# Laporan Praktikum WSE #6

| | |
|---|---|
| Mata Kuliah | : Web Service Engineering |
| Dosen Pengampu | : Muhayat, M.IT |
| Praktikum | : P6 – **RESTful API Best Practice (Express)** |
| Nama Mahasiswa | : SITI MAGFIRATUN WARAHMAH |
| NIM | : 230104040059 |
| Kelas | : TI23B |
| Tanggal Praktikum | : 03-11-2025 |

## A. Tujuan Praktikum

1. Memahami penerapan prinsip RESTful pada API Express.
2. Menggunakan HTTP Method dan Status Code secara tepat.
3. Mengimplementasikan 7 RESTful Principles dalam desain endpoint.
4. Menangani validasi input dan error secara terstandar.
5. Menyiapkan dokumentasi endpoint yang mudah dibaca dan diuji..

## B. Lingkungan & Tools

- ✓ Node.js 18+ & npm
- ✓ Express.js
- ✓ VS Code / Postman / Thunder Client
- ✓ Nodemon (dev dependency)
- ✓ morgan → logging request
- ✓ Middleware: `validateProduct.js`, `errorHandler.js`

## C. Arsitektur Singkat

- ✓ Client (Postman / Thunder Client / Frontend nanti) → mengirim HTTP request untuk CRUD + uji validasi + uji error.
- ✓ API Server (Express) → jadi pusat kontrol, menerima request, mem-parsing JSON, meneruskan ke router sesuai path /api/....
- ✓ Router (src/routes/products.routes.js) → mendefinisikan endpoint RESTful lengkap: GET, POST, PUT, PATCH, DELETE, plus /api/health.
- ✓ Controller / Handler → di dalam route, berisi logika ambil data, update, hapus, lalu membentuk response dalam format standar ({ success, message, data }).
- ✓ Middleware Validasi (src/middlewares/validateProduct.js) → dipasang khusus di POST dan PUT untuk menolak request yang tidak punya name atau price → balas 400 Bad Request.
- ✓ Middleware Error Global (src/middlewares/errorHandler.js) → menangkap error tak terduga (typo variabel, throw manual) dan mengembalikan 500 Server error tanpa menjatuhkan server.
- ✓ Data Layer (src/data/products.data.js) → masih pakai array in-memory sebagai sumber data sementara, tapi sudah dipisah agar nanti gampang diganti DB.
- ✓ Utility Response (src/utils/apiResponse.js) → (opsional) menyamakan bentuk response sukses dan error supaya API konsisten.
- ✓ Logging (morgan) → mencatat setiap request (method, path, status) untuk kebutuhan observability di praktikum berikutnya.

✓ Response JSON → semua endpoint mengembalikan JSON terstandar + status code yang sesuai (200, 201, 400, 404, 500).

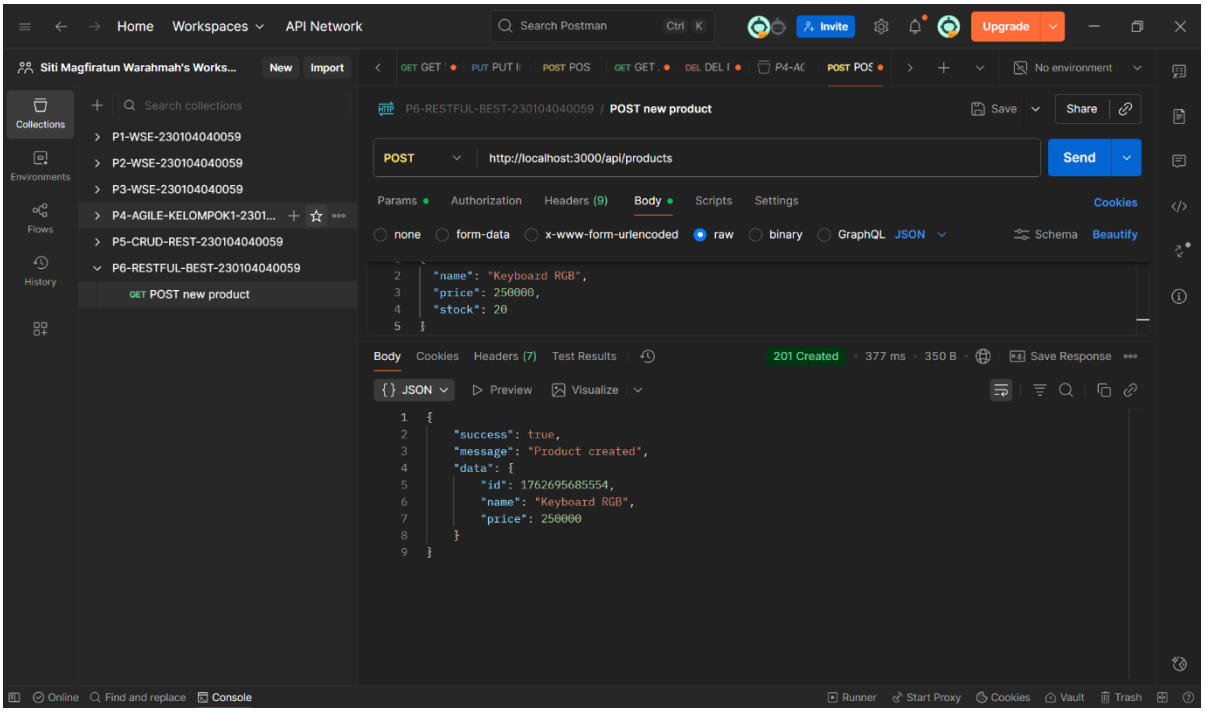
## D. Langkah Implementasi (ringkas)

1. Membuat struktur project Express di folder src/
2. Membuat file products.routes.js dengan endpoint CRUD + PATCH
3. Menambahkan middleware validateProduct untuk validasi POST dan PUT
4. Menambahkan middleware errorHandler untuk menangani error global
5. Menguji endpoint di Postman: GET, POST, PUT, PATCH, DELETE
6. Menguji validasi input dan simulasi error 500
7. Menulis dokumentasi endpoint di README.md

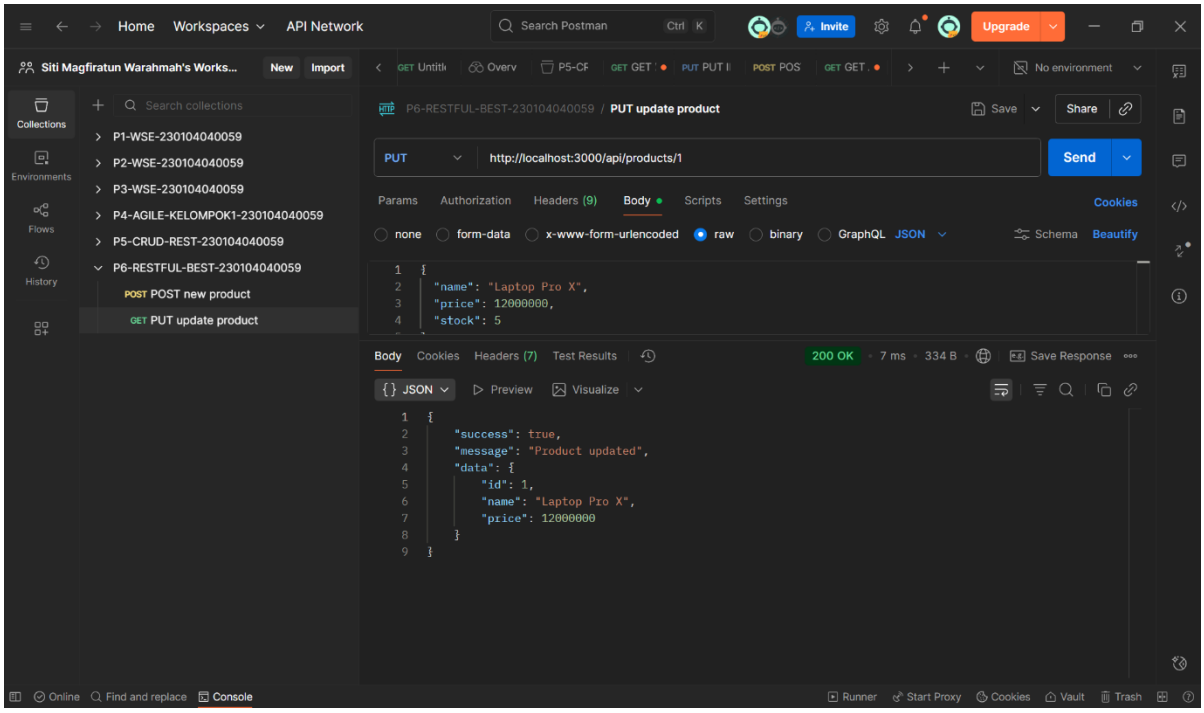
## E. Hasil & Bukti

Lampirkan **Screenshot Hasil Uji** Endpoint di Postman

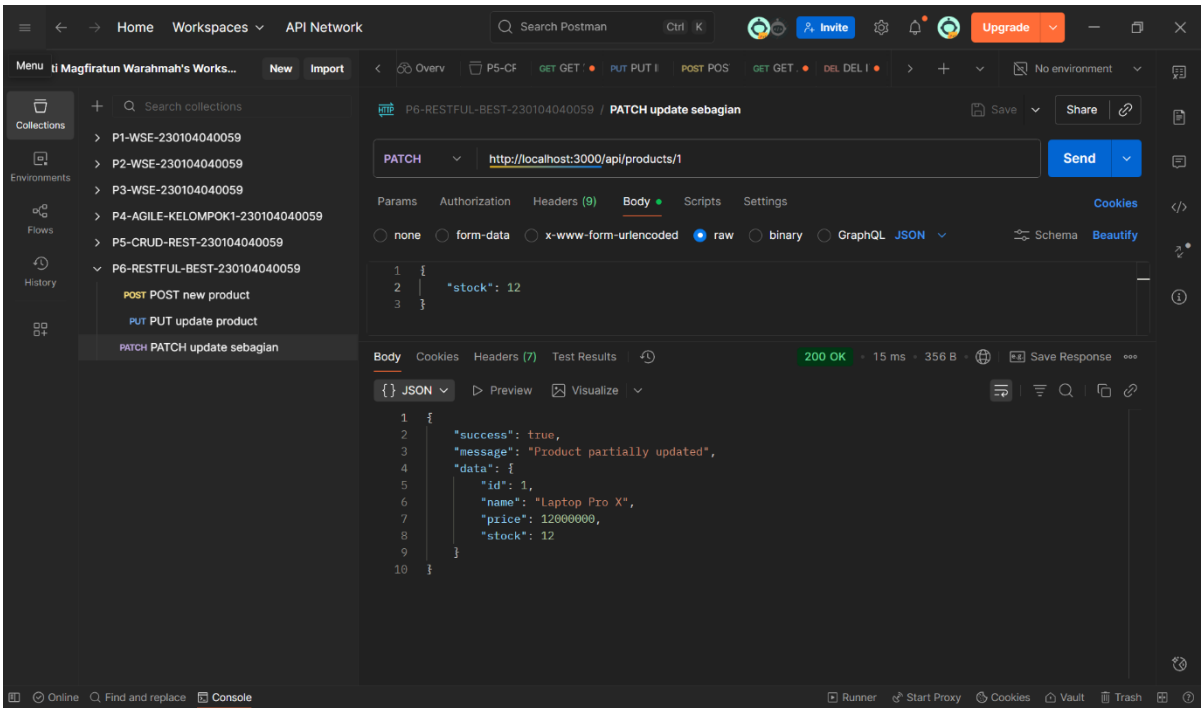
☑ POST /api/products (Berhasil tambah produk) → 201 Created



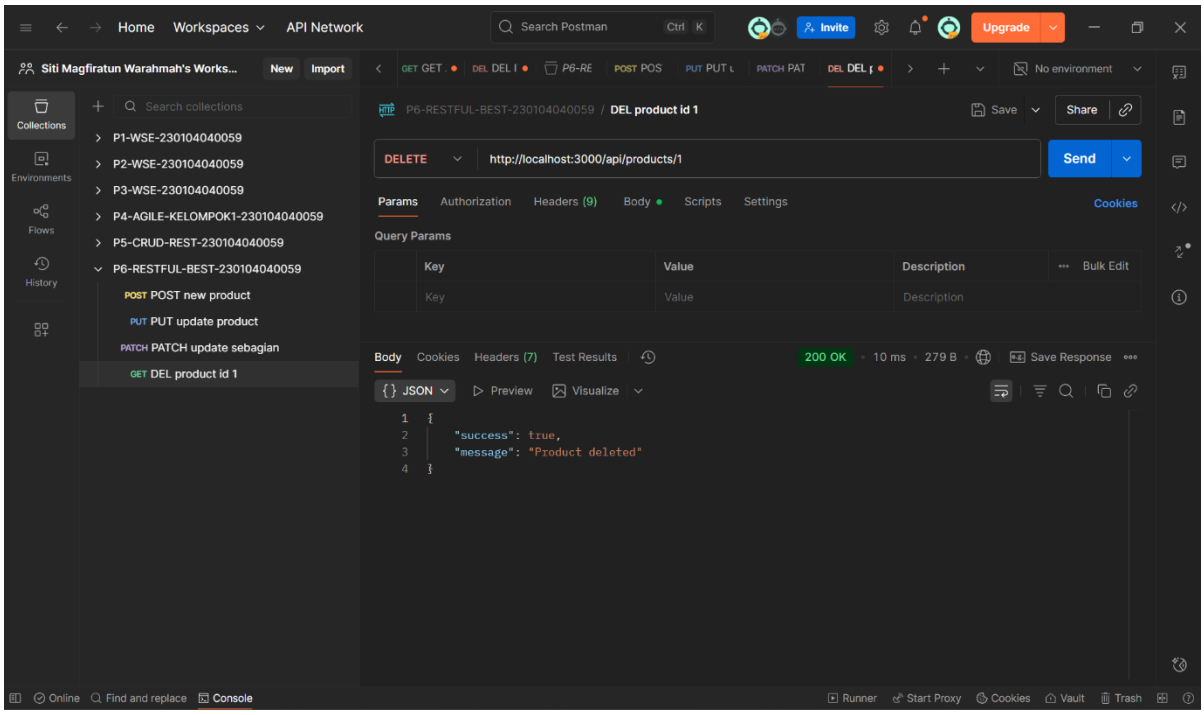
☑ PUT /api/products/1 (Berhasil update produk) → 200 OK



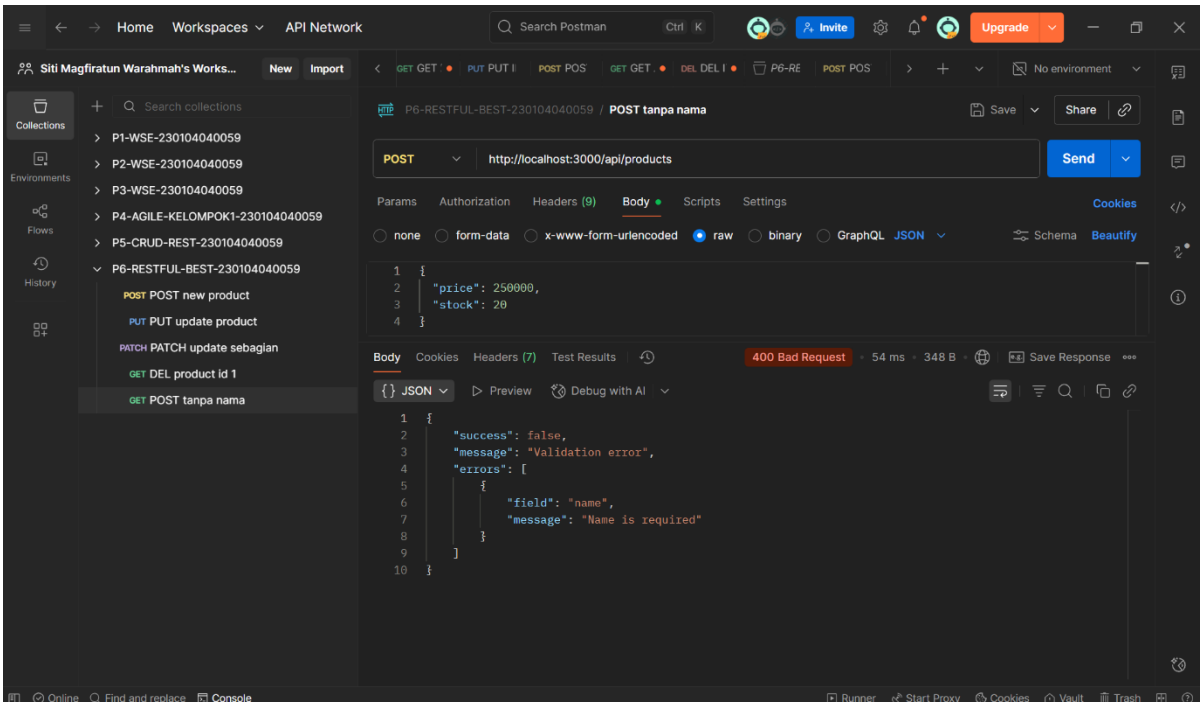

☑ PATCH /api/products/1 (Ubah sebagian data) → 200 OK



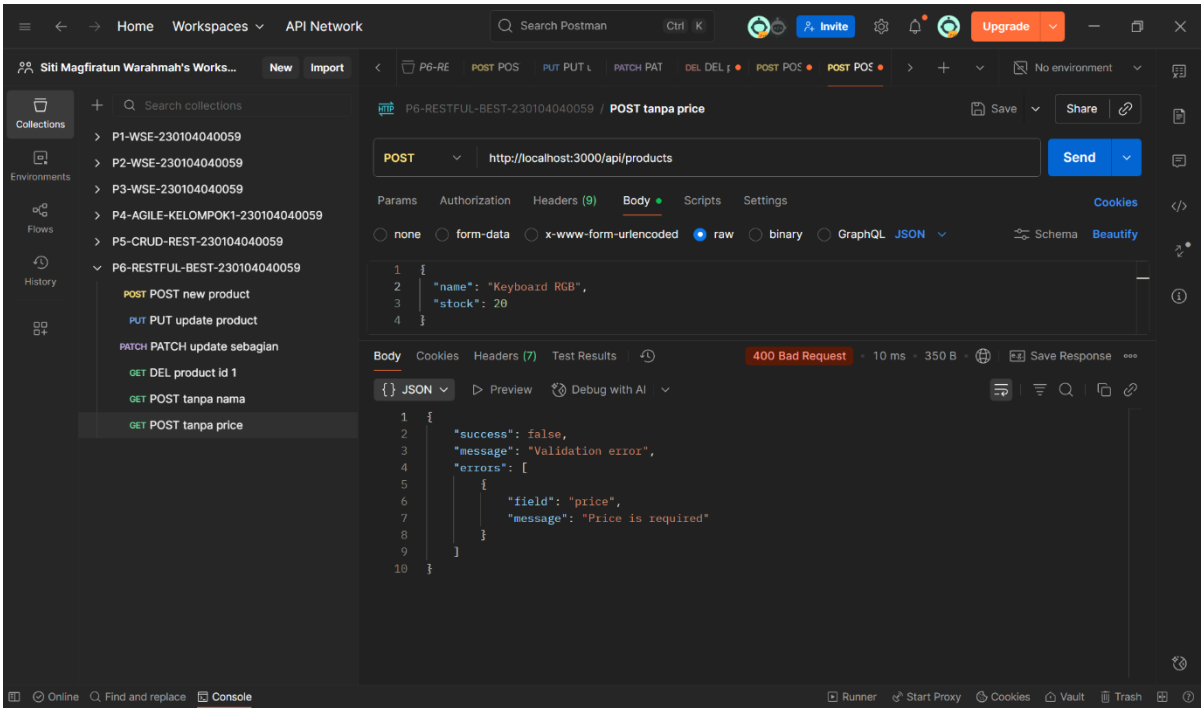
☑ DELETE /api/products/1 (Produk terhapus) → 200 OK



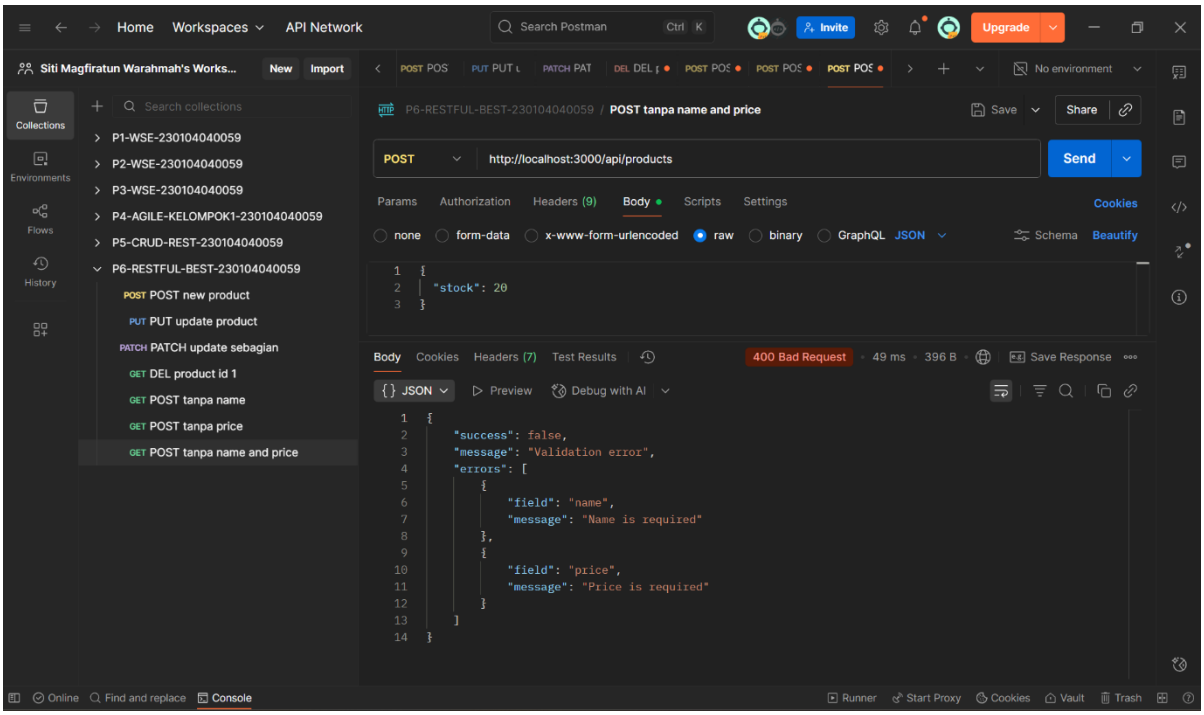

⚠ POST tanpa name → 400 Bad Request (Validation error)



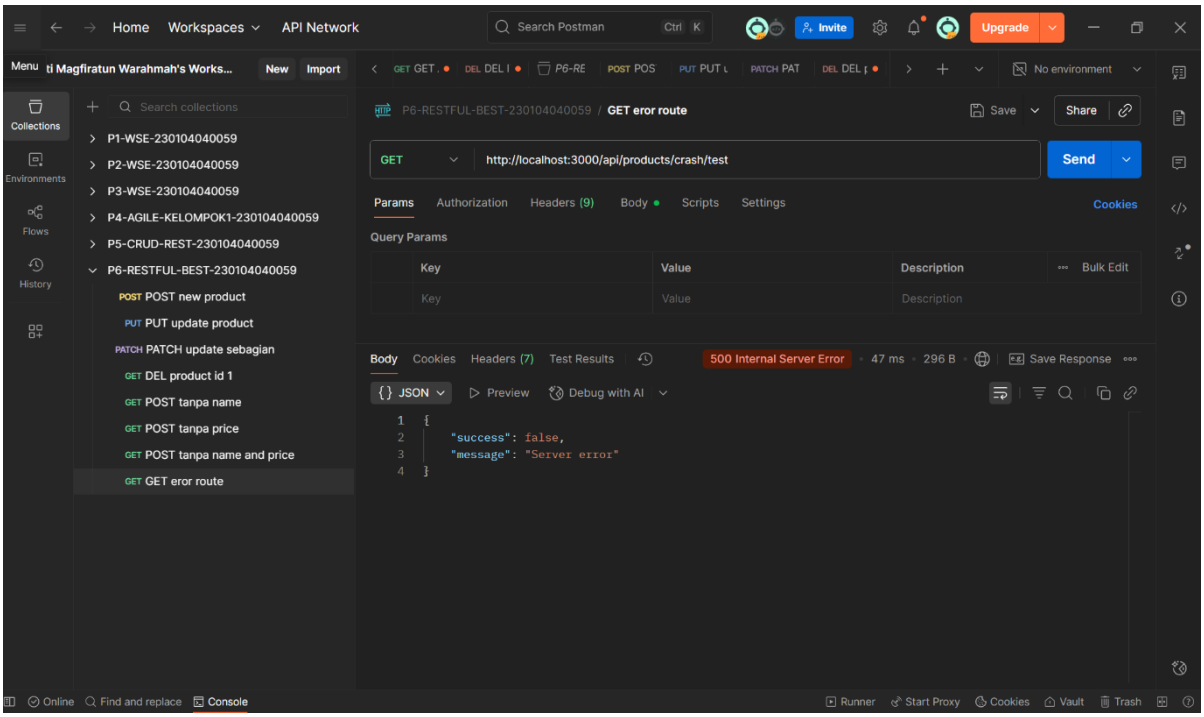
⚠ POST tanpa price → 400 Bad Request (Validation error)



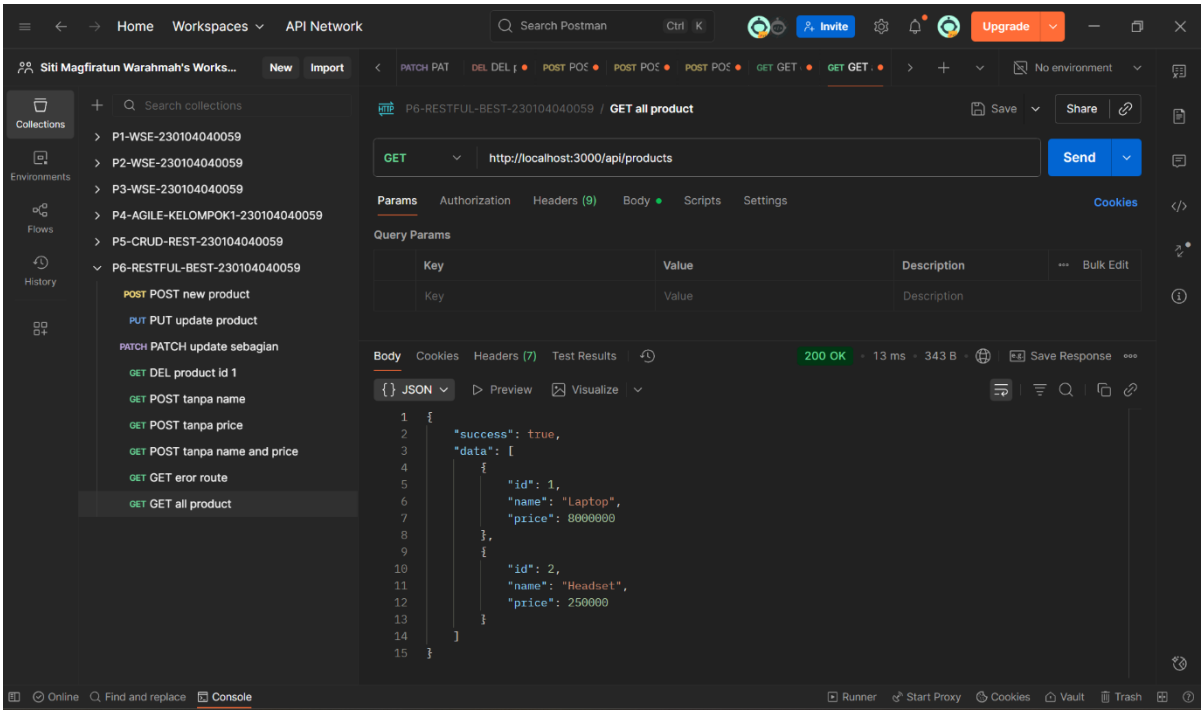

⚠ POST tanpa name/price → 400 Bad Request (Validation error)



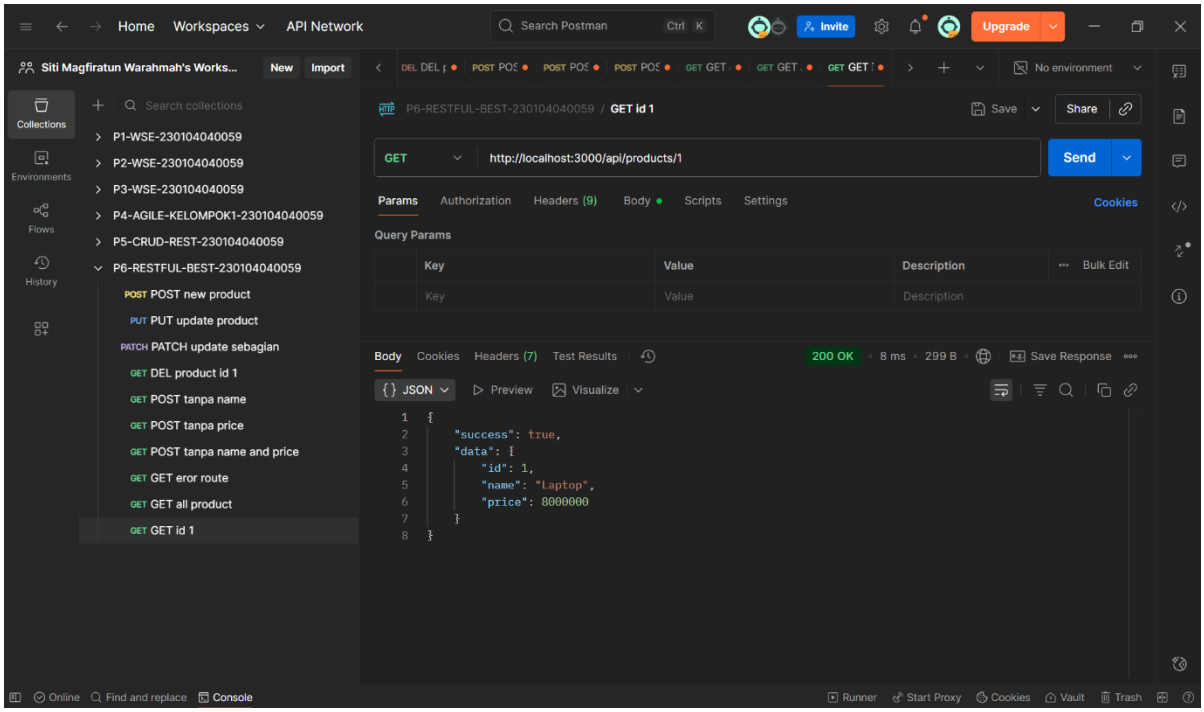
✴️ Simulasi typo variabel → 500 Internal Server Error (Error handler aktif)




☑️ GET /api/products → Menampilkan seluruh produk (200 OK)



☑ GET /api/products/1 → Menampilkan produk dengan ID tertentu (200 OK / 404 Not Found)




## F. Analisis

API telah memenuhi 7 RESTful Principles, memiliki struktur modular, dan menampilkan status code yang konsisten. Middleware validasi berhasil mencegah input kosong, dan error handler menampilkan pesan 500 tanpa mematikan server.

## G. Kesimpulan

Melalui praktikum ini, dapat dipahami bahwa RESTful API tidak hanya tentang CRUD, tetapi juga tentang penerapan prinsip desain yang konsisten, aman, dan mudah digunakan oleh client.

## H. Checklist Praktikum

- ✓ Endpoint CRUD lengkap
- ✓ PATCH berfungsi
- ✓ Middleware validasi aktif
- ✓ Error handler berjalan
- ✓ Status code konsisten
- ✓ Dokumentasi README.md selesai