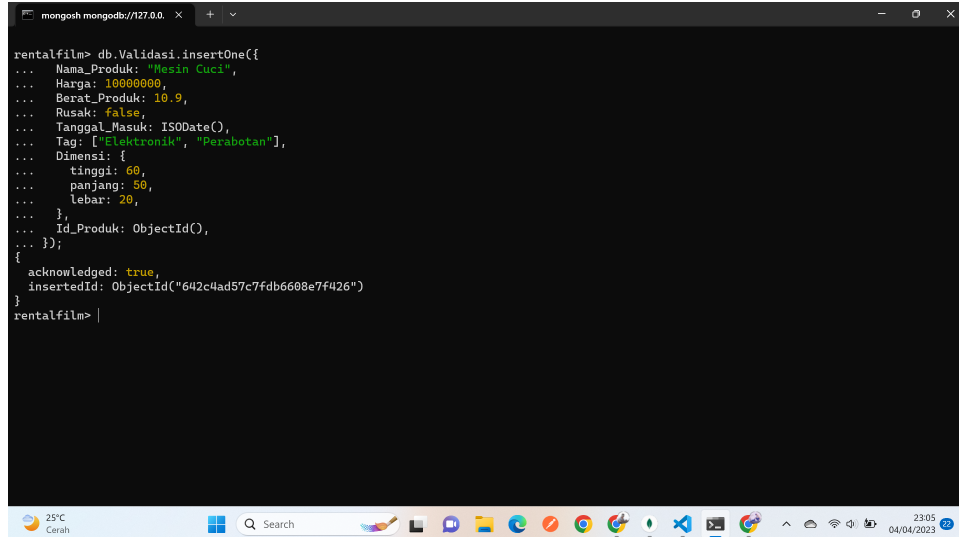


Praktikum Basis Data Lanjut

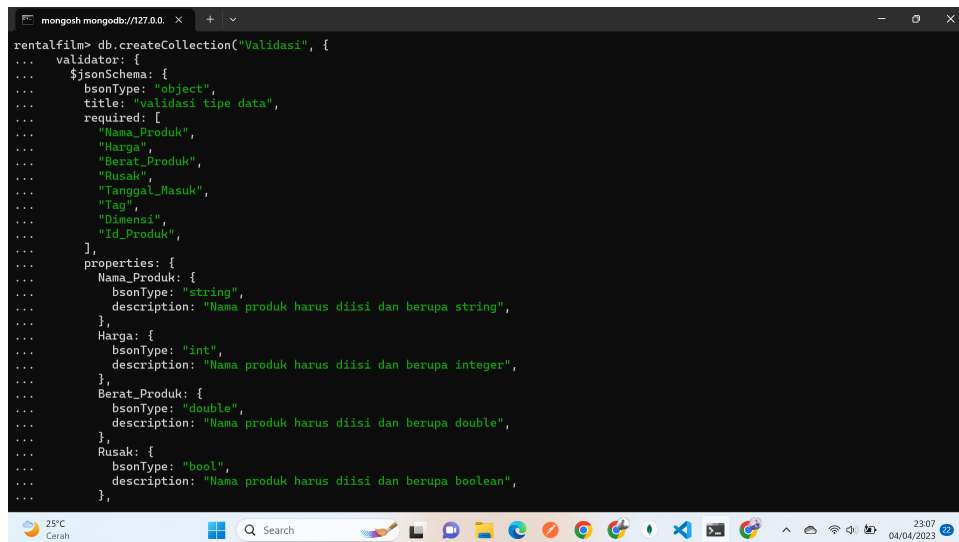
Tugas - 4

1. Buatlah serta jalankan query untuk membuat lebih dari 1 document baru yang menggunakan semua jenis tipe data.



```
mongosh mongodb://127.0.0.1:27017/rentalfilm> db.Validasi.insertOne({
...   Nama_Produk: "Mesin Cuci",
...   Harga: 10000000,
...   Berat_Produk: 10.9,
...   Rusak: false,
...   Tanggal_Masuk: ISODate(),
...   Tag: ["Elektronik", "Perabotan"],
...   Dimensi: {
...     tinggi: 60,
...     panjang: 50,
...     lebar: 20,
...   },
...   Id_Produk: ObjectId(),
... });
{
  acknowledged: true,
  insertedId: ObjectId("642c4ad57c7fdb6608e7f426")
}
rentalfilm> |
```

2. Buat dan jalankan query untuk menambahkan validator serta coba validator nya dengan data yang valid dan tidak valid kedalam collection yang sudah ada validator nya.



```
rentalfilm> db.createCollection("Validasi", {
...   validator: {
...     $jsonSchema: {
...       bsonType: "object",
...       title: "validasi tipe data",
...       required: [
...         "Nama_Produk",
...         "Harga",
...         "Berat_Produk",
...         "Rusak",
...         "Tanggal_Masuk",
...         "Tag",
...         "Dimensi",
...         "Id_Produk",
...       ],
...     },
...     properties: {
...       Nama_Produk: {
...         bsonType: "string",
...         description: "Nama produk harus diisi dan berupa string",
...       },
...       Harga: {
...         bsonType: "int",
...         description: "Nama produk harus diisi dan berupa integer",
...       },
...       Berat_Produk: {
...         bsonType: "double",
...         description: "Nama produk harus diisi dan berupa double",
...       },
...       Rusak: {
...         bsonType: "bool",
...         description: "Nama produk harus diisi dan berupa boolean",
...       },
...     },
...   },
... });
```

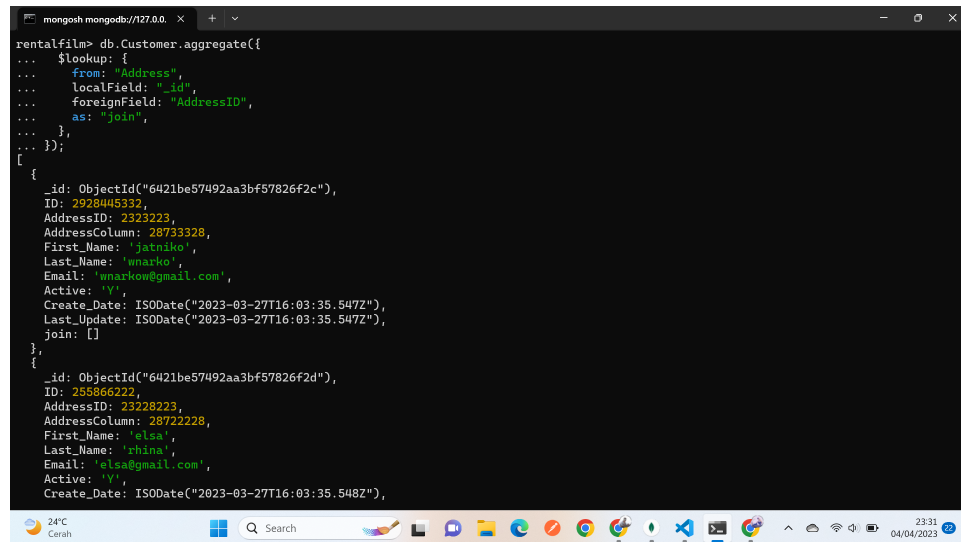
Jika data valid

```
mongosh mongodb://127.0.0.1:27027
> use rentalfilm
> db.Validasi.insertOne({
...   Nama_Produk: "Mesin Cuci",
...   Harga: 10000000,
...   Berat_Produk: 10.9,
...   Rusak: false,
...   Tanggal_Masuk: ISODate(),
...   Tag: ["Elektronik", "Perabotan"],
...   Dimensi: {
...     tinggi: 60,
...     panjang: 50,
...     lebar: 20,
...   },
...   Id_Produk: ObjectId(),
... });
{
  acknowledged: true,
  insertedId: ObjectId("642c4baf7c7fdb6608e7f428")
}
rentalfilm>
```

Jika data tidak valid

```
mongosh mongodb://127.0.0.1:27027
> use rentalfilm
> db.Validasi.insertOne({
...   Nama_Produk: "Mesin Cuci",
...   Harga: 10000000,
...   Berat_Produk: 10.9,
...   Rusak: false,
...   Tanggal_Masuk: 22-12-2000,
...   Tag: ["Elektronik", "Perabotan"],
...   Dimensi: {
...     tinggi: 60,
...     panjang: 50,
...     lebar: 20,
...   },
...   Id_Produk: ObjectId(),
... });
Uncaught:
MongoServerError: Document failed validation
Additional information: {
  failingDocumentId: ObjectId("642c4df17c7fdb6608e7f42a"),
  details: {
    operatorName: '$jsonSchema',
    title: 'validasi time data',
    schemaRulesNotSatisfied: [
      {
        operatorName: 'properties',
        propertiesNotSatisfied: [
          {
            propertyName: 'Tanggal_Masuk',
            description: 'Nama produk harus diisi dan berupa date',
            details: [ [Object] ]
          }
        ]
      }
    ]
  }
}
```

3. Buatlah query lookup untuk relasi many to many.

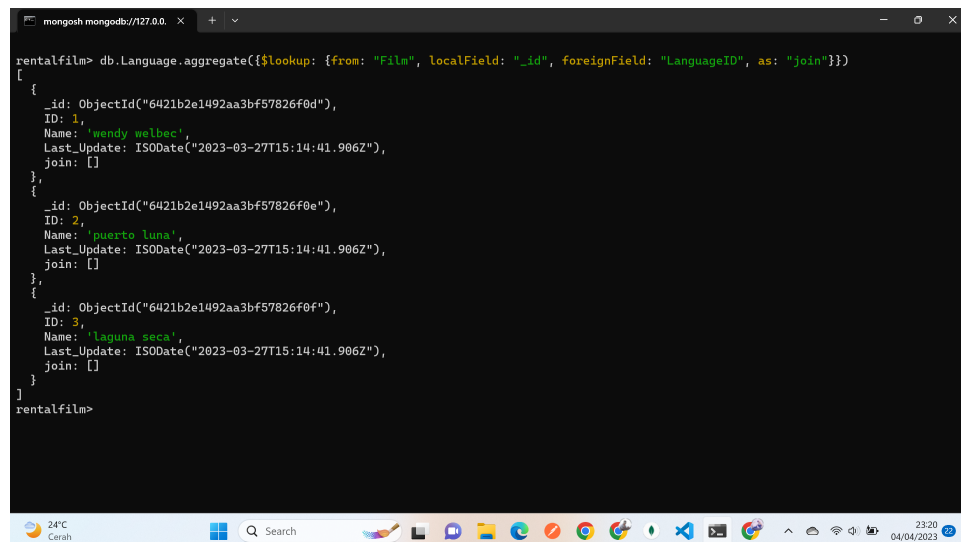


```

rentalfilm> db.Customer.aggregate({
...   $lookup: {
...     from: "Address",
...     localField: "_id",
...     foreignField: "AddressID",
...     as: "join",
...   },
... });
[
  {
    _id: ObjectId("6421be57492aa3bf57826f2c"),
    ID: 2928445332,
    AddressID: 23232223,
    AddressColumn: 28733328,
    First_Name: 'jatniwo',
    Last_Name: 'wanarko',
    Email: 'wanarkow@gmail.com',
    Active: 'Y',
    Create_Date: ISODate("2023-03-27T16:03:35.547Z"),
    Last_Update: ISODate("2023-03-27T16:03:35.547Z"),
    join: []
  },
  {
    _id: ObjectId("6421be57492aa3bf57826f2d"),
    ID: 255866222,
    AddressID: 23228223,
    AddressColumn: 28722228,
    First_Name: 'elsa',
    Last_Name: 'rhina',
    Email: 'elsa@gmail.com',
    Active: 'Y',
    Create_Date: ISODate("2023-03-27T16:03:35.548Z"),
    join: []
  }
]

```

4. Buat dan jalankan dua buah query lookup pada database rentalfilm yang telah dibuat selain pada collection country dan city.



```

rentalfilm> db.Language.aggregate({$lookup: {$from: "Film", localField: "_id", foreignField: "LanguageID", as: "join"}})
[
  {
    _id: ObjectId("6421b2e1492aa3bf57826f0d"),
    ID: 1,
    Name: 'wendy welbec',
    Last_Update: ISODate("2023-03-27T15:14:41.906Z"),
    join: []
  },
  {
    _id: ObjectId("6421b2e1492aa3bf57826f0e"),
    ID: 2,
    Name: 'puerto luna',
    Last_Update: ISODate("2023-03-27T15:14:41.906Z"),
    join: []
  },
  {
    _id: ObjectId("6421b2e1492aa3bf57826f0f"),
    ID: 3,
    Name: 'laguna seca',
    Last_Update: ISODate("2023-03-27T15:14:41.906Z"),
    join: []
  }
]
rentalfilm>

```

```

mongosh mongodb://127.0.0.1:27020
rentalfilm> db.Customer.aggregate({$lookup: {from: "Address", localField: "_id", foreignField: "AddressID", as: "join"}})
[
  {
    _id: ObjectId("6421be57492aa3bf57826f2c"),
    ID: 2928445332,
    AddressID: 2323223,
    AddressColumn: 28733328,
    First_Name: 'janiwo',
    Last_Name: 'mmarko',
    Email: 'mmarkow@gmail.com',
    Active: 'Y',
    Create_Date: ISODate("2023-03-27T16:03:35.547Z"),
    Last_Update: ISODate("2023-03-27T16:03:35.547Z"),
    join: []
  },
  {
    _id: ObjectId("6421be57492aa3bf57826f2d"),
    ID: 255866222,
    AddressID: 23228223,
    AddressColumn: 28722228,
    First_Name: 'elsa',
    Last_Name: 'rhina',
    Email: 'elsa@gmail.com',
    Active: 'Y',
    Create_Date: ISODate("2023-03-27T16:03:35.548Z"),
    Last_Update: ISODate("2023-03-27T16:03:35.548Z"),
    join: []
  },
  {
    _id: ObjectId("6421c0b6492aa3bf57826f2f"),
    ID: 2928445332,
    AddressID: 2323223,
    join: []
  }
]

```

5. Buatlah query untuk untuk semua tipe relasi.

One to one

```

mongosh mongodb://127.0.0.1:27020
rentalfilm> db.Customer.insertMany([
...
  {
    _id: 13,
    customername: "Andrea Lawson",
    address: { street: "Last Road", city: "Britania", country: "Narnia" },
  },
  {
    _id: 14,
    customerName: "Axcell Misk",
    city: "Unknown Dam",
    country: "Atlantis",
  },
...
]);
{ acknowledged: true, insertedIds: { '0': 13, '1': 14 } }
rentalfilm>

```

One to many

```

mongosh mongodb://127.0.0.1:27027
> use rentalfilm
> db.Film_Category.insertMany([
...   {
...     _id: 3,
...     name: "Jeanne",
...     genre: [
...       { genreId: 1, genreName: "Adventure" },
...       { genreId: 2, genreName: "Romance" },
...     ],
...   },
...   {
...     _id: 4,
...     name: "Walter",
...     genre: [
...       { genreId: 1, genreName: "Sci-fi" },
...       { genreId: 2, genreName: "Adventure" },
...     ],
...   },
... ]);
{ acknowledged: true, insertedIds: { '0': 3, '1': 4 } }
>

```

Many to many

```

mongosh mongodb://127.0.0.1:27027
> use rentalfilm
> db.Film_Category.find({ name: "Jeanne" }, { genre: 1, name: 1 })
[
  {
    _id: 1,
    name: "Jeanne",
    genre: [
      { genreId: 1, genreName: "Adventure" },
      { genreId: 2, genreName: "Romance" }
    ]
  },
  {
    _id: 3,
    name: "Jeanne",
    genre: [
      { genreId: 1, genreName: "Adventure" },
      { genreId: 2, genreName: "Romance" }
    ]
  }
]
>

```