

Database & Collection

Database:

Database di MongoDB adalah sebuah wadah logis dan fisik untuk menyimpan sekumpulan *collection*. Anggap saja ini sebagai direktori atau folder level tertinggi untuk proyek atau aplikasi Anda. Semua data yang berhubungan, seperti data pengguna, produk, dan pesanan untuk sebuah aplikasi *e-commerce*, idealnya disimpan dalam satu database yang sama untuk kemudahan pengelolaan dan isolasi data antar proyek.

- Kontainer untuk Collections: Fungsi utamanya adalah mengelompokkan *collections* yang saling terkait. Hirarkinya sederhana: Database berisi Collections, dan Collections berisi Documents.
- Pembuatan Implisit (Otomatis): Anda tidak perlu membuat database secara manual. MongoDB akan otomatis membuatnya saat Anda pertama kali menyimpan data di dalamnya. Perintah `use <nama_database>` digunakan untuk membuat sekaligus menggunakan database tersebut.

```
> use Latihan  
< switched to db Latihan
```

Method Database	Keterangan
<code>db.dropDatabase()</code>	Menghapus Database yang sedang aktif.
<code>db.getName()</code>	Mengambil nama database yang sedang aktif.
<code>db.stats()</code>	Mengambil statistik penggunaan database.
<code>db.version()</code>	Mengambil versi database.
<code>db.hostInfo()</code>	Mengambil informasi host tempat MongoDB berjalan.

Contoh penggunaan method:

```
> db.getName()
< Latihan
> db.hostInfo()
< {
  system: {
    currentTime: 2025-06-14T12:17:43.841Z,
    hostname: 'YASIR',
    cpuAddrSize: 64,
    memSizeMB: Long('15713'),
    memLimitMB: Long('15713'),
    numCores: 12,
    numCoresAvailableToProcess: 12,
    numPhysicalCores: 6,
    numCpuSockets: 1,
    cpuArch: 'x86_64',
    numaEnabled: false,
    numNumaNodes: 1
  },
  os: {
    type: 'Windows',
    name: 'Microsoft Windows 10',
    version: '10.0 (build 26100)'
  },
  extra: {
    pageSize: Long('4096'),
    cpuString: 'AMD Ryzen 5 5600H with Radeon Graphics'
  },
  ok: 1
}
```

```

> db.version()
< 8.0.9
> db.stats()
< {
  db: 'Latihan',
  collections: Long('0'),
  views: Long('0'),
  objects: Long('0'),
  avgObjSize: 0,
  dataSize: 0,
  storageSize: 0,
  indexes: Long('0'),
  indexSize: 0,
  totalSize: 0,
  scaleFactor: Long('1'),
  fsUsedSize: 0,
  fsTotalSize: 0,
  ok: 1
}
> db.dropDatabase()
< { ok: 1, dropped: 'Latihan' }

```

Collection:

Collection adalah kumpulan dokumen di dalam sebuah database, bisa dianalogikan seperti tabel pada database SQL. Keunikan utama *collection* adalah sifatnya yang memiliki skema fleksibel, yang berarti dokumen-dokumen di dalamnya tidak harus memiliki struktur yang seragam. Fleksibilitas ini sangat ideal untuk data yang berkembang atau memiliki banyak variasi.

Method untuk Collection	Keterangan
db.getCollectionNames()	Mengambil semua nama <i>collection</i> .
db.createCollection(name)	Membuat <i>collection</i> baru secara eksplisit.
db.getCollection(name)	Mendapatkan objek <i>collection</i> .
db.getCollectionInfos()	Mendapatkan informasi semua <i>collection</i> .

Contoh penggunaan :

```
> db.createCollection("Costumers")
< { ok: 1 }
> db.getCollectionNames()
< [ 'Costumers', 'Costumer', 'Products', 'Orders' ]
> db.getCollection("Orders")
< Latihan.Orders
> db.getCollectionInfos()
< [
  {
    name: 'Costumers',
    type: 'collection',
    options: {},
    info: {
      readOnly: false,
      uuid: UUID('bc998ec5-668a-4814-88af-be13ce675960')
    },
    idIndex: { v: 2, key: [Object], name: '_id_' }
  },
  {
    name: 'Costumer',
    type: 'collection',
    options: {},
    info: {
      readOnly: false,
      uuid: UUID('e30b792e-1756-4d68-8e17-8b4ce2fb72a3')
    },
    idIndex: { v: 2, key: [Object], name: '_id_' }
  },
]
```

```

{
  name: 'Products',
  type: 'collection',
  options: {},
  info: {
    readOnly: false,
    uuid: UUID('f45d00b4-228a-4de7-81fc-06ffa2826806')
  },
  idIndex: { v: 2, key: [Object], name: '_id_' },
},
{
  name: 'Orders',
  type: 'collection',
  options: {},
  info: {
    readOnly: false,
    uuid: UUID('fa56999b-75f7-4c08-a460-ea0e0bd0541b')
  },
  idIndex: { v: 2, key: [Object], name: '_id_' },
}
]

```

Latihan > |

Method pada Collection	Keterangan
db.<collection>.find()	Mengambil semua dokumen/data.
db.<collection>.count()	Mengambil jumlah dokumen.
db.<collection>.drop()	Menghapus <i>collection</i> .
db.<collection>.stats()	Mengambil informasi statistik <i>collection</i> .

Contoh penggunaan query :

```
> db.Orders.find()
<
> db.Orders.count()
< DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
< 0
> db.Orders.stats()
< {
  ok: 1,
  capped: false,
  wiredTiger: {
    metadata: { formatVersion: 1 },
    creationString: 'access_pattern_hint=none,allocation_size=4KB,app_metadata=(formatVersion=1),as
type: 'file',
    uri: 'statistics:table:collection-4-18414706907800404701',
    LSM: {
      'bloom filter false positives': 0,
      'bloom filter hits': 0,
      'bloom filter misses': 0,
      'bloom filter pages evicted from cache': 0,
      'bloom filter pages read into cache': 0,
      'bloom filters in the LSM tree': 0,
    }
  }
}
> db.Costumer.drop()
< true
```