Nama: Yosan Pratiwi

NIM: 10231091

UTS PROWEB

Aplikasi CRUD dengan React, Express, dan PostgreSQL

Aplikasi ini merupakan proyek latihan untuk mengimplementasikan fitur CRUD menggunakan **React (Vite) di frontend**, **Express.js di backend**, dan **PostgreSQL sebagai database**.

📌 Fitur Aplikasi

- **Menampilkan daftar produk dengan tampilan responsif**
- **Menambahkan produk baru dengan validasi input**
- **Mengedit produk yang sudah ada dengan modal form**
- **Menghapus produk dengan konfirmasi sebelum eksekusi**
- **Menerapkan notifikasi sukses/gagal pada operasi CRUD**
- **Menghubungkan frontend dan backend menggunakan API**
- **Menampilkan data dalam tabel dengan fitur pencarian dan pagination**
- **Menggunakan state management dengan React Hooks (useState, useEffect)**

🚀 Cara Install dan Menjalankan Aplikasi

1. Clone Repository

```
git clone https://github.com/yosanpratiwi/ecommerce-kecil.git
cd repository-name
```

2. Install Dependensi

Backend

```
cd backend
npm install
```

Frontend

```

```

```
cd frontend
npm install
```

3. Konfigurasi Database

1. Pastikan PostgreSQL sudah terinstall.
2. Buat database dengan nama `ecommerce_kecil`.
3. Sesuaikan konfigurasi di file `db.js`.
4. Jalankan migrasi database:

```
cd backend
node migrate.js
```

4. Menjalankan Aplikasi

Menjalankan Backend

```
cd backend
node index.js
```

Backend akan berjalan di `http://localhost:3001`.

Menjalankan Frontend

```
cd frontend
npm run dev
```

Frontend akan berjalan di `http://localhost:5173`.

📌 Daftar Endpoint API

1. Endpoint Produk

Method	Endpoint	Deskripsi
GET	/api/produk	Mendapatkan semua data produk
GET	/api/produk/:id	Mendapatkan satu produk berdasarkan ID
POST	/api/produk	Menambahkan produk baru
PUT	/api/produk/:id	Mengupdate produk berdasarkan ID

Method	Endpoint	Deskripsi
DELETE	/api/produk/:id	Menghapus produk berdasarkan ID

2. Contoh Payload untuk POST dan PUT

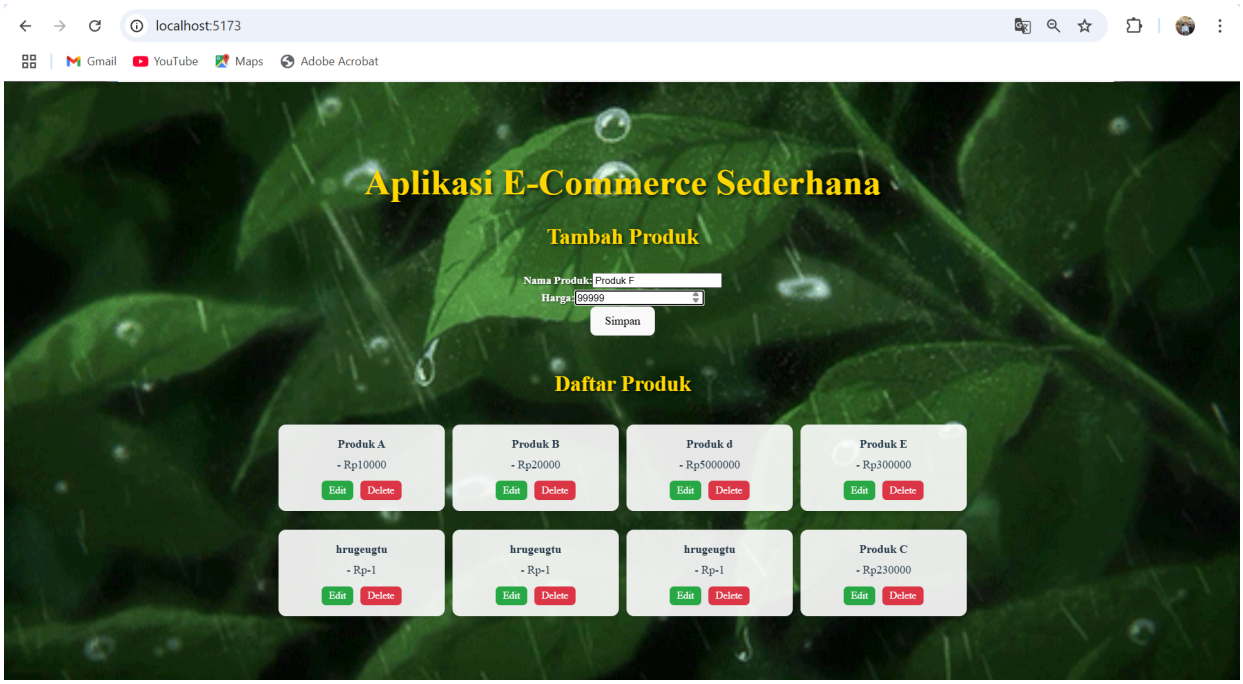
```
{
  "nama": "Produk A",
  "harga": 10000
}
```

Teknologi yang Digunakan

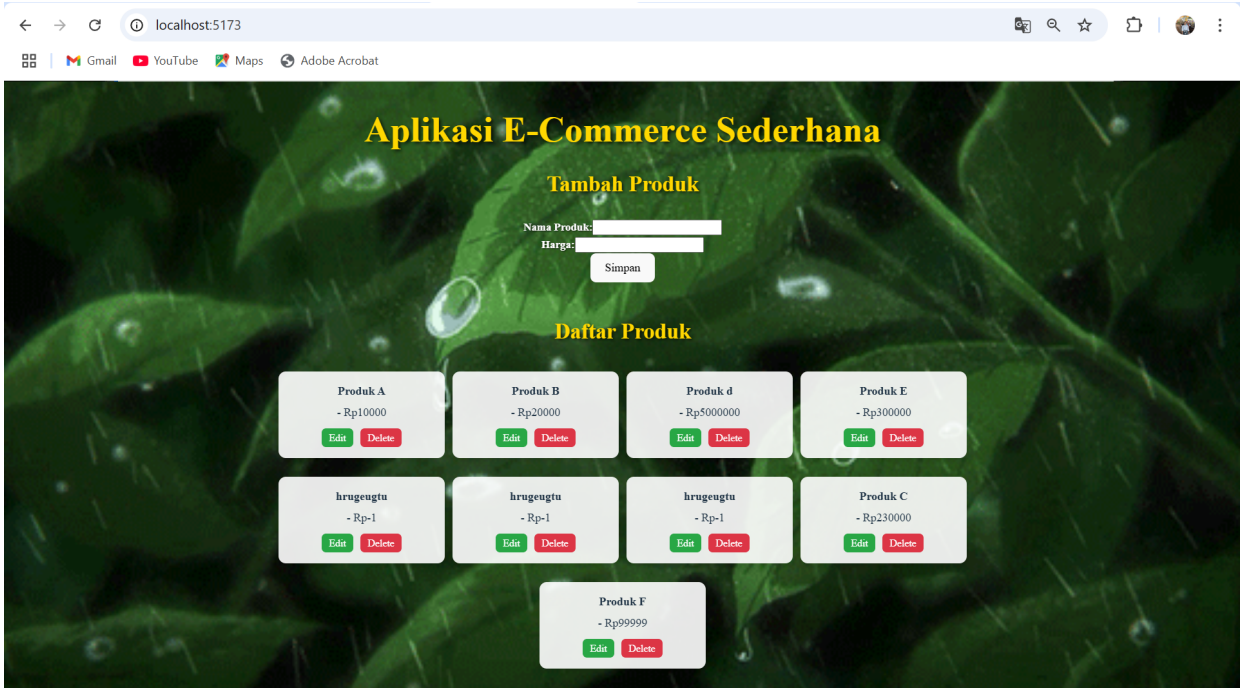
- **Frontend:** React (Vite), Tailwind CSS, Axios, React Hooks
- **Backend:** Express.js, Node.js, CORS
- **Database:** PostgreSQL, pgAdmin
- **Tools:** Postman, GitHub, Railway, Vercel
- **State Management:** useState, useEffect, Context API
- **Deployment:** GitHub Actions, Vercel, Railway

Screenshot Aplikasi

1. **Tampilan Tambah Produk dengan Form Validasi**

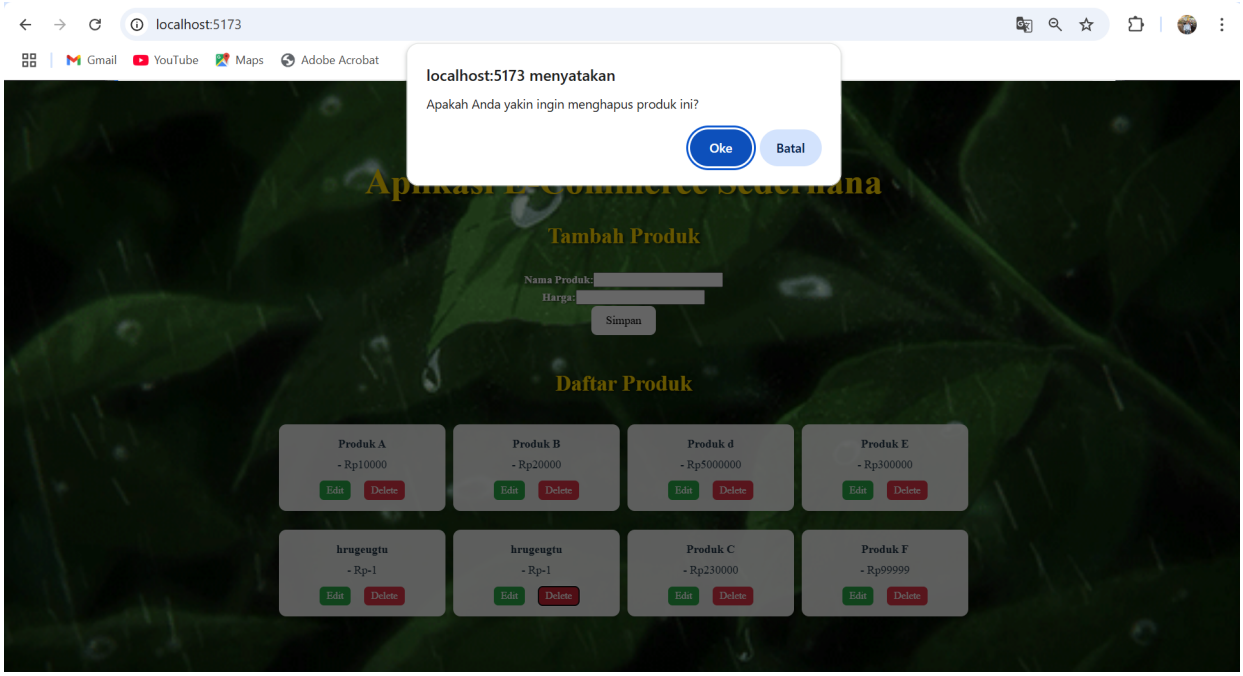


2. **Tampilan Read Data dalam Tabel dengan Pagination**

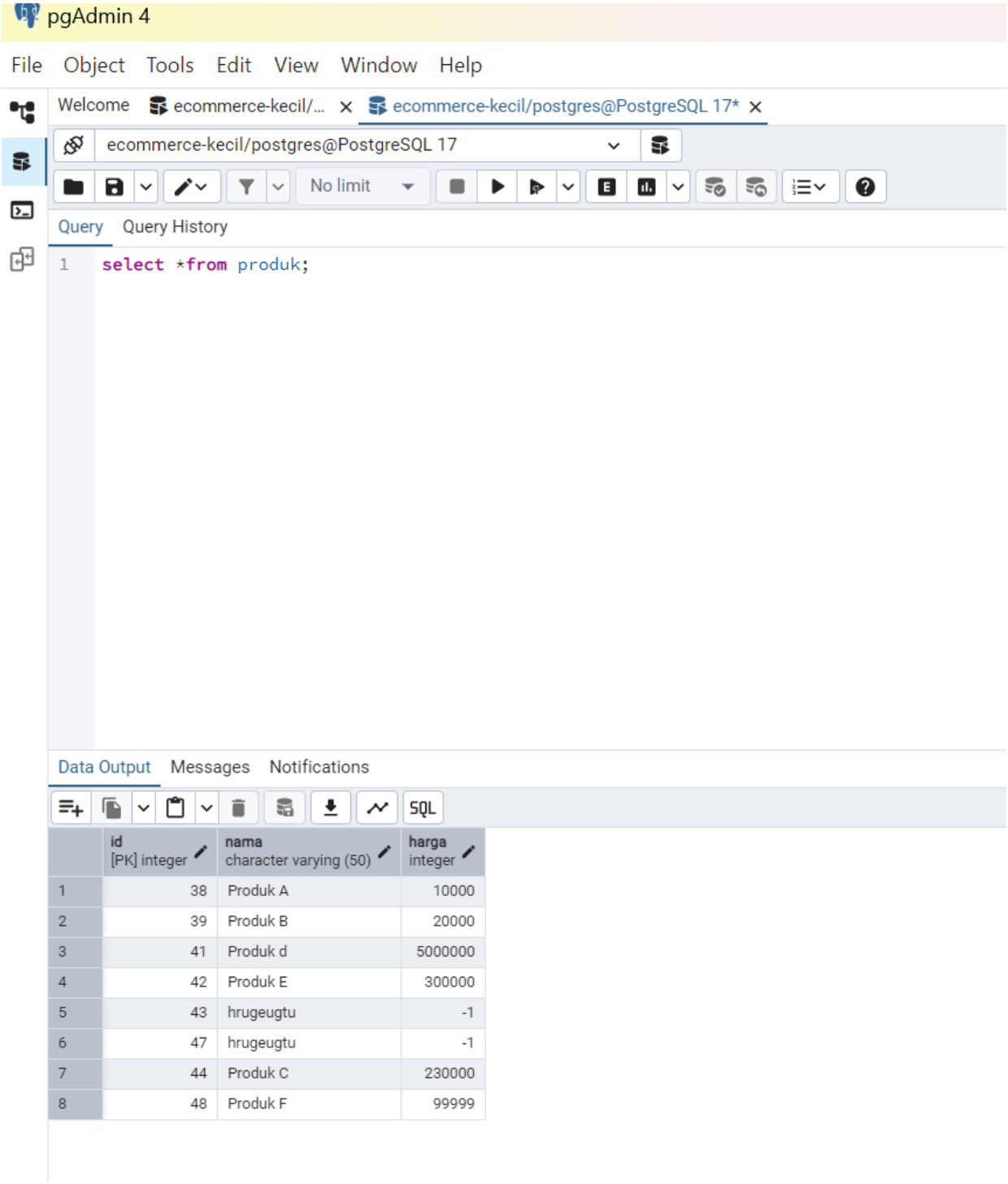


3. **Tampilan Edit Produk dengan Modal Pop-up**

4. **Tampilan Hapus Produk dengan Notifikasi Konfirmasi**

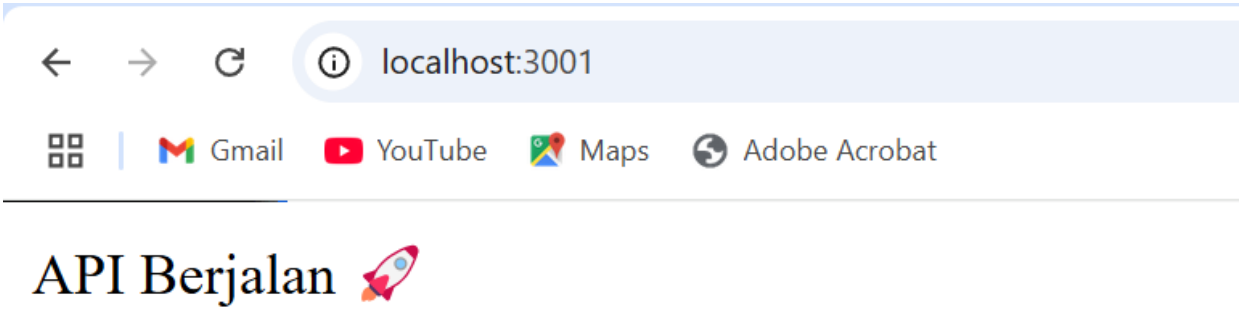


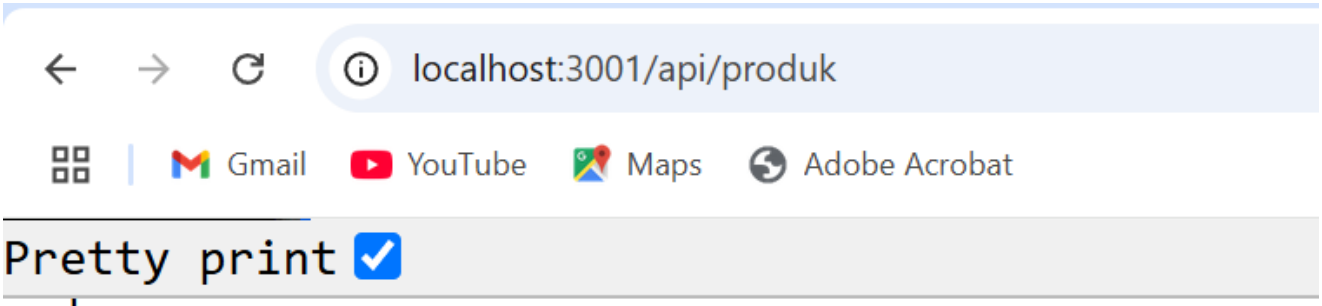
5. **Tampilan Database di pgAdmin setelah CRUD**



	id [PK] integer	nama character varying (50)	harga integer
1	38	Produk A	10000
2	39	Produk B	20000
3	41	Produk d	5000000
4	42	Produk E	300000
5	43	hrugeugtu	-1
6	47	hrugeugtu	-1
7	44	Produk C	230000
8	48	Produk F	99999

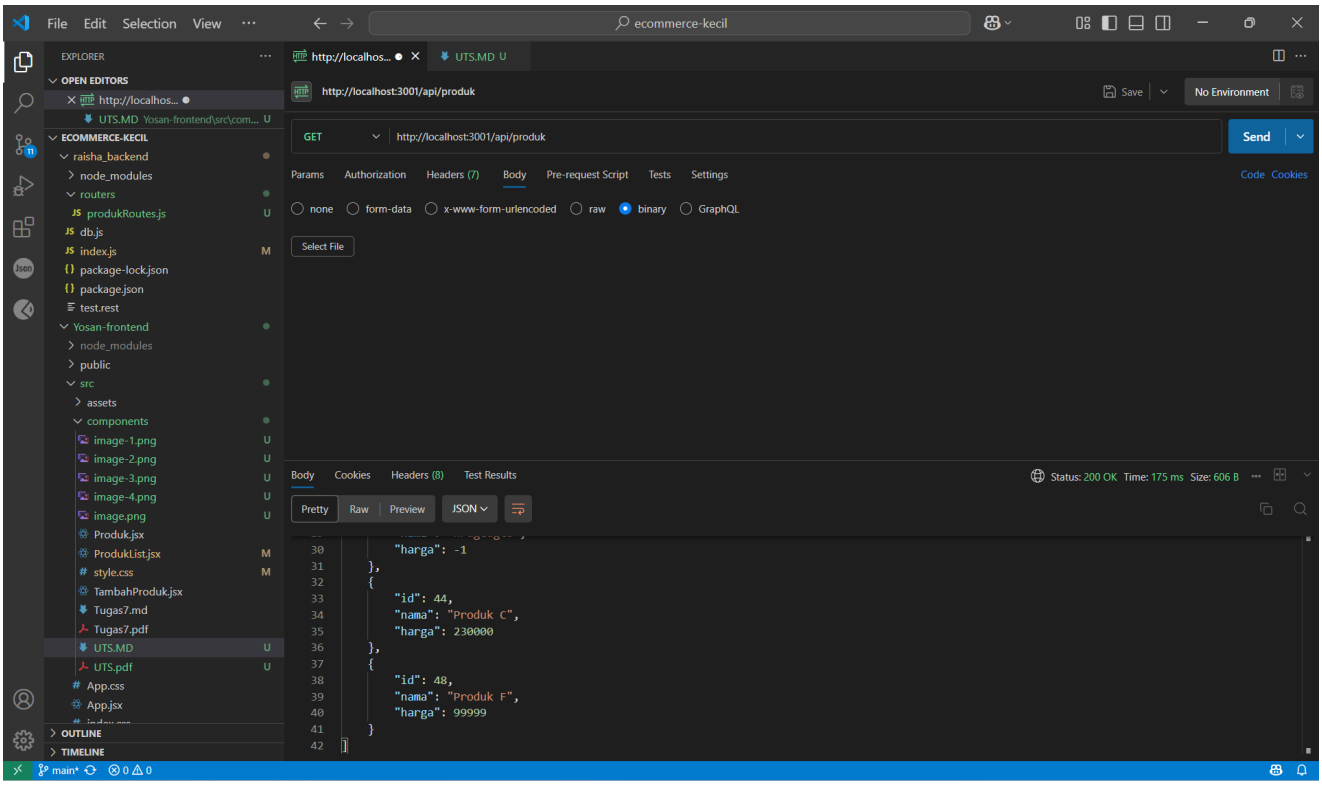
6. **Hasil API Request di Postman**





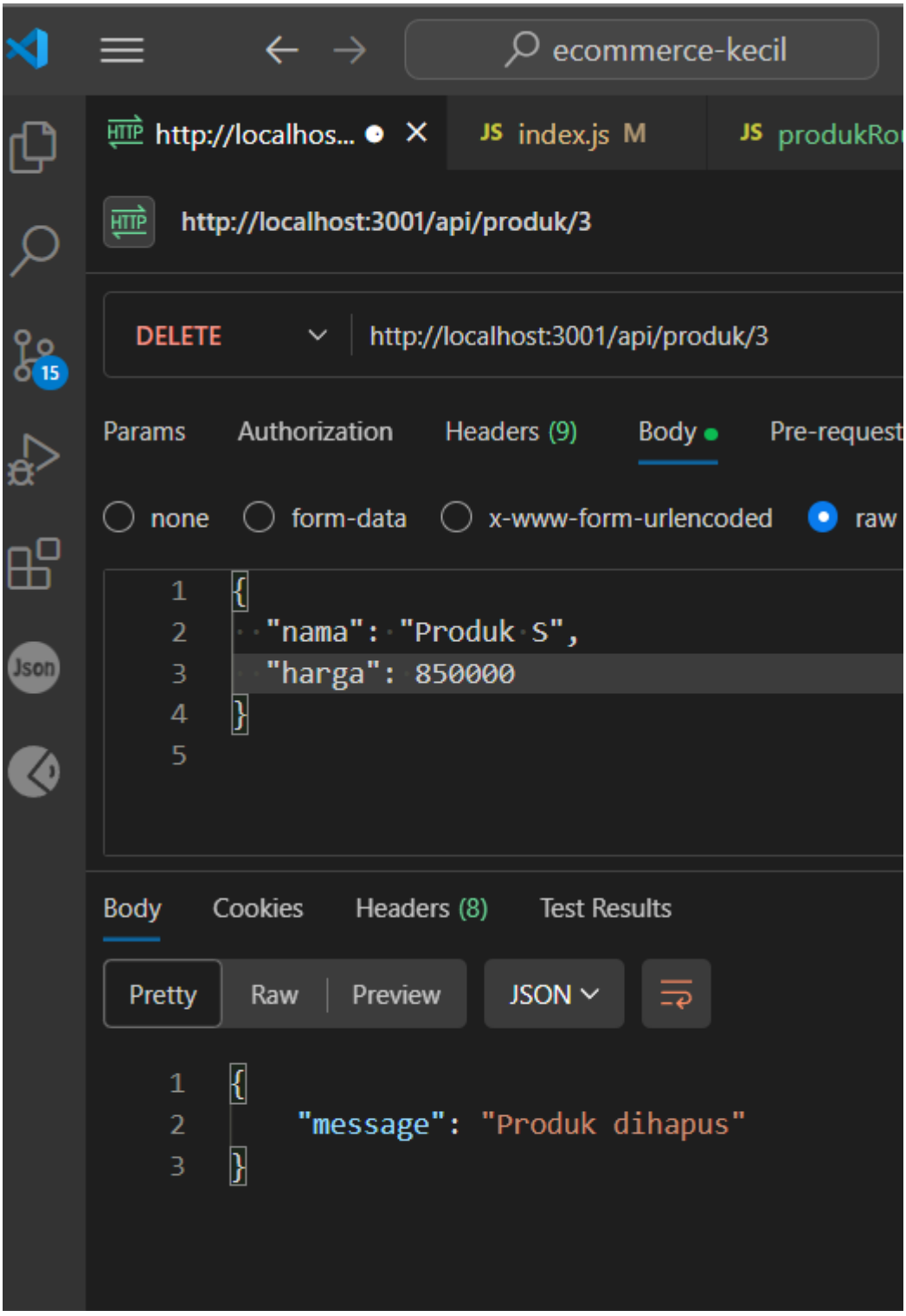

```
  },
  {
    "id": 42,
    "nama": "Produk E",
    "harga": 300000
  },
  {
    "id": 43,
    "nama": "hrugeugtu",
    "harga": -1
  },
  {
    "id": 47,
    "nama": "hrugeugtu",
    "harga": -1
  },
  {
    "id": 44,
    "nama": "Produk C",
    "harga": 230000
  },
  {
    "id": 48,
    "nama": "Produk F",
    "harga": 99999
  },
```


- GET


File Edit Selection View
ecommerce-kecil
EXPLORER
OPEN EDITORS
http://localhos...
UTS.MD Yosan-frontend\src\com... U
ECOMMERCE-KECIL
raisha_backend
node_modules
routers
produkRoutes.js
db.js
index.js
package-lock.json
package.json
test.rest
Yosan-frontend
node_modules
public
src
assets
components
image-1.png
image-2.png
image-3.png
image-4.png
image.png
Produk.jsx
ProdukList.jsx
style.css
TambahProduk.jsx
Tugas7.md
Tugas7.pdf
UTS.MD
UTS.pdf
App.css
App.jsx
OUTLINE
TIMELINE
http://localhos...
UTS.MD U
http://localhost:3001/api/produk
Save
No Environment
GET
http://localhost:3001/api/produk
Send
Params Authorization Headers (7) Body Pre-request Script Tests Settings
Code Cookies
none form-data x-www-form-urlencoded raw binary GraphQL
Select File
Body Cookies Headers (8) Test Results
Status: 200 OK Time: 175 ms Size: 606 B
Pretty Raw Preview JSON
"harga": -1
},
{
"id": 44,
"nama": "Produk C",
"harga": 230000
},
{
"id": 48,
"nama": "Produk F",
"harga": 99999
}
]
main*

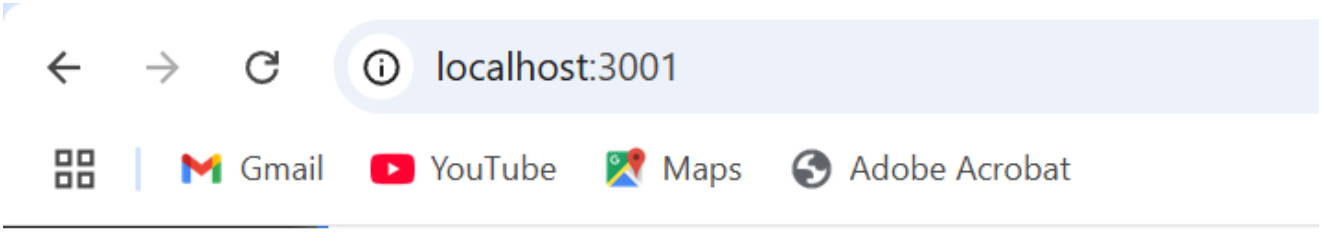
- POST
- PUT

- DELETE



📌 Troubleshooting

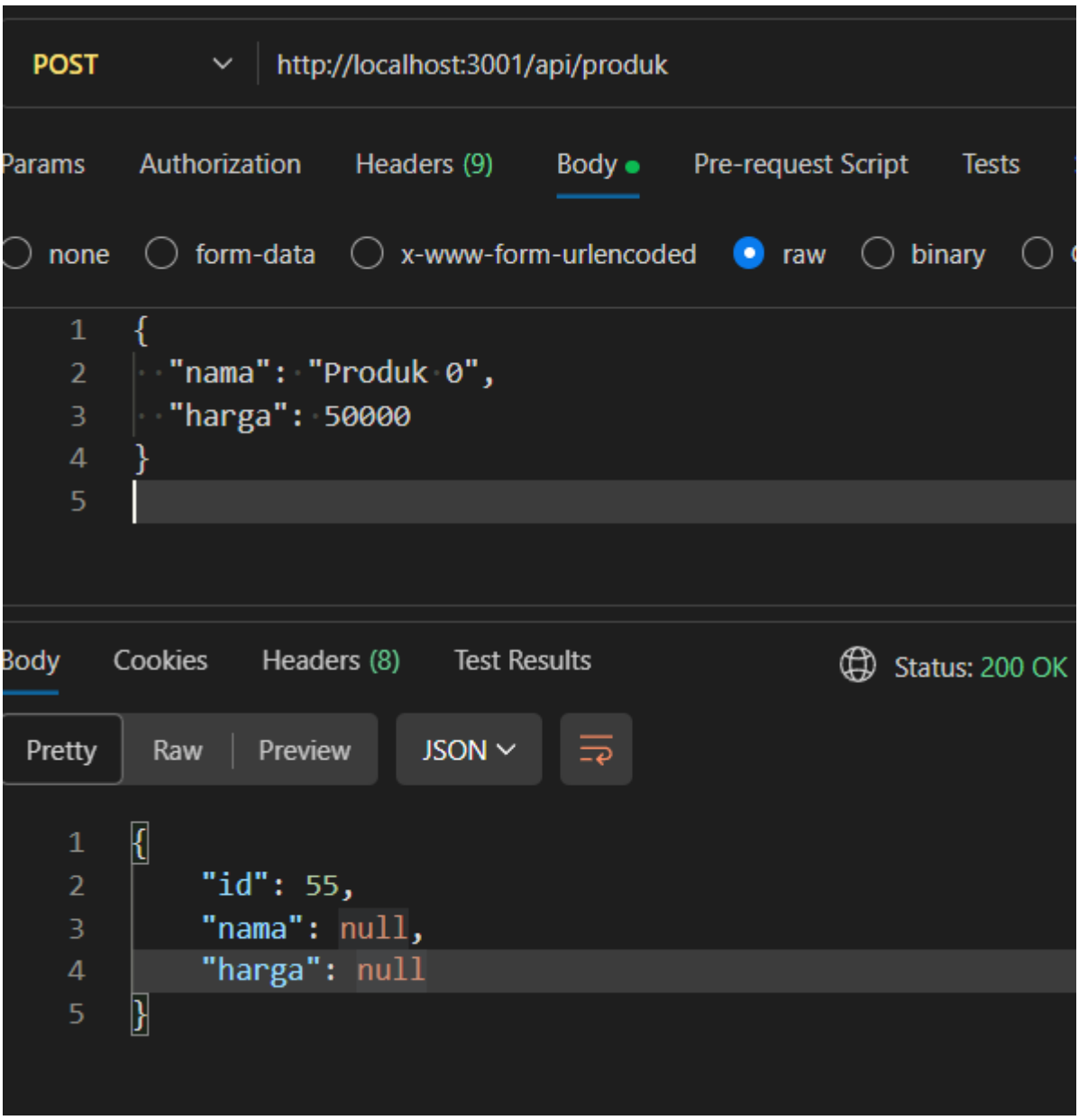
1. Node.js Tidak Berjalan



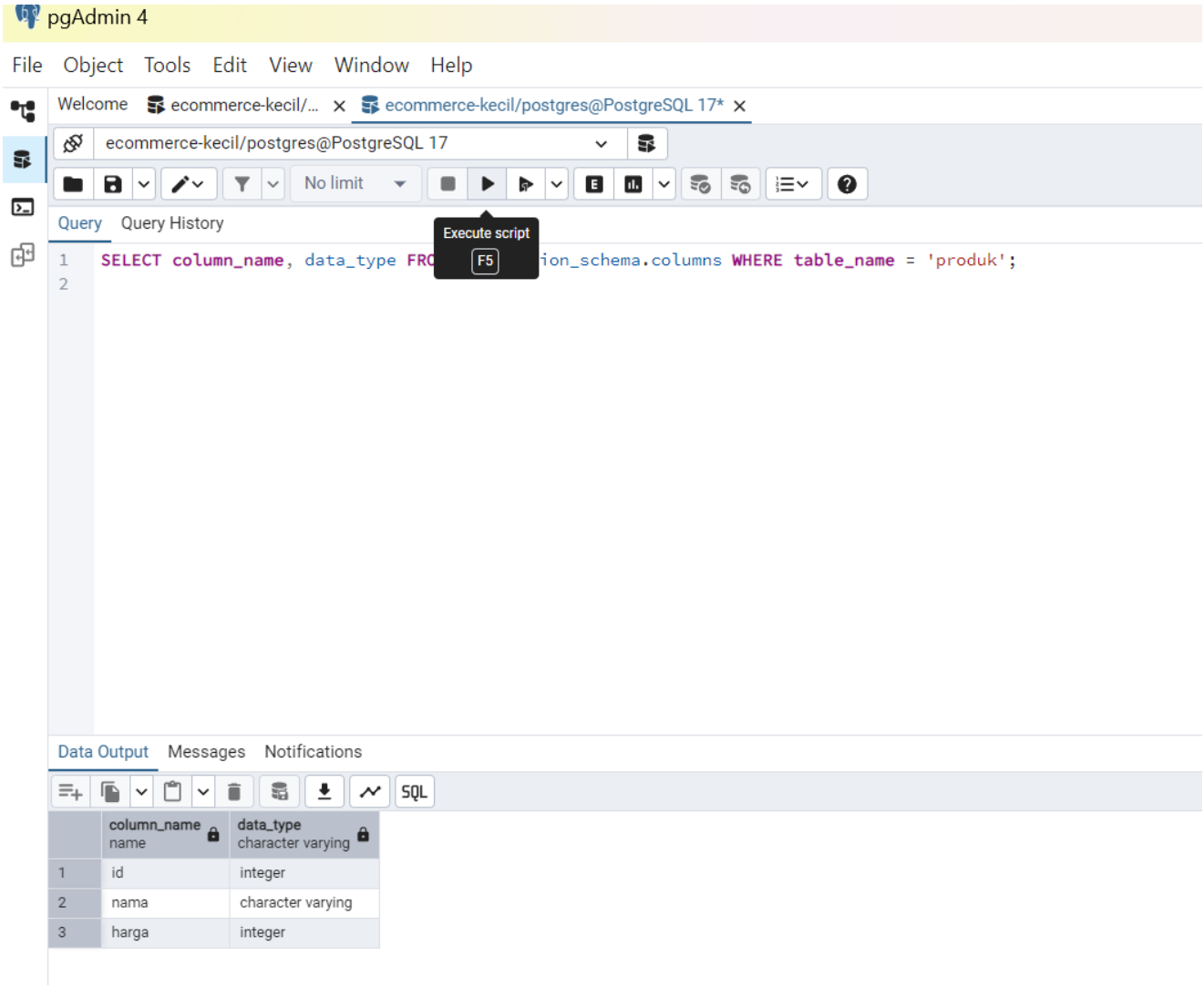

```
node -v
```

Pastikan Node.js sudah terinstall dengan benar.

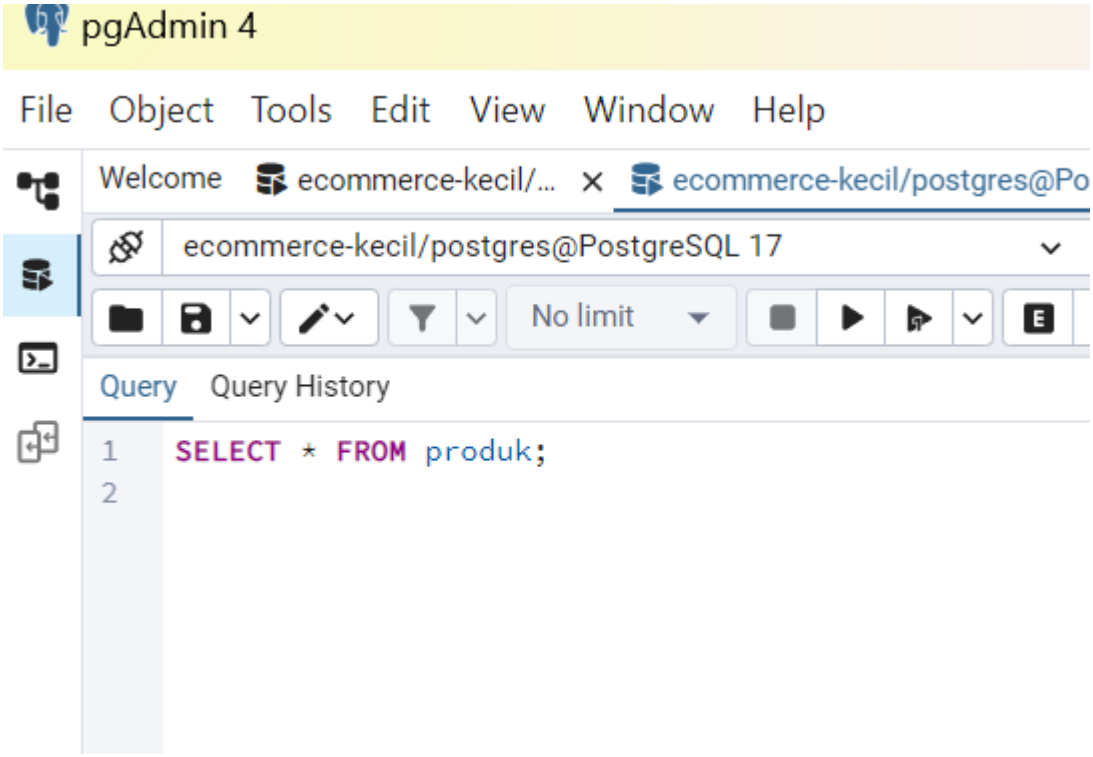
1. Node.js Tidak Berjalan

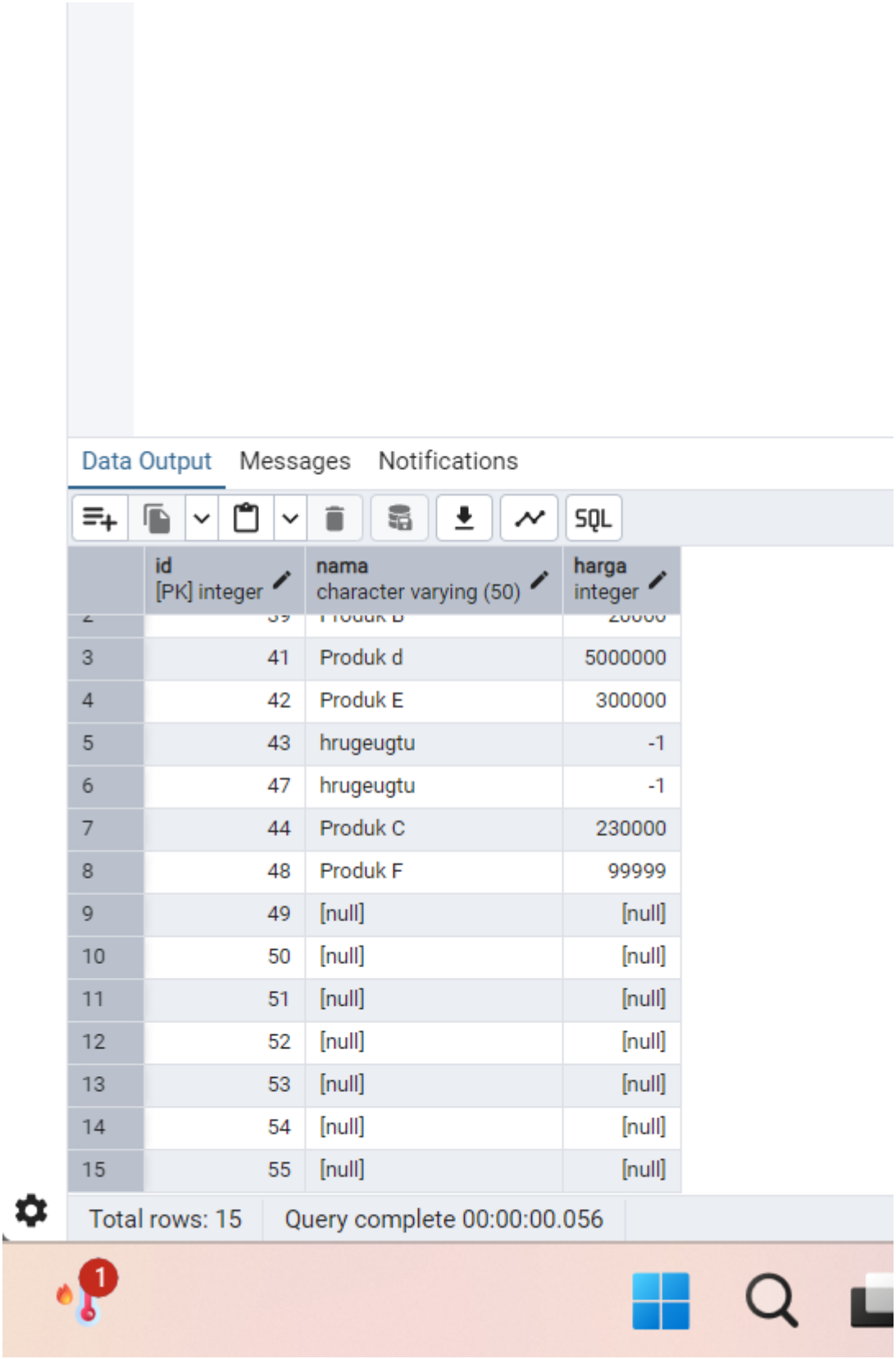


1. Node.js Tidak Berjalan

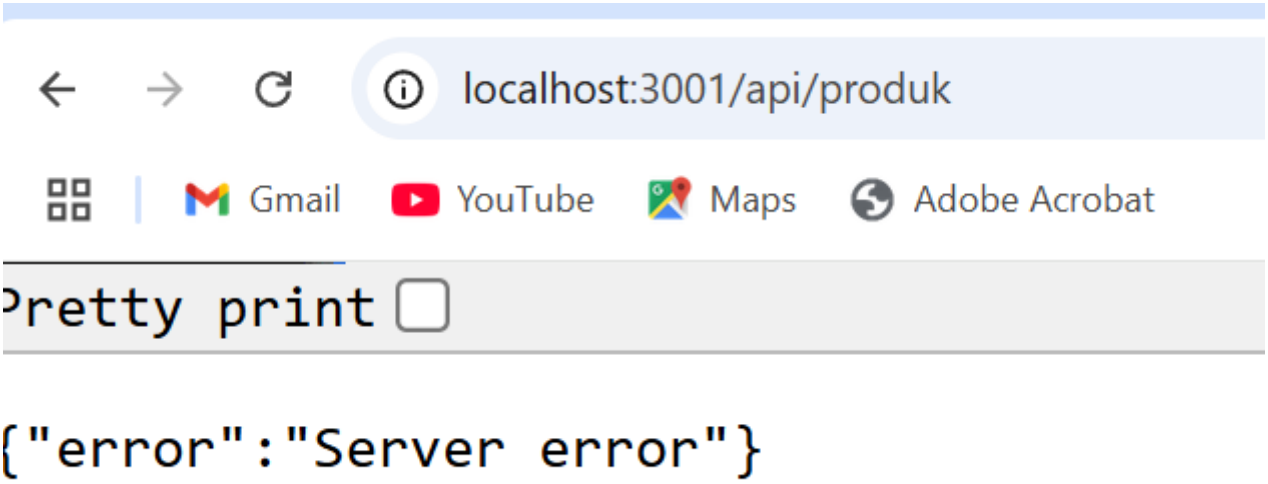


1. Node.js Tidak Berjalan





1. Node.js Tidak Berjalan



1. Node.js Tidak Berjalan

2. Database PostgreSQL Tidak Terhubung

- Pastikan PostgreSQL sudah berjalan dan user `postgres` memiliki akses.
- Cek konfigurasi di `db.js`.
- Jalankan perintah berikut di PostgreSQL untuk memastikan database sudah ada:

```
SELECT * FROM produk;
```

3. Masalah Fetch Data di Frontend

- Pastikan backend berjalan dengan benar di `http://localhost:3001`.
- Gunakan `npm install axios` jika Axios belum terinstall.
- Pastikan CORS sudah diaktifkan di backend.

📌 Best Practices dalam Pengembangan

- **Gunakan Environment Variables:** Simpan kredensial database di `.env`.
- **Gunakan GitHub Issues & Projects:** Kelola tugas dengan jelas.
- **Gunakan Middleware di Express:** Untuk menangani error dan autentikasi.
- **Optimasi Query di PostgreSQL:** Gunakan indeks jika perlu untuk performa tinggi.
- **Pastikan UI Responsif:** Gunakan media queries dengan Tailwind CSS.

📌 Commit dan Push ke GitHub

```
git add .
git commit -m "Menambahkan dokumentasi dan fitur baru"
git push origin main
```

Inspirasi Hari Ini

> "Mengetahui saja tidak cukup; kita harus menerapkannya. Berkeinginan saja tidak cukup; kita harus melakukannya." – Goethe