

## Schema Validation

Meskipun fleksibilitas skema adalah kekuatan MongoDB, terkadang konsistensi data tetap diperlukan. Schema Validation berfungsi sebagai "penjaga gerbang" yang memastikan setiap dokumen yang masuk atau diperbarui harus mematuhi aturan struktur yang telah Anda tentukan. Ini adalah cara untuk menyeimbangkan fleksibilitas dengan integritas data.

### Contoh:

Membuat *collection* users dengan aturan validasi.

```
db.createCollection("users", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["username", "email", "age"],
      properties: {
        username: {
          bsonType: "string",
          description: "harus berupa string dan wajib diisi"
        },
        age: {
          bsonType: "int",
          minimum: 17,
          description: "harus berupa angka integer minimal 17"
        },
        status: {
          enum: ["active", "pending", "disabled"],
          description: "hanya bisa salah satu dari nilai enum yang ada"
        }
      }
    }
  }
});
```

```
// Percobaan INSERT ini akan GAGAL karena umur kurang dari 17
db.users.insertOne({ username: "ana", email: "ana@mail.com", age: 16 });
```

// Percobaan INSERT ini akan BERHASIL karena umur lebih dari atau sama dengan 17

```
db.users.insertOne({ username: "anas", email: "anas@mail.com", age: 19 });
```

### Contoh Implementasi :

```
> db.createCollection("users", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["username", "email", "age"],
      properties: {
        username: {
          bsonType: "string",
          description: "harus berupa string dan wajib diisi"
        },
        age: {
          bsonType: "int",
          minimum: 17,
          description: "harus berupa angka integer minimal 17"
        },
        status: {
          enum: ["active", "pending", "disabled"],
          description: "hanya bisa salah satu dari nilai enum yang ada"
        }
      }
    }
  }
});
< { ok: 1 }
```

- Contoh gagal :

```
> db.users.insertOne({ username: "ana", email: "ana@mail.com", age: 16 });
MongoServerError: Document failed validation
```

- Contoh berhasil :

```
> db.users.insertOne({ username: "anas", email: "anas@mail.com", age: 19 });
< {
  acknowledged: true,
  insertedId: ObjectId('684d9c32beada2df13484c40')
}
```