Laporan Praktikum WSE #8

Mata Kuliah	:	Web Service Engineering
Dosen Pengampu	:	Muhayat, M.IT
Praktikum	:	P8 – **Secure & Observable RESTful CRUD API (JWT + Hardening + Observability)**
Nama Mahasiswa	:	Husna Norgina
NIM	:	230104040056
Kelas	:	TI23B
Link Repo Github	:	https://github.com/husna-norgina/P8-SecureArticles-230104040056
Tanggal Praktikum	:	24-11-2025

A. Tujuan Praktikum

1. Merancang API CRUD sesuai 7 RESTful Principles secara konsisten (resource, method, status code, HATEOAS ringan, stateless, caching, layered system).
2. Mengimplementasikan JWT Authentication (register/login/refresh/logout) dengan keamanan standar industri.
3. Menerapkan role-based authorization (admin vs user).
4. Melakukan hardening API: validation, rate limit, security headers, CORS, error hygiene, env secrets, dsb.
5. Membangun observability: structured logging + correlation-id + health/metrics endpoint.
6. Menyusun dokumentasi OpenAPI yang akurat, test-ready, dan bisa dipakai integrasi lintas layanan.

B. Aktivitas Praktikum (Workflow Advance)

Gunakan pola kerja modern:

1. Design-First (OpenAPI 3.1 + Spectral Linting)
2. Mock-First (Prism/Postman Mock Server)
3. Test-First (RED with Jest/Supertest + auth tests)
4. Implementasi (GREEN)
5. Hardening (Security + Reliability)
6. Observability (Logs + Metrics + Healthcheck)
7. CI (GitHub Actions: lint → test → build)
8. Demo & Dokumentasi Final

C. Tools yang Dipakai

Wajib:

- Node.js 18+ / 20 LTS
- Express.js
- MongoDB / PostgreSQL (pilih salah satu)
- JWT library (mis. jsonwebtoken)
- bcrypt
- Joi/Zod/Express-validator (validation)
- Jest + Supertest
- OpenAPI + Spectral
- Postman/Insomnia

• GitHub Actions

Observability/Hardening (advance layer):
• pino / winston (structured logger)
• express-rate-limit
• helmet
• cors
• morgan diganti structured log
• uuid / nanoid (correlation-id)
• swagger-ui-express (docs)
• optional: prom-client (metrics)

D. Struktur Project yang Direkomendasikan (Advance)

```
src/
  app.js
  server.js

  config/
    env.js
    db.js

  routes/
    auth.routes.js
    articles.routes.js
    system.routes.js

  controllers/
    auth.controller.js
    articles.controller.js
    system.controller.js

  services/
    auth.service.js
    articles.service.js

  repositories/
    articles.repo.js
    users.repo.js

  middlewares/
    auth.middleware.js
    role.middleware.js
    validate.middleware.js
    correlationId.middleware.js
    rateLimit.middleware.js
    error.middleware.js
    notFound.middleware.js


  utils/
    jwt.js
    logger.js
    response.js

  docs/
    openapi.yaml

tests/
  auth.test.js
  articles.test.js
```

E. Tabel Endpoint Project Praktikum #8

AUTH ENDPOINTS

Method	Endpoint	Auth	Deskripsi
POST	/api/auth/register	✗ Public	Register user baru (role: user/admin)
POST	/api/auth/login	✗ Public	Login → dpt accessToken+refreshToken
POST	/api/auth/refresh	✗ Public	Minta accessToken baru via refreshToken
POST	/api/auth/logout	✓ Access Token	Logout & invalidate refreshToken
GET	/api/auth/me	✓ Access Token	Ambil profil user dari JWT

ARTICLES ENDPOINTS (CRUD + RBAC)

Method	Endpoint	Auth	Role	Deskripsi
GET	/api/articles	✗ Public	public	List all articles + pagination + search
POST	/api/articles	✓ Access Token	user/admin	Create article (author otomatis dari JWT)
PUT	/api/articles/:id	✓ Access Token	owner/admin	Update article
DELETE	/api/articles/:id	✓ Access Token	admin	Hapus article

SYSTEM / OBSERVABILITY

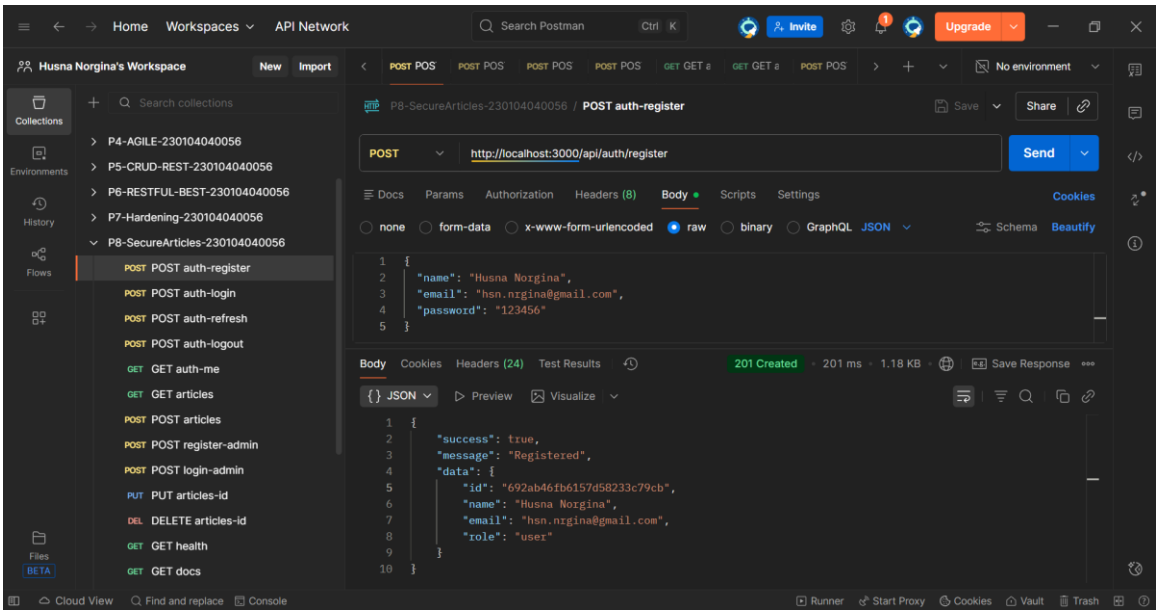
Method	Endpoint	Auth	Deskripsi
GET	/health	✗ Public	Cek status server
GET	/docs	✗ Public	Swagger UI (OpenAPI Documentation)

F. Hasil & Bukti

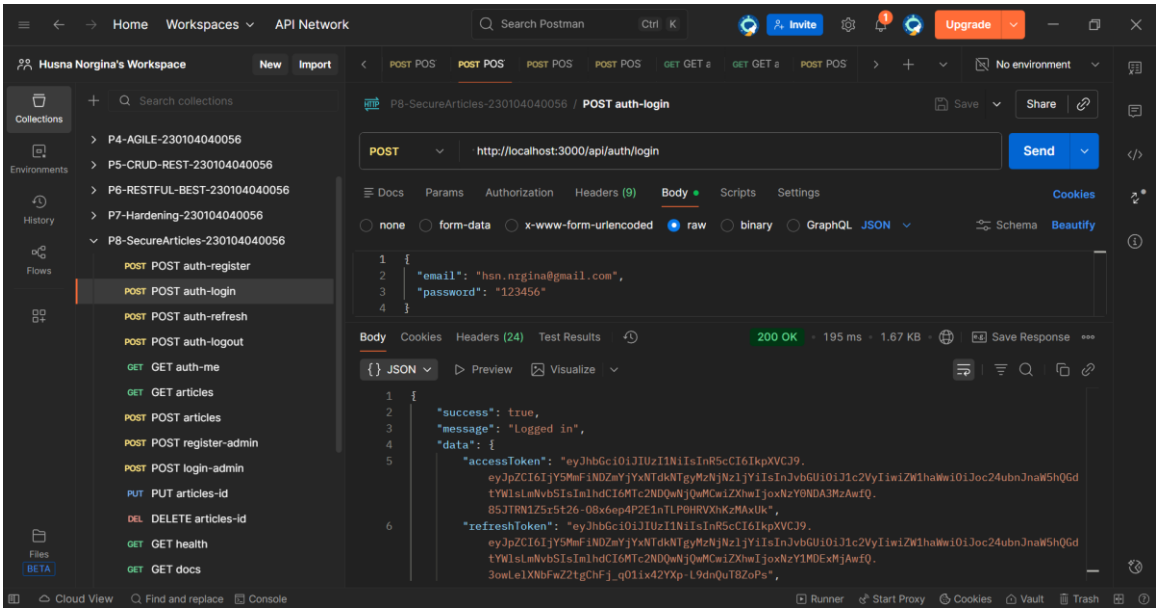
Screenshot Hasil Uji Endpoint di Postman

🛡️ AUTH ENDPOINTS

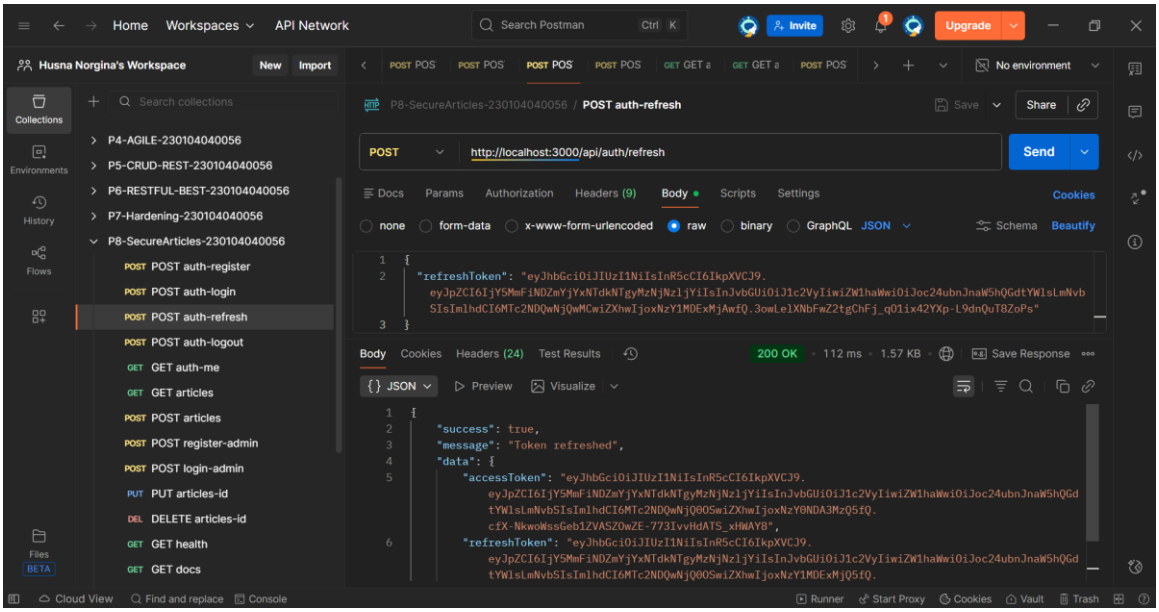
✅ POST http://localhost:3000/api/auth/register → Register user baru (201 Created)



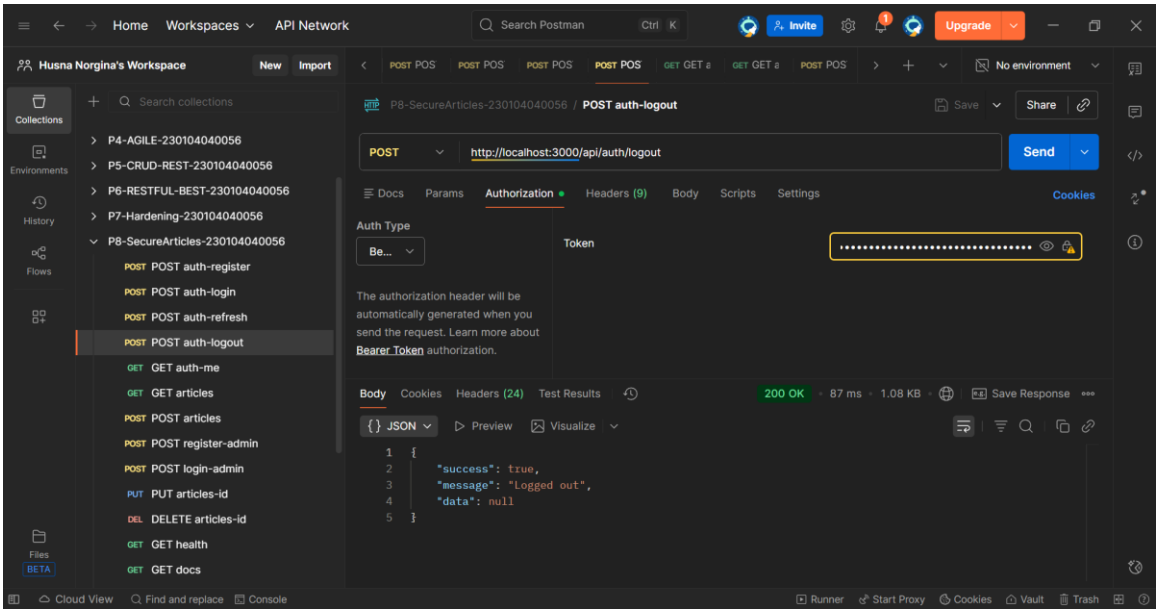
✅ POST http://localhost:3000/api/auth/login → Login user (200 OK)



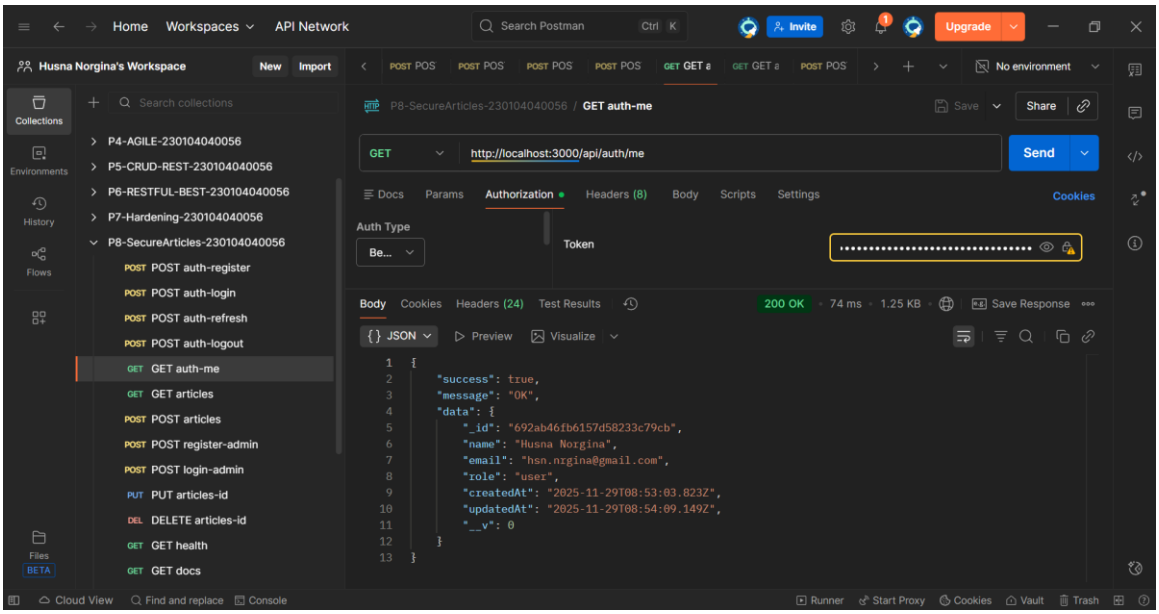
✅ POST http://localhost:3000/api/auth/refresh → Refresh access token (200 OK)



✅ POST http://localhost:3000/api/auth/logout → Logout & invalidate token (200 OK)

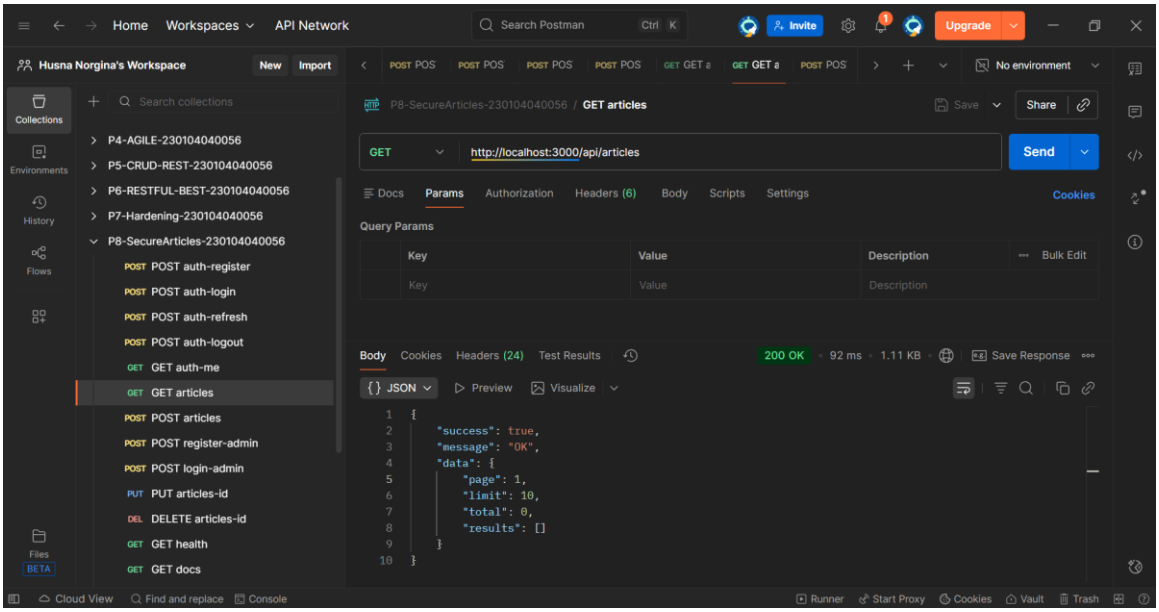


✅ GET http://localhost:3000/api/auth/me → Get profile user (200 OK)

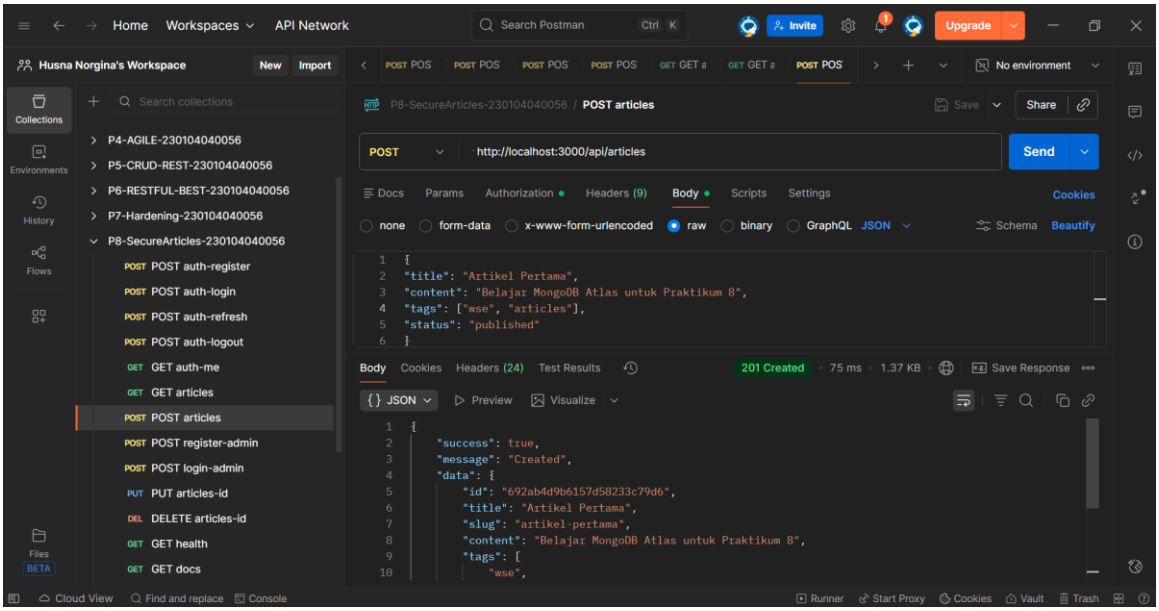


📰 ARTICLES ENDPOINTS (CRUD + RBAC)

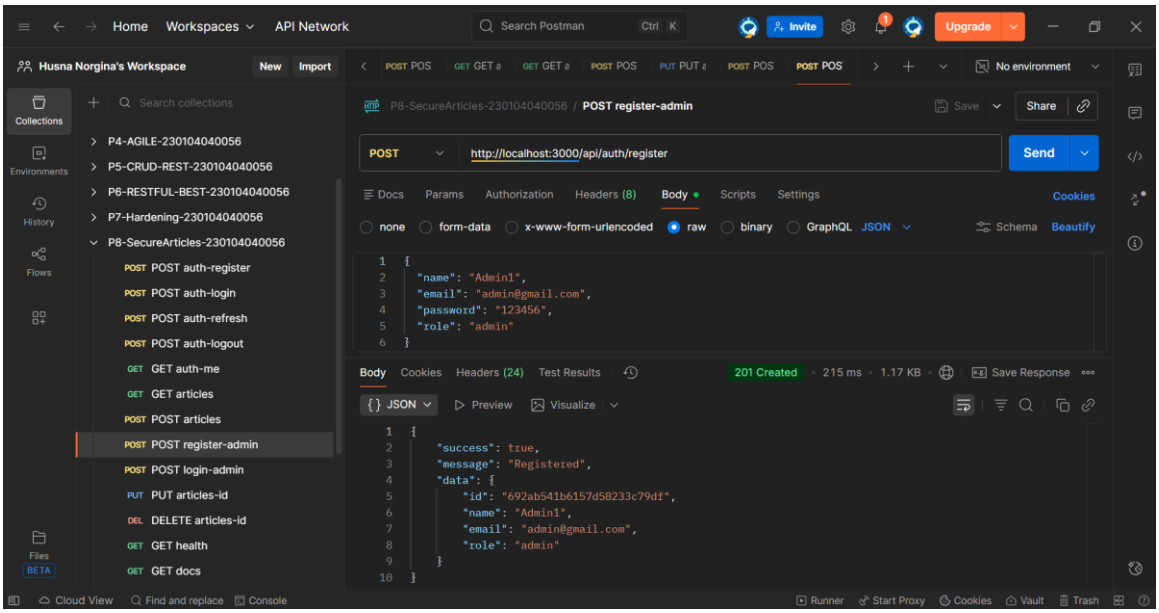
✅ GET http://localhost:3000/api/articles → List artikel (200 OK)



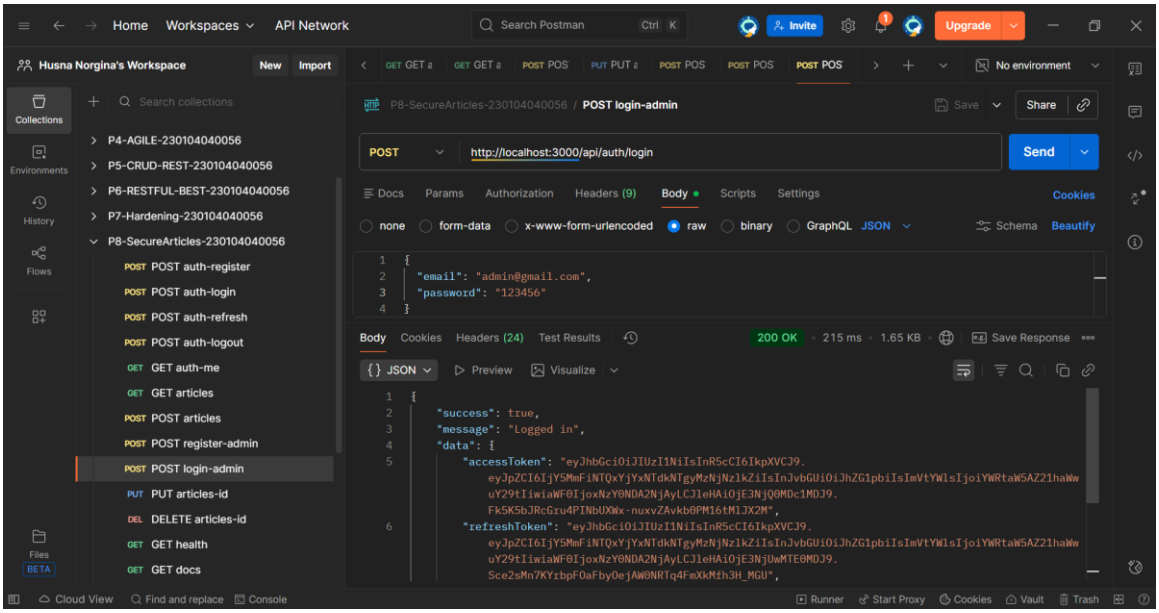
✅ POST http://localhost:3000/api/articles → Created article (201 Created)



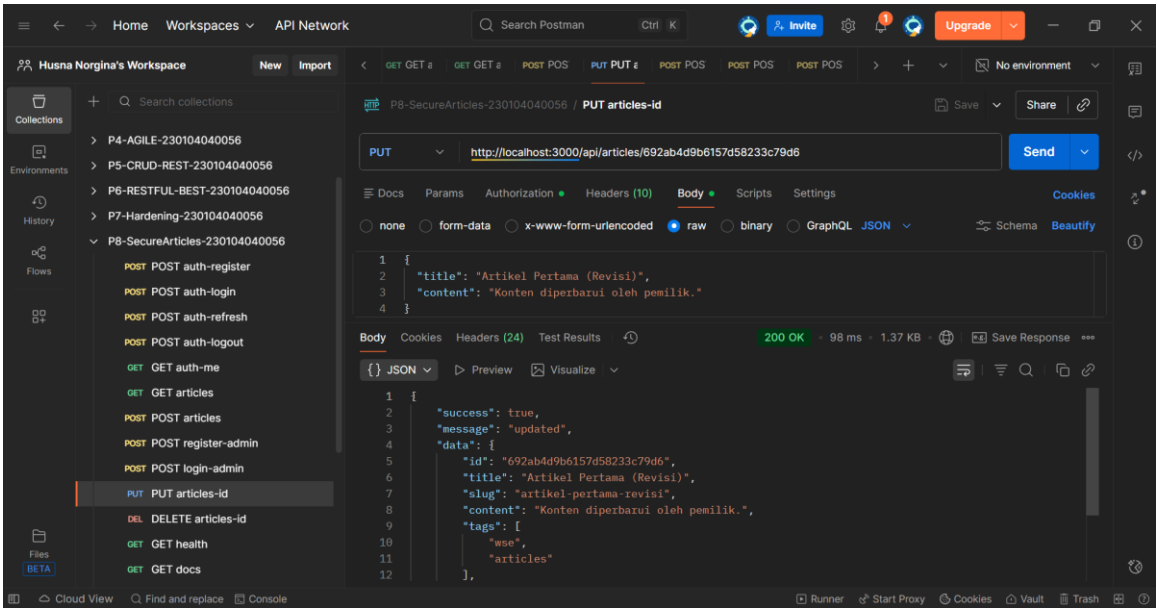
✅ POST http://localhost:3000/api/auth/register → Register admin baru (201 Created)



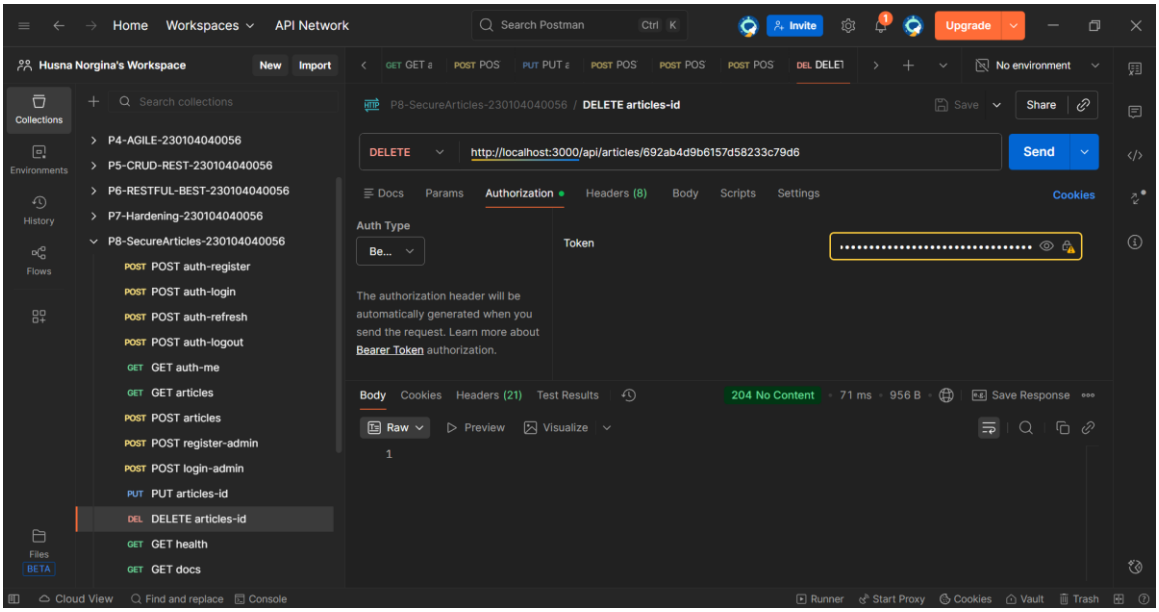
✅ POST http://localhost:3000/api/auth/login → Login admin (200 OK)



✅ PUT http://localhost:3000/api/articles → Update article (200 OK)

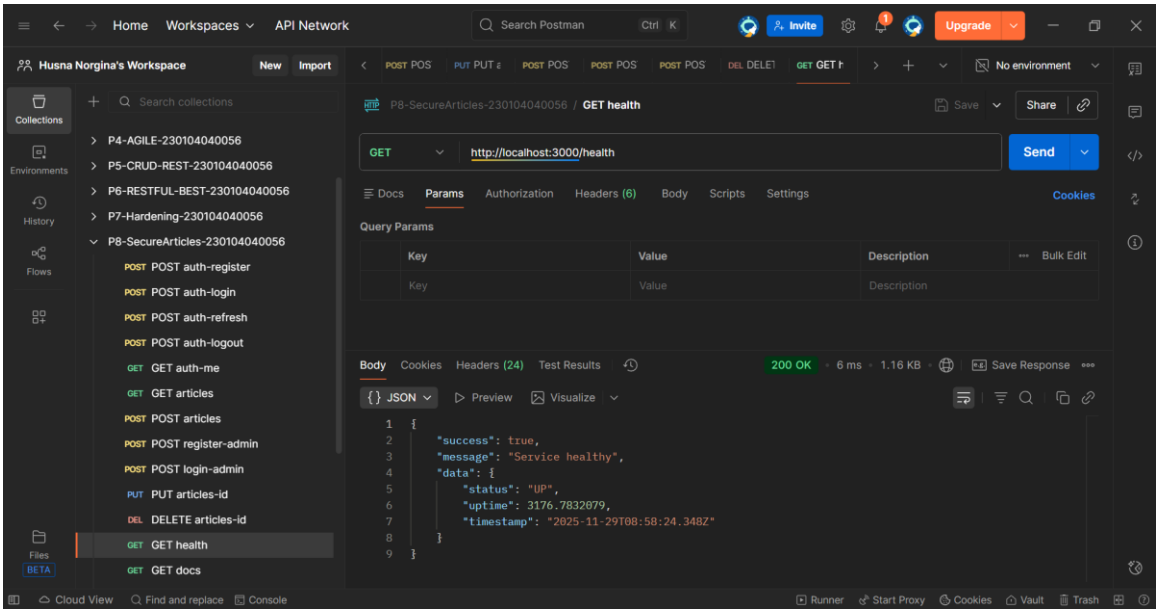


✅ DELETE http://localhost:3000/api/articles → Hapus article (204 No Content)

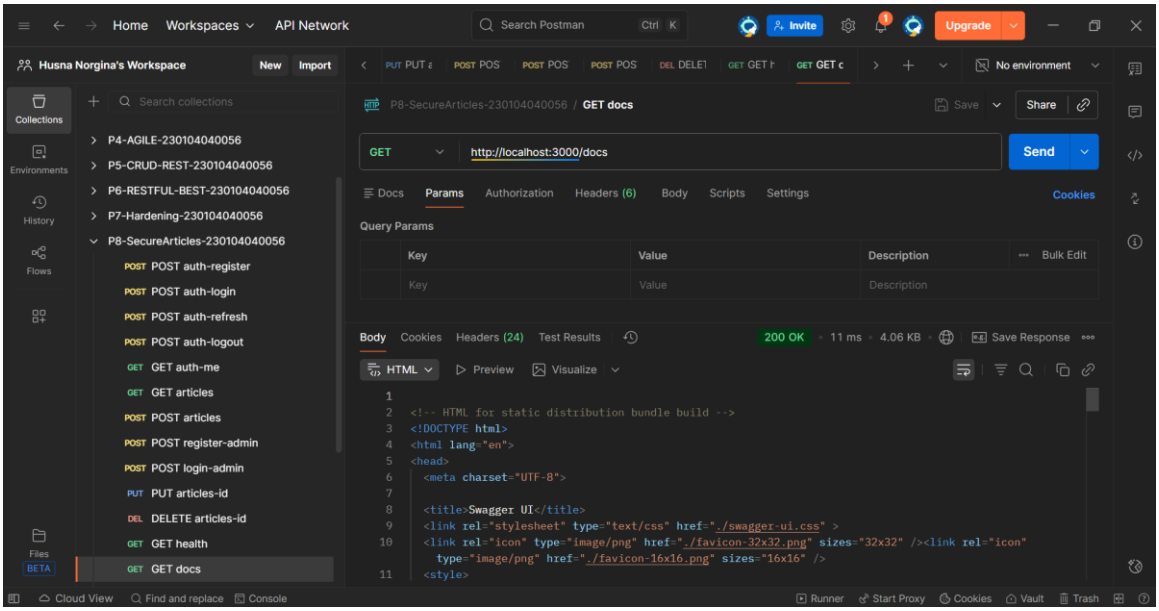


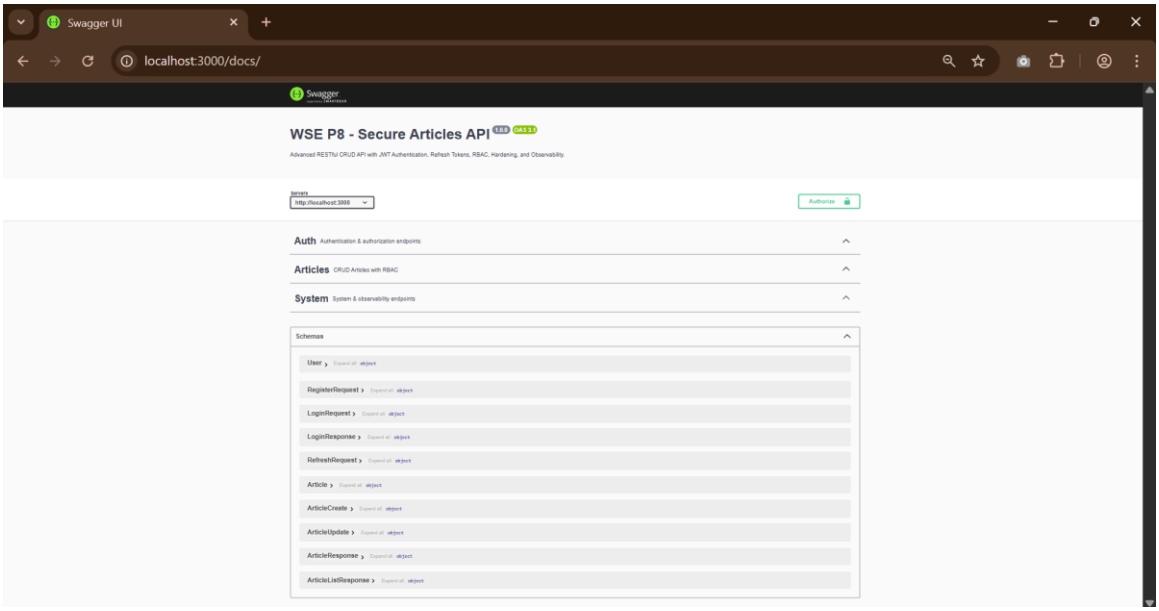
⚙️ SYSTEM / OBSERVABILITY

✅ GET http://localhost:3000/health → Cek status server (200 OK)



✅ GET http://localhost:3000/docs → Swagger UI (200 OK)





G. Analisis

API Articles pada Praktikum #8 telah memenuhi prinsip RESTful dengan struktur modular (routes, controllers, services, repositories) dan status code konsisten (200, 201, 204, 400, 401, 403, 404, 500). Autentikasi dan otorisasi berbasis JWT berjalan benar, termasuk refresh token dan RBAC untuk membatasi akses create, update, delete artikel. Middleware keamanan seperti Helmet, CORS, rate limiter, input validation membuat API tahan serangan, observability diperkuat dengan Pino logger, correlation-id, /health, dan dokumentasi OpenAPI/Swagger memudahkan pengujian serta siap production.

H. Kesimpulan

Praktikum 8 berhasil membangun Secure & Observable RESTful API tingkat lanjut untuk resource Articles dengan menerapkan JWT Authentication (access & refresh token), role-based authorization (admin dan user), serta hardening menyeluruh seperti validation, rate limit, security headers Helmet, CORS whitelist, sanitasi input, dan pengelolaan secret melalui .env. Observability ditingkatkan dengan structured logging menggunakan Pino, middleware correlation-id, healthcheck endpoint, serta dokumentasi OpenAPI yang konsisten. Seluruh operasi CRUD Articles berjalan sesuai 7 RESTful Principles, dilengkapi pagination, filter, sort, dan search. Endpoint Auth (register, login, refresh, logout, me) berfungsi dengan baik, integrasi RBAC berjalan aman, dan struktur project tetap modular, production-ready, serta sesuai standar industri.

I. Checklist Praktikum

- ✅ JWT Authentication
- ✅ Refresh Token + revoke
- ✅ RBAC (user/admin)
- ✅ CRUD Articles
- ✅ Helmet + CORS whitelist
- ✅ Rate limiting
- ✅ Request logging (Pino)
- ✅ Correlation ID
- ✅ Health check
- ✅ Dokumentasi OpenAPI
- ✅ Evidence Postman lengkap
- ✅ Dokumentasi README.md selesai