

## DATABASE

### dbperpustakaan

- ❖ **Hapus database :**
  - 1) use dbperpustakaan
  - 2) db.dropDatabase()
- ❖ **Hapus collections / tabel :**  
db.tb\_penjaga.drop()

## CRUD

- ❖ Lihat data tabel yg tersedia :  
show collections
- ❖ Tambah 1 document / data :  
db.tb\_peminjaman.insert(  
{id\_peminjaman:11, id\_anggota:1, id\_penjaga:5 id\_buku:10,  
tanggal\_peminjaman:'08/07/2022', tanggal\_pengembalian:'08/08/2022',  
denda:0, status\_peminjaman:'dipinjam'} )
- ❖ Hapus 1 document / data :  
db.tb\_peminjaman.remove( {id\_peminjaman:11} )
- ❖ Update 1 document / data :  
db.tb\_peminjaman.update( {id\_peminjaman:11} {\$set:{id\_buku:1}} )

## PENCARIAN

- ❖ Cari like:  
db.tb\_peminjaman.find(  
"tanggal\_pengembalian": {\$regex: "/.\* 02 .\*/"}  
)
- ❖ Cari isi jumlah seluruh data tabel :  
db.tb\_buku.find().size()
- ❖ Cari isi seluruh data tabel :  
db.tb\_buku.find()
- ❖ Cari hanya 1 isi data tabel, menampilkan data dengan (id\_buku:1) :  
db.tb\_buku.findOne( {id\_buku:1} )
- ❖ Cari data tabel, limit menampilkan hanya 3 data, dari ke-1 sampai ke-3 :  
db.tb\_buku.find().limit(3)
- ❖ Cari data tabel, skip 2 data, menampilkan data mulai dari data ke-3 :  
db.tb\_buku.find().skip(2)

- ❖ Cari data tabel, limit 5 data, skip 1 data, menampilkan data dari ke-2 sampai ke-6 :  
`db.tb_buku.find().limit(5).skip(1)`
- ❖ Cari data tabel buku dengan nama\_kategori Komputer, menampilkan secara Descending (terbalik) :  
`db.tb_buku.find({nama_kategori:'Komputer'}).sort({'id_buku':-1})`
- ❖ Cari data tabel buku dengan nama\_kategori Komputer, menampilkan secara Ascending (berurut) :  
`db.tb_buku.find({nama_kategori:'Komputer'}).sort({'id_buku':1})`
- ❖ Cari data tabel, menampilkan tahun\_terbit kurang dari 2018 :  
`db.tb_buku.find( {tahun_terbit:{$lt:2018}} )`
- ❖ Cari data tabel, menampilkan tahun\_terbit kurang dari sama dengan 2018 :  
`db.tb_buku.find( {tahun_terbit:{$lte:2018}} )`
- ❖ Cari data tabel, menampilkan tahun\_terbit lebih dari 2018 :  
`db.tb_buku.find( {tahun_terbit:{$gt:2018}} )`
- ❖ Cari data tabel, menampilkan tahun\_terbit lebih dari sama dengan 2018 :  
`db.tb_buku.find( {tahun_terbit:{$gte:2018}} )`
- ❖ Cari data tabel, menampilkan yang bukan tahun\_terbit 2018 :  
`db.tb_buku.find( {tahun_terbit:{$ne:2018}} )`
- ❖ Cari data tabel, dengan kondisi AND (semua true), menampilkan data yang bernilai (judul\_buku:'Bulan') dan (tahun\_terbit:2015) :  
`db.tb_buku.find( {$and: [{judul_buku:'Bulan'},{tahun_terbit:2015}] } )`
- ❖ Cari data tabel, dengan kondisi OR (salah satu true), menampilkan semua data yang bernilai (judul\_buku:'Bulan') atau (tahun\_terbit:2016) :  
`db.tb_buku.find( {$or: [{judul_buku:'Bulan'},{tahun_terbit:2016}] } )`

## AGGREGATE

- ❖ Aggregate semua data tahun\_terbit (yang sama), kemudian menampilkan dari id\_buku yang tertinggi di tahun tersebut:  
`db.tb_buku.aggregate([ {$group: {_id:'$tahun_terbit', bukuid:{$max:'$id_buku'}} } ])`
- ❖ Aggregate semua data tahun\_terbit (yang sama), kemudian menampilkan dari id\_buku yang terendah di tahun tersebut:  
`db.tb_buku.aggregate([ {$group: {_id:'$tahun_terbit', bukuid:{$min:'$id_buku'}} } ])`
- ❖ Aggregate \$match, menyaring dokumen hanya menampilkan yang terkena denda 10000 :  
`db.tb_peminjaman.aggregate([ {$match:{denda:10000}} ])`
- ❖ Aggregate, grouping, dan \$sum (menghitung total) :

```
db.tb_peminjaman.aggregate([ {$group: {_id: '$status_peminjaman',  
Denda: {$sum: '$denda'}}} ])
```

```
> db.tb_peminjaman.aggregate([ {$match: {status_peminjaman: "dikembalikan terlambat"},  
  {$group: {_id: '$status_peminjaman', Denda: {$sum: '$denda'}}  
    }  
  ])  
< { _id: 'dikembalikan terlambat', Denda: 20000 }
```

- ❖ Aggregate, menampilkan data dengan periode :

```
db.tb_peminjaman.aggregate([  
  {$match: {tanggal_peminjaman: {$gte: "02012022", $lte: "07012022"}} } ])
```

## AGGREGATE (RELASI TABEL)

- ❖ Relasi tb\_peminjaman dengan tb\_buku (menampilkan semua):

```
db.tb_peminjaman.aggregate([ {$lookup: {from: "tb_buku", localField:  
"id_buku", foreignField: "id_buku", as: "tb_buku"}} ])
```

- ❖ Relasi tb\_peminjaman dengan tb\_buku (menampilkan id 1):

```
db.tb_peminjaman.aggregate([ {$match: {id_peminjaman: 1}}, {$lookup:  
{from: "tb_buku", localField: "id_buku", foreignField: "id_buku",  
as: "tb_buku"}} ])
```

- ❖ Relasi 3 tabel (tampil semua):

```
db.tb_peminjaman.aggregate([  
  {$lookup: {from: 'tb_anggota', localField: 'id_anggota',  
foreignField: 'id_anggota', as: 'tb_anggota'}} ,  
  {$lookup: {from: 'tb_penjaga', localField: 'id_penjaga',  
foreignField: 'id_penjaga', as: 'tb_penjaga'}} ])
```

- ❖ Relasi 3 tabel (tampil data id 1) :

```
db.tb_peminjaman.aggregate([ {$match: {id_peminjaman: 1}},  
  {$lookup: {from: 'tb_anggota', localField: 'id_anggota',  
foreignField: 'id_anggota', as: 'tb_anggota'}} ,  
  {$lookup: {from: 'tb_penjaga', localField: 'id_penjaga',  
foreignField: 'id_penjaga', as: 'tb_penjaga'}} ])
```