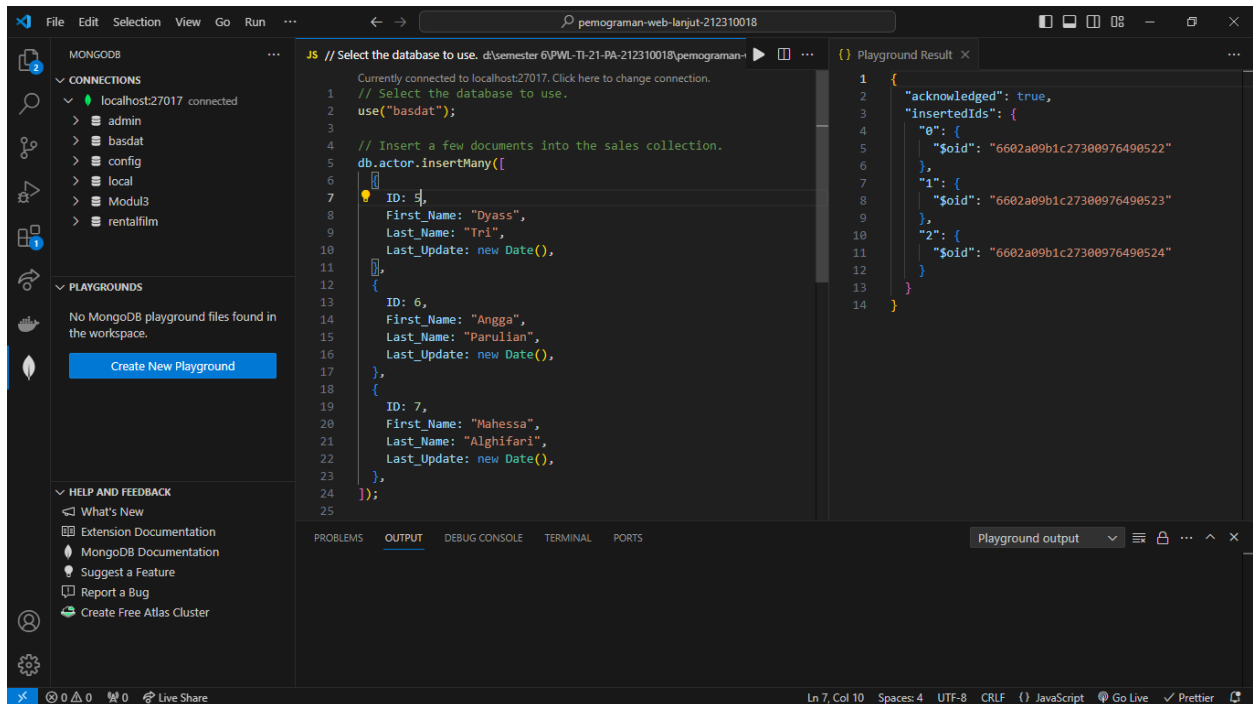


Fathurahman AL Faridzi

212310018

TI 21 PA

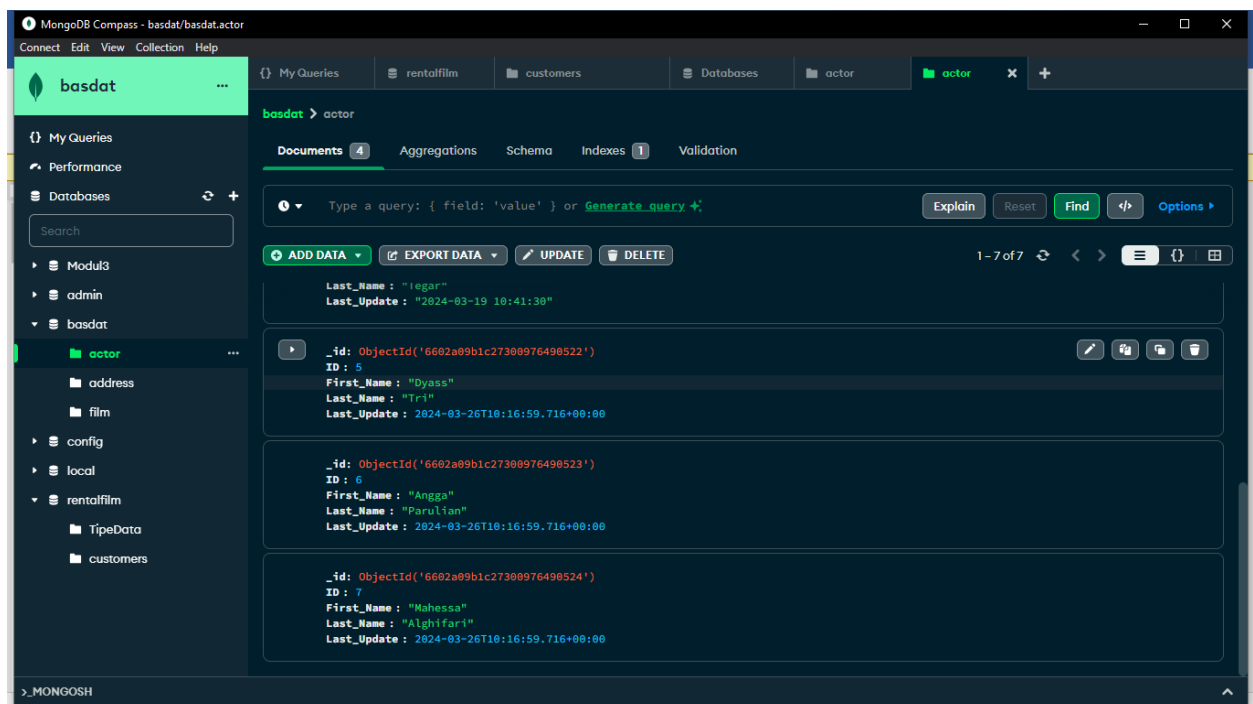
## Insert Query actor



The screenshot shows the VS Code editor with a MongoDB playground script. The script is connected to localhost:27017 and uses the 'basdat' database. It inserts three documents into the 'actor' collection. The playground results show the inserted documents with their IDs and timestamps.

```
1 // Select the database to use.
2 use("basdat");
3
4 // Insert a few documents into the sales collection.
5 db.actor.insertMany([
6   {
7     ID: 5,
8     First_Name: "Dyass",
9     Last_Name: "Tri",
10    Last_Update: new Date(),
11  },
12  {
13    ID: 6,
14    First_Name: "Angga",
15    Last_Name: "Parulian",
16    Last_Update: new Date(),
17  },
18  {
19    ID: 7,
20    First_Name: "Mahessa",
21    Last_Name: "Alghifari",
22    Last_Update: new Date(),
23  },
24 ]);
```

```
1 {
2   "acknowledged": true,
3   "insertedIds": {
4     "0": {
5       "soid": "6602a09b1c27300976490522"
6     },
7     "1": {
8       "soid": "6602a09b1c27300976490523"
9     },
10    "2": {
11      "soid": "6602a09b1c27300976490524"
12    }
13  }
14 }
```



The screenshot shows the MongoDB Compass interface. The 'actor' collection is selected, and the 'Documents' tab is active. It displays three documents with their IDs, first names, last names, and last update timestamps.

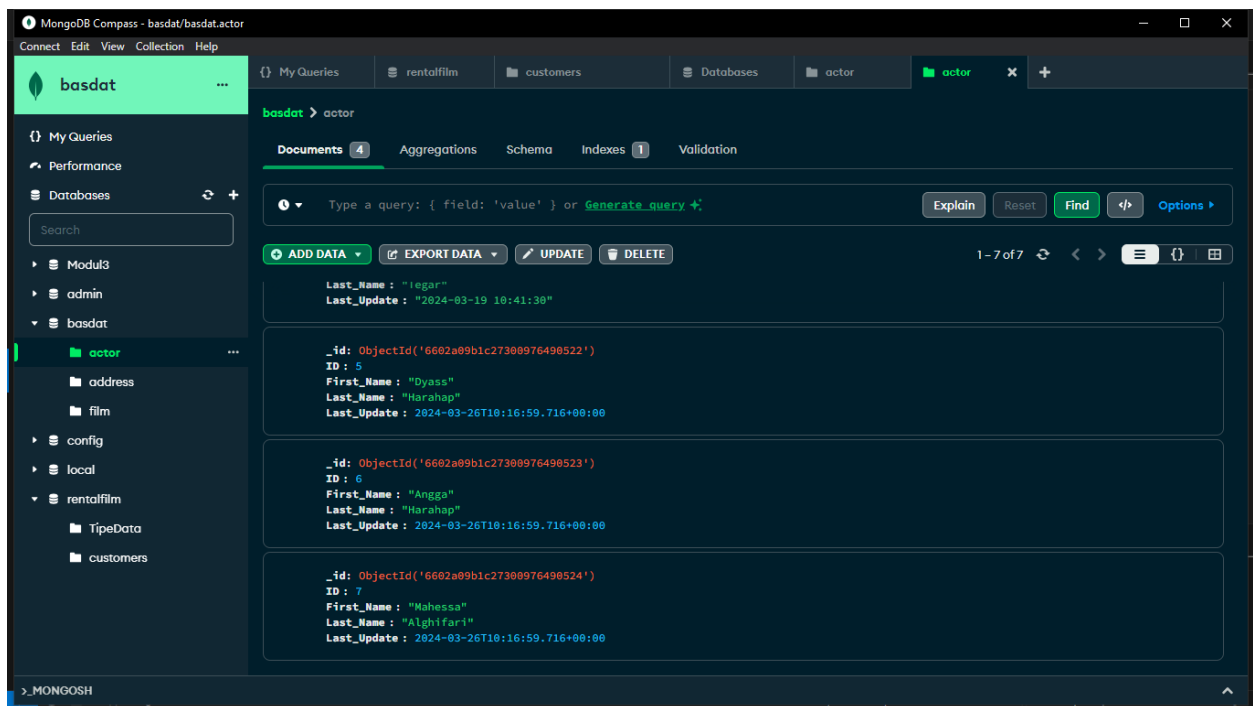
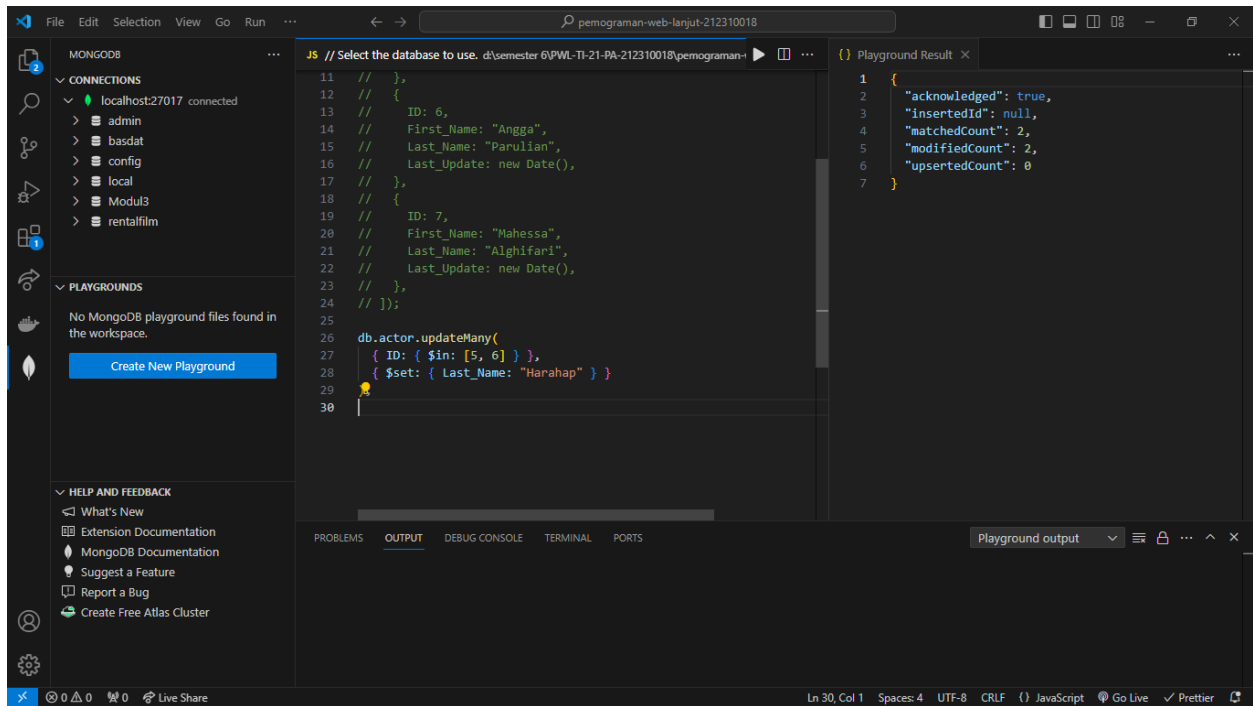
```
{ "Last_Name": "Iegar", "Last_Update": "2024-03-19 10:41:30" }
```

```
{ "_id": ObjectId('6602a09b1c27300976490522'), ID: 5, First_Name: "Dyass", Last_Name: "Tri", Last_Update: 2024-03-26T10:16:59.716+00:00 }
```

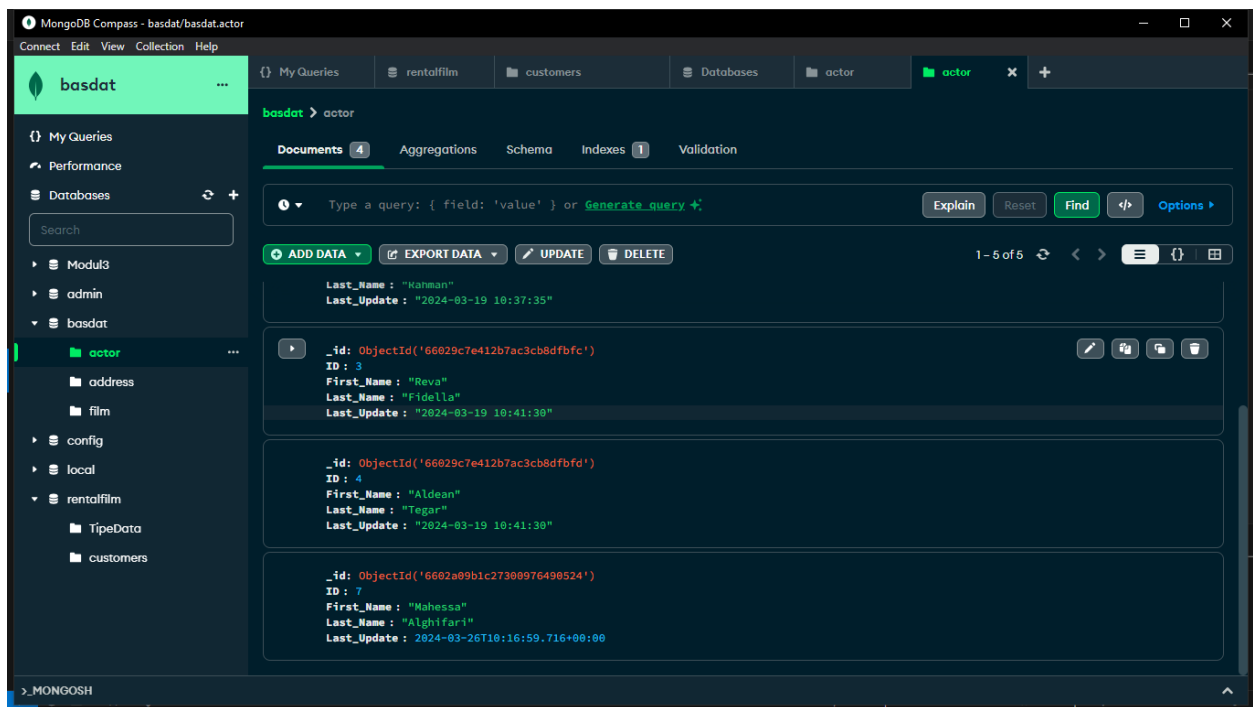
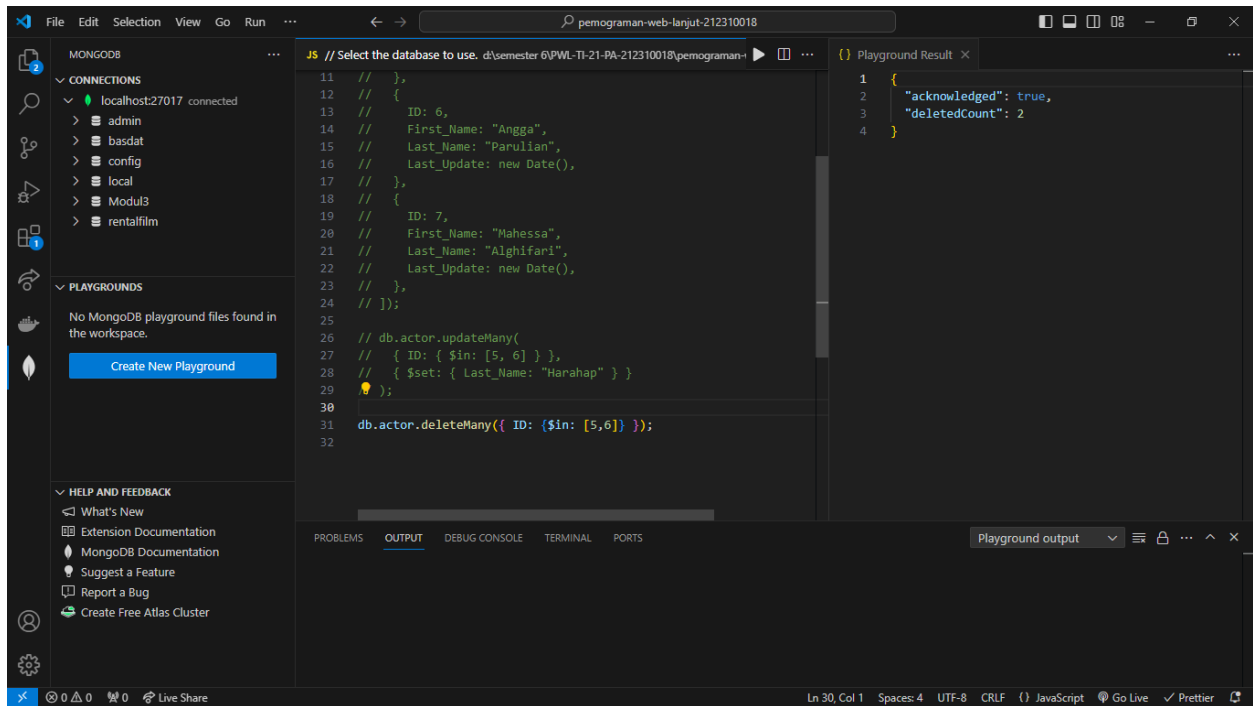
```
{ "_id": ObjectId('6602a09b1c27300976490523'), ID: 6, First_Name: "Angga", Last_Name: "Parulian", Last_Update: 2024-03-26T10:16:59.716+00:00 }
```

```
{ "_id": ObjectId('6602a09b1c27300976490524'), ID: 7, First_Name: "Mahessa", Last_Name: "Alghifari", Last_Update: 2024-03-26T10:16:59.716+00:00 }
```

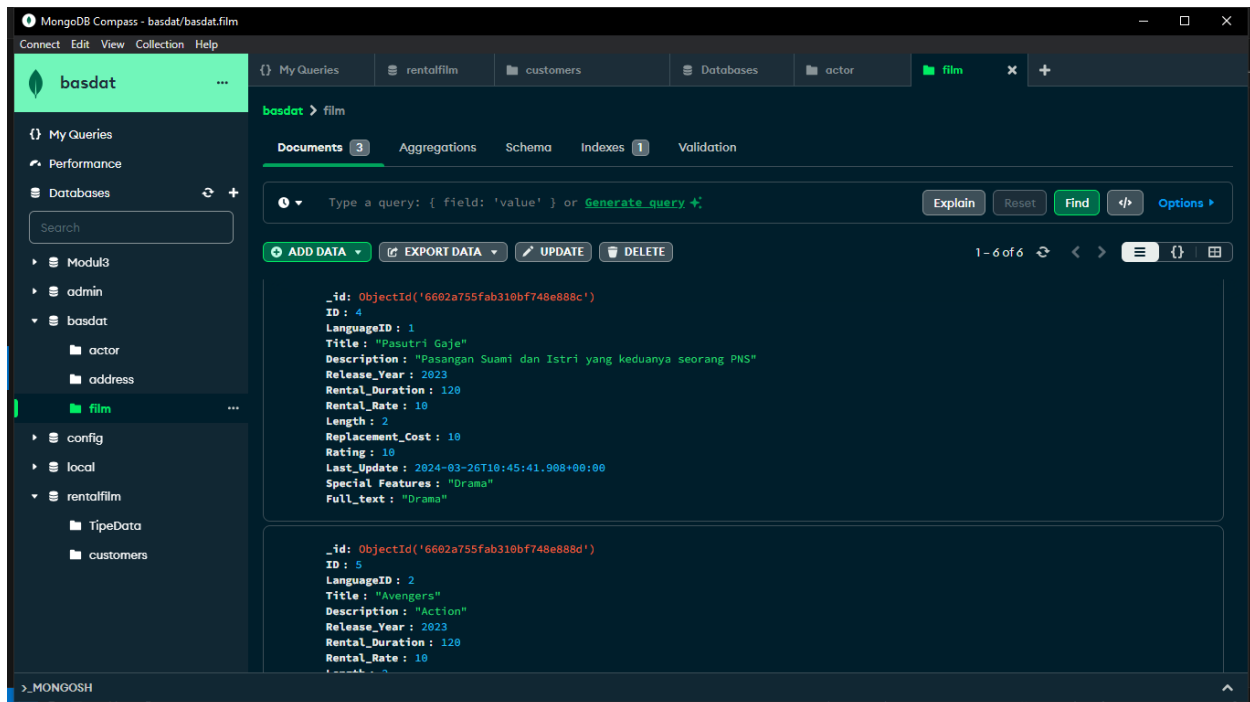
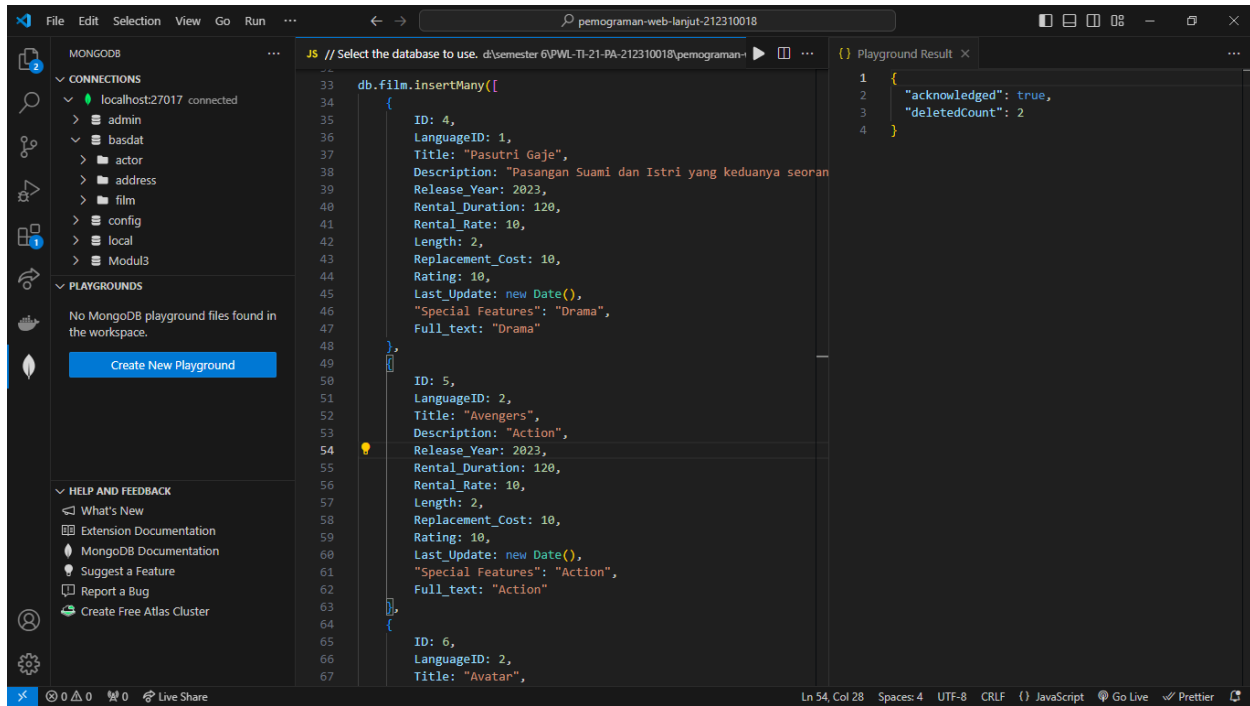
## Update Query actor



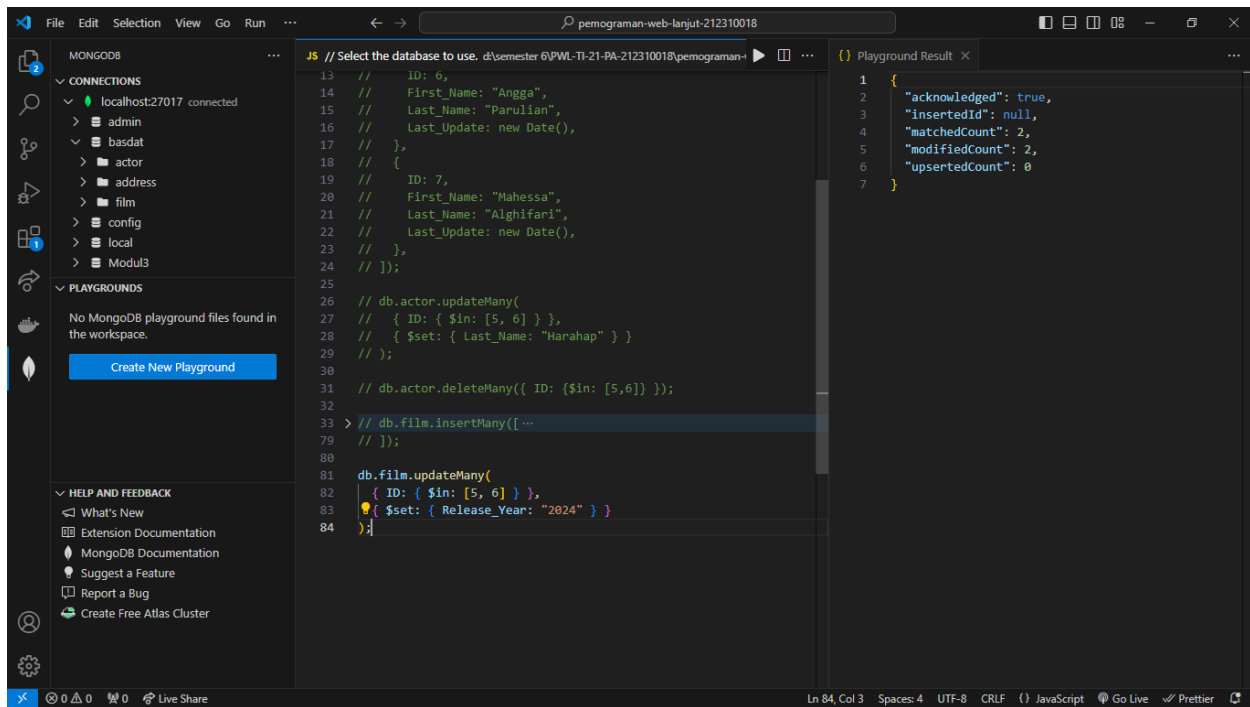
## Delete Query actor



## Insert Query film



## Update Query film

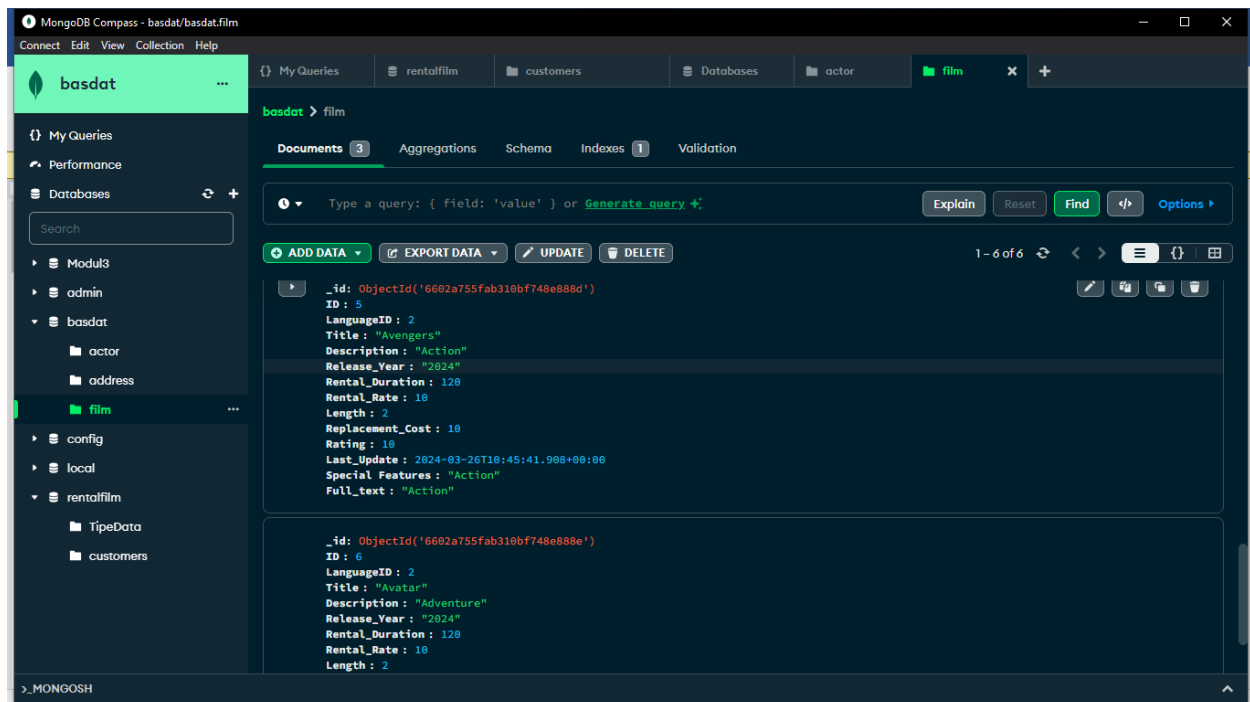


The screenshot shows the MongoDB Playground interface. On the left, the 'CONNECTIONS' panel shows a connection to 'localhost:27017' with a database named 'basdat'. The 'PLAYGROUNDS' panel shows 'No MongoDB playground files found in the workspace.' The main editor displays a JavaScript query to update a film document. The query is as follows:

```
13 // Select the database to use. d:\semester 6\PWL-TI-21-PA-212310018\pemograman-
14 // First_Name: "Angga",
15 // Last_Name: "Parullian",
16 // Last_Update: new Date(),
17 // },
18 // {
19 // ID: 7,
20 // First_Name: "Mahessa",
21 // Last_Name: "Alghifari",
22 // Last_Update: new Date(),
23 // },
24 // });
25
26 // db.actor.updateMany(
27 // { ID: { $in: [5, 6] } },
28 // { $set: { Last_Name: "Harahap" } }
29 // );
30
31 // db.actor.deleteMany({ ID: { $in: [5,6] } });
32
33 > db.film.insertMany([...
34 // });
35
36 db.film.updateMany(
37 { ID: { $in: [5, 6] } },
38 { $set: { Release_Year: "2024" } }
39 );
```

The 'Playground Result' panel on the right shows the result of the query:

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

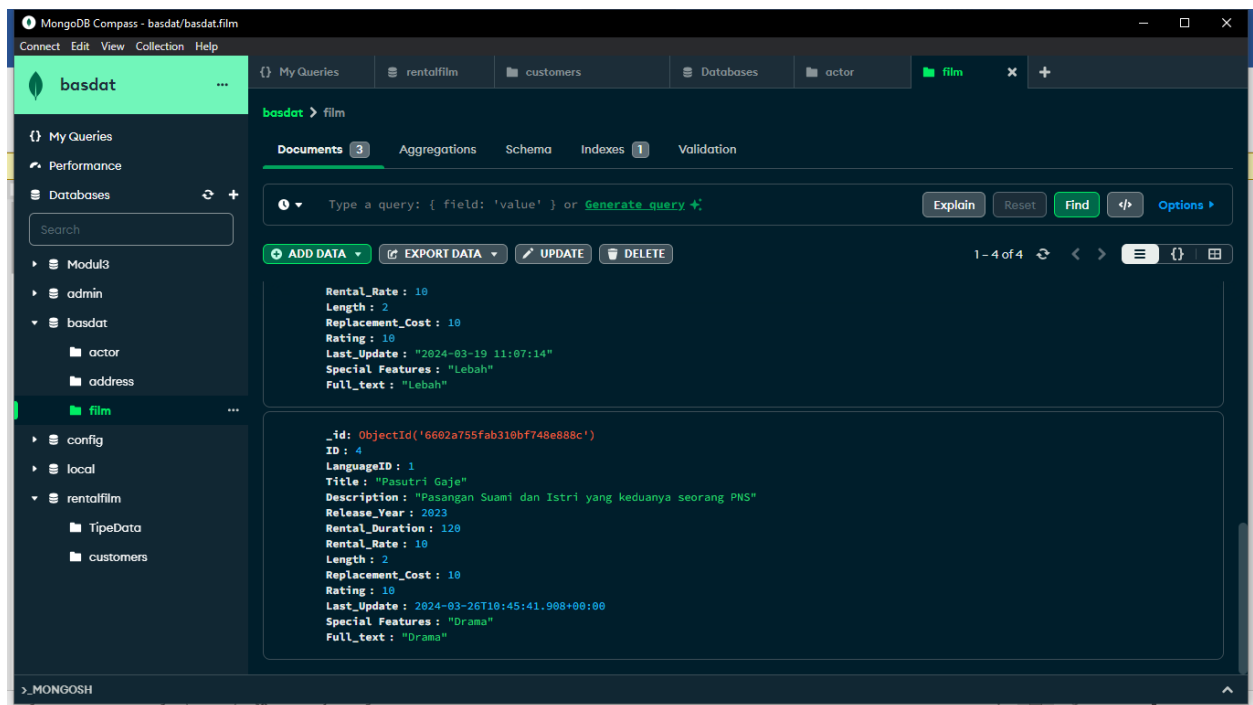
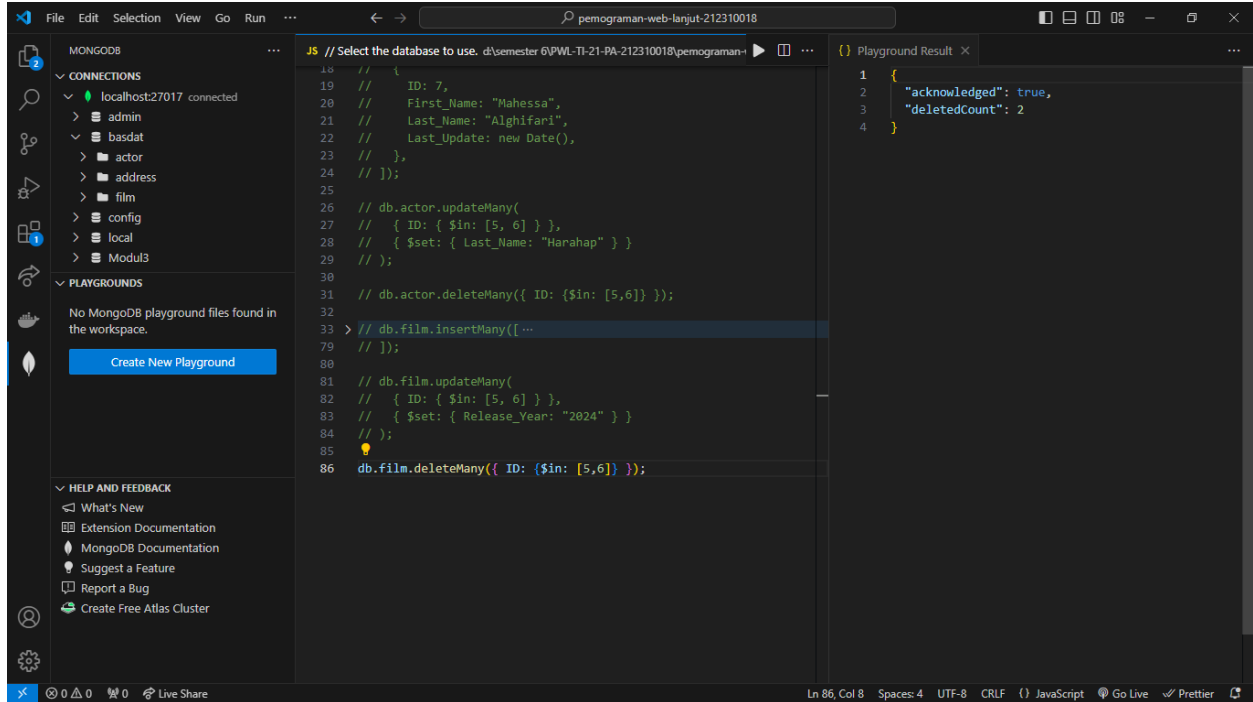


The screenshot shows the MongoDB Compass interface. The left sidebar shows the 'basdat' database selected. The main panel shows the 'film' collection. The 'Documents' tab is active, displaying two documents:

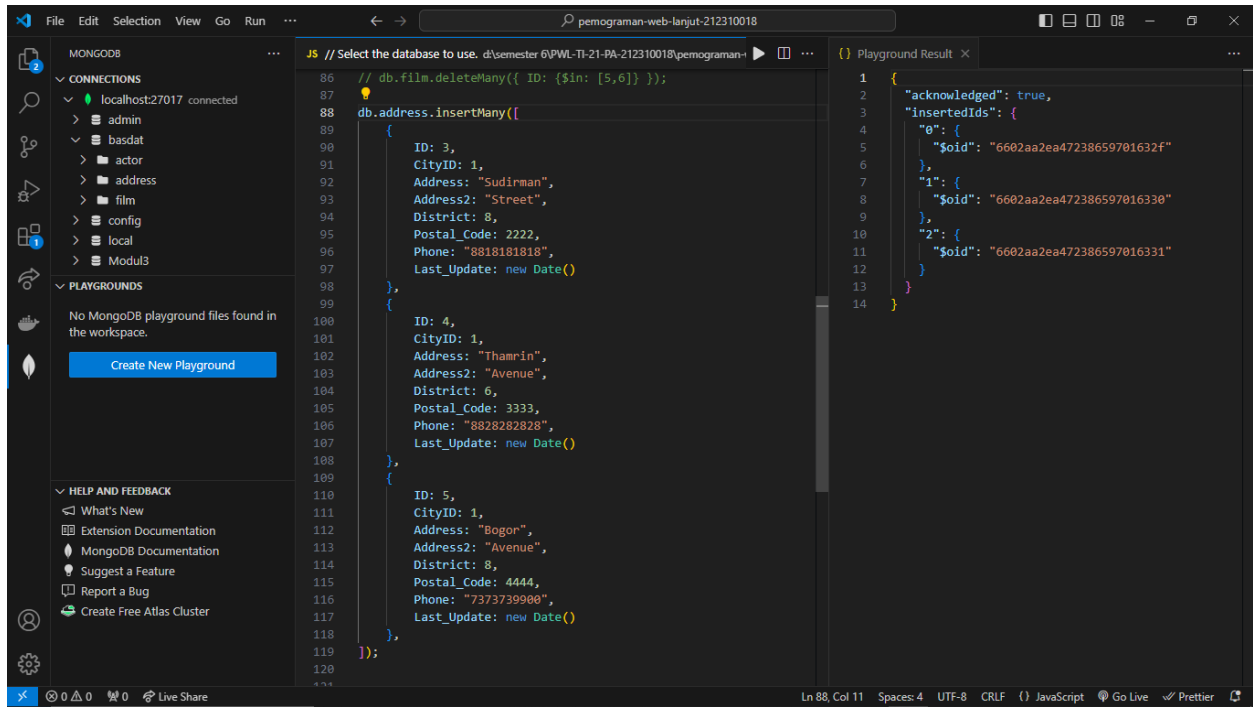
```
{ "_id": ObjectId('6602a755fab310bf748e888d'),
  "ID": 5,
  "LanguageID": 2,
  "Title": "Avengers",
  "Description": "Action",
  "Release_Year": "2024",
  "Rental_Duration": 120,
  "Rental_Rate": 10,
  "Length": 2,
  "Replacement_Cost": 10,
  "Rating": 10,
  "Last_Update": "2024-03-26T10:45:41.908+00:00",
  "Special_Features": "Action",
  "Full_text": "Action" }

{ "_id": ObjectId('6602a755fab310bf748e888e'),
  "ID": 6,
  "LanguageID": 2,
  "Title": "Avatar",
  "Description": "Adventure",
  "Release_Year": "2024",
  "Rental_Duration": 120,
  "Rental_Rate": 10,
  "Length": 2 }
```

## Delete Quey film



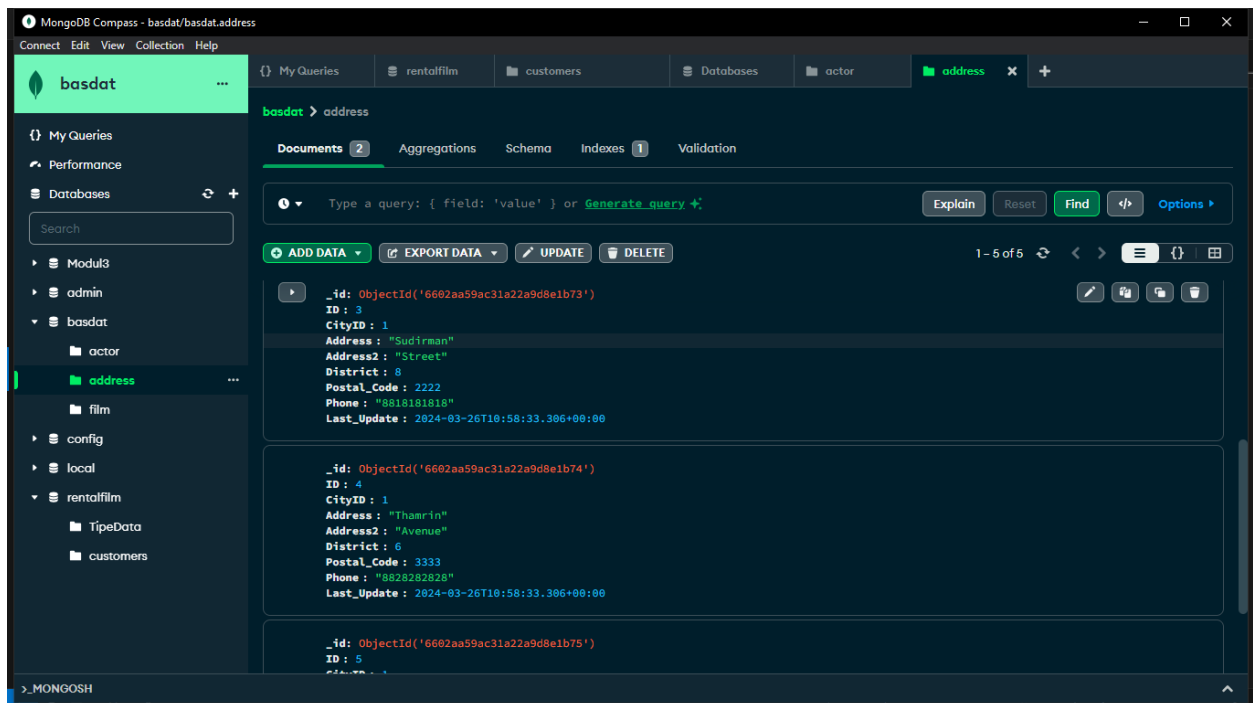
## Insert Query address



The screenshot shows the VS Code editor with a MongoDB playground script. The script is in JavaScript and uses the `db.address.insertMany()` method to insert three documents into the `address` collection. The documents contain fields like `ID`, `CityID`, `Address`, `Address2`, `District`, `Postal_Code`, `Phone`, and `Last_Update`. The playground results on the right show the response from the database, including `acknowledged` status and `insertedIds`.

```
JS // Select the database to use. d:\semester 6\PWL-TI-21-PA-212310018\pemograman-
86 // db.film.deleteMany({ ID: {$in: [5,6]} });
87
88 db.address.insertMany([
89   {
90     ID: 3,
91     CityID: 1,
92     Address: "Sudirman",
93     Address2: "Street",
94     District: 8,
95     Postal_Code: 2222,
96     Phone: "8818181818",
97     Last_Update: new Date()
98   },
99   {
100     ID: 4,
101     CityID: 1,
102     Address: "Thamrin",
103     Address2: "Avenue",
104     District: 6,
105     Postal_Code: 3333,
106     Phone: "8828282828",
107     Last_Update: new Date()
108   },
109   {
110     ID: 5,
111     CityID: 1,
112     Address: "Bogor",
113     Address2: "Avenue",
114     District: 8,
115     Postal_Code: 4444,
116     Phone: "7373739900",
117     Last_Update: new Date()
118   }
119 ]);
120
```

```
1 {
2   "acknowledged": true,
3   "insertedIds": {
4     "0": {
5       "$oid": "6602aa2ea47238659701632f"
6     },
7     "1": {
8       "$oid": "6602aa2ea472386597016330"
9     },
10    "2": {
11      "$oid": "6602aa2ea472386597016331"
12    }
13  }
14 }
```



The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure, including the `basdat` database and the `address` collection. The main panel shows the `Documents` tab for the `address` collection, displaying three documents with their fields and values. The documents are:

- `_id: ObjectId('6602aa59ac31a22a9d8e1b73')`, `ID: 3`, `CityID: 1`, `Address: "Sudirman"`, `Address2: "Street"`, `District: 8`, `Postal_Code: 2222`, `Phone: "8818181818"`, `Last_Update: 2024-03-26T10:58:33.306+00:00`
- `_id: ObjectId('6602aa59ac31a22a9d8e1b74')`, `ID: 4`, `CityID: 1`, `Address: "Thamrin"`, `Address2: "Avenue"`, `District: 6`, `Postal_Code: 3333`, `Phone: "8828282828"`, `Last_Update: 2024-03-26T10:58:33.306+00:00`
- `_id: ObjectId('6602aa59ac31a22a9d8e1b75')`, `ID: 5`, `CityID: 1`, `Address: "Bogor"`, `Address2: "Avenue"`, `District: 8`, `Postal_Code: 4444`, `Phone: "7373739900"`, `Last_Update: 2024-03-26T10:58:33.306+00:00`

## Update Query address

The screenshot shows the MongoDB Playground interface. On the left, the 'CONNECTIONS' panel shows a connection to 'localhost:27017' with a database named 'basdat'. The 'PLAYGROUNDS' panel shows 'No MongoDB playground files found in the workspace.' and a 'Create New Playground' button. The main editor shows a JavaScript query:

```
28 // { $set: { Last_Name: "Marahap" } }
29 // };
30
31 // db.actor.deleteMany( { ID: { $in: [5,6] } });
32
33 > db.film.insertMany([...
79 // ]);
80
81 // db.film.updateMany(
82 //   { ID: { $in: [5, 6] } },
83 //   { $set: { Release_Year: "2024" } }
84 // );
85
86 // db.film.deleteMany( { ID: { $in: [5,6] } });
87
88 > db.address.insertMany([...
119 // ]);
120
121 db.address.updateMany(
122   { ID: { $in: [4, 5] } },
123   { $set: { Postal_Code: "1416" } }
124
125
```

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

```
1 {
2   "acknowledged": true,
3   "insertedIds": {
4     "0": {
5       "$oid": "6602aa59ac31a22a9d8e1b73"
6     },
7     "1": {
8       "$oid": "6602aa59ac31a22a9d8e1b74"
9     },
10    "2": {
11      "$oid": "6602aa59ac31a22a9d8e1b75"
12    }
13  }
14 }
```

The status bar at the bottom shows 'Ln 125, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'JavaScript', 'Go Live', and 'Prettier'.

The screenshot shows the MongoDB Compass interface. The left sidebar shows the 'basdat' database with collections: 'Modul3', 'admin', 'basdat', 'actor', 'address', 'film', 'config', 'local', and 'rentalfilm'. The 'address' collection is selected. The main panel shows the 'Documents' tab with two documents:

```
Last_Update : 2024-03-26T10:58:33.396+00:00

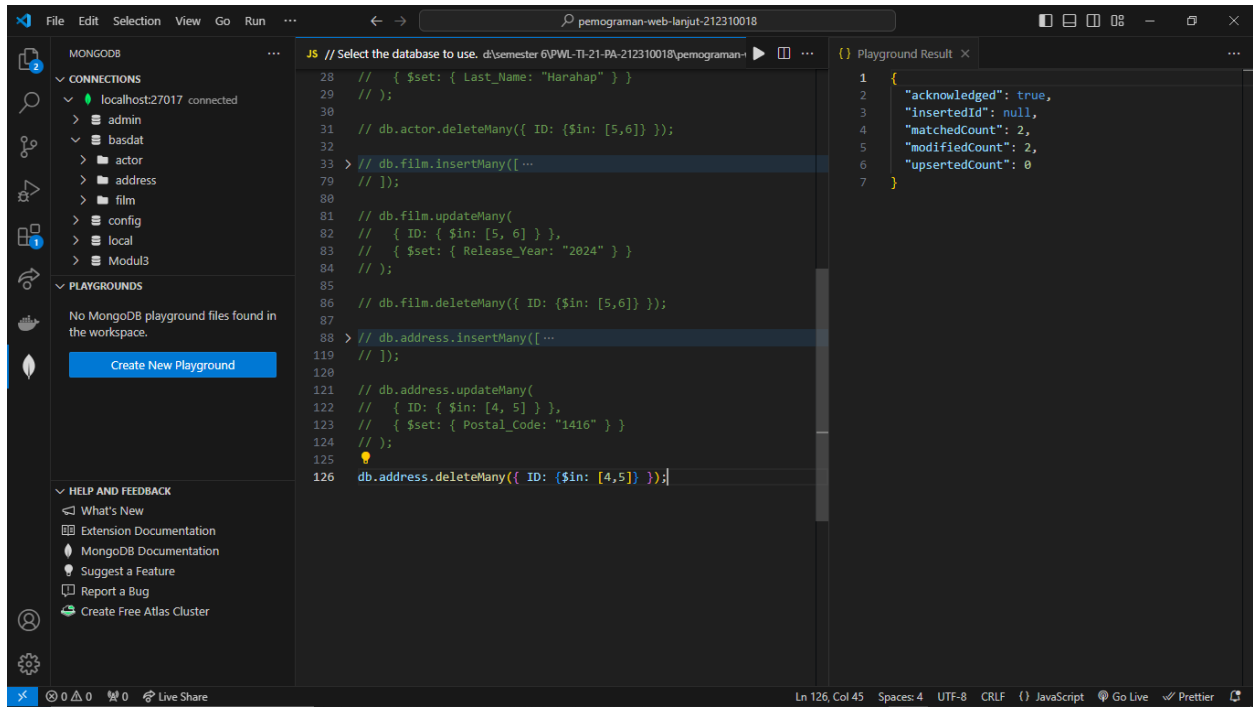
{
  "_id": ObjectId('6602aa59ac31a22a9d8e1b74'),
  "ID": 4,
  "CityID": 1,
  "Address": "Thamrin",
  "Address2": "Avenue",
  "District": 6,
  "Postal_Code": "1416",
  "Phone": "8828282828",
  "Last_Update": 2024-03-26T10:58:33.396+00:00
}

{
  "_id": ObjectId('6602aa59ac31a22a9d8e1b75'),
  "ID": 5,
  "CityID": 1,
  "Address": "Bogor",
  "Address2": "Avenue",
  "District": 8,
  "Postal_Code": "1416",
  "Phone": "7373739988",
  "Last_Update": 2024-03-26T10:58:33.396+00:00
}
```

The status bar at the bottom shows '> MONGODB'.



## Delete Query address



The screenshot shows the MongoDB Playground interface. On the left, the 'CONNECTIONS' panel shows a connection to 'localhost:27017' with a database named 'basdat'. The 'PLAYGROUNDS' panel shows 'No MongoDB playground files found in the workspace.' The main editor displays a JavaScript query to delete an address document. The query is as follows:

```
28 // { $set: { Last_Name: "Marahap" } }
29 // };
30
31 // db.actor.deleteMany({ ID: {$in: [5,6]} });
32
33 > db.film.insertMany([...
79 // ]);
80
81 // db.film.updateMany(
82 //   { ID: { $in: [5, 6] } },
83 //   { $set: { Release_Year: "2024" } }
84 // );
85
86 // db.film.deleteMany({ ID: {$in: [5,6]} });
87
88 > db.address.insertMany([...
119 // ]);
120
121 // db.address.updateMany(
122 //   { ID: { $in: [4, 5] } },
123 //   { $set: { Postal_Code: "1416" } }
124 // );
125
126 db.address.deleteMany({ ID: {$in: [4,5]} });
```

The 'Playground Result' panel on the right shows the result of the query:

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

