

**Nama: Muhammad Rizky Al Farabi**

**Kelas: 5-A FSWD**

**Kelompok: 4**

## **Tugas: HomeWork (Week 8)**

---

- **Soal**
  - **Soal Homework**

Buatlah sebuah aplikasi Express JS yang dapat menampilkan data text pada halaman web yang didapatkan dari sampel data ini dan schema di sini sebagai referensi.

Soal 1: Lakukan seeding untuk menambahkan 5 data baru pada data table actor yang disediakan

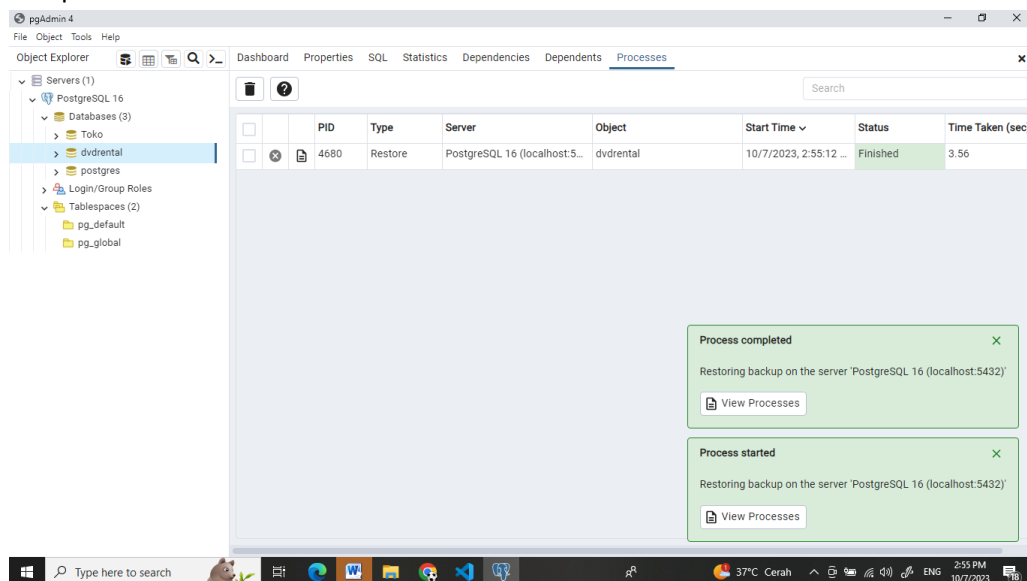
Soal 2 : Buatlah query untuk menampilkan data pada database sebagai berikut.

1. Menampilkan data seluruh list film
2. Menampilkan data film tertentu berdasarkan id
3. Menampilkan data list category
4. Menampilkan data list film berdasarkan category

Implementasikan routing yang sesuai dan benar pada aplikasi yang teman-teman buat.

Soal 3: Lakukan migrasi database dengan menambahkan column age pada tabel actor yang

- **Jawab**
  - Import Database



## -Inizialitation

```
TERMINAL  PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE

PS C:\Users\ASUS\Documents\Farabi Maba ULM\Semester 5\MSIB\Homework8> mkdir express-postgresql-app

Directory: C:\Users\ASUS\Documents\Farabi Maba ULM\Semester 5\MSIB\Homework8

Mode                LastWriteTime         Length Name
----                -
PS C:\Users\ASUS\Documents\Farabi Maba ULM\Semester 5\MSIB\Homework8> cd express-postgresql-app
PS C:\Users\ASUS\Documents\Farabi Maba ULM\Semester 5\MSIB\Homework8\express-postgresql-app> npm init -y
Wrote to C:\Users\ASUS\Documents\Farabi Maba ULM\Semester 5\MSIB\Homework8\express-postgresql-app\package.json:

{
  "name": "express-postgresql-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

PS C:\Users\ASUS\Documents\Farabi Maba ULM\Semester 5\MSIB\Homework8\express-postgresql-app> npm install express sequelize pg pg-hstore ejs
added 113 packages, and audited 114 packages in 28s

11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\ASUS\Documents\Farabi Maba ULM\Semester 5\MSIB\Homework8\express-postgresql-app>
```

## -Code

Selanjutnya saya membuat aplikasi Express.js dengan routing yang sesuai untuk mengakses data dari database dan menampilkan halaman web, bisa dilihat gambar dibawah ini.

```
JS indexjs x
JS indexjs > ...
1 // app.js
2 const express = require('express');
3 const Sequelize = require('sequelize');
4 const ejs = require('ejs');
5 const db = require('./models');
6
7 const app = express();
8
9 app.set('view engine', 'ejs');
10
11 // Middleware untuk mengizinkan penggunaan file statis (misalnya: CSS, JS)
12 app.use(express.static('public'));
13
14 // Routing untuk menampilkan seluruh list film
15 app.get('/films', async (req, res) => {
16   const films = await db.Film.findAll();
17   res.render('films', { films });
18 });
19
20 // Routing untuk menampilkan data film tertentu berdasarkan id
21 app.get('/films/:id', async (req, res) => {
22   const film = await db.Film.findById(req.params.id);
23   res.render('film', { film });
24 });
25
26 // Routing untuk menampilkan list category
27 app.get('/categories', async (req, res) => {
28   const categories = await db.Category.findAll();
29   res.render('categories', { categories });
30 });
31
32 // Routing untuk menampilkan list film berdasarkan category
33 app.get('/categories/:id/films', async (req, res) => {
34   const category = await db.Category.findById(req.params.id, {
35     include: [{ model: db.Film }],
36   });
37   res.render('category_films', { category });
38 });
39
40 app.listen(3000, () => {
41   console.log('Server berjalan pada port 3000');
42 });
43
```

-Selanjutnya, saya perlu mengkonfigurasi Sequelize untuk mengakses database PostgreSQL saya. Saya membuat file config.js dalam folder proyek dan menambahkan konfigurasi.

```
JS config.js X
express-postgresql-app > JS config.js > ...
1 // config.js
2
3 module.exports = {
4   development: {
5     username: 'postgres',
6     password: 'rizky123',
7     database: 'dvdrental',
8     host: 'localhost',
9     dialect: 'postgres',
10   },
11 };
12
```

-Setelah migrasi selesai, tambahkan kolom age secara manual ke dalam definisi model Actor di file models/actor.js:

```
JS actor.js X
express-postgresql-app > node_modules > models > JS actor.js > ...
1 // models/actor.js
2
3 'use strict';
4 const { Model } = require('sequelize');
5 module.exports = (sequelize, DataTypes) => {
6   class Actor extends Model {
7     static associate(models) {
8       // Define associations here
9     }
10  }
11  Actor.init(
12    {
13      name: DataTypes.STRING,
14      age: DataTypes.INTEGER, // tambahkan kolom age
15    },
16    {
17      sequelize,
18      modelName: 'Actor',
19    }
20  );
21  return Actor;
22 };
23
```

-Setelah itu, saya membuat file seeder untuk tabel actor, film, dan category menggunakan Sequelize CLI atau dengan membuat file JavaScript manual yang mengisi data. Untuk mempermudah, saya membuat file seed manual seperti ini:



```
JS seed.js X
express-postgresql-app > JS seed.js > categoryId
1 // seed.js
2 const db = require('./models');
3
4 db.Actor.bulkCreate([
5   { name: 'Lukman Sardi', age: 30 },
6   { name: 'Bryan Domani', age: 25 },
7   { name: 'Iko Uwais', age: 35 },
8   { name: 'Reza Rahadian', age: 35 },
9   { name: 'Iqbal Ramadhan', age: 28 },
10  ]);
11
12 db.Category.bulkCreate([
13   { name: 'Action' },
14   { name: 'Drama' },
15   { name: 'Comedy' },
16  ]);
17
18 db.Film.bulkCreate([
19   { title: 'HeadShot', categoryId: 1 },
20   { title: 'Habibie & Ainun', categoryId: 2 },
21   { title: 'Dilan 1990', categoryId: 2 },
22   { title: 'Orang Kaya Baru', categoryId: 3 },
23   { title: 'Bumi Manusia', categoryId: 2 },
24  ]);
25
```

- **Link Google Drive**  
<https://drive.google.com/drive/folders/1u8r-DgRO4tsRX1mgHBAsVLS9Qc4bAPPx?usp=sharing>
- **Link Repo**  
<https://github.com/rizkyfarabi03/MSIB-RAKAMIN-Farabi>