

LAPORAN PRAKTIKUM

PRAKTIKUM DATABASE PERTEMUAN 9

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 Aslamiah, 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

Soal Pertanyaan Dari Modul 7

5	Danu	Citra Digital	75
6	Zahra	Citra Digital	90
7	Danu	Data Mining	95
8	Sidqia	Struktur Data	95
9	Kiki	Struktur Data	70
10	Maira	Citra Digital	80

Dari 10 document tersebut coba lakukan aggregation untuk mendapatkan data :

- Berapa peserta tiap matakuliah ?
- Berapa nilai tertinggi tiap matakuliah ?
- Berapa nilai terendah tiap matakuliah ?
- Berapa nilai rata-rata tiap matakuliah ?

- Buat Collection nilaiMHS02 di MongoDB** Jalankan perintah berikut di MongoDB :

Dengan menggunakan Perintah `db.nilaiMHS02.find()` atau `db.nilaiMHS02.find().pretty()`

```
[
  { _id: 1, nama: 'Danu', matkul: 'Struktur Data', nilai: 80 },
  { _id: 2, nama: 'Zahra', matkul: 'Struktur Data', nilai: 85 },
  { _id: 3, nama: 'Maira', matkul: 'Data Mining', nilai: 90 },
  { _id: 4, nama: 'Kiki', matkul: 'Citra Digital', nilai: 75 },
  { _id: 5, nama: 'Danu', matkul: 'Citra Digital', nilai: 75 },
  { _id: 6, nama: 'Zahra', matkul: 'Citra Digital', nilai: 90 },
  { _id: 7, nama: 'Danu', matkul: 'Data Mining', nilai: 95 },
  { _id: 8, nama: 'Sidqia', matkul: 'Struktur Data', nilai: 95 },
  { _id: 9, nama: 'Kiki', matkul: 'Struktur Data', nilai: 70 },
  { _id: 10, nama: 'Maira', matkul: 'Citra Digital', nilai: 80 },
  { _id: 11, nama: 'Raka', matkul: 'Struktur Data', nilai: 79 },
  { _id: 12, nama: 'Tari', matkul: 'Citra Digital', nilai: 90 },
  { _id: 13, nama: 'Budi', matkul: 'Data Mining', nilai: 65 },
  { _id: 14, nama: 'Rina', matkul: 'Struktur Data', nilai: 43 },
  { _id: 15, nama: 'Adit', matkul: 'Citra Digital', nilai: 23 },
  { _id: 16, nama: 'Lala', matkul: 'Data Mining', nilai: 14 },
  { _id: 17, nama: 'Gilang', matkul: 'Citra Digital', nilai: 33 },
  { _id: 18, nama: 'Indah', matkul: 'Struktur Data', nilai: 99 }
]
```

```

NonRelationalPTM11> db.nilaiMHS02.find()
[
  { _id: 1, nama: 'Danu', matkul: 'Struktur Data', nilai: 80 },
  { _id: 2, nama: 'Zahra', matkul: 'Struktur Data', nilai: 85 },
  { _id: 3, nama: 'Maira', matkul: 'Data Mining', nilai: 90 },
  { _id: 4, nama: 'Kiki', matkul: 'Citra Digital', nilai: 75 },
  { _id: 5, nama: 'Danu', matkul: 'Citra Digital', nilai: 75 },
  { _id: 6, nama: 'Zahra', matkul: 'Citra Digital', nilai: 90 },
  { _id: 7, nama: 'Danu', matkul: 'Data Mining', nilai: 95 },
  { _id: 8, nama: 'Sidqia', matkul: 'Struktur Data', nilai: 95 },
  { _id: 9, nama: 'Kiki', matkul: 'Struktur Data', nilai: 70 },
  { _id: 10, nama: 'Maira', matkul: 'Citra Digital', nilai: 80 },
  { _id: 11, nama: 'Raka', matkul: 'Struktur Data', nilai: 79 },
  { _id: 12, nama: 'Tari', matkul: 'Citra Digital', nilai: 90 },
  { _id: 13, nama: 'Budi', matkul: 'Data Mining', nilai: 65 },
  { _id: 14, nama: 'Rina', matkul: 'Struktur Data', nilai: 43 },
  { _id: 15, nama: 'Adit', matkul: 'Citra Digital', nilai: 23 },
  { _id: 16, nama: 'Lala', matkul: 'Data Mining', nilai: 14 },
  { _id: 17, nama: 'Gilang', matkul: 'Citra Digital', nilai: 33 },
  { _id: 18, nama: 'Indah', matkul: 'Struktur Data', nilai: 99 }
]
NonRelationalPTM11> |

NonRelationalPTM11> db.nilaiMHS02.find().pretty()
[
  { _id: 1, nama: 'Danu', matkul: 'Struktur Data', nilai: 80 },
  { _id: 2, nama: 'Zahra', matkul: 'Struktur Data', nilai: 85 },
  { _id: 3, nama: 'Maira', matkul: 'Data Mining', nilai: 90 },
  { _id: 4, nama: 'Kiki', matkul: 'Citra Digital', nilai: 75 },
  { _id: 5, nama: 'Danu', matkul: 'Citra Digital', nilai: 75 },
  { _id: 6, nama: 'Zahra', matkul: 'Citra Digital', nilai: 90 },
  { _id: 7, nama: 'Danu', matkul: 'Data Mining', nilai: 95 },
  { _id: 8, nama: 'Sidqia', matkul: 'Struktur Data', nilai: 95 },
  { _id: 9, nama: 'Kiki', matkul: 'Struktur Data', nilai: 70 },
  { _id: 10, nama: 'Maira', matkul: 'Citra Digital', nilai: 80 },
  { _id: 11, nama: 'Raka', matkul: 'Struktur Data', nilai: 79 },
  { _id: 12, nama: 'Tari', matkul: 'Citra Digital', nilai: 90 },
  { _id: 13, nama: 'Budi', matkul: 'Data Mining', nilai: 65 },
  { _id: 14, nama: 'Rina', matkul: 'Struktur Data', nilai: 43 },
  { _id: 15, nama: 'Adit', matkul: 'Citra Digital', nilai: 23 },
  { _id: 16, nama: 'Lala', matkul: 'Data Mining', nilai: 14 },
  { _id: 17, nama: 'Gilang', matkul: 'Citra Digital', nilai: 33 },
  { _id: 18, nama: 'Indah', matkul: 'Struktur Data', nilai: 99 }
]
NonRelationalPTM11> |

```

2. **Aggregation** Queries Setelah collection nilaiMHS02 dibuat, kamu bisa menggunakan query berikut untuk menjawab pertanyaan :

1. **Berapa peserta tiap matakuliah?**

```

db.nilaiMHS02.aggregate([
  {
    $group: {
      _id: "$matkul",
      jumlahPeserta: { $sum: 1 }
    }
  }
])

```

```

NonRelationalPTM11> db.nilaiMHS02.aggregate([
...   {
...     $group: {
...       _id: "$matkul",
...       jumlahPeserta: { $sum: 1 }
...     }
...   }
... ])
...
[
  { _id: 'Struktur Data', jumlahPeserta: 7 },
  { _id: 'Data Mining', jumlahPeserta: 4 },
  { _id: 'Citra Digital', jumlahPeserta: 7 }
]

```

Artinya:

- Dokumen yang memiliki nilai matkul yang sama akan dikumpulkan ke dalam satu grup.
- Dalam setiap grup, akan dihitung jumlah dokumennya menggunakan \$sum: 1, yang pada dasarnya menjumlahkan 1 untuk setiap dokumen.

2. Berapa nilai tertinggi tiap matakuliah ?

```

db.nilaiMHS02.aggregate([
{
  $group: {
    _id: "$matkul",
    nilaiTertinggi: { $max: "$nilai" }
  }
}
])

```

```

NonRelationalPTM11> db.nilaiMHS02.aggregate([
...   {
...     $group: {
...       _id: "$matkul",
...       nilaiTertinggi: { $max: "$nilai" }
...     }
...   }
... ])
...
[
  { _id: 'Data Mining', nilaiTertinggi: 95 },
  { _id: 'Citra Digital', nilaiTertinggi: 90 },
  { _id: 'Struktur Data', nilaiTertinggi: 99 }
]

```

Artinya:

- **"_id": "\$matkul"**

Data dikelompokkan berdasarkan nama mata kuliah (matkul). Jadi semua mahasiswa dengan matkul yang sama akan masuk dalam satu grup.

- **"nilaiTertinggi": { \$max: "\$nilai" }**

Di setiap grup, MongoDB akan mencari nilai terbesar dari field nilai.

3. Berapa nilai terendah tiap matakuliah ?

```
db.nilaiMHS02.aggregate([
{
  $group: {
    _id: "$matkul",
    nilaiTerendah: { $min: "$nilai" }
  }
}]
```

```
NonRelationalPTM11> db.nilaiMHS02.aggregate([
...   {
...     $group: {
...       _id: "$matkul",
...       nilaiTerendah: { $min: "$nilai" }
...     }
...   }
... ])
...
... [
...   { _id: 'Citra Digital', nilaiTerendah: 23 },
...   { _id: 'Struktur Data', nilaiTerendah: 43 },
...   { _id: 'Data Mining', nilaiTerendah: 14 }
... ]
NonRelationalPTM11> |
```

1. aggregate([...])

Menggunakan aggregation pipeline untuk memproses data dalam beberapa tahap.

2. \$group

Mengelompokkan data berdasarkan field tertentu, dalam hal ini:

- **_id: "\$matkul"** → Data dikelompokkan berdasarkan nama mata kuliah (matkul).
- **nilaiTerendah: { \$min: "\$nilai" }** → Dari setiap kelompok (matkul), cari nilai terkecil dari field nilai.

4. Berapa nilai rata-rata tiap matakuliah ?

```
db.nilaiMHS02.aggregate([
  {
    $group: {
      _id: "$matkul",
      rataRataNilai: { $avg: "$nilai" }
    }
  }
])
```

```
NonRelationalPTM11> db.nilaiMHS02.aggregate([
... {
...   $group: {
...     _id: "$matkul",
...     rataRataNilai: { $avg: "$nilai" }
...   }
... }
... ])
...
... [
...   { _id: 'Citra Digital', rataRataNilai: 66.57142857142857 },
...   { _id: 'Struktur Data', rataRataNilai: 78.71428571428571 },
...   { _id: 'Data Mining', rataRataNilai: 66 }
... ]
```

a. aggregate([...])

Menjalankan aggregation pipeline, yaitu proses bertahap untuk menganalisis atau mengubah data.

b. \$group

Mengelompokkan dokumen berdasarkan kriteria tertentu.

Penjelasan isi \$group:

a. _id: "\$matkul"

Mengelompokkan data berdasarkan field matkul. Jadi semua mahasiswa yang mengambil mata kuliah yang sama akan dikumpulkan dalam satu grup.

b. rataRataNilai: { \$avg: "\$nilai" }

Menghitung rata-rata nilai dari semua dokumen (mahasiswa) dalam setiap grup.

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.