Laporan Praktikum WSE #10

Mata Kuliah : Web Service Engineering
Dosen Pengampu : Muhayat, M.IT
Praktikum : P10 – **Simulasi API Key & OAuth 2.0**
Nama Mahasiswa : Husna Norgina
NIM : 230104040056
Kelas : TI23B
Link Repo Github : https://github.com/husna-norgina/p10-oauth2-api-key-230104040056
Tanggal Praktikum : 08-12-2025

A. Tujuan Praktikum

1. Memahami konsep dan perbedaan antara API Key dan OAuth 2.0 (dengan fokus pada Authorization Code Grant atau Implicit Grant yang disederhanakan).
2. Mengimplementasikan Middleware dalam Express.js untuk validasi otentikasi.
3. Mengelola dan memvalidasi API Keys sederhana di sisi server.
4. Mensimulasikan proses pemberian token (Token Grant) dan akses sumber daya terproteksi (Protected Resource) menggunakan JWT (JSON Web Tokens).
5. Menggunakan MongoDB Atlas untuk menyimpan data pengguna, client aplikasi, dan API Key

B. Aktivitas Praktikum Tools

Kategori	Teknologi/Konsep	Keterangan
Backend	Node.js & Express.js	Server API utama.
Database	MongoDB Atlas	Menyimpan data pengguna, API Key, dan klien.
Autentikasi	JSON Web Tokens (JWT)	Untuk simulasi *Access Token* pada OAuth 2.0.
Alat	Postman / Insomnia	Untuk menguji *endpoint* API.
Prasyarat	Pemahaman dasar Node.js, Express.js, dan operasi CRUD MongoDB.	

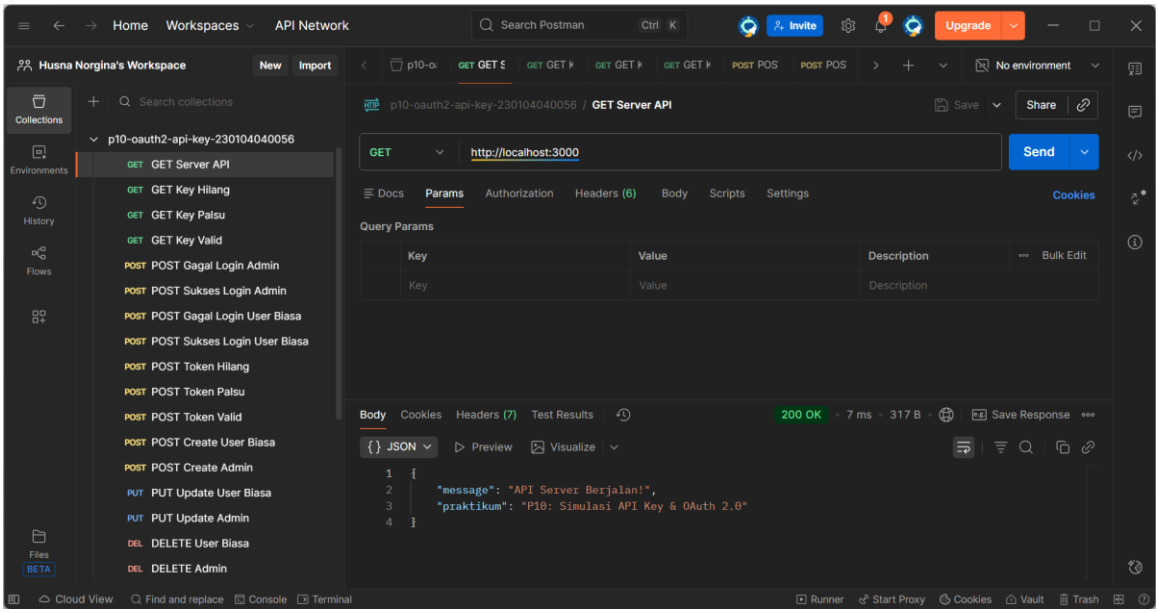
C. Struktur Project

```
p10-oauth2-api-key-230104040056/
├── controllers/
│   ├── authController.js     # [Langkah 4] Logika untuk login dan token generation (JWT)
│   └── productController.js  # [Langkah 3 & 6] Handler GET Public dan CRUD Private
├── middleware/
│   ├── validateApiKey.js     # [Langkah 3] Middleware validasi API Key
│   └── validateToken.js      # [Langkah 5] Middleware verifikasi JWT (Bearer Token)
├── models/
│   ├── ApiKey.js             # [Langkah 2] Model penyimpanan API Keys
│   ├── Product.js            # [Langkah 2] Model data produk
│   └── User.js               # [Langkah 2] Model pengguna + bcrypt pre-save hook
├── routes/
│   ├── authRoutes.js         # [Langkah 4] Route POST /auth/token
│   └── productRoutes.js      # [Langkah 3 & 6] Route GET Public + CRUD Private
├── seeders/
│   └── seed.js               # [Langkah 2] Seeder awal (User, API Key, Product)
├── utils/
│   └── generateToken.js      # [Langkah 4] Util untuk membuat JWT
├── .env                      # [Langkah 1] MONGODB_URI, PORT, JWT_SECRET
├── package.json              # Metadata project + dependencies
├── server.js                 # [Langkah 1] Entry point server + koneksi DB
└── README.md                 # Dokumentasi (opsional)
```

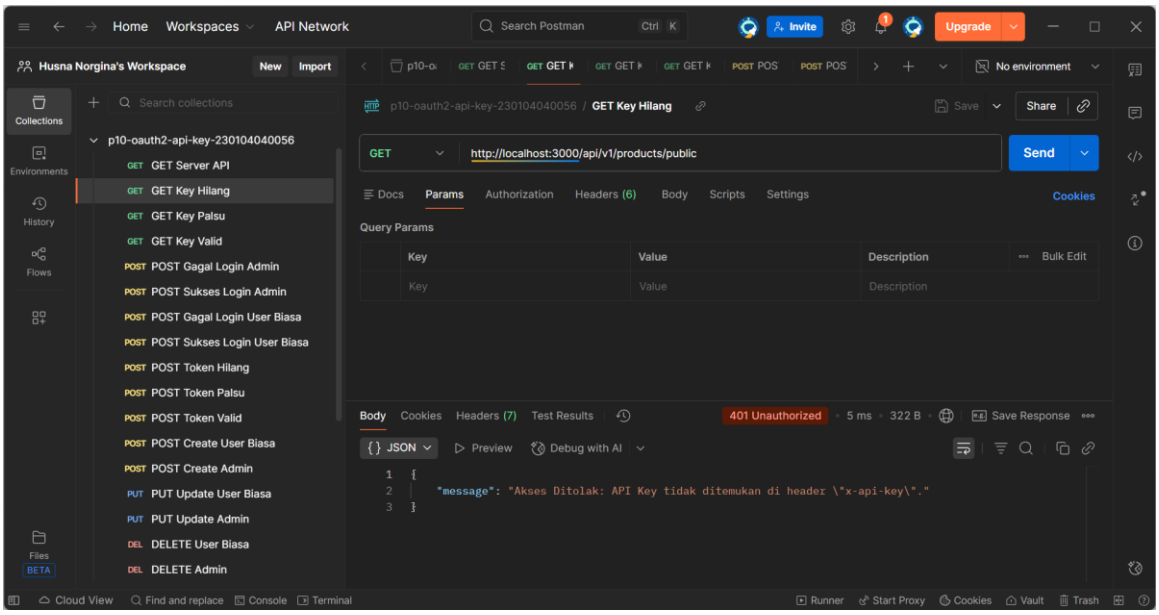
D. Hasil & Bukti

Screenshot Hasil Uji Endpoint di Postman

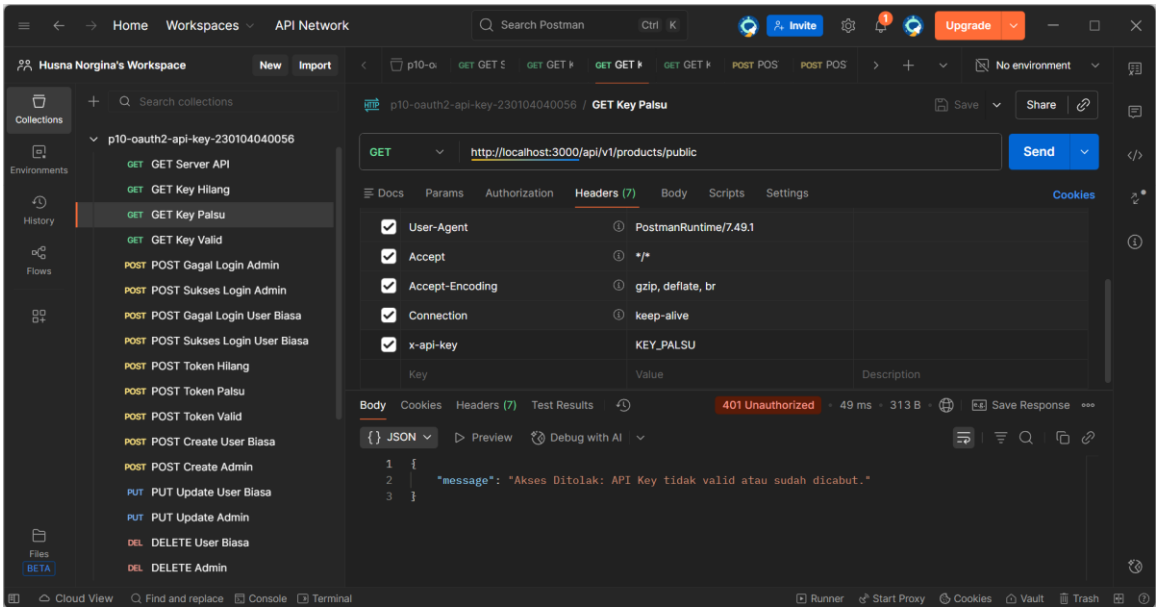
✅ GET http://localhost:3000 → Menampilkan status server API (200 OK)



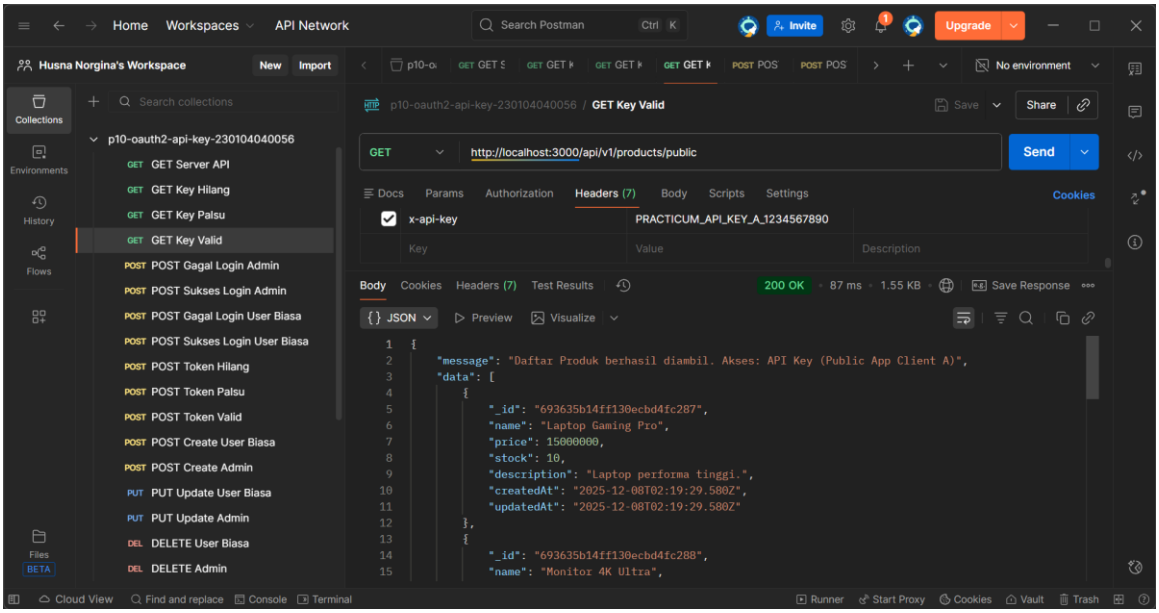
❌ GET http://localhost:3000/api/v1/products/public → Key hilang (401 Unauthorized)



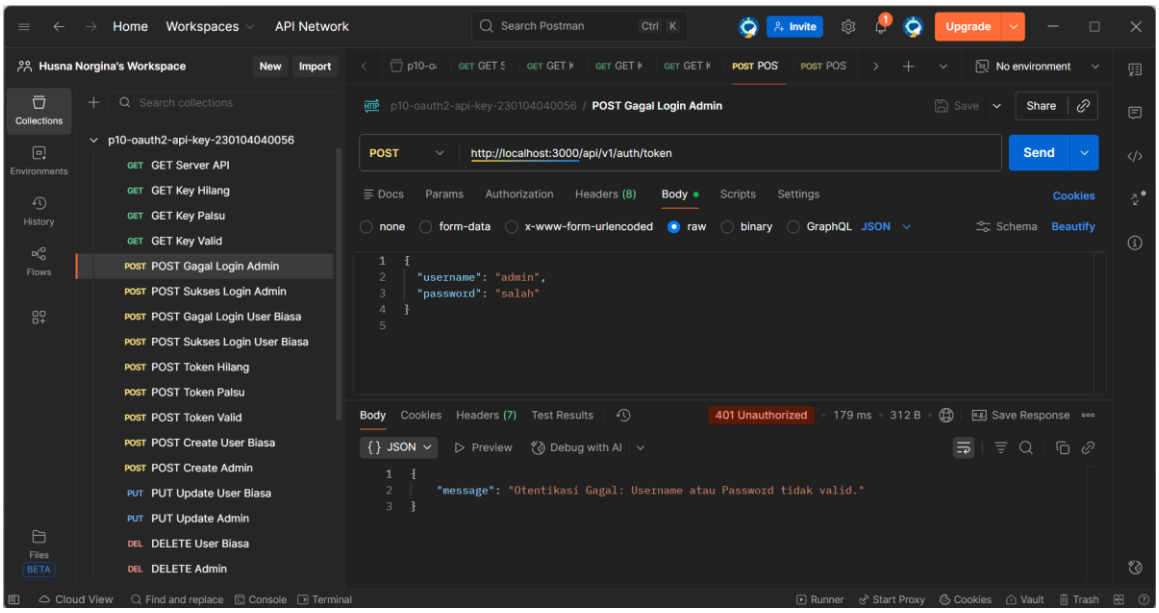
❌ GET http://localhost:3000/api/v1/products/public → Key palsu (401 Unauthorized)



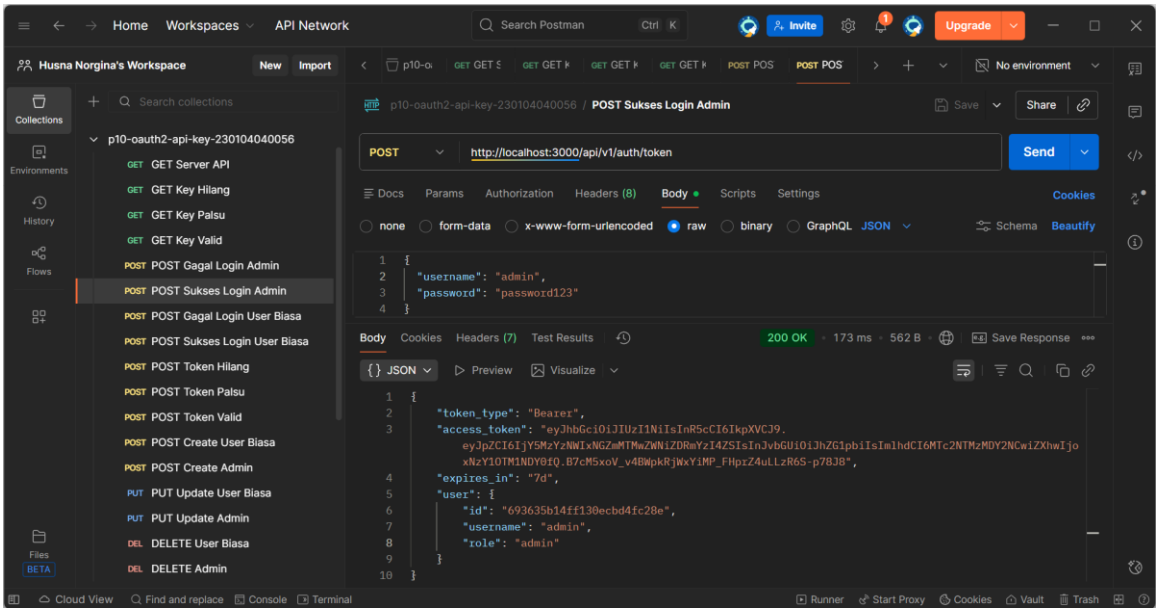
✅ GET http://localhost:3000/api/v1/products/public → Key valid (200 OK)



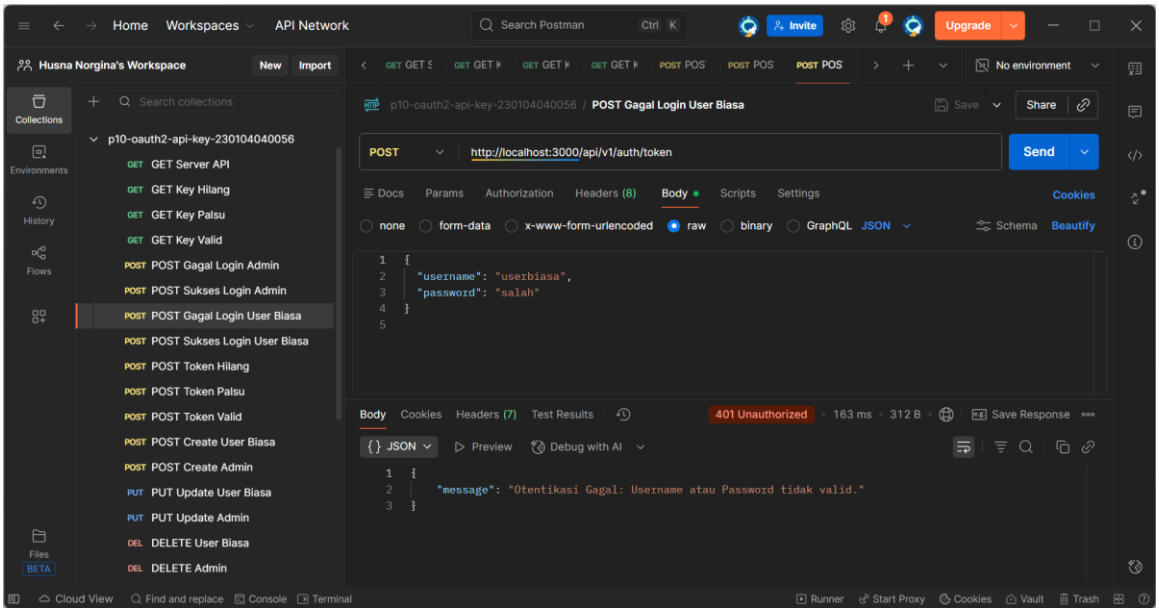
❌ POST http://localhost:3000/api/v1/auth/token → Gagal login admin (401 Unauthorized)



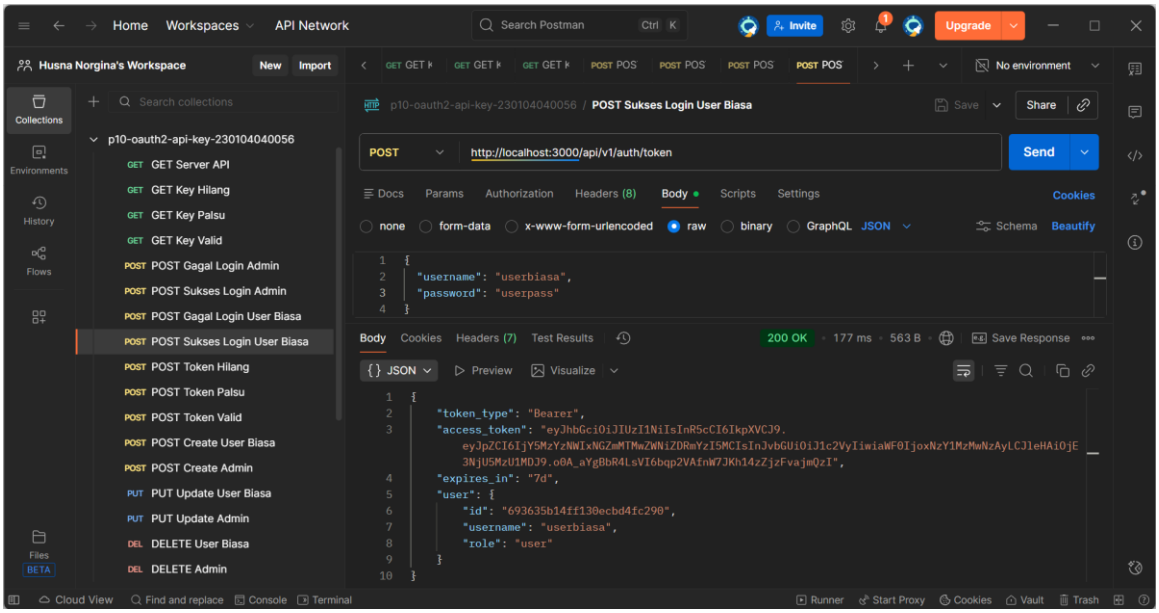
✅ POST http://localhost:3000/api/v1/auth/token → Sukses login admin (200 OK)



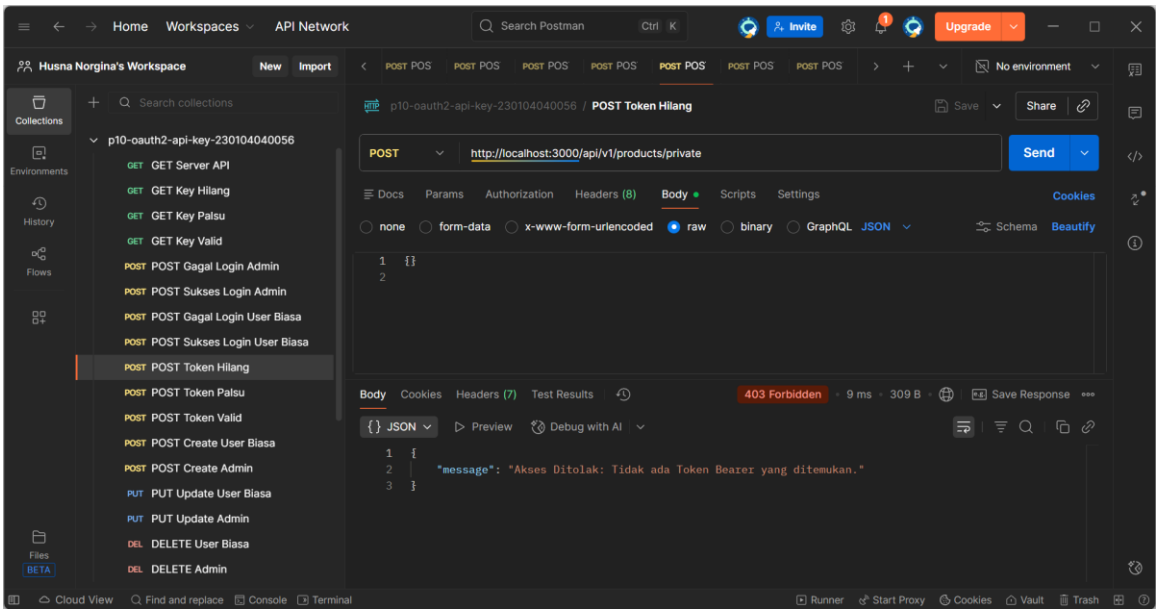
❌ POST http://localhost:3000/api/v1/auth/token → Gagal login user biasa (401 Unauthorized)



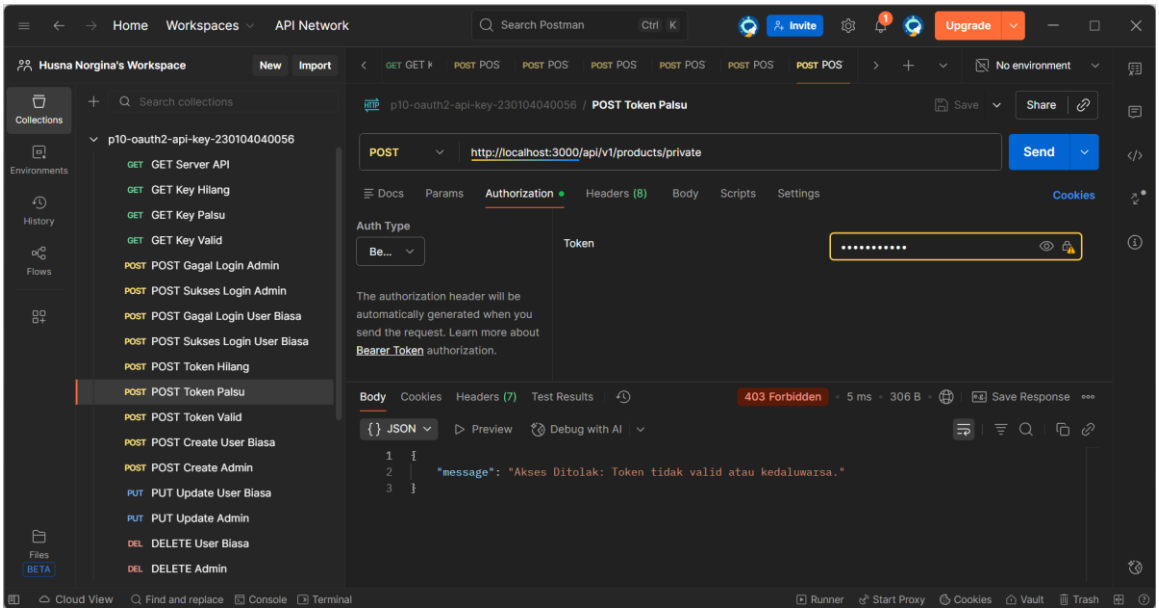
✅ POST http://localhost:3000/api/v1/auth/token → Sukses login user biasa (200 OK)



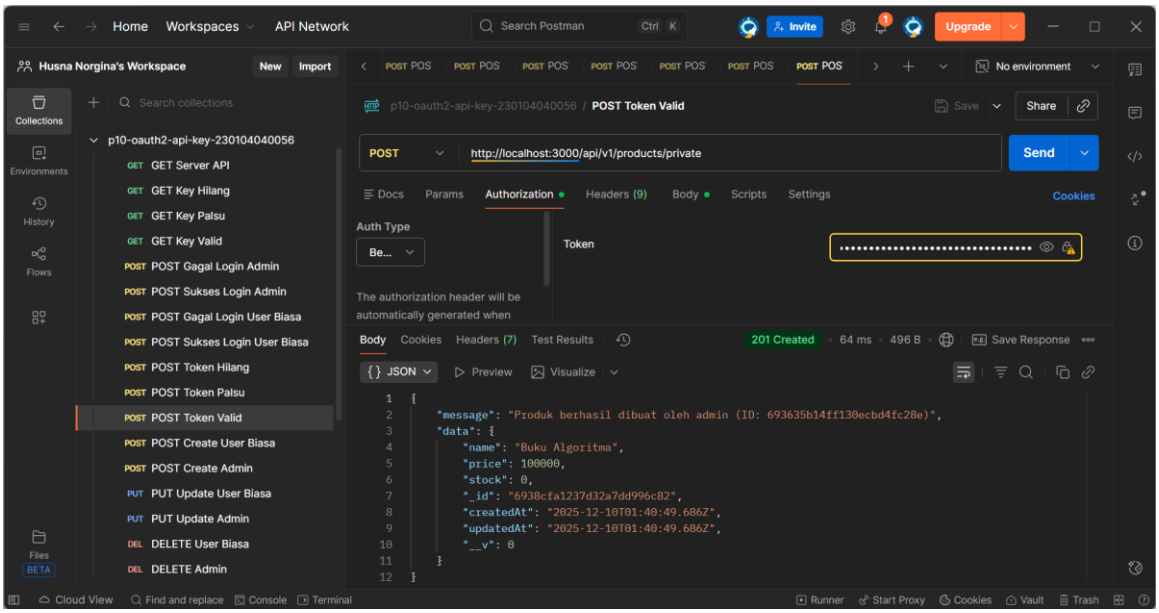
❌ POST http://localhost:3000/api/v1/products/private → Token hilang (403 Forbidden)



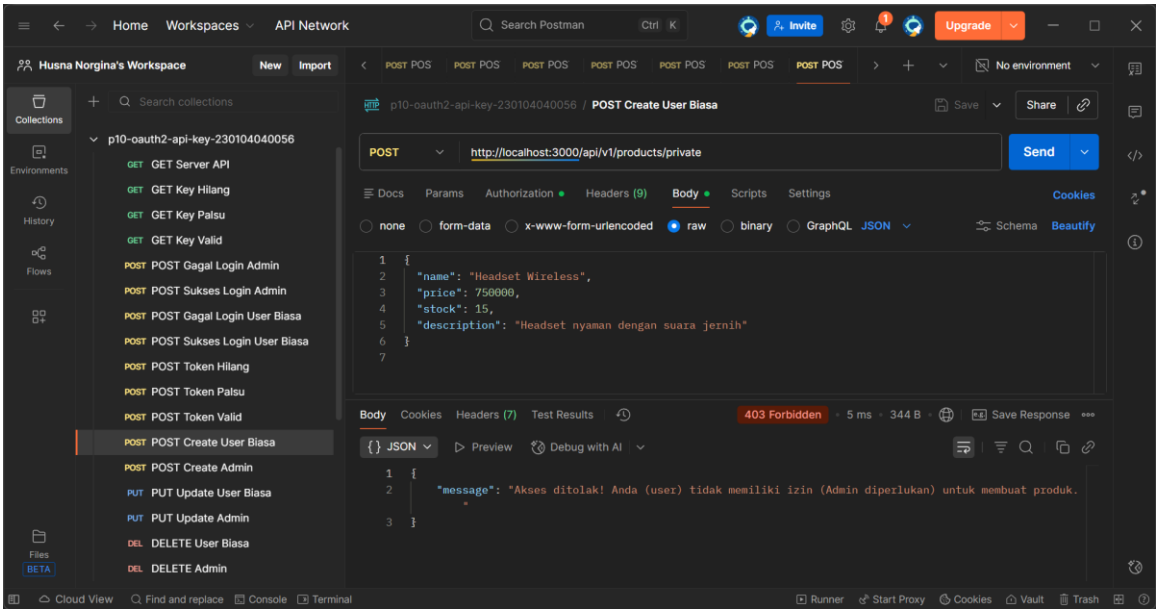
❌ POST http://localhost:3000/api/v1/products/private → Token palsu (403 Forbidden)



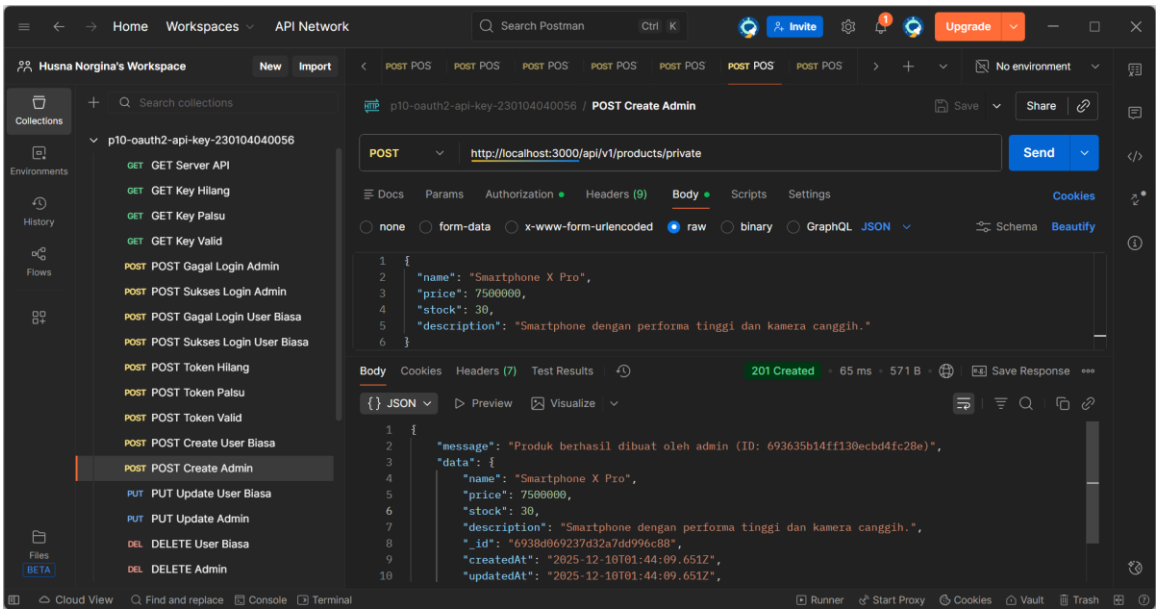
✅ POST http://localhost:3000/api/v1/products/private → Token valid (201 Created)



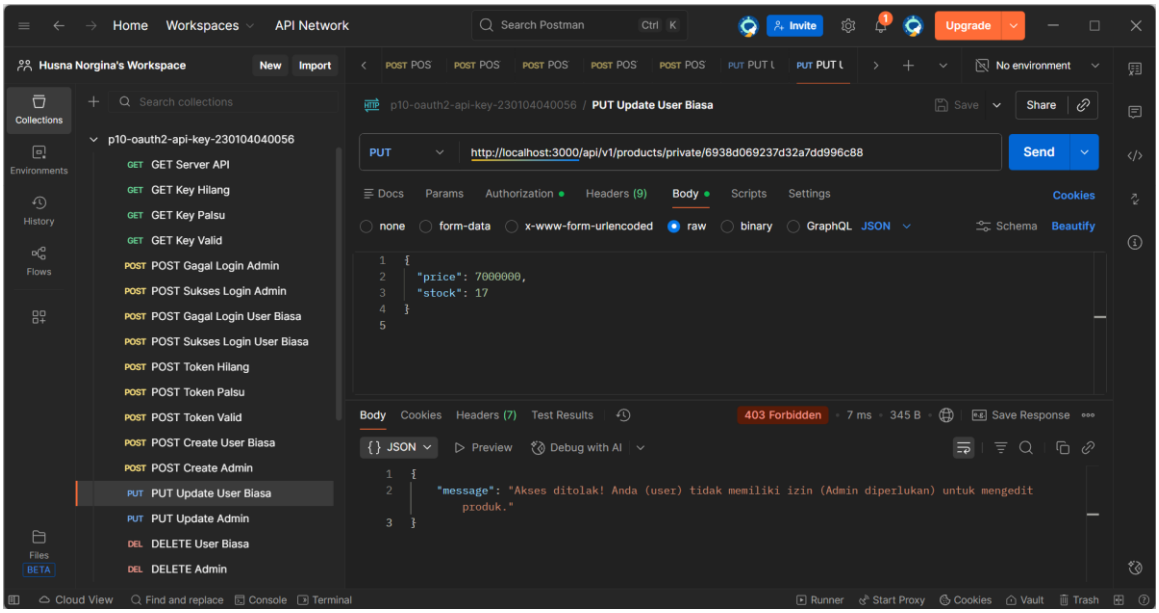
❌ POST http://localhost:3000/api/v1/products/private → Gagal create, user biasa (403 Forbidden)



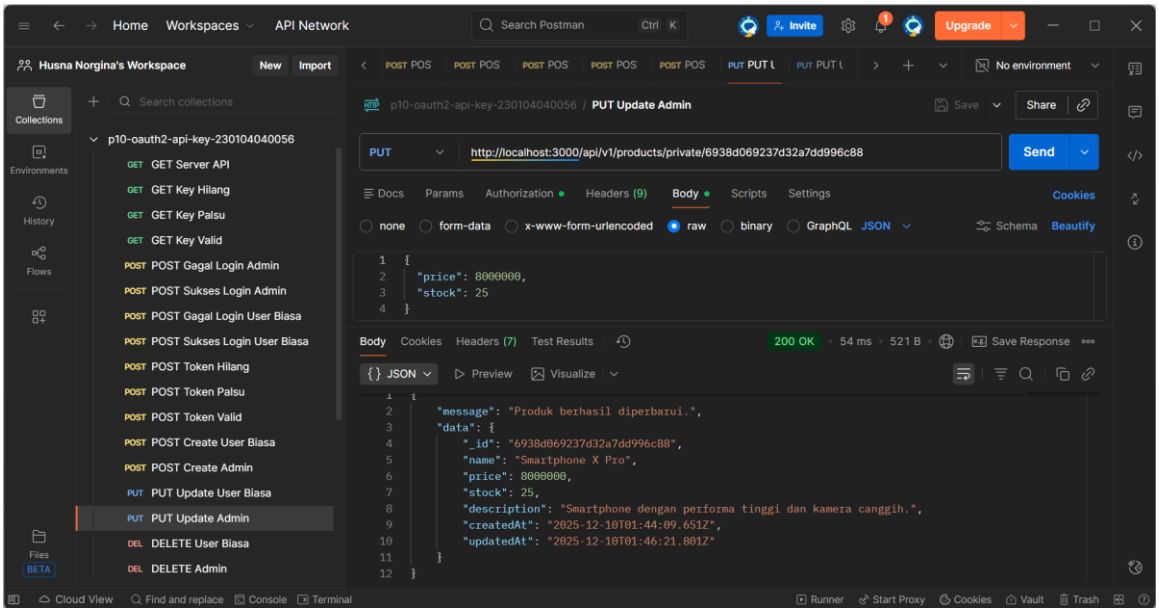
✅ POST http://localhost:3000/api/v1/products/private → Sukses create, admin (201 Created)



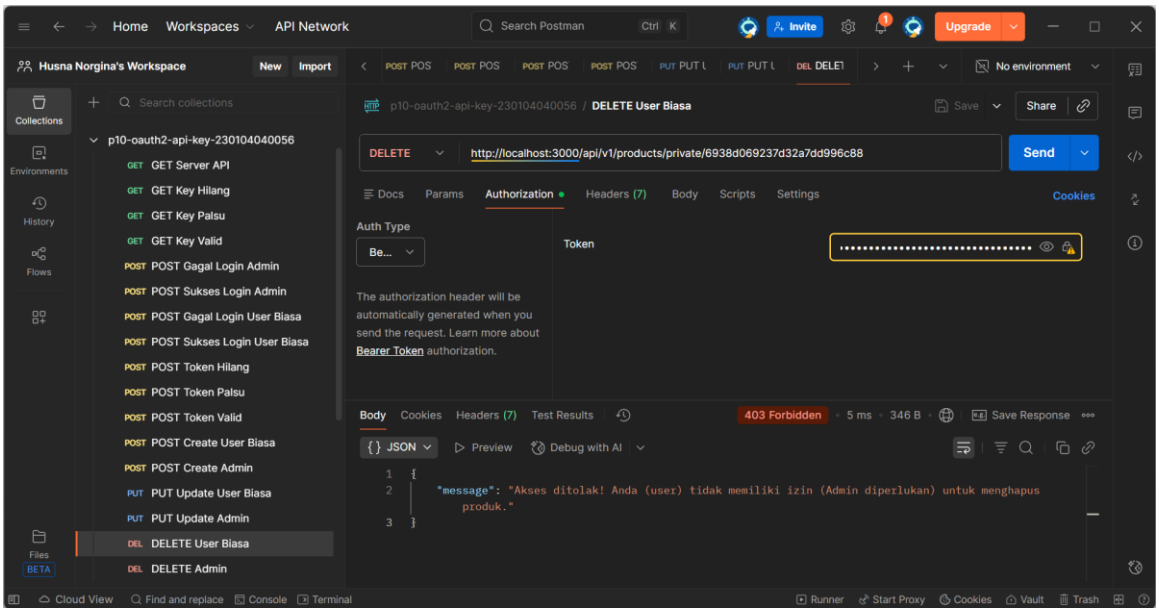
❌ PUT http://localhost:3000/api/v1/products/private/6938d069237d32a7dd996c88 → Gagal update, user biasa (403 Forbidden)



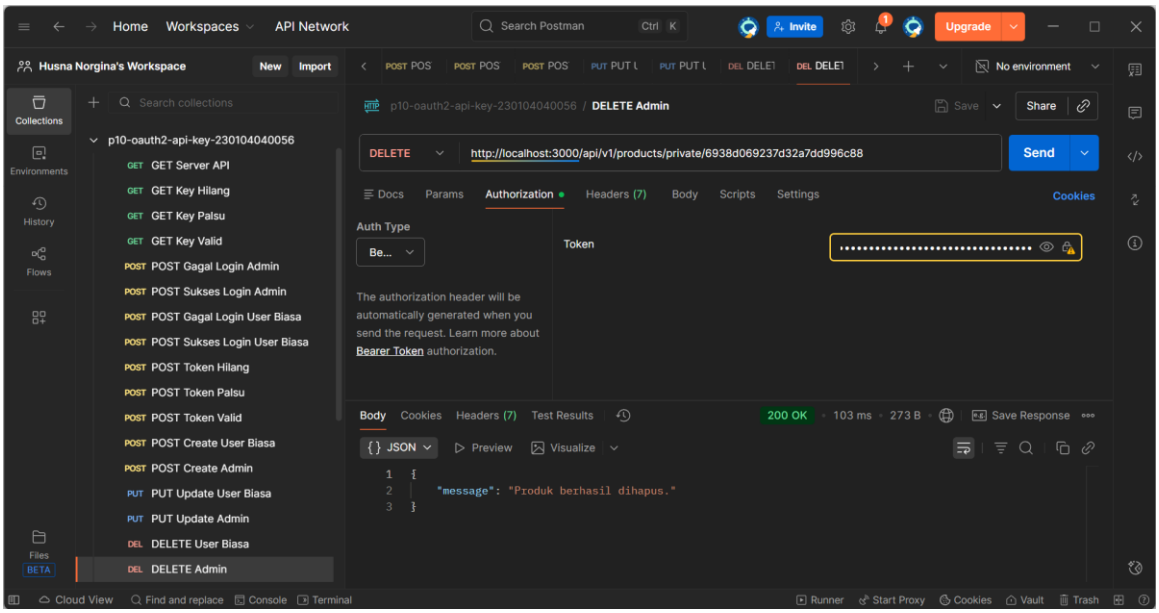
✅ PUT http://localhost:3000/api/v1/products/private/6938d069237d32a7dd996c88 → Sukses update, admin (200 OK)



❌ DEL http://localhost:3000/api/v1/products/private/6938d069237d32a7dd996c88 → Gagal delete, user biasa (403 Forbidden)



✅ DEL http://localhost:3000/api/v1/products/private/6938d069237d32a7dd996c88 → Sukses delete, admin (200 OK)



E. Analisis

Praktikum #10 berhasil menerapkan autentikasi dan otorisasi dengan API Key dan JWT secara benar, di mana middleware dapat memvalidasi API Key, memverifikasi token, dan membedakan hak akses antara user biasa dan admin. Pengujian Postman menunjukkan seluruh skenario berjalan sesuai harapan, request tanpa key atau token ditolak, token palsu terdeteksi invalid, dan admin dapat melakukan CRUD produk. Secara keseluruhan, sistem sudah aman, terstruktur, dan memenuhi tujuan praktikum, serta memberikan pemahaman yang lebih kuat tentang keamanan backend modern.

F. Kesimpulan

Praktikum 10 berhasil membangun sistem autentikasi dan otorisasi yang aman menggunakan API Key dan JWT dengan arsitektur backend Node.js yang terstruktur melalui pemisahan controllers, middleware, models, routes, dan utils. Validasi API Key pada endpoint publik serta verifikasi JWT pada endpoint private berjalan efektif, didukung kontrol akses berbasis role yang membedakan hak admin dan user biasa. Seluruh skenario pengujian Postman seperti saat key atau token hilang, key atau token palsu, dan token yang valid menunjukkan hasil yang sesuai standar keamanan. Error handling memberikan respons yang konsisten dan mudah dipahami, sedangkan seeder membantu mempermudah proses penyiapan data awal. Secara keseluruhan, sistem yang dibangun stabil, aman, mengikuti best practices, dan memenuhi tujuan praktikum.

G. Checklist Praktikum

- ✅ Validasi API Key berjalan pada endpoint publik
- ✅ Proses login menghasilkan JWT yang valid
- ✅ Middleware verifikasi token berfungsi untuk semua endpoint private
- ✅ Role admin dan user biasa terdeteksi dengan benar
- ✅ Akses CRUD hanya terbuka untuk admin sesuai aturan otorisasi
- ✅ Error handler mengembalikan respons JSON yang jelas dan konsisten
- ✅ Seed data (User, API Key, Product) berhasil dijalankan
- ✅ Semua skenario pengujian Postman berhasil sesuai ekspektasi
- ✅ Evidence Postman lengkap
- ✅ Dokumentasi README.md selesai
- ✅ Dokumentasi project tersusun rapi dalam struktur folder aplikasi