

Fathurahman AL Faridzi FSWD 5B

Melakukan seeding untuk menambahkan data baru

The image shows two screenshots illustrating the process of seeding a database.

Top Screenshot (VS Code): The editor shows a file named `seeding.sql` with the following SQL code:

```
1 INSERT INTO public.actor (actor_id, first_name, last_name, last_update)
2 VALUES (201, 'Fathurahman', 'AL Faridzi', NOW()),
3         (202, 'Angga', 'Parulian', NOW()),
4         (203, 'Dyas', 'Tri', NOW()),
5         (204, 'Zacky', 'Afif', NOW()),
6         (205, 'Radian', 'Rama', NOW());
7
8
9
```

The terminal output shows the command `npm run seed` being executed, resulting in the message `seeding complete`.

Bottom Screenshot (pgAdmin 4): The interface shows the `public` schema selected in the Object Explorer. The Query tool is used to execute the following SQL query:

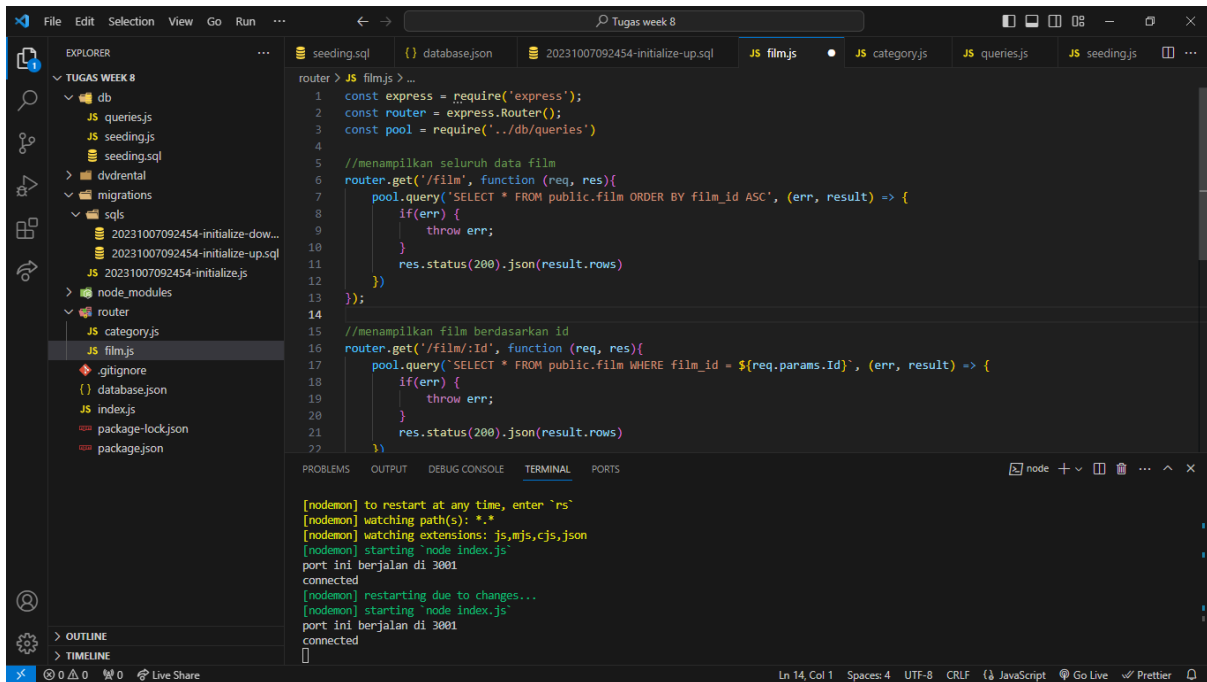
```
1 SELECT * FROM public.actor
2 ORDER BY actor_id ASC
```

The Data Output pane displays the results of the query, showing 205 rows. The columns are `actor_id`, `first_name`, `last_name`, and `last_update`.

actor_id	first_name	last_name	last_update
198	Mary	Keller	2013-05-26 14:47:57.62
199	Julia	Fawcett	2013-05-26 14:47:57.62
200	Thora	Temple	2013-05-26 14:47:57.62
201	Fathurahman	AL Faridzi	2023-10-07 13:19:15.378023
202	Angga	Parulian	2023-10-07 13:19:15.378023
203	Dyas	Tri	2023-10-07 13:19:15.378023
204	Zacky	Afif	2023-10-07 13:19:15.378023
205	Radian	Rama	2023-10-07 13:19:15.378023

Total rows: 205 of 205. Query complete 00:00:00.335.

Menampilkan data seluruh list film

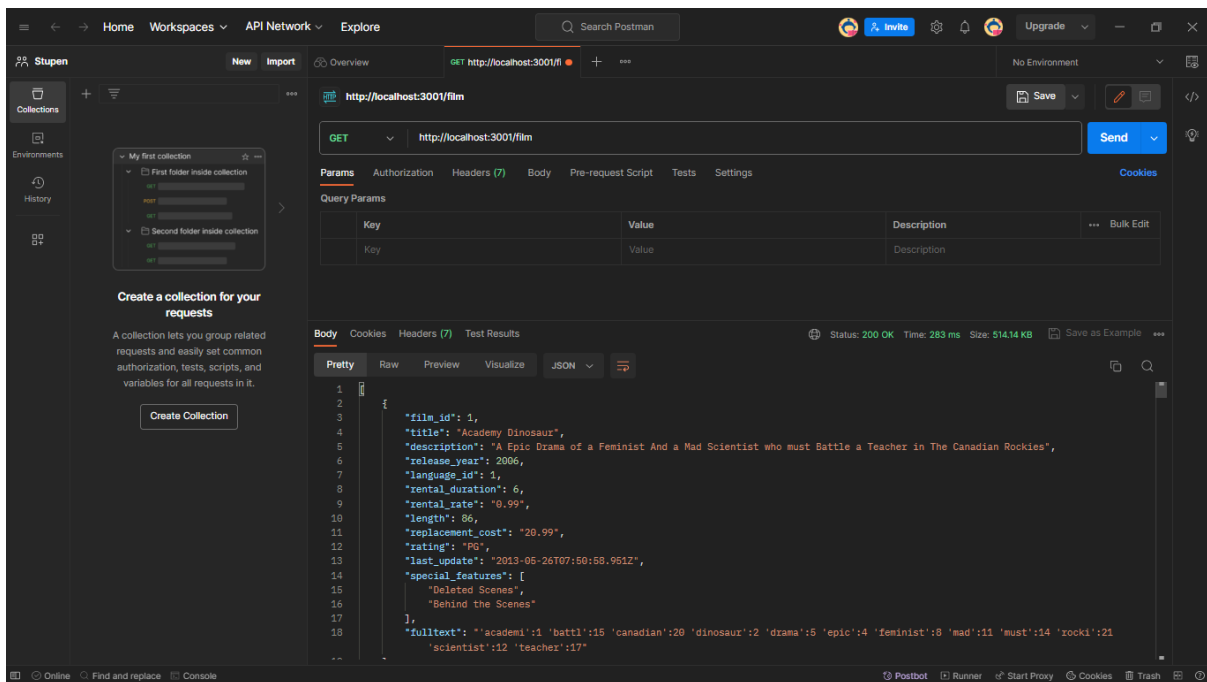


The screenshot shows a VS Code editor with a project named 'TUGAS WEEK 8'. The file explorer on the left shows a directory structure with files like 'db', 'queries.js', 'seeding.js', 'seeding.sql', 'migrations', 'sqls', '20231007092454-initialize-down...', '20231007092454-initialize-up.sql', '20231007092454-initialize.js', 'node_modules', 'router', 'category.js', 'film.js', '.gitignore', 'database.json', 'index.js', 'package-lock.json', and 'package.json'. The main editor shows the 'film.js' file with the following code:

```
router > JS film.js > ...
1  const express = require('express');
2  const router = express.Router();
3  const pool = require('../db/queries')
4
5  //menampilkan seluruh data film
6  router.get('/film', function (req, res){
7    pool.query('SELECT * FROM public.film ORDER BY film_id ASC', (err, result) => {
8      if(err) {
9        throw err;
10     }
11     res.status(200).json(result.rows)
12   })
13 });
14
15 //menampilkan film berdasarkan id
16 router.get('/film/:id', function (req, res){
17   pool.query('SELECT * FROM public.film WHERE film_id = ${req.params.id}', (err, result) => {
18     if(err) {
19       throw err;
20     }
21     res.status(200).json(result.rows)
22   })
23 });
```

The terminal at the bottom shows the output of the application running with nodemon:

```
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
port ini berjalan di 3001
connected
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
port ini berjalan di 3001
connected
```



The screenshot shows the Postman API client interface. The top bar indicates the current environment is 'No Environment'. The left sidebar shows the 'Stupen' collection, which is expanded to show 'My first collection' and 'Second folder inside collection'. The main area shows a GET request to 'http://localhost:3001/film'. The 'Send' button is visible. Below the request, the 'Body' tab is selected, showing the response in JSON format:

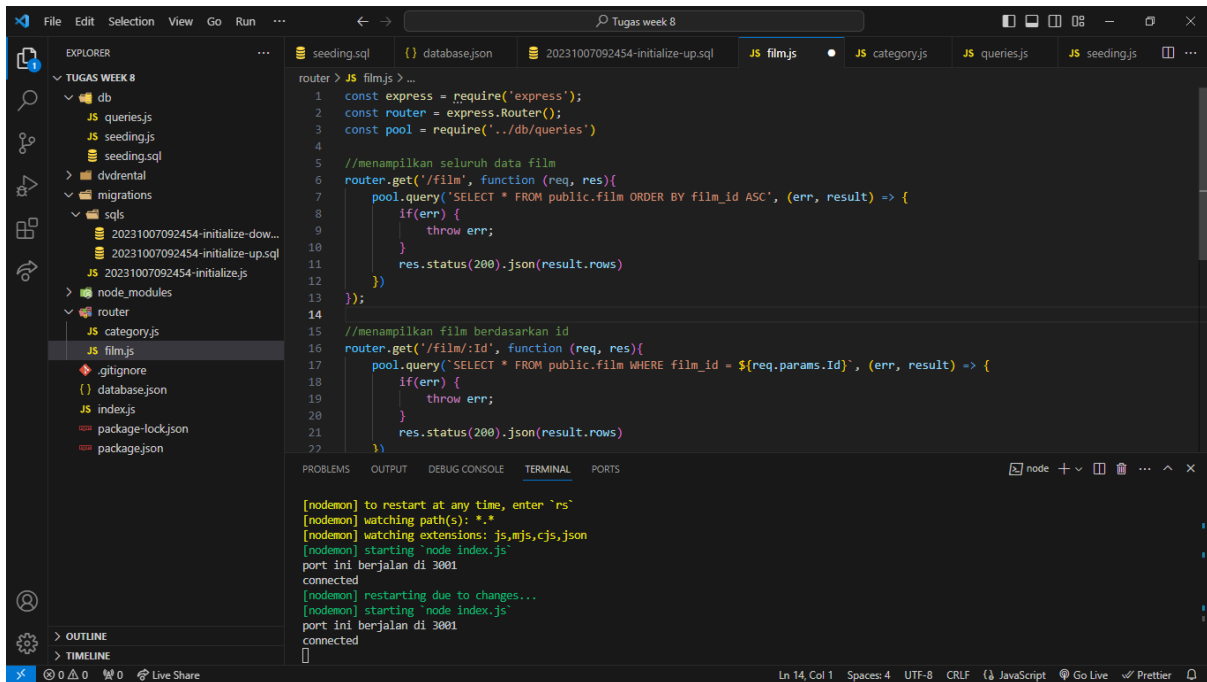
```
{
  "file_id": 1,
  "title": "Academy Dinosaur",
  "description": "A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies",
  "release_year": 2006,
  "language_id": 1,
  "rental_duration": 6,
  "rental_rate": "0.99",
  "length": 86,
  "replacement_cost": "20.99",
  "rating": "PG",
  "last_update": "2013-05-26T07:50:58.951Z",
  "special_features": [
    "Deleted Scenes",
    "Behind the Scenes"
  ],
  "fulltext": "academ:1 'battl':15 'canadian':20 'dinosaur':2 'drama':15 'epic':14 'feminist':8 'mad':11 'must':14 'rocki':21 'scientist':12 'teacher':17"
```

The status bar at the bottom shows 'Status: 200 OK', 'Time: 283 ms', and 'Size: 514.14 KB'. The bottom right corner shows the 'Postman' logo, 'Runner', 'Start Proxy', 'Cookies', 'Trash', and 'Help' icons.

```
localhost:3001/film
localhost:3001/film

[{"film_id":1,"title":"Academy Dinosaur","description":"A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies","release_year":2006,"language_id":1,"rental_duration":6,"rental_rate":"0.99","length":86,"replacement_cost":"20.99","rating":"PG","last_update":"2013-05-26T07:50:58.951Z","special_features":["Deleted Scenes","Behind the Scenes"],"fulltext":"academi:1 'battl':15 'canadian':20 'dinosaur':2 'drama':5 'epic':4 'feminist':8 'mad':11 'must':14 'rocki':21 'scientist':12 'teacher':17"}, {"film_id":2,"title":"Ace Goldfinger","description":"A Astounding Epistle of a Database Administrator And a Explorer who must Find a Car in Ancient China","release_year":2006,"language_id":1,"rental_duration":3,"rental_rate":"4.99","length":48,"replacement_cost":"12.99","rating":"G","last_update":"2013-05-26T07:50:58.951Z","special_features":["Trailers","Deleted Scenes"],"fulltext":"ace:1 'administr':9 'ancient':19 'astound':4 'car':17 'china':20 'databas':8 'epistl':5 'explor':12 'find':15 'goldfing':2 'must':14"}, {"film_id":3,"title":"Adaptation Holes","description":"A Astounding Reflection of a Lumberjack And a Car who must Sink a Lumberjack in A Baloon Factory","release_year":2006,"language_id":1,"rental_duration":7,"rental_rate":"2.99","length":50,"replacement_cost":"18.99","rating":"NC-17","last_update":"2013-05-26T07:50:58.951Z","special_features":["Trailers","Deleted Scenes"],"fulltext":"adapt:1 'astound':4 'baloon':19 'car':11 'factori':20 'hole':12 'lumberjack':8,16 'must':13 'reflect':5 'sink':14}), {"film_id":4,"title":"Affair Prejudice","description":"A Fanciful Documentary of a Frisbee And a Lumberjack who must Chase a Monkey in A Shark Tank","release_year":2006,"language_id":1,"rental_duration":5,"rental_rate":"2.99","length":117,"replacement_cost":"26.99","rating":"G","last_update":"2013-05-26T07:50:58.951Z","special_features":["Commentaries","Behind the Scenes"],"fulltext":"affair:1 'chase':14 'documentari':5 'fancil':4 'frisbe':8 'lumberjack':11 'monkey':16 'must':13 'prejudic':2 'shark':19 'tank':20"}, {"film_id":5,"title":"African Egg","description":"A Fast-Paced Documentary of a Pastry Chef And a Dentist who must Pursue a Forensic Psychologist in The Gulf of Mexico","release_year":2006,"language_id":1,"rental_duration":6,"rental_rate":"2.99","length":130,"replacement_cost":"22.99","rating":"G","last_update":"2013-05-26T07:50:58.951Z","special_features":["Deleted Scenes"],"fulltext":"african:1 'chef':11 'dentist':14 'documentari':7 'egg':2 'fast':5 'fast-pac':4 'forens':19 'gulf':23 'mexico':25 'must':16 'pace':6 'pastri':10 'psychologist':20 'pursu':17"}, {"film_id":6,"title":"Agent Truman","description":"A Intrepid Panorama of a Robot And a Boy who must Escape a Sumo Wrestler in Ancient China","release_year":2006,"language_id":1,"rental_duration":3,"rental_rate":"2.99","length":169,"replacement_cost":"17.99","rating":"PG","last_update":"2013-05-26T07:50:58.951Z","special_features":["Deleted Scenes"],"fulltext":"agent:1 'ancient':19 'boy':11 'china':20 'escap':14 'intrepid':4 'must':13 'panorama':5 'robot':8 'sumo':16 'truman':2 'wrestler':17}), {"film_id":7,"title":"Airplane Sierra","description":"A Touching Saga of a Hunter And a Butler who must Discover a Butler in A Jet Boat","release_year":2006,"language_id":1,"rental_duration":6,"rental_rate":"4.99","length":62,"replacement_cost":"28.99","rating":"PG-13","last_update":"2013-05-26T07:50:58.951Z","special_features":["Trailers","Deleted Scenes"],"fulltext":"airplan:1 'boat':20 'butler':11,16 'discov':14 'hunter':8 'jet':19 'must':13 'saga':5 'sierra':2 'touch':4}), {"film_id":8,"title":"Airport Pollock","description":"A Epic Tale of a Moose And a Girl who must Confront a Monkey in Ancient India","release_year":2006,"language_id":1,"rental_duration":6,"rental_rate":"4.99","length":54,"replacement_cost":"15.99","rating":"R","last_update":"2013-05-26T07:50:58.951Z","special_features":["Trailers"],"fulltext":"airport:1 'ancient':18 'confront':14 'epic':4 'girl':11 'india':19 'monkey':16 'moos':8 'must':13 'pollock':2 'tale':5}), {"film_id":9,"title":"Alabama Devil","description":"A Thoughtful Panorama of a Database Administrator And a Mad Scientist who must Outgun a Mad Scientist in A Jet Boat","release_year":2006,"language_id":1,"rental_duration":3,"rental_rate":"2.99","length":114,"replacement_cost":"21.99","rating":"PG-13","last_update":"2013-05-26T07:50:58.951Z","special_features":["Trailers","Deleted Scenes"],"fulltext":"alabama:1 'boat':23 'databas':8 'devil':2 'jet':22 'mad':12,18 'must':15 'outgun':16 'panorama':5 'scientist':13,19 'thought':4"}, {"film_id":10,"title":"Aladdin Calendar","description":"A Action-Packed Tale of a Men And a Lumberjack who must Reach a feminist in Ancient China","release_year":2006,"language_id":1,"rental_duration":6,"rental_rate":"4.99","length":63,"replacement_cost":"24.99","rating":"NC-17","last_update":"2013-05-26T07:50:58.951Z","special_features":["Trailers","Deleted Scenes"],"fulltext":"action:15 'action-pack':4 'aladdin':1 'ancient':20 'calenda':12 'china':21 'feminist':18 'lumberjack':13 'man':10 'must':15 'pack':6 'reach':16 'tale':7"}, {"film_id":11,"title":"Alamo Videotape","description":"A Boring Epistle of a Butler And a Cat who must Fight a Pastry Chef in A MySQL Convention","release_year":2006,"language_id":1,"rental_duration":6,"rental_rate":"0.99","length":126,"replacement_cost":"16.99","rating":"G","last_update":"2013-05-26T07:50:58.951Z","special_features":["Commentaries","Behind the Scenes"],"fulltext":"alamo:1 'bore':4 'butler':8 'cat':11 'chef':17 'convent':21 'epistl':5 'fight':14 'must':13 'mysql':20 'pastri':16 'videotap':2"}, {"film_id":12,"title":"Alaska Phantom","description":"A Fanciful Saga of a Hunter And a Pastry Chef who must Vanquish a Boy in Australia","release_year":2006,"language_id":1,"rental_duration":6,"rental_rate":"0.99","length":136,"replacement_cost":"22.99","rating":"PG","last_update":"2013-05-26T07:50:58.951Z","special_features":["Commentaries","Deleted Scenes"],"fulltext":"alaska:1 'australia':19 'boy':17 'chef':12 'fanci':4 'hunter':8 'must':14 'pastri':11 'phantom':2 'saga':5 'vanquish':15}), {"film_id":13,"title":"Ali Forever","description":"A Action-Packed Drama of a Dentist And a Crocodile who must Battle a Feminist in The Canadian Rockies","release_year":2006,"language_id":1,"rental_duration":4,"rental_rate":"4.99","length":150,"replacement_cost":"21.99","rating":"PG","last_update":"2013-05-26T07:50:58.951Z","special_features":["Deleted Scenes","Behind the Scenes"],"fulltext":"action:15 'action-pack':4 'all':1 'battl':16 'canadian':21 'crocodil':13 'dentist':10 'drama':7 'feminist':18 'forev':2 'must':15 'pack':6 'rocki':22"}, {"film_id":14,"title":"Alice Fantasia","description":"A Emotional Drama of a A Shark And a Database Administrator who must Vanquish a Pioneer in Soviet Georgia","release_year":2006,"language_id":1,"rental_duration":6,"rental_rate":"0.99","length":94,"replacement_cost":"23.99","rating":"NC-17","last_update":"2013-05-26T07:50:58.951Z","special_features":["Trailers","Deleted Scenes","Behind the Scenes"],"fulltext":"administr:13 'alic':1 'databas':12 'drama':5 'emot':14 'fantasia':2 'georgia':21 'must':15 'pioneer':18 'shark':19 'soviet':20 'vanquish':16"}, {"film_id":15,"title":"Alien Center","description":"A Brilliant Drama of a Cat And a Mad Scientist who must Battle a Feminist in A MySQL Convention","release_year":2006,"language_id":1,"rental_duration":5,"rental_rate":"2.99","length":46,"replacement_cost":"10.99","rating":"NC-17","last_update":"2013-05-26T07:50:58.951Z","special_features":["Trailers","Commentaries","Behind the Scenes"],"fulltext":"alien:1 'battl':15 'brilliant':4 'cat':8 'center':2 'convent':21 'drama':5 'feminist':17 'mad':11 'must':14 'mysql':20 'scientist':12"}, {"film_id":16,"title":"Alley Evolution","description":"A Fast-Paced Drama of a Robot And a Composer who must Battle a Astronaut in New Orleans","release_year":2006,"language_id":1,"rental_duration":6,"rental_rate":"2.99","length":100,"replacement_cost":"23.99","rating":"NC-17","last_update":"2013-05-26T07:50:58.951Z","special_features":}
```

Menampilkan data film tertentu berdasarkan id

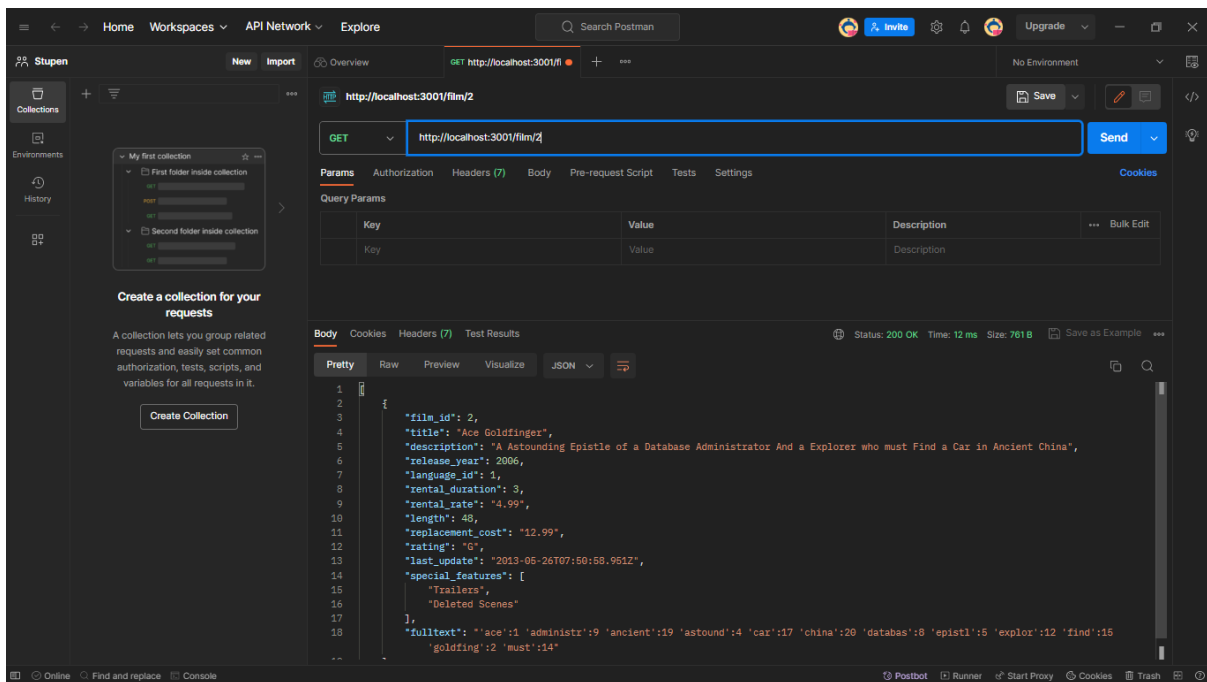


The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows a project structure for 'TUGAS WEEK 8' with folders 'db', 'migrations', 'sqls', 'node_modules', and 'router'. The 'router' folder contains 'category.js', 'film.js', 'index.js', 'package-lock.json', and 'package.json'. The 'film.js' file is selected and its content is displayed in the editor. The code defines an Express router with two endpoints: one for listing all films and another for listing a film by its ID. The terminal at the bottom shows the output of running the application, indicating that the server is running on port 3001 and has restarted due to changes.

```
router > JS film.js > ...
1  const express = require('express');
2  const router = express.Router();
3  const pool = require('../db/queries')
4
5  //menampilkan seluruh data film
6  router.get('/film', function (req, res){
7    pool.query('SELECT * FROM public.film ORDER BY film_id ASC', (err, result) => {
8      if(err) {
9        throw err;
10     }
11     res.status(200).json(result.rows)
12   });
13 });
14
15 //menampilkan film berdasarkan id
16 router.get('/film/:id', function (req, res){
17   pool.query('SELECT * FROM public.film WHERE film_id = ${req.params.id}', (err, result) => {
18     if(err) {
19       throw err;
20     }
21     res.status(200).json(result.rows)
22   });
23 });
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
port ini berjalan di 3001
connected
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
port ini berjalan di 3001
connected
```



The screenshot shows the Postman interface. The top bar indicates the environment is 'No Environment'. The left sidebar shows a 'Collections' panel with a collection named 'My First collection' containing two folders: 'First folder inside collection' and 'Second folder inside collection'. The main area shows a GET request to 'http://localhost:3001/film/2'. The 'Send' button is highlighted. Below the request, the 'Body' tab is selected, showing the response in JSON format. The response is a JSON object representing a film record.

GET http://localhost:3001/film/2

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

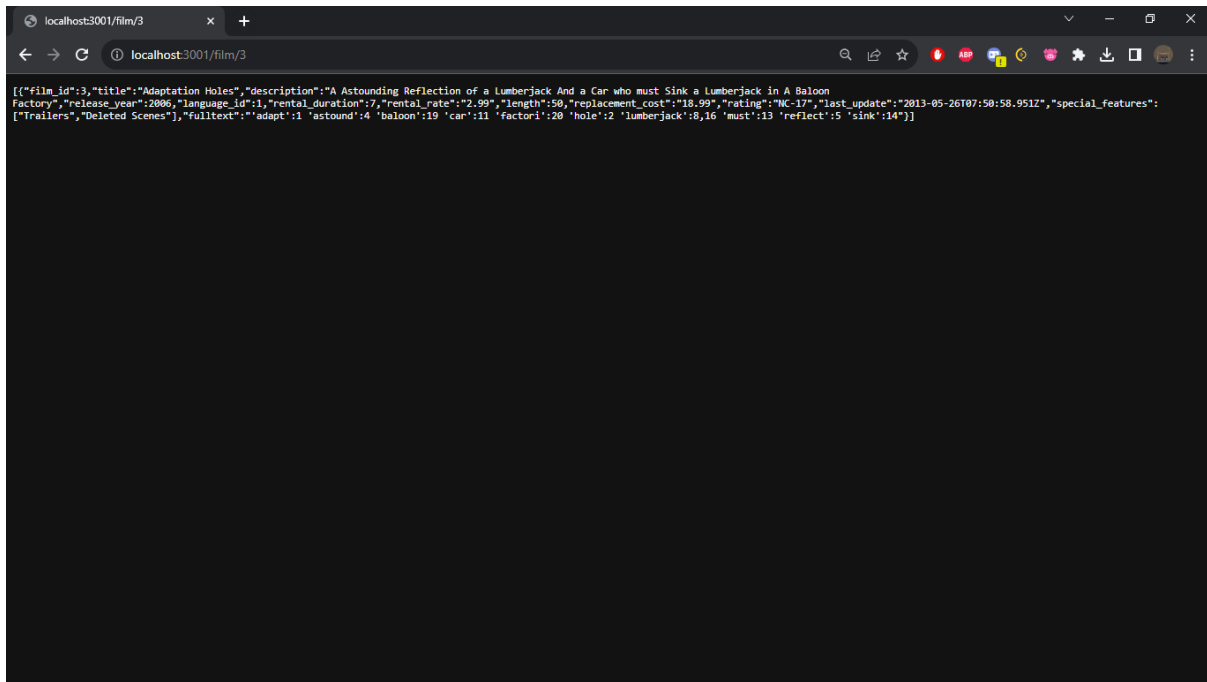
Key	Value	Description
Key	Value	Description

Body Cookies Headers (7) Test Results

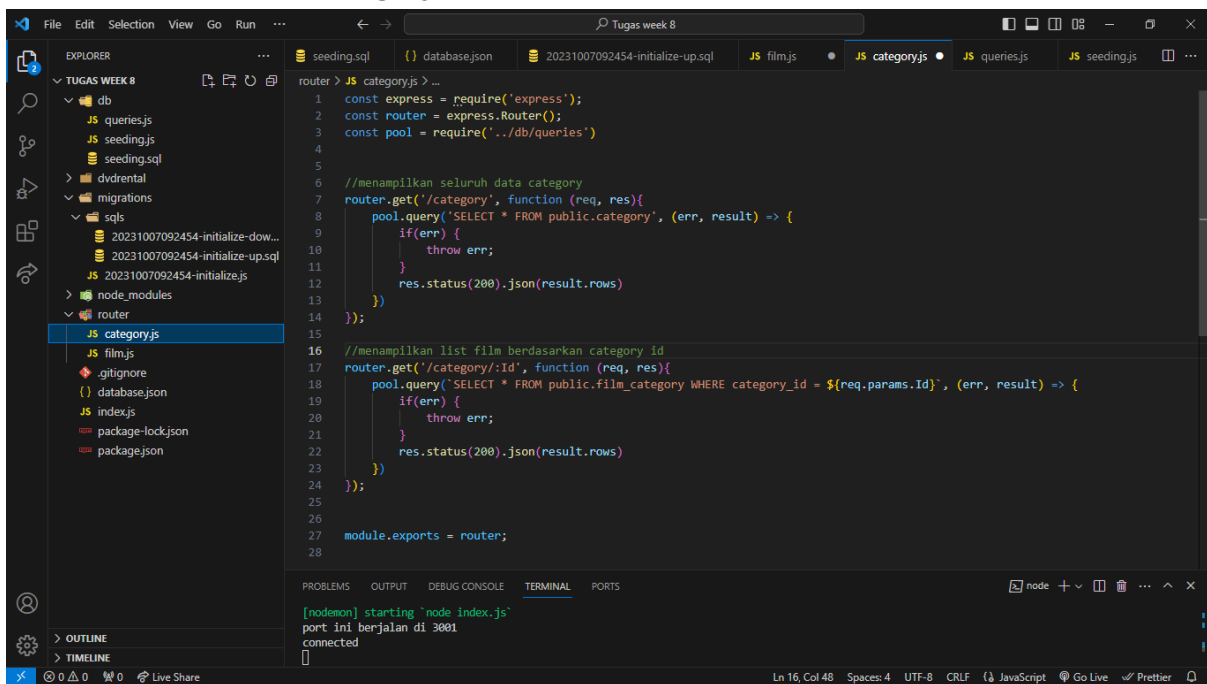
Status: 200 OK Time: 12 ms Size: 761 B Save as Example

Pretty Raw Preview Visualize JSON

```
1  {
2    "file_id": 2,
3    "title": "Ace Goldfinger",
4    "description": "A Astounding Epistle of a Database Administrator And a Explorer who must Find a Car in Ancient China",
5    "release_year": 2006,
6    "language_id": 1,
7    "rental_duration": 3,
8    "rental_rate": "4.99",
9    "length": 48,
10   "replacement_cost": "12.99",
11   "rating": "G",
12   "last_update": "2013-05-26T07:50:58.951Z",
13   "special_features": [
14     "Trailers",
15     "Deleted Scenes"
16   ],
17   "fulltext": "ace:1 'administr':9 'ancient':19 'astound':4 'car':17 'china':20 'databas':8 'epistl':5 'explor':12 'find':15
18     'goldfing':2 'must':14"
```



Menampilkan data list category



Stupen API Network Explorer

GET http://localhost:3001/category

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "category_id": 1,
3   "name": "Action",
4   "last_update": "2006-02-15T02:46:27.000Z"
5 },
6 {
7   "category_id": 2,
8   "name": "Animation",
9   "last_update": "2006-02-15T02:46:27.000Z"
10 },
11 {
12   "category_id": 3,
13   "name": "Children",
14   "last_update": "2006-02-15T02:46:27.000Z"
15 },
16 {
17   "category_id": 4,
18   "name": "Classics",
19   "last_update": "2006-02-15T02:46:27.000Z"
20 },
21 {
22   "category_id": 5,
23   "name": "Comedy",
24   "last_update": "2006-02-15T02:46:27.000Z"
25 },
26 {
27   "category_id": 6,
28   "name": "Documentary",
29   "last_update": "2006-02-15T02:46:27.000Z"
30 },
31 {
32   "category_id": 7,
33   "name": "Drama",
34   "last_update": "2006-02-15T02:46:27.000Z"
35 },
36 {
37   "category_id": 8,
38   "name": "Family",
39   "last_update": "2006-02-15T02:46:27.000Z"
40 },
41 {
42   "category_id": 9,
43   "name": "Foreign",
44   "last_update": "2006-02-15T02:46:27.000Z"
45 },
46 {
47   "category_id": 10,
48   "name": "Games",
49   "last_update": "2006-02-15T02:46:27.000Z"
50 },
51 {
52   "category_id": 11,
53   "name": "Horror",
54   "last_update": "2006-02-15T02:46:27.000Z"
55 },
56 {
57   "category_id": 12,
58   "name": "Music",
59   "last_update": "2006-02-15T02:46:27.000Z"
60 },
61 {
62   "category_id": 13,
63   "name": "New",
64   "last_update": "2006-02-15T02:46:27.000Z"
65 },
66 {
67   "category_id": 14,
68   "name": "Sci-Fi",
69   "last_update": "2006-02-15T02:46:27.000Z"
70 },
71 {
72   "category_id": 15,
73   "name": "Sports",
74   "last_update": "2006-02-15T02:46:27.000Z"
75 },
76 {
77   "category_id": 16,
78   "name": "Travel",
79   "last_update": "2006-02-15T02:46:27.000Z"
80 },
81 }
```

Status: 200 OK Time: 377 ms Size: 1.42 KB Save as Example

localhost:3001/category

```
[{"category_id":1,"name":"Action","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":2,"name":"Animation","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":3,"name":"Children","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":4,"name":"Classics","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":5,"name":"Comedy","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":6,"name":"Documentary","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":7,"name":"Drama","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":8,"name":"Family","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":9,"name":"Foreign","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":10,"name":"Games","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":11,"name":"Horror","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":12,"name":"Music","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":13,"name":"New","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":14,"name":"Sci-Fi","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":15,"name":"Sports","last_update":"2006-02-15T02:46:27.000Z"}, {"category_id":16,"name":"Travel","last_update":"2006-02-15T02:46:27.000Z"}]
```

Menampilkan data list film berdasarkan category

The screenshot shows the VS Code editor with a project named 'TUGAS WEEK 8'. The file explorer on the left shows a directory structure with files like 'queries.js', 'seeding.js', 'migrations', 'sqls', 'node_modules', 'router', 'film.js', 'database.json', 'index.js', 'package-lock.json', and 'package.json'. The main editor displays the 'category.js' file, which contains the following code:

```
1 const express = require('express');
2 const router = express.Router();
3 const pool = require('../db/queries')
4
5
6 //menampilkan seluruh data category
7 router.get('/category', function (req, res){
8   pool.query('SELECT * FROM public.category', (err, result) => {
9     if(err) {
10       throw err;
11     }
12     res.status(200).json(result.rows)
13   });
14 });
15
16 //menampilkan list film berdasarkan category id
17 router.get('/category/:id', function (req, res){
18   pool.query('SELECT * FROM public.film_category WHERE category_id = ${req.params.id}', (err, result) => {
19     if(err) {
20       throw err;
21     }
22     res.status(200).json(result.rows)
23   });
24 });
25
26
27 module.exports = router;
28
```

The terminal at the bottom shows the command '[nodemon] starting "node index.js"' and the message 'port ini berjalan di 3001 connected'.

The screenshot shows the Postman interface. The URL bar is set to 'http://localhost:3001/category/2'. The request method is 'GET'. The 'Send' button is clicked, and the response is displayed in the 'Body' tab. The response is a JSON array of movie objects, each containing 'file_id', 'category_id', and 'last_update'.

Key	Value	Description
Key	Value	Description

```
1 {
2   "file_id": 19,
3   "category_id": 2,
4   "last_update": "2006-02-15T03:07:09.000Z"
5 },
6 {
7   "file_id": 23,
8   "category_id": 2,
9   "last_update": "2006-02-15T03:07:09.000Z"
10 },
11 {
12   "file_id": 36,
13   "category_id": 2,
14   "last_update": "2006-02-15T03:07:09.000Z"
15 },
16 {
17   "file_id": 70,
18   "category_id": 2,
19   "last_update": "2006-02-15T03:07:09.000Z"
20 }
```

[illegible]

igrasi database dengan menambahkan column age pada tabel actor

The image shows a two-part workflow for migrating a database. The top part is a Visual Studio Code editor window with a file explorer on the left showing a project structure for 'TUGAS WEEK 8'. The main editor displays a SQL migration file '20231007092454-initialize-up.sql' with the following content:

```
1 /* Replace with your SQL commands */
2
3 ALTER TABLE public.actor
4 ADD age integer DEFAULT 0;
5
6
```

The bottom part of the image shows the pgAdmin 4 interface. The 'Object Explorer' on the left shows the 'public' schema. The 'Query' tab is active, showing the same SQL command: `SELECT * FROM public.actor ORDER BY actor_id ASC`. The 'Data Output' tab shows the result of the query, which is a table with 8 rows and 6 columns: `actor_id` (integer), `first_name` (character varying (45)), `last_name` (character varying (45)), `last_update` (timestamp without time zone), `age` (integer), and an unnamed column. The data is as follows:

actor_id	first_name	last_name	last_update	age	
1	Penelope	Guinness	2013-05-26 14:47:57.62	0	
2	Nick	Wahlberg	2013-05-26 14:47:57.62	0	
3	Ed	Chase	2013-05-26 14:47:57.62	0	
4	Jennifer	Davis	2013-05-26 14:47:57.62	0	
5	Johnny	Loiobrigida	2013-05-26 14:47:57.62	0	
6	Bette	Nicholson	2013-05-26 14:47:57.62	0	
7	Grace	Mostel	2013-05-26 14:47:57.62	0	
8	Matthew	Johansson	2013-05-26 14:47:57.62	0	

The status bar at the bottom of pgAdmin 4 indicates 'Total rows: 205 of 205' and 'Query complete 00:00:00.921'.