

LAPORAN PRAKTIKUM

PRAKTIKUM DATABASE PERTEMUAN 7

Disusun untuk Memenuhi Matakuliah Praktikum {Nama Matakuliah}

Dibimbing oleh : Sulaibatul Aslamiyah, M.Kom



Oleh:

RAFLI RAHMAN.EFENDY

1124102162

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

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM

Matakuliah : Praktikum Database
Oleh : Rafli Rahman.Efendy
NIM : 1124102162

Telah disahkan pada

Hari : SELASA
Tanggal : 27/05/2025

Mengetahui/ Menyetujui :

Dosen Pengampu Mata Kuliah

Sulaibatul Aslamiyah, M.Kom

NIDN. 0712058304

KATA PENGANTAR

Berisikan kata pengantar yang dituliskan oleh mahasiswa untuk laporan praktikum matakuliah yang diikuti.

MODUL PRAKTIKUM 1

1.1 Capaian Praktikum Pertemuan 1

Mahasiswa dapat mencatat hasil praktikum dengan baik dan membuat laporan praktikum yang mencakup tujuan, alat, bahan, prosedur, dan hasil praktikum.

1.2 Indikator Capaian

- Mahasiswa mampu menerapkan model relasi One-to-One dengan Embedded Document.
- Mahasiswa mampu menerapkan model relasi One-to-Many dengan Embedded Document.
- Mahasiswa mampu menerapkan model relasi One-to-Many dengan References Document

1.3 Landasan Teori

Databases adalah kumpulan data yang terstruktur dan disimpan secara terorganisir untuk memudahkan pengelolaan dan pencarian informasi

1.4 Pelaksanaan Praktikum

Model data embedded cocok untuk digunakan ketika : Antar document saling ketergantungan Kita jarang melakukan perubahan isi data pada embedded document Embedded document selalu dibutuhkan ketika mengambil data document

1.4.1 Percobaan Pertama

a.Script / Setting Program

```
PertemuanTujuh> show dbs
admin          40.00 KiB
config         96.00 KiB
local          72.00 KiB
matakuliah     80.00 KiB
soallatihan    184.00 KiB
PertemuanTujuh> use PertemuanTujuh
already on db PertemuanTujuh
PertemuanTujuh> db.createCollection("Penjual")
{ ok: 1 }
PertemuanTujuh> db.createCollection("Kota")
{ ok: 1 }
PertemuanTujuh> |
```

```
PertemuanTujuh> db.Penjual.insertOne(
... {
...   _id:"P1",
...   nama:"SupKen",
...   alamat:"Jl Borobudur No 143"
... });
{ acknowledged: true, insertedId: 'P1' }
PertemuanTujuh> |
```

```
PertemuanTujuh> db.Kota.insertOne(
... {
...   _id:"K1",
...   nama:"Surayabaya",
...   Provinsi:"Jawa Timur"}});
{ acknowledged: true, insertedId: 'K1' }
PertemuanTujuh> |
```

```
db.Penjual.insertOne(
{
  _id: "P1",
  nama: "SupKen",
  alamat: "Jl Borobudur No 143"
})
```

```
);
```

A. Penjelasan Code ini:

- db.Penjual Menunjukkan bahwa Anda akan melakukan operasi pada koleksi bernama Penjual.
- insertOne({...}) Memasukkan satu dokumen ke dalam koleksi tersebut.
- Objek yang dimasukkan:
 - _id: "P1" ID unik untuk penjual ini. Jika tidak ditentukan, MongoDB akan membuat ID sendiri.
 - nama: "SupKen" Nama penjual.
 - alamat: "Jl Borobudur No 143" → Alamat penjual.

Hasil nya: " { acknowledged: true, insertedId: 'P1' }"

B. Penjelasan Code ini:

```
db.Kota.insertOne(  
  {  
    _id: "K1",  
    nama: "Surayabaya",  
    Provinsi: "Jawa Timur"  
  }  
);
```

- db.Kota Operasi dilakukan pada koleksi Kota.
- Objek yang dimasukkan:
 - _id: "K1" ID unik untuk kota ini.
 - nama: "Surayabaya" Nama kota. (kemungkinan salah ketik, seharusnya "Surabaya").
 - Provinsi: "Jawa Timur" Provinsi tempat kota berada.

```
PertemuanTujuh> db.Penjual.insertOne({  
...   _id: "P2", // ganti dengan ID yang belum digunakan  
...   nama: "SupKen",  
...   Provinsi: "Jawa Timur",  
...   Kota: {  
...     nama: "Surabaya", // juga perbaiki typo dari "Surayabaya"  
...     Provinsi: "Jawa Timur"  
...   }  
... });  
...  
{ acknowledged: true, insertedId: 'P2' }  
PertemuanTujuh> |
```

Ringkasan nya:

Koleksi	Data yang Disimpan
---------	--------------------

Koleksi	<code>_id: "P1", nama: "SupKen", alamat: "Jl Borobudur No 143"</code>
Kota	<code>_id: "K1", nama: "Surabaya", Provinsi: "Jawa Timur"</code>

```
PertemuanTujuh> db.Penjual.find().pretty();
[
  { _id: 'P1', nama: 'SupKen', alamat: 'Jl Borobudur No 143' },
  {
    _id: 'P2',
    nama: 'SupKen',
    Provinsi: 'Jawa Timur',
    Kota: { nama: 'Surabaya', Provinsi: 'Jawa Timur' }
  }
]
PertemuanTujuh> |
```

1.4.2 Percobaan 2

```
PertemuanTujuh> db.Penjual.insertOne( { _id: "K1", nama: "SupKen", Provinsi: "Jawa Timur", Kota: { nama: "Surabaya", Provinsi: "Jawa Timur" } });
{ acknowledged: true, insertedId: 'K1' }
PertemuanTujuh> |
```

```
PertemuanTujuh> db.Produk.insertMany([
...   {
...     _id: "S3",
...     nama: "RAM DDR4 8GB 3200MHz",
...     harga: 600000
...   },
...   {
...     _id: "S4",
...     nama: "Monitor LED Samsung 24 Inch",
...     harga: 1750000
...   }
... ]);
{ acknowledged: true, insertedIds: { '0': 'S3', '1': 'S4' } }
PertemuanTujuh> |
```

```

PertemuanTujuh> db.Penjual.insertOne({
...   _id: "P3",
...   nama: "Dina Rahayu",
...   alamat: "Jl Melati No 88",
...   Produk: [
...     {
...       _id: "S3",
...       id_penjual: "P3",
...       nama: "RAM DDR4 8GB 3200MHz",
...       harga: 600000
...     },
...     {
...       _id: "S4",
...       id_penjual: "P3",
...       nama: "Monitor LED Samsung 24 Inch",
...       harga: 1750000
...     }
...   ]
... });
{ acknowledged: true, insertedId: 'P3' }
PertemuanTujuh> |

```

A. Penjelasan Code ini:

```

db.Penjual.insertOne({
  _id: "K1",
  nama: "SupKen",
  Provinsi: "Jawa Timur",
  Kota: {
    nama: "Surabaya",
    Provinsi: "Jawa Timur"
  }
});

```

- Menyisipkan dokumen ke dalam koleksi Penjual.
- Kota di sini merupakan embedded document (dokumen bersarang) yang menyimpan informasi kota penjual secara langsung di dalam dokumen penjual.
- Field:
 - `_id`: ID penjual.
 - `nama`: Nama penjual.
 - `Provinsi`: Provinsi tempat tinggal (juga terdapat di dalam Kota, agak redundan).
 - `Kota`: Menyimpan nama dan provinsi kota (nested object).

B. Penjelasan Code ini:

```
db.Produk.insertMany([
  {
    _id: "S3",
    nama: "RAM DDR4 8GB 3200MHz",
    harga: 600000
  },
  {
    _id: "S4",
    nama: "Monitor LED Samsung 24 Inch",
    harga: 1750000
  }
]);
```

- Menambahkan dua produk sekaligus ke dalam koleksi Produk.
- insertMany digunakan ketika ingin menyisipkan lebih dari satu dokumen.
- Masing-masing dokumen produk memiliki:
 - _id: ID produk unik.
 - nama: Nama produk.
 - harga: Harga produk dalam satuan rupiah.

```
PertemuanTujuh> db.Penjual.find().pretty();
[
  {
    _id: 'P1', nama: 'SupKen', alamat: 'Jl Borobudur No 143' },
  {
    _id: 'P2',
    nama: 'SupKen',
    Provinsi: 'Jawa Timur',
    Kota: { nama: 'Surabaya', Provinsi: 'Jawa Timur' }
  },
  {
    _id: 'K1',
    nama: 'SupKen',
    Provinsi: 'Jawa Timur',
    Kota: { nama: 'Surabaya', Provinsi: 'Jawa Timur' }
  },
  {
    _id: 'P3',
    nama: 'Dina Rahayu',
    alamat: 'Jl Melati No 88',
    Produk: [
      {
        _id: 'S3',
        id_penjual: 'P3',
        nama: 'RAM DDR4 8GB 3200MHz',
        harga: 600000
      },
      {
        _id: 'S4',
        id_penjual: 'P3',
        nama: 'Monitor LED Samsung 24 Inch',
        harga: 1750000
      }
    ]
  }
]
```

1.4.3 1.4.2 Percobaan 3

```
PertemuanTujuh> db.Penjual.insertOne({ _id: "M1", nama: "SupKen", alamat: "Jl Borobudur No 143" });
{ acknowledged: true, insertedId: 'M1' }
PertemuanTujuh> |
```

```
PertemuanTujuh> db.Produk.insertMany([
...   {
...     _id: "LP1",
...     nama: "RAM DDR4 8GB 3200MHz",
...     harga: 600000
...   },
...   {
...     _id: "LP2",
...     nama: "Monitor LED Samsung 24 Inch",
...     harga: 1750000
...   }
... ]);
...
{ acknowledged: true, insertedIds: { '0': 'LP1', '1': 'LP2' } }
PertemuanTujuh>
```

```
PertemuanTujuh> db.Penjual.aggregate([
...   { $unwind: "$Produk" },
...   {
...     $project: {
...       _id: 0,
...       nama_penjual: "$nama",
...       nama_produk: "$Produk.nama",
...       harga: "$Produk.harga"
...     }
...   }
... ]);
...
[
  {
    nama_penjual: 'Dina Rahayu',
    nama_produk: 'RAM DDR4 8GB 3200MHz',
    harga: 600000
  },
  {
    nama_penjual: 'Dina Rahayu',
    nama_produk: 'Monitor LED Samsung 24 Inch',
    harga: 1750000
  }
]
PertemuanTujuh> |
```

A. Penjelasan Code ini:

```
db.Penjual.insertOne({
  _id: "M1",
  nama: "SupKen",
  alamat: "Jl Borobudur No 143"
});
```

Penjelasan:

- Fungsi: Menyisipkan dokumen baru ke dalam koleksi Penjual.
- Dokumen:
 - `_id`: Menentukan ID unik untuk dokumen, disini "M1".
 - `nama`: Nama penjual, disini "SupKen".
 - `alamat`: Alamat penjual, disini "Jl Borobudur No 143".
- Hasil: Mengembalikan objek dengan atribut `acknowledged` dan `insertedId`, yang menunjukkan bahwa dokumen telah berhasil disisipkan.

B. Penjelasan Code ini:

```
db.Produk.insertMany([
  {
    _id: "LP1",
    nama: "RAM DDR4 8GB 3200MHz",
    harga: 600000
  },
  {
    _id: "LP2",
    nama: "Monitor LED Samsung 24 Inch",
    harga: 1750000
  }
]);
```

Penjelasan:

- Fungsi: Menyisipkan beberapa dokumen ke dalam koleksi Produk.
- Dokumen:
 - Dokumen pertama memiliki `_id`: "LP1", nama: "RAM DDR4 8GB 3200MHz", dan harga: 600000.
 - Dokumen kedua memiliki `_id`: "LP2", nama: "Monitor LED Samsung 24 Inch", dan harga: 1750000.
- Hasil: Mengembalikan objek dengan `acknowledged` dan `insertedIds`, menunjukkan bahwa dokumen bergerak berhasil dan menyertakan ID yang disisipkan.

3. Aggregating Data with aggregate

- Fungsi: Menghasilkan data teragregasi dari koleksi Penjual.
- Langkah:
 - `$unwind`: Menguraikan array Produk sehingga setiap elemen dalam array menjadi dokumen terpisah.
 - `$project`: Memproyeksikan hasil akhir dengan menentukan field yang ingin ditampilkan:
 - `nama_penjual`: Nama penjual.
 - `nama_produk`: Nama produk dari array Produk.
 - `harga`: Harga dari produk.
 - `_id`: Diatur ke 0 agar tidak ditampilkan di hasil.
- Hasil: Keluaran adalah daftar objek dengan `nama_penjual`, `nama_produk`, dan `harga`.

Kesimpulan

Kode-kode di atas menunjukkan bagaimana cara menambahkan data ke dalam koleksi dalam MongoDB melalui fungsi `insertOne` dan `insertMany`, serta bagaimana mengolah dan menampilkan data dengan menggunakan metode `aggregate`.

1. Pada relasi One-to-Many, kapankah sebaiknya kita memilih pemakaian antara model data Embedded Document dengan References Document?
2. Diketahui sebuah relasi seperti gambar berikut :



- No. Anggota
- Nama Anggota
- Tgl. Lahir
- Alamat
- Tgl. Masuk

- Tgl. Pinjam
- Tgl. Kembali

- Kode Buku
- Judul
- Pengarang
- Penerbit
- Tahun Terbit

- a. Buat semua collection yang dibutuhkan dalam kasus diatas beserta isikan minimal 1 document untuk anggota dan buku dan 3 document untuk pinjam!
- b. Buatkan juga query untuk menampilkan isi document pada collection pinjam!

Latihan Soal Modul 6 Pertemuan 7

1.Hasil Percobaan 1

Pada relasi One-to-Many, kapankah sebaiknya kita memilih pemakaian antara model data Embedded Document dengan References Document?

```

PertemuanTujuh> show dbs
PertemuanTujuh  128.00 KiB
admin            40.00 KiB
config          108.00 KiB
local           72.00 KiB
matakuliah      80.00 KiB
PertemuanTujuh> use soallatihan
switched to db soallatihan
soallatihan> db.createCollection("anggota")
{ ok: 1 }
soallatihan> db.createCollection("pinjam")
{ ok: 1 }
soallatihan> db.createCollection("buku")
{ ok: 1 }
soallatihan> |
  
```

A.Hasil Uji Coba/Hasil Nya

- Use = Untuk masuk kedalam databases
- db.Bukul.insertOne ({Isi data pada kolom ini}) = Funsinya adalah untuk membuat dokumen pada collection (Buku) dan hanya satu data karena syntexnya One.
- db.Anggota.insertOne ({Isi data pada kolom ini}) = Funsinya adalah untuk membuat dokumen pada collection (Anggota) dan hanya satu data karena syntexnya One.
- db.Pinjam.insertMany({Isi data pada kolom ini}) = Funsinya adalah untuk membuat dokumen pada collection (Pinjam).

Kesimpulan

Kesimpulan Percobaan 1

Dengan study kasus ini Saya bisa membuat Relasi One-to-One adalah relasi dimana suatu baris tabel A hanya berhubungan dengan suatu baris tabel B. Kita dapat menerapkan model data Embedded Document untuk mempresentasikan relasi One-to-One tersebut pada database Non Relational.

Kesimpulan Percobaan 2

Dengan study kasus ini Saya bisa membuat Mirip dengan implementasi embedded document pada relasi One-to-One, pada kasus One-to-Many, document dengan relasi "Many" akan di-embedded kedalam document dengan relasi "One".

Kesimpulan Percobaan 3

Dengan study kasus ini Saya bisa membuat Untuk menampilkan data kita dapat menggunakan method \$lookup dengan aggregation.

Kesimpulan Percobaan 4

Dengan study kasus ini Saya bisa membuat Collection dan juga menambahkan dokumen menggunakan one to one dan many to one, dan juga query untuk menampilkan isi document pada collection

BAB 1

PENUTUP

Kesimpulan

teori MySQL adalah Structured Query Language (SQL) sebagai bahasa interaktif untuk mengelola data. MySQL adalah sistem manajemen basis data relasional (RDBMS) yang menggunakan SQL untuk menjalankan fungsinya..

Saran

Saran saya tidak ada perubahan pembelajaran sama seperti di Semester 1 dan masih masuk di praktikum kali kecuali pada pengisian laporan praktikum database ke word itu yang perlu di tanyakan dan cara pengisian laporan untuk kedepannya.

DAFTAR PUSTAKA

Tuliskan rujukan yang anda gunakan baik website maupun buku seperti contoh dibawah.

1. Tim Asisten Dosen. 2014. Modul 1 Pengenalan Sistem Operasi, Ide Visual C++, Dan Algoritma Pemrograman. Malang: Universitas Negeri Malang.
2. Program Konversi Suhu (online)
<http://bondanoky.blogspot.com/2012/10/program-konversi-suhu-c.html>. Di akses 8 September.

LAMPIRAN

Berisikan syntax atau gambar yang dibutuhkan dalam tiap pertemuan praktikum.