

DATABASE NOSQL

POKOK BAHASAN

- Instalasi MongoDB
- CRUD pada database NoSQL

TUJUAN BELAJAR:

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

- Melakukan Instalasi MongoDB
- Memahami CRUD pada database NoSQL

PENDAHULUAN

NoSQL (Not Only SQL)

adalah jenis sistem manajemen basis data yang tidak menggunakan model relasional seperti halnya SQL (Structured Query Language). NoSQL dirancang untuk menangani data yang besar dan tidak terstruktur atau semi-terstruktur, serta untuk mendukung skalabilitas horizontal dan performa tinggi.

Berikut adalah beberapa karakteristik utama dari basis data NoSQL:

1. **Tidak Relasional:** Tidak menggunakan tabel dengan baris dan kolom seperti pada basis data relasional. Struktur data yang digunakan bisa berupa dokumen, pasangan kunci-nilai, graf, atau kolom yang lebar.
2. **Bersifat Skema-Less:** NoSQL memungkinkan fleksibilitas dalam penyimpanan data tanpa harus mendefinisikan skema yang ketat. Artinya, data dapat berubah bentuk seiring waktu tanpa perlu mengubah struktur keseluruhan.
3. **Skalabilitas Horizontal:** NoSQL dirancang untuk mendukung replikasi dan partisi data dengan mudah, sehingga dapat di-skala secara horizontal, yakni dengan menambah lebih banyak server untuk menangani lebih banyak data.
4. **Kinerja Tinggi:** Basis data NoSQL umumnya dioptimalkan untuk operasi baca/tulis cepat, bahkan pada skala data yang sangat besar.
5. **Cocok untuk Big Data dan Aplikasi Terdistribusi:** NoSQL digunakan untuk aplikasi yang membutuhkan pengolahan data besar secara real-time, seperti media sosial, analitik big data, dan aplikasi IoT.

Jenis-Jenis Database NoSQL:

1. **Dokumen (Document Store):** Data disimpan dalam format dokumen (biasanya JSON, BSON, atau XML). Contoh: MongoDB, CouchDB.
2. **Key-Value Store:** Data disimpan sebagai pasangan kunci-nilai. Contoh: Redis, DynamoDB.
3. **Basis Data Kolom Lebar (Wide-Column Store):** Data disimpan dalam kolom, mirip dengan spreadsheet. Contoh: Cassandra, HBase.
4. **Basis Data Graf:** Digunakan untuk merepresentasikan hubungan antar entitas. Contoh: Neo4j, ArangoDB.

MongoDB

adalah salah satu sistem manajemen basis data NoSQL yang paling populer, yang menggunakan model penyimpanan dokumen berbasis JSON (JavaScript Object Notation) atau BSON (Binary JSON). MongoDB dikembangkan untuk mendukung kebutuhan aplikasi modern yang memerlukan fleksibilitas tinggi, kemampuan untuk menangani data besar, serta skalabilitas yang baik.

Karakteristik Utama MongoDB:

1. **Penyimpanan Berbasis Dokumen (Document-Oriented Database):**
 - MongoDB menyimpan data dalam dokumen berbentuk **JSON** atau **BSON**. Dokumen-dokumen ini dapat memiliki struktur yang fleksibel dan tidak perlu memiliki skema yang sama antar satu dengan yang lain.
 - Setiap dokumen adalah pasangan kunci-nilai, mirip seperti objek dalam pemrograman.
2. **Skema Dinamis (Schema-Less):**
 - Tidak seperti database relasional yang memerlukan skema tetap, MongoDB memungkinkan setiap dokumen dalam satu koleksi (collection) memiliki struktur yang berbeda. Ini memberikan fleksibilitas bagi pengembang untuk menyesuaikan data secara dinamis sesuai kebutuhan aplikasi.
3. **Koleksi (Collection):**
 - Dokumen-dokumen dalam MongoDB dikelompokkan dalam koleksi (collection). Koleksi ini serupa dengan tabel dalam database relasional, tetapi tanpa skema tetap.
4. **Skalabilitas Horizontal (Horizontal Scalability):**
 - MongoDB mendukung **sharding**, sebuah teknik untuk mendistribusikan data ke beberapa server (nodes). Hal ini memungkinkan MongoDB untuk menangani volume data yang sangat besar dan lalu lintas permintaan yang tinggi dengan efisien.
 - Selain itu, MongoDB juga mendukung replikasi untuk ketersediaan data yang tinggi melalui konfigurasi **replica set**.
5. **Performansi Tinggi:**
 - MongoDB dirancang untuk menangani operasi baca dan tulis secara cepat. Ini ideal untuk aplikasi yang membutuhkan respons real-time atau yang memerlukan penyimpanan data dalam volume besar.
6. **Indeksasi dan Querying yang Kuat:**
 - MongoDB mendukung berbagai jenis indeksasi untuk mempercepat pencarian data. Selain itu, MongoDB memiliki kemampuan querying yang cukup kuat untuk mencari dokumen berdasarkan berbagai kriteria (misalnya menggunakan operator perbandingan, logika, dan pencocokan pola).

7. Aggregation Framework:

- MongoDB memiliki **aggregation framework** yang memungkinkan pengguna untuk melakukan analisis data yang kompleks, seperti pengelompokan data (grouping), pengurutan (sorting), dan transformasi data melalui pipeline.

Install MongoDB di Windows

1. Download Installer MongoDB:

- Kunjungi situs resmi MongoDB dan unduh *installer* untuk Windows.
- Pilih versi yang sesuai dengan sistem operasi Anda (biasanya Windows x64).

2. Jalankan Installer:

- Buka file installer yang telah diunduh.
- Ikuti petunjuk di layar untuk menyelesaikan proses instalasi.
- Pilih "Complete" untuk *install* semua fitur MongoDB.

3. Konfigurasi Environment Variable:

- Setelah instalasi selesai, tambahkan path MongoDB ke *environment variable* agar dapat diakses dari *command prompt*.
- Buka "Control Panel" -> "System and Security" -> "System" -> "Advanced system settings" -> "Environment Variables".
- Tambahkan path ke folder `bin` MongoDB, biasanya `C:\Program Files\MongoDB\Server<versi>\bin`.

4. Mulai MongoDB:

- Buka *Command Prompt*.
- Jalankan perintah `mongod` untuk memulai server MongoDB.
- Server MongoDB sekarang berjalan di `localhost:27017`.

Contoh Data pada MongoDB

Untuk memahami bagaimana MongoDB menyimpan data, mari kita lihat contoh dokumen yang disimpan dalam koleksi MongoDB. Misalnya, kita memiliki koleksi berisi data pengguna:

```
{
  "_id": "5f1d7f3e2ab79c3b90c3a4e7",
  "nama": "Novan Junaedi",
  "email": "novan@example.com",
  "alamat": {
    "jalan": "Jalan Kebon Jeruk",
    "kota": "Jakarta",
    "kode_pos": "11530"
  },
  "telepon": "08123456789",
  "tanggal_lahir": "1990-01-01"
}
```

Dokumen di atas menunjukkan fleksibilitas MongoDB dalam menyimpan data kompleks yang mencakup subdokumen dan array. Tidak seperti database relasional yang memerlukan beberapa tabel dan hubungan antar tabel, MongoDB memungkinkan penyimpanan data yang lebih natural dan sesuai dengan struktur aplikasi.

Dalam basis data NoSQL seperti **MongoDB**, operasi **CRUD** (Create, Read, Update, Delete) dilakukan menggunakan berbagai perintah atau fungsi yang berbeda tergantung pada implementasi masing-masing jenis NoSQL. Berikut contoh implementasi CRUD pada MongoDB, karena MongoDB adalah salah satu NoSQL yang paling populer.

1. Create (Membuat Data Baru)

Operasi **Create** digunakan untuk menambahkan dokumen baru ke dalam koleksi. Di MongoDB, kita bisa menggunakan metode `insertOne()` atau `insertMany()` untuk menambahkan satu atau lebih dokumen.

Contoh:

Js

```
// Menambahkan satu dokumen ke dalam koleksi "users"
db.users.insertOne({
  name: "John Doe",
  age: 30,
  email: "john.doe@example.com"
});

// Menambahkan banyak dokumen ke dalam koleksi "users"
db.users.insertMany([
  { name: "Jane Doe", age: 25, email: "jane.doe@example.com" },
  { name: "Robert Smith", age: 40, email: "robert.smith@example.com" }
]);
```

2. Read (Membaca/Mengambil Data)

Operasi **Read** digunakan untuk mengambil data dari basis data. Kita bisa menggunakan metode `find()` untuk mengambil data berdasarkan kriteria tertentu. Jika kita ingin mendapatkan semua dokumen dalam koleksi, kita cukup menggunakan `find({})`.

Contoh:

```
// Mengambil semua dokumen dalam koleksi "users"
db.users.find({});

// Mengambil dokumen pengguna dengan nama "John Doe"
db.users.find({ name: "John Doe" });

// Mengambil pengguna yang berusia di atas 30 tahun
db.users.find({ age: { $gt: 30 } });
```

Hasil dari operasi `find()` dapat difilter lebih lanjut menggunakan berbagai operator, seperti `$gt` (lebih besar), `$lt` (lebih kecil), `$eq` (sama dengan), dan banyak lagi.

3. Update (Memperbarui Data)

Operasi `Update` digunakan untuk memperbarui dokumen yang sudah ada dalam koleksi. MongoDB memiliki metode `updateOne()`, `updateMany()`, dan `replaceOne()` untuk memperbarui data. Kita juga bisa menggunakan operator seperti `$set` untuk mengubah nilai spesifik dari dokumen.

Contoh:

```
// Memperbarui email dari pengguna dengan nama "John Doe"
db.users.updateOne(
  { name: "John Doe" },
  { $set: { email: "new.johndoe@example.com" } }
);

// Memperbarui umur semua pengguna yang bernama "Jane Doe"
db.users.updateMany(
  { name: "Jane Doe" },
  { $set: { age: 28 } }
);
```

4. Delete (Menghapus Data)

Operasi **Delete** digunakan untuk menghapus dokumen dari koleksi. Di MongoDB, kita bisa menggunakan `deleteOne()` untuk menghapus satu dokumen atau `deleteMany()` untuk menghapus beberapa dokumen yang cocok dengan kriteria.

Contoh:

```
// Menghapus satu pengguna yang bernama "John Doe"  
db.users.deleteOne({ name: "John Doe" });  
  
// Menghapus semua pengguna yang berusia di bawah 20 tahun  
db.users.deleteMany({ age: { $lt: 20 } });
```

LATIHAN

1. Buatlah database baru bernama perpustakaan. Di dalam database tersebut, buatlah sebuah koleksi bernama buku yang menyimpan informasi buku. Setiap dokumen di dalam koleksi tersebut memiliki field sebagai berikut:
 - judul (tipe string): judul buku
 - pengarang (tipe string): pengarang buku
 - tahun_terbit (tipe integer): tahun terbit buku
 - jumlah_halaman (tipe integer): jumlah halaman buku
2. Tambahkan 5 data baru pada database tersebut.
3. Perbarui data buku pada baris ke-3 menjadi judul "Pemrograman Web" untuk mengubah jumlah_halaman menjadi 500 halaman.
4. Hapus buku dengan judul "Pemrograman Web" dari koleksi buku.