

LAPORAN PRAKTIKUM WSE #4

Mata Kuliah : Web Service Engineering

Dosen Pengampu : Muhayat, M.IT

Praktikum : P4 – Web Service Development Methodologies (AGILE)

Kelompok 4 :

- Muhammad Riduwan (Project Manager) – NIM: 230104040080
- Muna Rahimatul 'Olya – NIM: 230104040081
- Siti Alayda Azzahro – NIM: 230104040084
- Noor Ahmad Naufal – NIM: 230104040074

Kelas : TI23B

Tanggal Praktikum : 20 Oktober 2025

A. Tujuan Praktikum

1. Menerapkan metodologi Agile dalam pengembangan *Web Service* berbasis kontrak (OpenAPI).
2. Melakukan proses Design-First dan memastikan spesifikasi API tervalidasi (lint 0 error).
3. Mengimplementasikan Mock-First (Prism) untuk menguji skenario 201, 200, 400, dan 401.
4. Melakukan pendekatan Test-First (RED) hingga seluruh tes berhasil (*GREEN*).
5. Menyusun *Continuous Integration (CI)* untuk lint, typecheck, dan test otomatis menggunakan GitHub Actions (opsional mahasiswa).
6. Melakukan proses Hardening untuk meningkatkan keamanan dan observabilitas layanan, meliputi logging terstruktur, correlation-id, security headers, rate-limit, serta validasi error.

B. Lingkungan & Tools

- Node.js v18+ dan npm
- Visual Studio Code sebagai IDE utama
- Spectral (OpenAPI linter) dan Prism (mock server)
- TypeScript, Jest, ts-jest, Supertest
- Express, Zod, Helmet, express-rate-limit
- Pino / pino-http untuk logging
- GitHub Actions (opsional CI) dan Postman untuk uji manual

C. Arsitektur Singkat

- Services: order-service (POST /orders) dan notification-service (GET /notifications)
- Kontrak API: openapi/api.yaml (lint 0 error)
- Shared utilities: utils.ts berisi logger, correlation-id, auth dummy, validator, dan error handler.

D. Langkah Implementasi

1. Design-First
Menulis file openapi/api.yaml dan melakukan linting menggunakan npx spectral lint openapi/api.yaml dengan hasil 0 error.
2. Mock-First
Menjalankan npx prism mock openapi/api.yaml dan merekam hasil 201, 200, 401, 400 di folder mock_logs/.
3. Test-First(RED)
Menulis tes menggunakan Jest dan Supertest hingga semua tes gagal sesuai ekspektasi.
4. Implementasi (GREEN)
Mengembangkan middleware dan route hingga semua tes berhasil (*npm test* lulus semua).
5. CI Setup (*opsional mahasiswa*)
Menambahkan workflow .github/workflows/ci.yml dan memastikan seluruh lint dan tes berjalan otomatis.
6. Hardening
Menambahkan Helmet, rate-limit, redaksi auth di log, serta validasi JSON untuk mengembalikan error 400 jika format salah. Setiap respons mengandung header x-correlation-id.
7. Demo & Dokumentasi
Seluruh bukti hasil lint, mock, dan tes disimpan di docs/, mock_logs/, dan hardening_logs/.

E. Hasil & Bukti

1. OpenAPI Lint

- Perintah: `npx spectral lint openapi/api.yaml`
 - Hasil: **0 error / 0 warning**
 - Bukti: docs/spectral_pass.png
- Berikut Screenshoot nya

```
PS C:\Users\User\Documents\PRAKTIKUM-WSE\P4-AGILE-KELOMPOK4> npx spectral lint openapi/api.yaml
No results with a severity of 'error' found!
PS C:\Users\User\Documents\PRAKTIKUM-WSE\P4-AGILE-KELOMPOK4>
```

2. Unit Test (Jest)

- Perintah: `npm test`
 - Hasil: **2 suites, 5 tests passed**
 - Bukti: docs/npm_test_pass.png
- Berikut Screenshoot nya

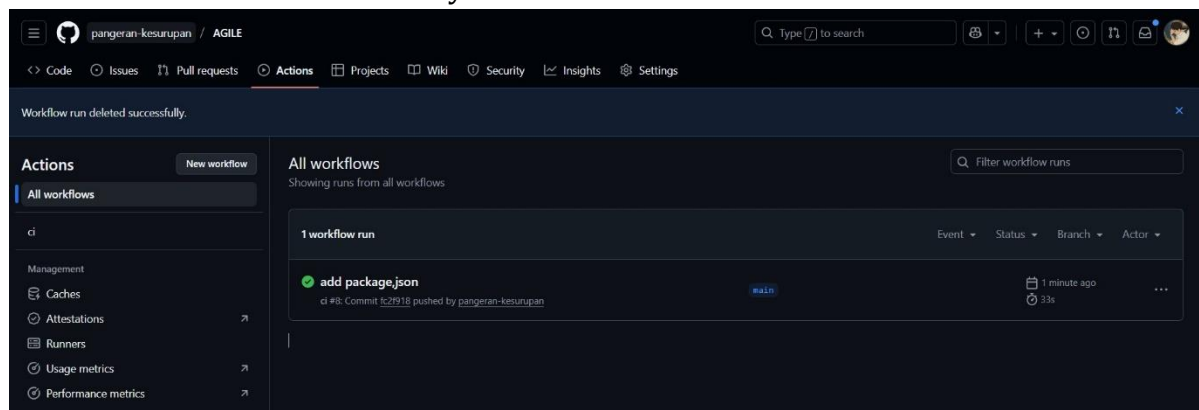
```
PS C:\Users\User\Documents\PRAKTIKUM-WSE\P4-AGILE-KELOMPOK4> npm test

GET /notifications
  ✓ 401 tanpa bearer (8 ms)
  ✓ 200 sukses (6 ms)

Test Suites: 2 passed, 2 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        0.89 s
Ran all test suites.
PS C:\Users\User\Documents\PRAKTIKUM-WSE\P4-AGILE-KELOMPOK4>
```

3. CI (GitHub Actions) (opsional)

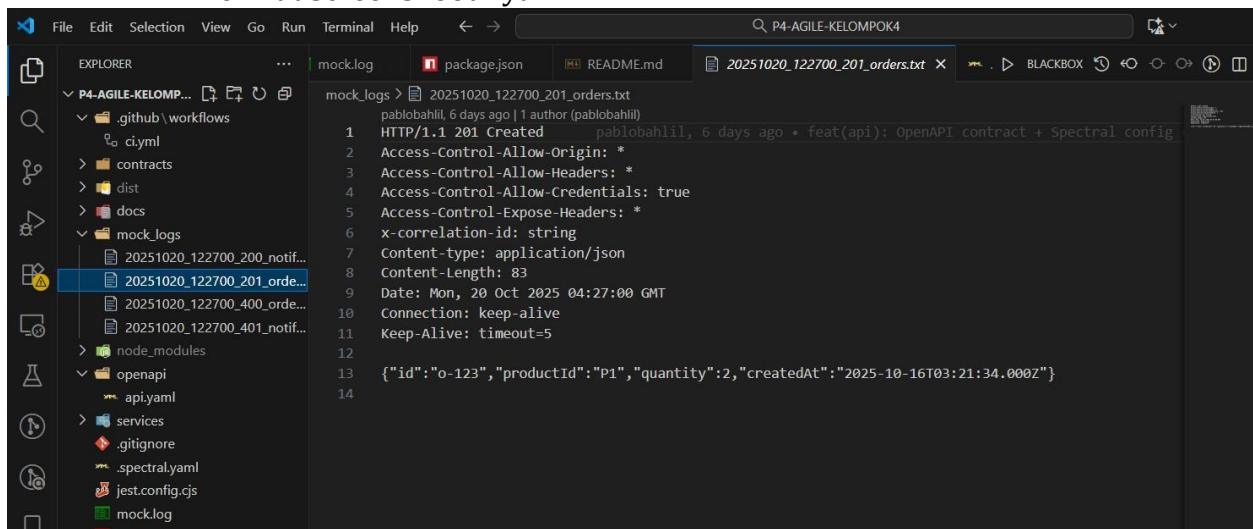
- Workflow: `.github/workflows/ci.yml`
 - Status: **Hijau / Tidak digunakan**
- Berikut Screenshoot nya



4. Mock Logs (mock_logs/)

- 201 Created (orders)

Berikut Screenshoot nya

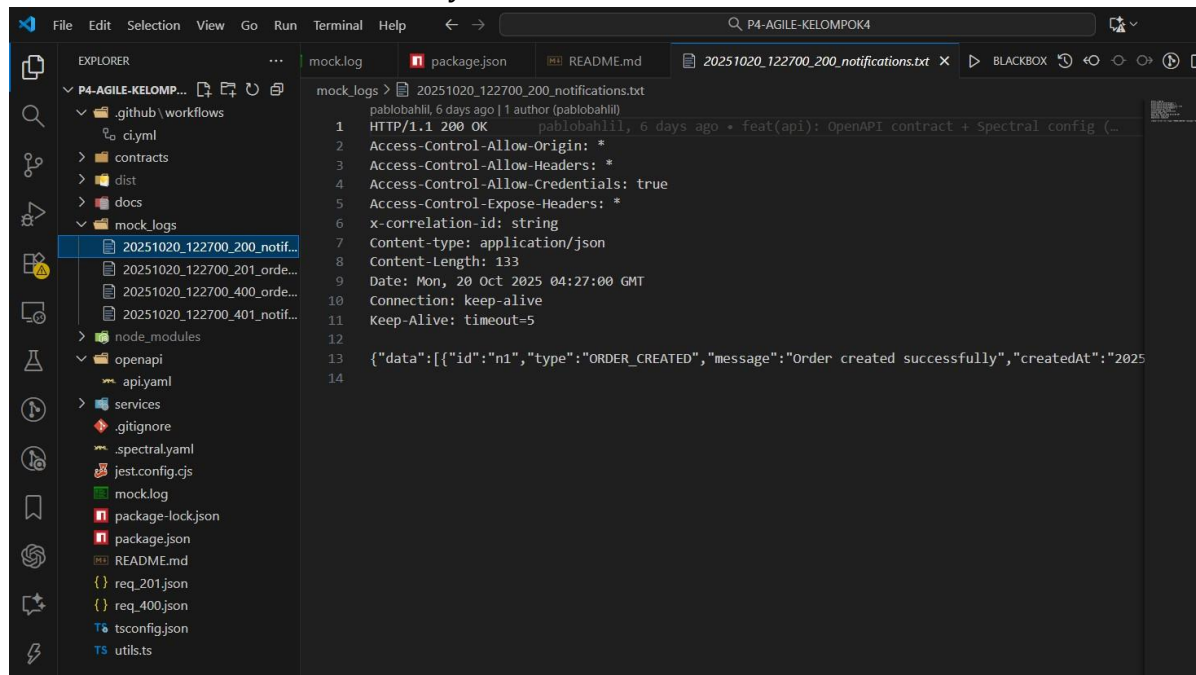


The screenshot shows the Visual Studio Code interface with the Explorer view on the left and the Editor view on the right. The Explorer view shows the file structure of the project, with the `mock_logs` folder expanded. The Editor view shows the content of the `20251020_122700_201_orders.txt` file, which contains a mock log entry for a 201 Created response.

```
mock_logs > 20251020_122700_201_orders.txt
pablohahil, 6 days ago | 1 author (pablohahil)
1 HTTP/1.1 201 Created pablohahil, 6 days ago * feat(api): OpenAPI contract + Spectral config
2 Access-Control-Allow-Origin: *
3 Access-Control-Allow-Headers: *
4 Access-Control-Allow-Credentials: true
5 Access-Control-Expose-Headers: *
6 x-correlation-id: string
7 Content-type: application/json
8 Content-Length: 83
9 Date: Mon, 20 Oct 2025 04:27:00 GMT
10 Connection: keep-alive
11 Keep-Alive: timeout=5
12
13 {"id":"o-123","productId":"P1","quantity":2,"createdAt":"2025-10-16T03:21:34.000Z"}
14
```

- 200 OK (notifications)

Berikut Screenshoot nya

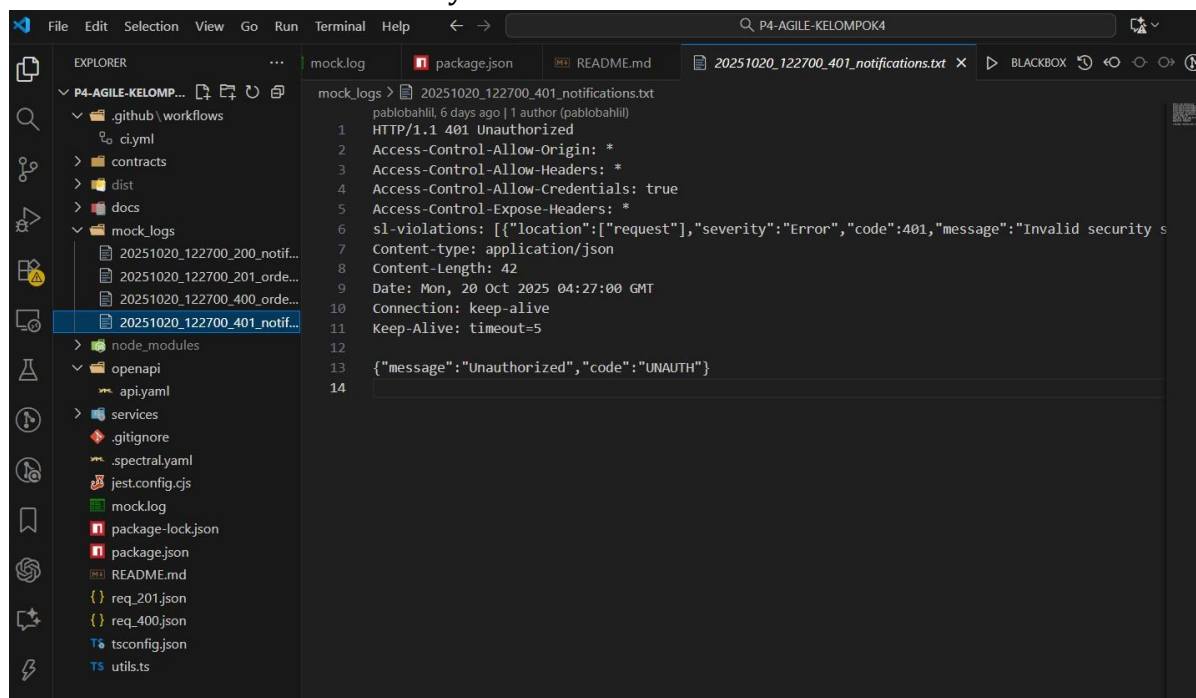


The screenshot shows the Visual Studio Code interface with the Explorer view on the left and the mock_logs file open in the editor. The file 20251020_122700_200_notifications.txt is selected in the Explorer. The editor displays the following content:

```
mock_logs > 20251020_122700_200_notifications.txt
pablobahlil, 6 days ago | 1 author (pablobahlil)
1 HTTP/1.1 200 OK pablobahlil, 6 days ago * feat(api): OpenAPI contract + Spectral config (...)
2 Access-Control-Allow-Origin: *
3 Access-Control-Allow-Headers: *
4 Access-Control-Allow-Credentials: true
5 Access-Control-Expose-Headers: *
6 x-correlation-id: string
7 Content-type: application/json
8 Content-Length: 133
9 Date: Mon, 20 Oct 2025 04:27:00 GMT
10 Connection: keep-alive
11 Keep-Alive: timeout=5
12
13 {"data":[{"id":"n1","type":"ORDER_CREATED","message":"Order created successfully","createdAt":"2025
14
```

- 401 Unauthorized

Berikut Screenshoot nya

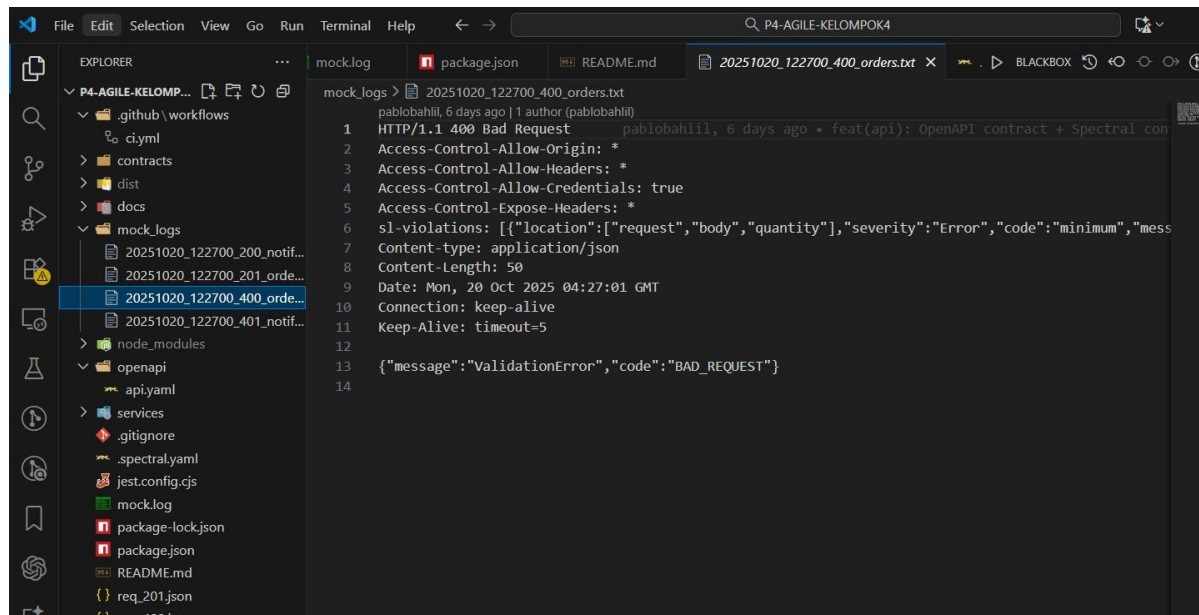


The screenshot shows the Visual Studio Code interface with the Explorer view on the left and the mock_logs file open in the editor. The file 20251020_122700_401_notifications.txt is selected in the Explorer. The editor displays the following content:

```
mock_logs > 20251020_122700_401_notifications.txt
pablobahlil, 6 days ago | 1 author (pablobahlil)
1 HTTP/1.1 401 Unauthorized
2 Access-Control-Allow-Origin: *
3 Access-Control-Allow-Headers: *
4 Access-Control-Allow-Credentials: true
5 Access-Control-Expose-Headers: *
6 sl-violations: [{"location":["request"],"severity":"Error","code":401,"message":"Invalid security s
7 Content-type: application/json
8 Content-Length: 42
9 Date: Mon, 20 Oct 2025 04:27:00 GMT
10 Connection: keep-alive
11 Keep-Alive: timeout=5
12
13 {"message":"Unauthorized","code":"UNAUTH"}
14
```

- 400 Bad Request

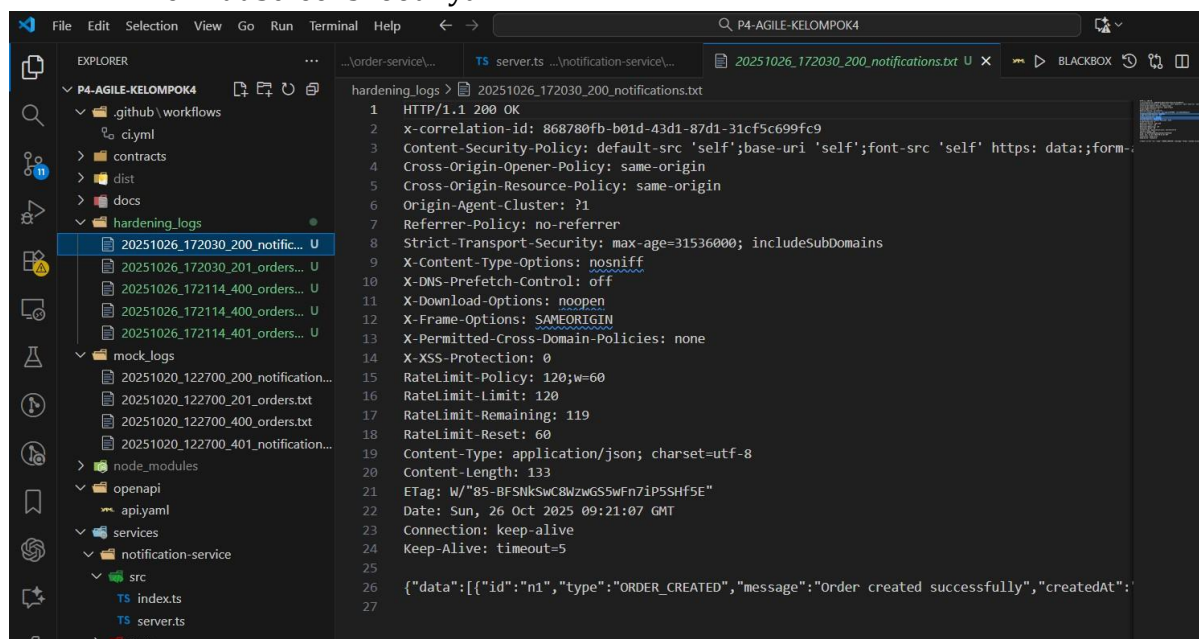
Berikut Screenshoot nya



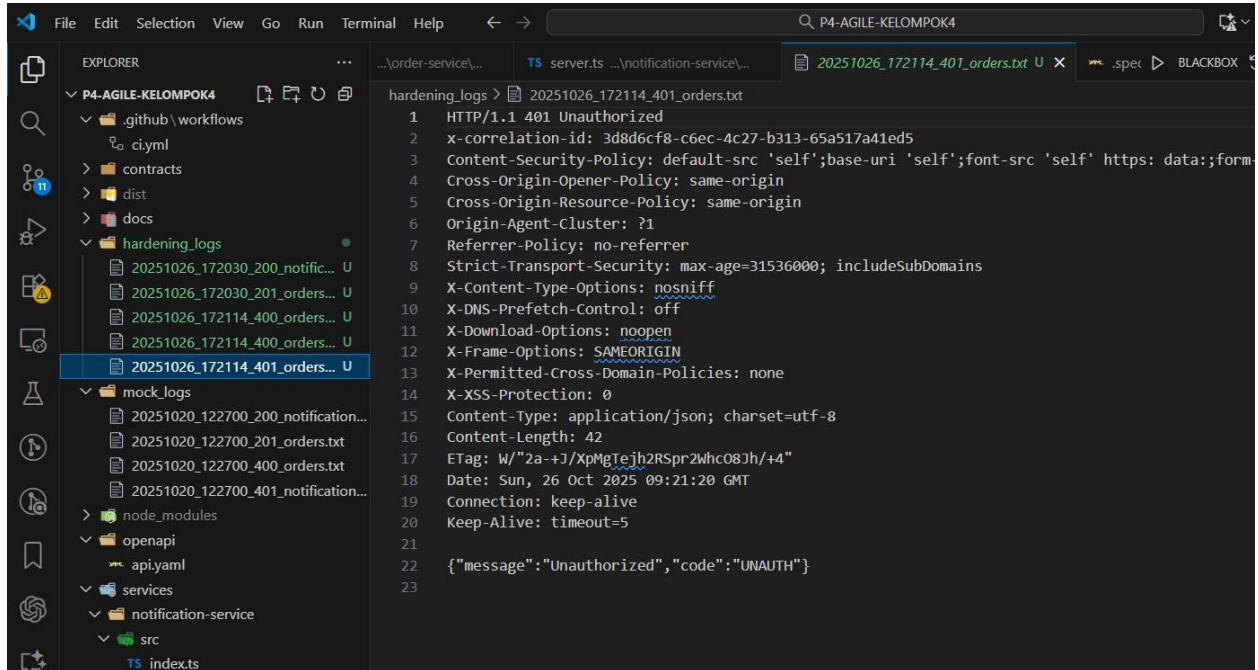
5. Hardening Logs (hardening_logs/)

- 201 Created, 200 OK,

Berikut Screenshoot nya



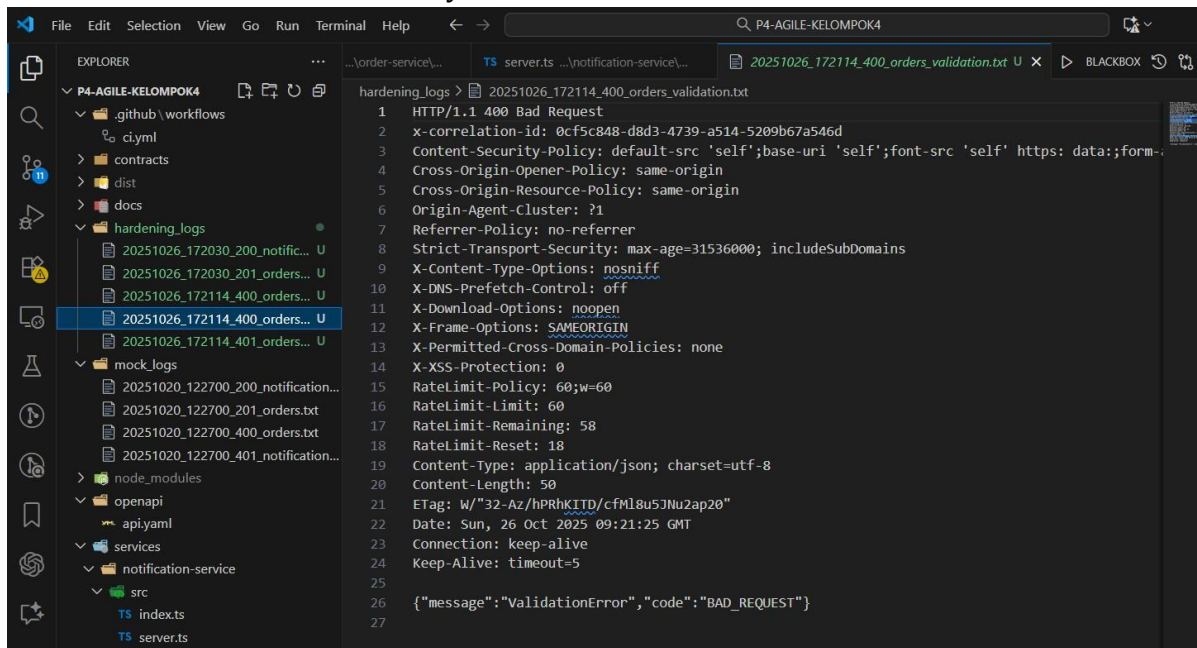
- 401 Unauthorized,
Berikut Screenshoot nya



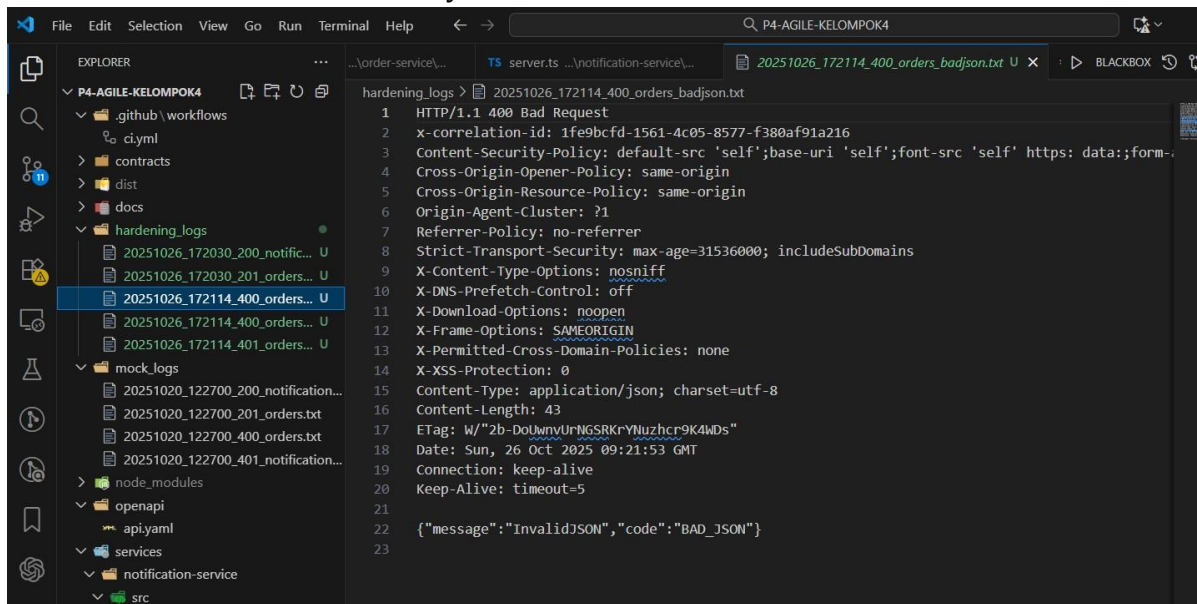
The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane displays a file tree for a project named 'P4-AGILE-KELOMPOK4'. The file '20251026_172114_401_orders... U' is selected under the 'hardening_logs' folder. The main editor pane shows the content of this file, which is a log entry for an HTTP 401 Unauthorized response. The log entry includes various headers and a JSON body indicating the reason for the unauthorized access.

```
1 HTTP/1.1 401 Unauthorized
2 x-correlation-id: 3d8d6cf8-c6ec-4c27-b313-65a517a41ed5
3 Content-Security-Policy: default-src 'self';base-uri 'self';font-src 'self' https://data:;form
4 Cross-Origin-Opener-Policy: same-origin
5 Cross-Origin-Resource-Policy: same-origin
6 Origin-Agent-Cluster: ?1
7 Referrer-Policy: no-referrer
8 Strict-Transport-Security: max-age=31536000; includeSubDomains
9 X-Content-Type-Options: nosniff
10 X-DNS-Prefetch-Control: off
11 X-Download-Options: noopen
12 X-Frame-Options: SAMEORIGIN
13 X-Permitted-Cross-Domain-Policies: none
14 X-XSS-Protection: 0
15 Content-Type: application/json; charset=utf-8
16 Content-Length: 42
17 ETag: W/"2a-+J/XpMgTejh2RSpr2Whc08Jh/+4"
18 Date: Sun, 26 Oct 2025 09:21:20 GMT
19 Connection: keep-alive
20 Keep-Alive: timeout=5
21
22 {"message":"Unauthorized","code":"UNAUTH"}
23
```


- 400 ValidationError,
Berikut Screenshoot nya



- 400 Bad JSON
Berikut Screenshoot nya



- Setiap response memiliki x-correlation-id dan *Helmet headers* aktif.

F. Observability & Security Verification

Skenario	Endpoint	Ekspektasi	Status	Bukti
201 Created	POST /orders	201 + x-correlation-id	✓	hardening_logs/_201_orders.txt
401 Unauthorized	POST /orders tanpa bearer	401 Unauthorized	✓	hardening_logs/_401_orders.txt
400 ValidationError	POST /orders (quantity=0)	400 Bad Request	✓	hardening_logs/_400_orders_validation.txt
400 Bad JSON	POST /orders (JSON rusak)	400 InvalidJSON	✓	hardening_logs/_400_orders_badjson.txt
200 OK	GET /notifications	200 OK + x-correlation-id	✓	hardening_logs/_200_notifications.txt

Semua log menunjukkan keamanan aktif: authorization: [REDACTED] dan header Helmet lengkap (X-Frame-Options, X-Content-Type-Options, Content-Security-Policy).

G. Kendala & Solusi

1. PowerShell sering memecah argumen saat curl, solusi: gunakan curl.exe + single quotes.
2. Jest config tidak kompatibel dengan TypeScript tanpa ts-node, solusi: ubah ke jest.config.cjs.
3. Tes gagal karena lokasi file salah, solusi: pindahkan ke folder services/order-service/test/.
4. Error JSON rusak menghasilkan 500, solusi: tambahkan handler parse error agar mengembalikan 400 (*BAD_JSON*).

H. Cara Reproduksi Proyek

```
npm ci
npm run dev:orders      # jalankan di port 5002
npm run dev:notif       # jalankan di port 5003
npm test
npx spectral lint openapi/api.yaml
```

I. Kesimpulan

Praktikum berhasil menerapkan seluruh tahapan metodologi Agile pada pengembangan *Web Service* berbasis kontrak: mulai dari Design-First, Mock-First, Test-First, hingga Implementasi (GREEN) dan Hardening.

Setiap tahap terverifikasi dengan bukti nyata berupa log, lint tanpa error, dan tes yang seluruhnya lulus. Proyek ini menunjukkan bahwa pendekatan Agile mampu menghasilkan layanan web yang terstruktur, aman, mudah diuji, dan siap dikembangkan lebih lanjut.