiswa.DataDiri

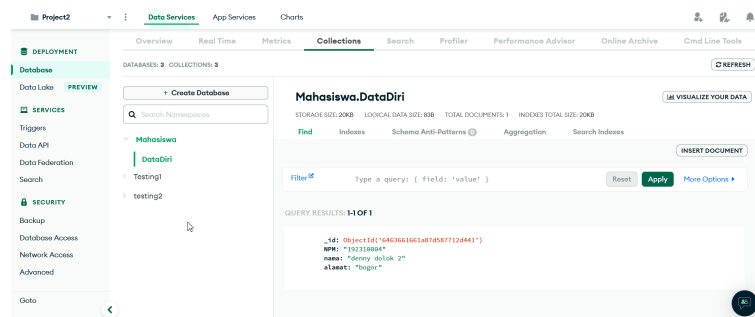
    local.insert_one({"NPM": "192310004", "nama" : "denny dolok 2", "alamat" :
    "bogor"}, session = session)

with client.start_session() as session:
    session.with_transaction(
        callback,
        read_concern=ReadConcern("local"),
        write_concern=wc_majority,
        read_preference=ReadPreference.PRIMARY,
    )

```

Pada def callback(session) dimasukkan semua operasi yang akan dieksekusi oleh MongoDB yang kemudian akan di commit pada saat session.with_transactions.

2. Apabila proses transaksi berhasil maka data akan masuk kedalam database MongoDB di Atlas.



7.5. Tugas dan Latihan

1. Buat dan jalankan proses transaksi untuk memasukan data diri mahasiswa minimal 5 buah data kedalam database MongoDB Atlas menggunakan pymongo.
2. Buat dan jalankan query aggregate untuk melakukan map reduce pada sebuah collection.