

Update Document

Untuk mengubah data yang sudah ada, MongoDB menyediakan operator pembaruan (seperti \$set, \$inc) agar Anda bisa memodifikasi field tertentu tanpa harus mengganti seluruh dokumen. Ini membuat proses update lebih efisien.

- `updateOne(filter, update)`: Memperbarui satu dokumen pertama yang cocok dengan kriteria filter.
- `updateMany(filter, update)`: Memperbarui semua dokumen yang cocok dengan kriteria filter.
- `replaceOne(filter, replacement)`: Berbeda dengan update, metode ini mengganti seluruh isi dokumen dengan dokumen baru.

Contoh:

```
// 1. Mengubah harga pada satu produk bernama "Laptop Pro"
db.products.updateOne(
  { name: "Laptop Pro" },
  { $set: { price: 14500000, stock: 20 } } // $set hanya mengubah field yang
ditentukan
);

// 2. Menambah stok sebanyak 10 untuk semua produk kategori
'Electronics'
db.products.updateMany(
  { category: "Electronics" },
  { $inc: { stock: 10 } } // $inc digunakan untuk operasi
penambahan/pengurangan
);

// 3. Mengganti seluruh dokumen produk bernama "Mouse Gaming"
// Filter: cari produk dengan nama "Mouse Gaming"
db.products.replaceOne(
  { name: "Mouse Gaming" },
  // Replacement: dokumen baru yang akan menggantikan dokumen lama
  // secara keseluruhan
  {
```

```

    name: "Mouse Ergonomic",
    price: 600000,
    category: "Peripherals",
    brand: "Logi",
    in_warranty: true
  }
);

```

Contoh implementasi updateOne() & updateMany() :

Collection sebelum di update :

★ products						
	_id ObjectId	name String	price Int32	category String	stock Int32	tags Arrs
1	ObjectId('684d7ec7beada2d...	"Laptop Pro"	1500000	"Electronics"	25	[] 2 elem / [] [] []
2	ObjectId('684d7ed5beada2d...	"Mouse Gaming"	500000	"Electronics"	150	No field / [] [] []
3	ObjectId('684d7ed5beada2d...	"Buku MongoDB"	125000	"Books"	200	No field / [] [] []

Collection setelah update :

```

> db.products.updateOne({ name: "Laptop Pro" }, { $set: { price: 1450000, stock: 20 } });
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
> db.products.updateMany({category: "Electronics"}, { $inc: { stock: 10 } });
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}

```

★ products						
	_id ObjectId	name String	price Int32	category String	stock Int32	tags Arrs
1	ObjectId('684d7ec7beada2d...	"Laptop Pro"	1450000	"Electronics"	30	[] 2 elem / [] [] []
2	ObjectId('684d7ed5beada2d...	"Mouse Gaming"	500000	"Electronics"	160	No field / [] [] []
3	ObjectId('684d7ed5beada2d...	"Buku MongoDB"	125000	"Books"	200	No field / [] [] []

Collection setelah replace :

```

> db.products.replaceOne(
  { name: "Mouse Gaming" },
  {
    name: "Mouse Ergonomic",
    price: 600000,
    category: "Peripherals",
    brand: "Logi",
    in_warranty: true
  }
);
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

products						
	_id ObjectId	name String	price Int32	category String	stock Int32	tags Arra
1	ObjectId('684d7ec7beada2d...	"Laptop Pro"	14500000	"Electronics"	30	[] 2 elem
2	ObjectId('684d7ed5beada2d...	"Mouse Ergonomic"	600000	"Peripherals"	No field	No field
3	ObjectId('684d7ed5beada2d...	"Buku MongoDB"	125000	"Books"	200	No field