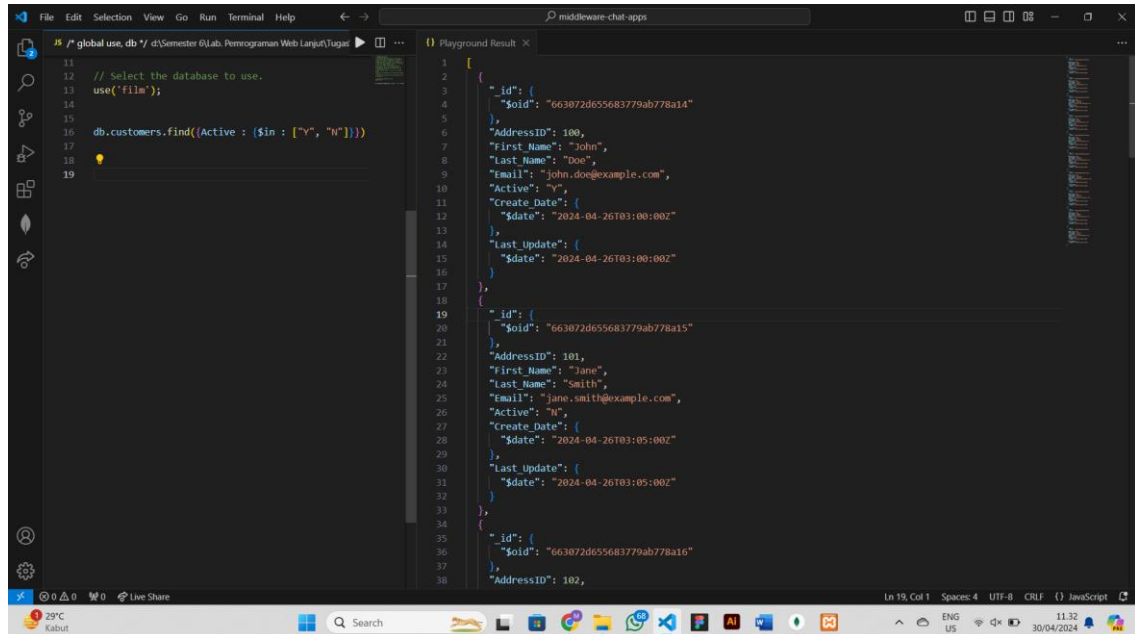


1. **Buat dan jalankan perintah untuk mencari data dari collection Customer menggunakan minimal 2 buah variasi dari perintah \$in dan \$nin**

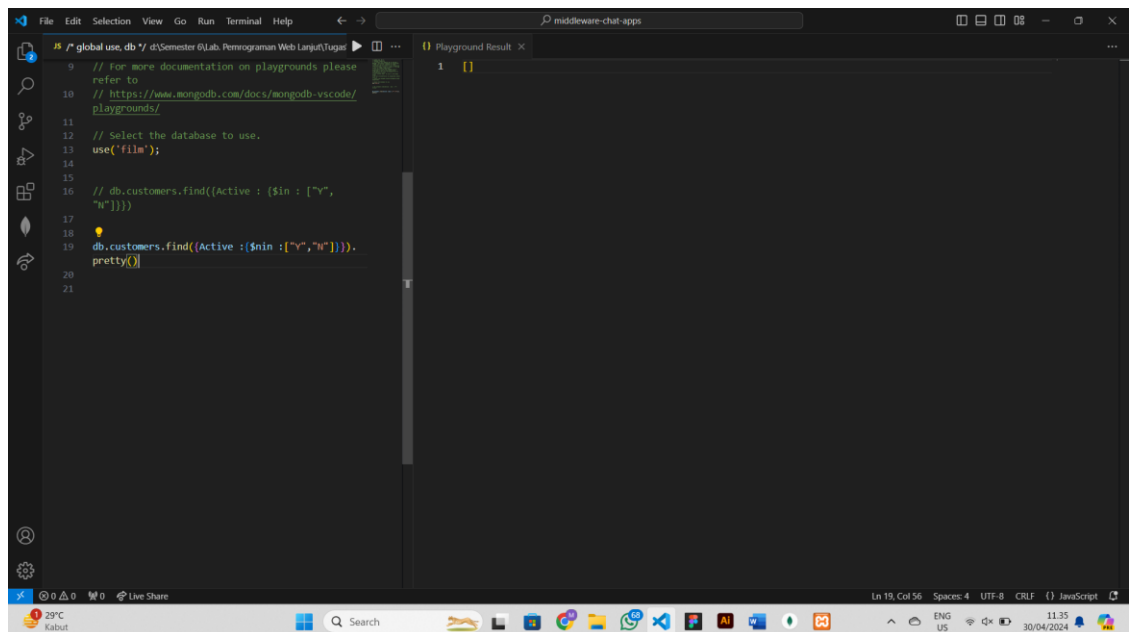
```
db.customers.find({Active : {$in : ["Y", "N"]}})
```



The screenshot shows a VS Code editor with a terminal window. The terminal output shows the command `db.customers.find({Active : {$in : ["Y", "N"]}})` being executed. The results are displayed in a JSON array format, showing three customer records with their IDs, addresses, names, emails, and active status.

```
1 {
2   {
3     "_id": {
4       "$oid": "663872d655683779ab778a14"
5     },
6     "AddressID": 100,
7     "First_Name": "John",
8     "Last_Name": "Doe",
9     "Email": "john.doe@example.com",
10    "Active": "Y",
11    "Create_Date": {
12      "$date": "2024-04-26T03:00:00Z"
13    },
14    "Last_Update": {
15      "$date": "2024-04-26T03:00:00Z"
16    }
17  },
18  {
19    "_id": {
20      "$oid": "663872d655683779ab778a15"
21    },
22    "AddressID": 101,
23    "First_Name": "Jane",
24    "Last_Name": "Smith",
25    "Email": "jane.smith@example.com",
26    "Active": "N",
27    "Create_Date": {
28      "$date": "2024-04-26T03:05:00Z"
29    },
30    "Last_Update": {
31      "$date": "2024-04-26T03:05:00Z"
32    }
33  },
34  {
35    "_id": {
36      "$oid": "663872d655683779ab778a16"
37    },
38    "AddressID": 102,
```

```
db.customers.find({Active : {$nin : ["Y", "N"]}}).pretty()
```



The screenshot shows a VS Code editor with a terminal window. The terminal output shows the command `db.customers.find({Active : {$nin : ["Y", "N"]}}).pretty()` being executed. The results are displayed in a JSON array format, showing an empty array, indicating that no records were found.

```
1 []
```

2. Buatlah dan Jalankan perintah untuk mencari data dari collection Film untuk menampilkan data Title yang diakhiri dengan huruf G.

```
db.film.find(  
  { Title: { $regex: /g$/i } },{ Title: 1, _id: 0 }  
);
```



The screenshot shows the MongoDB Playground interface. On the left, the code editor contains the following JavaScript code:

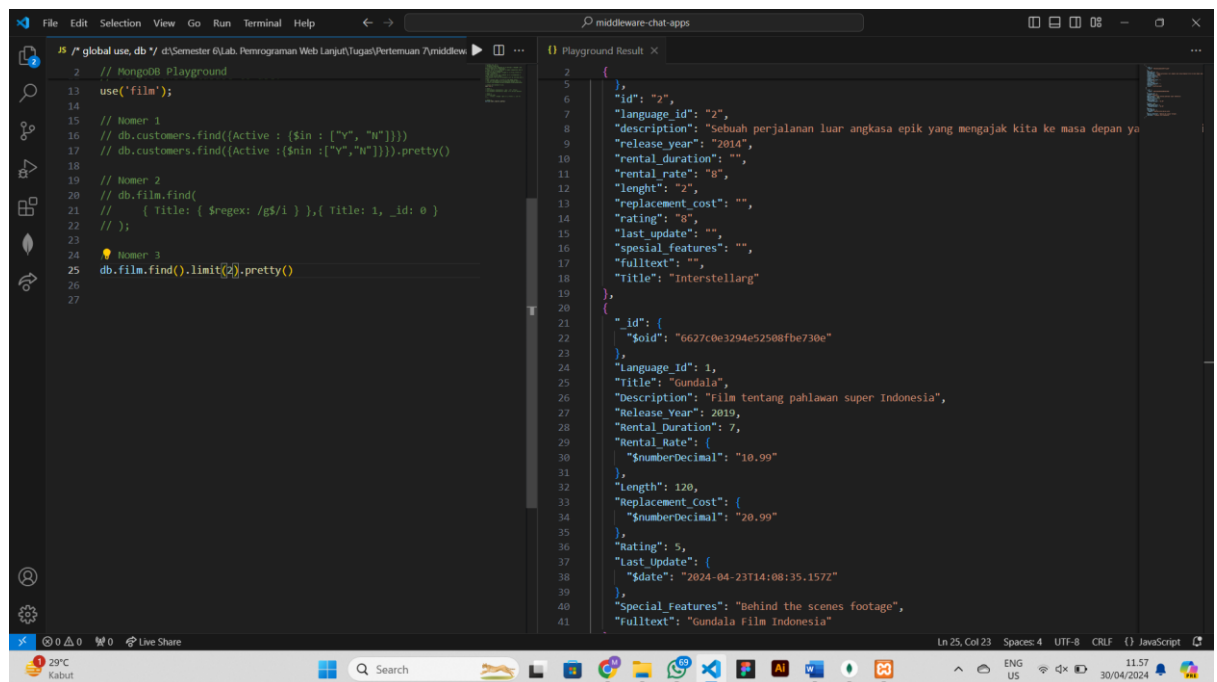
```
1 use('film');  
2  
3 // Nomer 1  
4 // db.customers.find({Active : {$in : ["Y", "N"]}})  
5 // db.customers.find({Active : {$nin : ["Y", "N"]}}).pretty()  
6  
7 db.film.find(  
8   { Title: { $regex: /g$/i } },{ Title: 1, _id: 0 }  
9 );
```

On the right, the 'Playground Result' tab shows the output of the query:

```
1 {  
2   "Title": "Interstellarg"  
3 }  
4  
5 ]
```

3. Buatlah dan jalankan perintah untuk menampilkan 2 buah data dari collection Customer.

```
db.film.find().limit(2).pretty()
```



The screenshot shows the VS Code editor with the MongoDB Playground extension. The code editor on the left contains the following JavaScript code:

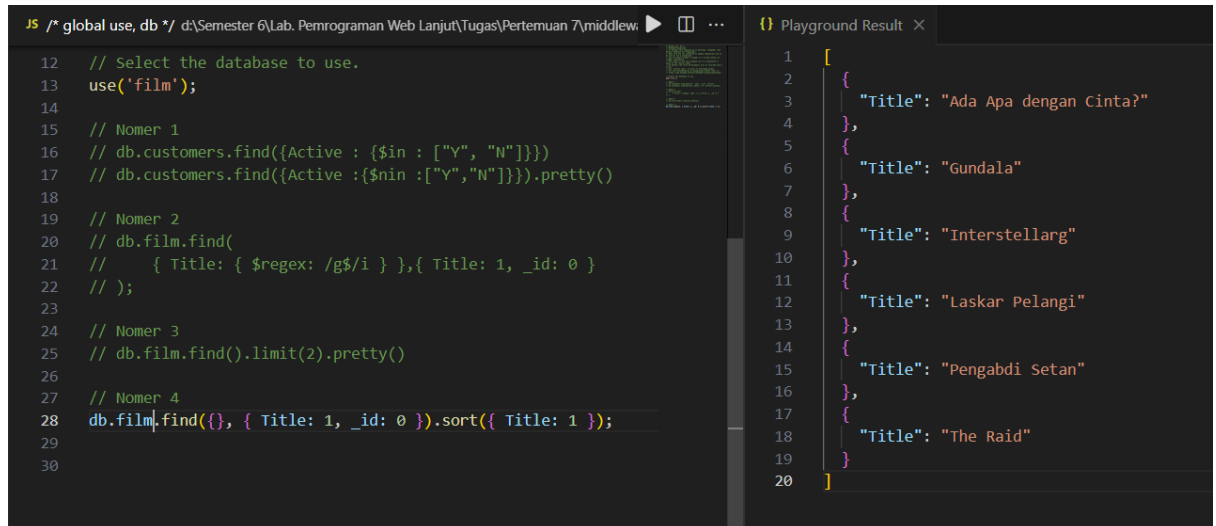
```
1 // MongoDB Playground  
2  
3 use('film');  
4  
5 // Nomer 1  
6 // db.customers.find({Active : {$in : ["Y", "N"]}})  
7 // db.customers.find({Active : {$nin : ["Y", "N"]}}).pretty()  
8  
9 // Nomer 2  
10 // db.film.find(  
11 //   { Title: { $regex: /g$/i } },{ Title: 1, _id: 0 }  
12 // );  
13  
14 // Nomer 3  
15 db.film.find().limit(2).pretty()
```

The 'Playground Result' tab on the right shows the output of the query, displaying two document objects in JSON format:

```
1 {  
2   "id": "2",  
3   "language_id": "2",  
4   "description": "Sebuah perjalanan luar angkasa epik yang mengajak kita ke masa depan ya",  
5   "release_year": "2014",  
6   "rental_duration": "",  
7   "rental_rate": "8",  
8   "length": "2",  
9   "replacement_cost": "",  
10  "rating": "8",  
11  "last_update": "",  
12  "special_features": "",  
13  "fulltext": "",  
14  "title": "Interstellarg"  
15 },  
16 {  
17   "_id": {  
18     "$oid": "6627c0e3294e52508fbc730e"  
19   },  
20   "language_id": 1,  
21   "title": "Gundala",  
22   "description": "Film tentang pahlawan super Indonesia",  
23   "release_year": 2019,  
24   "rental_duration": 7,  
25   "rental_rate": {  
26     "$numberDecimal": "10.99"  
27   },  
28   "length": 120,  
29   "replacement_cost": {  
30     "$numberDecimal": "20.99"  
31   },  
32   "rating": 5,  
33   "last_update": {  
34     "$date": "2024-04-23T14:08:35.157Z"  
35   },  
36   "special_features": "Behind the scenes footage",  
37   "fulltext": "Gundala Film Indonesia"  
38 }  
39 ]
```

4. Buatlah dan jalankan perintah untuk menampilkan semua data di film berdasarkan urutan alfabet.

```
db.film.find({}, { Title: 1, _id: 0 }).sort({ Title: 1 });
```



The screenshot shows a MongoDB Playground interface. On the left, the command window contains the following JavaScript code:

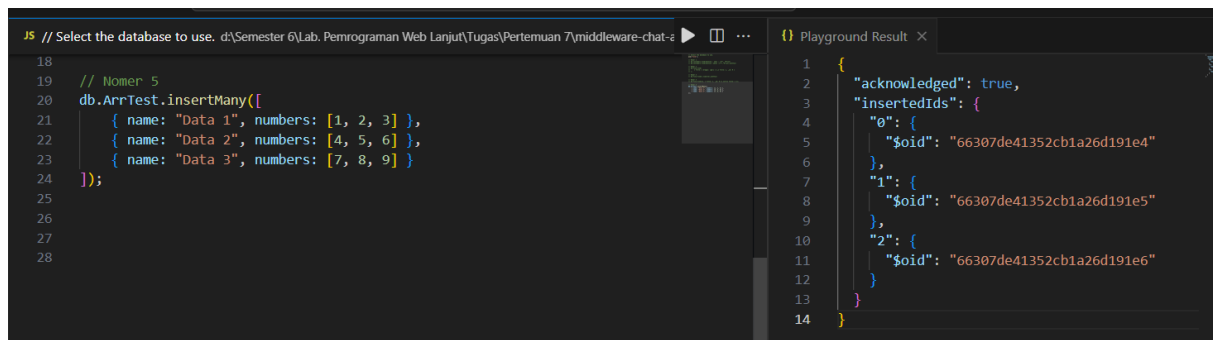
```
12 // Select the database to use.
13 use('film');
14
15 // Nomer 1
16 // db.customers.find({Active : {$in : ["Y", "N"]}})
17 // db.customers.find({Active : {$nin : ["Y", "N"]}}).pretty()
18
19 // Nomer 2
20 // db.film.find(
21 //   { Title: { $regex: /g$/i } }, { Title: 1, _id: 0 }
22 // );
23
24 // Nomer 3
25 // db.film.find().limit(2).pretty()
26
27 // Nomer 4
28 db.film.find({}, { Title: 1, _id: 0 }).sort({ Title: 1 });
29
30
```

On the right, the 'Playground Result' tab shows the output of the query, which is a JSON array of film documents sorted by title:

```
1 [
2   {
3     "Title": "Ada Apa dengan Cinta?"
4   },
5   {
6     "Title": "Gundala"
7   },
8   {
9     "Title": "Interstellarg"
10  },
11  {
12    "Title": "Laskar Pelangi"
13  },
14  {
15    "Title": "Pengabdi Setan"
16  },
17  {
18    "Title": "The Raid"
19  }
20 ]
```

5. Buatlah collection baru dengan nama ArrTest masukan beberapa data yang memiliki setidaknya 1 array kemudian buat dan jalankan perintah Push, Pull dan addToSet secara bergiliran.

```
db.ArrTest.insertMany([
  { name: "Data 1", numbers: [1, 2, 3] },
  { name: "Data 2", numbers: [4, 5, 6] },
  { name: "Data 3", numbers: [7, 8, 9] }
]);
```



The screenshot shows a MongoDB Playground interface. On the left, the command window contains the following JavaScript code:

```
18 // Select the database to use. d:\Semester 6\Lab. Pemrograman Web Lanjut\Tugas\Pertemuan 7\middleware-chat-z
19 // Nomer 5
20 db.ArrTest.insertMany([
21   { name: "Data 1", numbers: [1, 2, 3] },
22   { name: "Data 2", numbers: [4, 5, 6] },
23   { name: "Data 3", numbers: [7, 8, 9] }
24 ]);
25
26
27
28
```

On the right, the 'Playground Result' tab shows the output of the insertMany operation, which is a JSON object indicating success and listing the inserted documents with their IDs:

```
1 {
2   "acknowledged": true,
3   "insertedIds": {
4     "0": {
5       "$oid": "66307de41352cb1a26d191e4"
6     },
7     "1": {
8       "$oid": "66307de41352cb1a26d191e5"
9     },
10    "2": {
11      "$oid": "66307de41352cb1a26d191e6"
12    }
13  }
14 }
```

```
db.ArrTest.updateMany({}, { $push: { numbers: 10 } });
```

```
JS // Select the database to use. d:\Semester 6\Lab. Pemrograman Web Lanjut\Tugas\Pertemuan 7\middleware-chat-2
18
19 // Nomer 5
20 // db.ArrTest.insertMany([
21 //   { name: "Data 1", numbers: [1, 2, 3] },
22 //   { name: "Data 2", numbers: [4, 5, 6] },
23 //   { name: "Data 3", numbers: [7, 8, 9] }
24 // ]);
25
26 db.ArrTest.updateMany({}, { $push: { numbers: 10 } });
27
```

```
1 {
2   "acknowledged": true,
3   "insertedId": null,
4   "matchedCount": 3,
5   "modifiedCount": 3,
6   "upsertedCount": 0
7 }
```

```
db.ArrTest.updateMany({}, { $pull: { numbers: 10 } });
```

```
JS // Select the database to use. d:\Semester 6\Lab. Pemrograman Web Lanjut\Tugas\Pertemuan 7\middleware-chat-2
18
19 // Nomer 5
20 // db.ArrTest.insertMany([
21 //   { name: "Data 1", numbers: [1, 2, 3] },
22 //   { name: "Data 2", numbers: [4, 5, 6] },
23 //   { name: "Data 3", numbers: [7, 8, 9] }
24 // ]);
25
26 // db.ArrTest.updateMany({}, { $push: { numbers: 10 } });
27 db.ArrTest.updateMany({}, { $pull: { numbers: 10 } });
28
```

```
1 {
2   "acknowledged": true,
3   "insertedId": null,
4   "matchedCount": 3,
5   "modifiedCount": 0,
6   "upsertedCount": 0
7 }
```

```
db.ArrTest.updateMany({}, { $addToSet: { numbers: 10 } });
```

```
JS // Select the database to use. d:\Semester 6\Lab. Pemrograman Web Lanjut\Tugas\Pertemuan 7\middleware-chat-2
25
26 // db.ArrTest.updateMany({}, { $push: { numbers: 10 } });
27 // db.ArrTest.updateMany({}, { $pull: { numbers: 10 } });
28
29 db.ArrTest.updateMany({}, { $addToSet: { numbers: 10 } });
30
31
32
33
```

```
1 {
2   "acknowledged": true,
3   "insertedId": null,
4   "matchedCount": 3,
5   "modifiedCount": 3,
6   "upsertedCount": 0
7 }
```