

Praktikum Basis Data: CRUD dan Query pada MongoDB

Disusun untuk Memenuhi Matakuliah Praktikum Database

Dibimbing oleh : Sulaibatul Aslamiyah, M.Kom



Oleh:

Nama : M RAVI ACHYAR TRISTA ZHAID

Nim : 1124102193

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
SEKOLAH TINGGI ILMU KOMPUTER PGRI BANYUWANGI
2025**

**LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM**

Matakuliah : PRATIKUM DATABASE NON RELATIONAL
Oleh : M RAVI ACHYAR TRISTA ZHAID
NIM : 1124102193
Kelas : SP1 Informatika

Telah disahkan pada

Hari : Rabu
Tanggal : 7 Mei 2025

Mengetahui/ Menyetujui :

Dosen Pengampu Mata Kuliah

Sulaibatul Aslamiyah, M.Kom

NIDN. 0712058304

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga laporan praktikum ini dapat diselesaikan sebagai bagian dari penilaian Ujian Tengah Semester pada mata kuliah *Database Non Relational*.

Laporan ini berisi hasil implementasi operasi CRUD dan comparison query pada MongoDB untuk pengelolaan data produk. Praktikum ini bertujuan untuk meningkatkan pemahaman konsep basis data NoSQL serta keterampilan teknis dalam pengelolaan data berbasis dokumen.

Penulis mengucapkan terima kasih kepada Ibu Sulaibatul Aslamiyah, S.Kom., M.Kom. selaku dosen pengampu mata kuliah *Database Non Relational*, atas bimbingan dan arahan yang telah diberikan. Semoga laporan ini dapat memberikan manfaat bagi semua pihak yang membacanya.

Banyuwangi, 7 Mei 2025

M Ravi Achyar Trista Zhaid

1.1 Tujuan

- Menerapkan operasi CRUD (Create, Read, Update, Delete) pada MongoDB.
- Memahami cara melakukan pencarian data menggunakan comparison query seperti \$gt, \$lt, \$in, dan lainnya.
- Memahami pengelolaan data non-relasional berbentuk dokumen di MongoDB.

1.2 Implementasi

1. Membuat database dengan nama db_toko dan membuat collection dengan nama produk.

```
test> use db_toko
switched to db db_toko
db_toko> db.createCollection("produk")
{ ok: 1 }
```

Deskripsi : Perintah **use db_toko** berfungsi untuk membuat atau memilih database bernama **db_toko**. Jika database belum ada, MongoDB akan otomatis membuatnya saat data dimasukkan. Perintah **db.createCollection("produk")** digunakan untuk membuat koleksi produk secara eksplisit.

2. Data yang dikelola meliputi informasi seperti:

- id, nama_produk
- kategori
- stok
- harga
- status (aktif / nonaktif)

```
db_toko> db.produk.insertOne({
... _id: "P001",
... nama_produk: "Mouse Wireless Logitech",
... kategori: "Aksesori",
... stok: 35,
... harga: 75000,
... status: "aktif"
... })
{ acknowledged: true, insertedId: 'P001' }
```

Deskripsi : Perintah ini digunakan untuk memasukkan satu dokumen produk ke dalam koleksi **"produk"**. Produk ini memiliki informasi lengkap seperti **ID, nama, kategori, jumlah stok, harga, dan status**.

3. Melakukan operasi :
 - a. **Insert** minimal 5 data produk dengan variasi kategori dan harga.

```

db_toko> db.produk.insertMany([
... {
...   _id: "P002",
...   nama_produk: "Laptop Lenovo ThinkPad",
...   kategori: "Laptop",
...   stok: 12,
...   harga: 8500000,
...   status: "aktif"
... },
... {
...   _id: "P003",
...   nama_produk: "Keyboard Mechanical",
...   kategori: "Aksesoris",
...   stok: 20,
...   harga: 300000,
...   status: "aktif"
... },
... {
...   _id: "P004",
...   nama_produk: "RAM DDR4 8GB",
...   kategori: "Komponen",
...   stok: 8,
...   harga: 420000,
...   status: "aktif"
... },
... {
...   _id: "P005",
...   nama_produk: "Monitor AOC 24 inch",
...   kategori: "Periferal",
...   stok: 18,
...   harga: 1250000,
...   status: "nonaktif"
... },
... {
...   _id: "P006",
...   nama_produk: "SSD Samsung 512GB",
...   kategori: "Komponen",
...   stok: 11,
...   harga: 670000,
...   status: "aktif"
... },
... {
...   _id: "P007",
...   nama_produk: "Cooling Fan CPU",
...   kategori: "Aksesori",
...   stok: 20,
...   harga: 200000,
...   status: "aktif"
... }
... ])

{
  acknowledged: true,
  insertedIds: {
    '0': 'P002',
    '1': 'P003',
    '2': 'P004',
    '3': 'P005',
    '4': 'P006',
    '5': 'P007'
  }
}

```

Deskripsi : **insertMany** digunakan untuk memasukkan beberapa dokumen sekaligus ke dalam koleksi. Ini mempercepat proses pengisian data awal, terutama untuk simulasi data toko.

b. Read/Query seluruh data dan data dengan kondisi tertentu

a) Produk dengan harga lebih dari Rp 50.000

```

db_toko> db.produk.find({ harga: { $gt: 50000 } })
[
  {
    _id: 'P001',
    nama_produk: 'Mouse Wireless Logitech',
    kategori: 'Aksesori',
    stok: 35,
    harga: 75000,
    status: 'aktif'
  },
  {
    _id: 'P002',
    nama_produk: 'Laptop Lenovo ThinkPad',
    kategori: 'Laptop',
    stok: 12,
    harga: 8500000,
    status: 'aktif'
  },
  {
    _id: 'P003',
    nama_produk: 'Keyboard Mechanical',
    kategori: 'Aksesoris',
    stok: 20,
    harga: 300000,
    status: 'aktif'
  }
]

```

```

{
  id: 'P004',
  nama_produk: 'RAM DDR4 8GB',
  kategori: 'Komponen',
  stok: 8,
  harga: 420000,
  status: 'aktif'
},
{
  id: 'P005',
  nama_produk: 'Monitor AOC 24 inch',
  kategori: 'Periferal',
  stok: 18,
  harga: 1250000,
  status: 'nonaktif'
},
{
  id: 'P006',
  nama_produk: 'SSD Samsung 512GB',
  kategori: 'Komponen',
  stok: 11,
  harga: 670000,
  status: 'aktif'
},
{
  id: 'P007',
  nama_produk: 'Cooling Fan CPU',
  kategori: 'Aksesori',
  stok: 20,
  harga: 200000,
  status: 'aktif'
}
]

```

Deskripsi : Menggunakan operator **\$gt (greater than)** untuk menampilkan produk yang memiliki harga di atas Rp 50.000.

b) Produk dengan stok kurang dari 20

```
db_toko> db.produk.find({ stok: {$lt: 20} })
[
  {
    _id: 'P002',
    nama_produk: 'Laptop Lenovo ThinkPad',
    kategori: 'Laptop',
    stok: 12,
    harga: 8500000,
    status: 'aktif'
  },
  {
    _id: 'P004',
    nama_produk: 'RAM DDR4 8GB',
    kategori: 'Komponen',
    stok: 8,
    harga: 420000,
    status: 'aktif'
  },
  {
    _id: 'P005',
    nama_produk: 'Monitor AOC 24 inch',
    kategori: 'Periferal',
    stok: 18,
    harga: 1250000,
    status: 'nonaktif'
  },
  {
    _id: 'P006',
    nama_produk: 'SSD Samsung 512GB',
    kategori: 'Komponen',
    stok: 11,
    harga: 670000,
    status: 'aktif'
  }
]
```

Deskripsi : Menggunakan operator \$lt (less than) untuk memfilter produk dengan stok rendah dari 20.

c) Update

- mengubah harga salah satu produk

```
db_toko> db.produk.updateOne(
... { _id: "P003" },
... {$set: { harga: 60000 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Deskripsi : Menggunakan \$set untuk memperbarui harga menjadi **60000** pada produk dengan **_id: "P003"**

- menambahkan kategori pada salah satu produk.

```

db_toko> db.produk.updateOne(
... { _id: "P003" },
... { $set: { kategori: ["Aksesori", "Gaming"] } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

Deskripsi : Menggunakan **\$set** untuk mengganti kategori menjadi array baru, tanpa mengubah field lainnya.

d) **Delete** menghapus 1 produk dari database berdasarkan **_id**.

```

db_toko> db.produk.deleteOne({ _id: "P005" })
{ acknowledged: true, deletedCount: 1 }

```

Deskripsi : Menghapus satu data produk berdasarkan **_id** menggunakan **deleteOne**.

e) **Comparison Query**

- menampilkan produk dengan harga di antara Rp 30.000 hingga Rp 70.000

```

db_toko> db.produk.find({
... harga: { $gte: 30000, $lte: 70000 }
... })
[
  {
    _id: 'P003',
    nama_produk: 'Keyboard Mechanical',
    kategori: [ 'Aksesori', 'Gaming' ],
    stok: 20,
    harga: 60000,
    status: 'aktif'
  }
]

```

Deskripsi : Menggunakan operator **\$gte** dan **\$lte** untuk menampilkan produk dengan harga dalam rentang tertentu.

- menampilkan produk dengan kategori tertentu dengan menggunakan **\$in**.

```

db_toko> db.produk.find({
... kategori: { $in: ["Aksesori", "Laptop"] }
... })
[
  {
    _id: 'P001',
    nama_produk: 'Mouse Wireless Logitech',
    kategori: 'Aksesori',
    stok: 35,
    harga: 75000,
    status: 'aktif'
  },
  {
    _id: 'P002',
    nama_produk: 'Laptop Lenovo ThinkPad',
    kategori: 'Laptop',
    stok: 12,
    harga: 8500000,
    status: 'aktif'
  },
  {
    _id: 'P003',
    nama_produk: 'Keyboard Mechanical',
    kategori: [ 'Aksesori', 'Gaming' ],
    stok: 20,
    harga: 60000,
    status: 'aktif'
  },
  {
    _id: 'P007',
    nama_produk: 'Cooling Fan CPU',
    kategori: 'Aksesori',
    stok: 20,
    harga: 200000,
    status: 'aktif'
  }
]

```

Deskripsi : Operator **\$in** digunakan untuk mencari produk yang memiliki kategori tertentu.

1.3 Kesimpulan

- MongoDB memberikan keleluasaan dalam penyimpanan data non-relasional melalui format dokumen.
- Operasi CRUD sangat intuitif dan fleksibel, cocok untuk pengembangan aplikasi berbasis data dinamis.
- Comparison query memudahkan pencarian dan analisis data berdasarkan kondisi tertentu, yang sangat relevan untuk sistem manajemen produk toko