REFERENSI CEPAT MONGODB

CRUD, query, agregasi, indeks, desain schema

Koneksi

Connection String

```
mongosh "mongodb://localhost:27017"
mongosh "mongodb://user:pass@host:27017/mydb"
mongosh "mongodb+srv://user:pass@cluster.mongodb.net/mydb"
```

Koneksi Driver (Node.js)

```
const { MongoClient } = require('mongodb');
const client = new MongoClient(uri);
await client.connect();
const db = client.db('mydb');
```

Database & Collection

Operasi Database

```
show dbs
use mydb
db.dropDatabase()
```

Operasi Collection

```
db.createCollection("users")
show collections
db.users.drop()
```

Capped Collection

```
db.createCollection("logs", {
  capped: true, size: 10485760, max: 5000
})
```

Operasi CRUD

Insert

```
db.users.insertOne({ name: "Alice", age: 30 })
db.users.insertMany([
  { name: "Bob", age: 25 },
  { name: "Carol", age: 28 }
])
```

Find

```
db.users.findOne({ name: "Alice" })
db.users.find({ age: { $gte: 25 } })
db.users.find({}, { name: 1, _id: 0 })
db.users.find().sort({ age: -1 }).limit(10)
```

Update

```
db.users.updateOne(
  { name: "Alice" },
  { $set: { age: 31, city: "Boston" } }
)
db.users.updateMany(
  { active: false },
  { $set: { archived: true } }
)
```

Delete

```
db.users.deleteOne({ name: "Alice" })
db.users.deleteMany({ active: false })
```

Replace & Upsert

```
db.users.replaceOne(
  { name: "Alice" },
  { name: "Alice", age: 32, city: "NYC" }
)
db.users.updateOne(
  { email: "a@ex.com" },
  { $set: { name: "Alice" } },
  { upsert: true }
)
```

Operator Query

Perbandingan

`$eq` / `$ne`	Sama dengan / tidak sama dengan
`$gt` / `$gte`	Lebih dari / lebih dari atau sama dengan
`$lt` / `$lte`	Kurang dari / kurang dari atau sama dengan
`$in` / `$nin`	Ada dalam array / tidak ada dalam array

Logika

`$and`	Semua kondisi harus cocok
`$or`	Setidaknya satu kondisi cocok
`$not`	Negasi kondisi
`$exists`	Field ada (true/false)
`$regex`	Kecocokan ekspresi reguler

Operator Update

`$set`	Atur nilai field
`$unset`	Hapus field
`$inc`	Tambah nilai numerik
`$push` / `$pull`	Tambah / hapus elemen array
`$addToSet`	Tambah ke array jika belum ada
`$rename`	Ganti nama field

Agregasi

Tahap Pipeline

`$match`	Filter dokumen (seperti WHERE)
`$group`	Kelompokkan dan agregasi
`$project`	Ubah bentuk dokumen (seperti SELECT)
`$sort`	Urutkan hasil
`$limit` / `$skip`	Pagination
`$lookup`	Left outer join dengan collection lain
`$unwind`	Dekonstruksi array menjadi dokumen

Contoh Agregasi

```
db.orders.aggregate([
  { $match: { status: "completed" } },
  { $group: {
      _id: "$customer_id",
      total: { $sum: "$amount" },
      count: { $sum: 1 }
  }},
  { $sort: { total: -1 } },
  { $limit: 10 }
])
```

Indeks

Buat & Hapus

```
db.users.createIndex({ email: 1 }, { unique: true })
db.users.createIndex({ name: 1, age: -1 })
db.users.createIndex({ location: "2dsphere" })
db.users.dropIndex("email_1")
```

Tipe Indeks

`Single field`	Indeks pada satu field ({ name: 1 })
`Compound`	Beberapa field ({ a: 1, b: -1 })
`Text`	Pencarian teks penuh ({ field: 'text' })
`2dsphere`	Query geospasial
`TTL`	Auto-expire dokumen setelah waktu tertentu

Info Indeks

```
db.users.getIndexes()
db.users.find({ name: "Alice" }).explain()
```

Desain Schema

Embedding vs Referencing

`Embed`	1:1 atau 1:sedikit, data dibaca bersama
`Reference`	1:banyak, data diakses secara independen
`Embed`	Sub-dokumen jarang melebihi 16 MB
`Reference`	Relasi banyak-ke-banyak

Validasi Schema

```
db.createCollection("users", {
  validator: { $jsonSchema: {
    bsonType: "object",
    required: ["name", "email"],
    properties: {
      name: { bsonType: "string" },
      email: { bsonType: "string" }
}}})
```

Replikasi

Konsep Replica Set

`Primary`	Menerima semua penulisan
`Secondary`	Mereplikasi dari primary, bisa melayani pembacaan
`Arbiter`	Memilih dalam pemilihan, tidak menyimpan data

Perintah Replica Set

```
rs.initiate()
rs.add("mongo2:27017")
rs.addArb("mongo3:27017")
rs.status()
rs.conf()
```

Pola Umum

Transaksi

```
const session = client.startSession();
session.startTransaction();
await db.collection("accounts").updateOne(
  { _id: 1 }, { $inc: { bal: -100 } }, { session });
await db.collection("accounts").updateOne(
  { _id: 2 }, { $inc: { bal: 100 } }, { session });
await session.commitTransaction();
```

Bulk Write

```
db.users.bulkWrite([
  { insertOne: { document: { name: "Dan" } } },
  { updateOne: {
      filter: { name: "Alice" },
      update: { $set: { age: 31 } }
  }},
  { deleteOne: { filter: { name: "old" } } }
])
```

Change Stream

```
const stream = db.collection("orders")
  .watch([{ $match: { "fullDocument.status": "new" } }]);
stream.on("change", (change) => {
  console.log(change.fullDocument);
});
```

Perintah mongosh

Helper Shell

`show dbs`	Daftar database
`show collections`	Daftar collection di db saat ini
`db.stats()`	Statistik database
`db.collection.stats()`	Statistik collection
`db.collection.countDocuments({})`	Hitung dokumen
`db.collection.distinct('field')`	Nilai berbeda untuk sebuah field

Export & Import

```
mongoexport --db=mydb --collection=users \
  --out=users.json
mongoimport --db=mydb --collection=users \
  --file=users.json
mongodump --db=mydb --out=/backup/
mongorestore --db=mydb /backup/mydb/
```