Nama : Leonardho R Sitanggang

Assignment Guidance: Case Study Running web server dan route in golang

Golang Backend Development Bootcamp

Deskripsi Assignment

Dalam tugas ini, student akan membangun sistem backend untuk mengelola aplikasi manajemen inventaris. Proyek ini akan melibatkan perancangan dan query pada database relasional untuk mengelola produk, inventaris, dan pesanan. Student akan membuat dan mengintegrasikan RESTful API untuk berinteraksi dengan database, memungkinkan operasi CRUD untuk data inventaris menggunakan Golang.

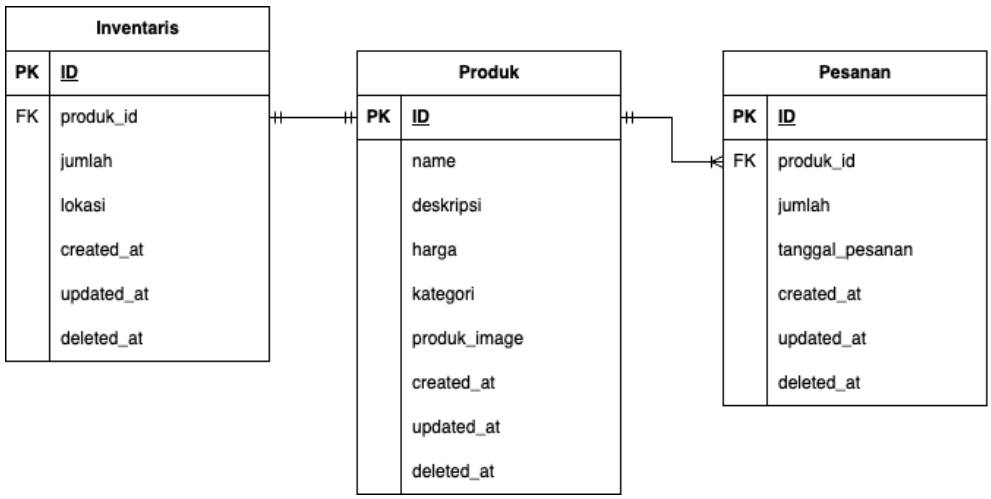
Selain itu, student akan mengimplementasikan penanganan file untuk mengunggah dan mengunduh gambar produk, serta menyusun route menggunakan framework Gin untuk menyajikan API secara efektif. Dengan menyelesaikan tugas ini, student akan mendapatkan pengalaman langsung dalam pengembangan backend, integrasi API, dan penanganan penyimpanan file, serta menerapkan konsep pemrograman inti dalam skenario dunia nyata.

Detail Assignment

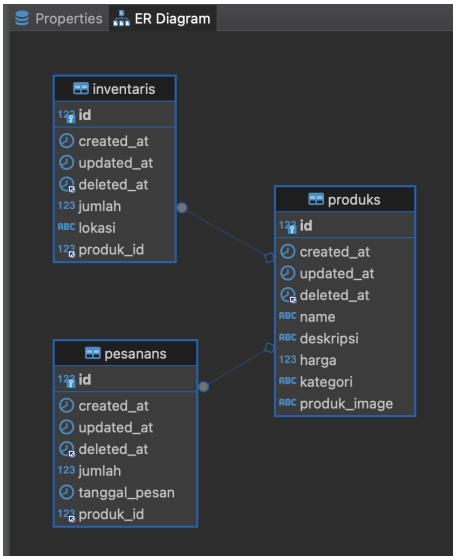
1. Database Design and Queries

a. Desain sebuah database relasional dengan setidaknya tabel-tabel berikut::

- Produk: Menyimpan detail produk (ID, nama, deskripsi, harga, dan kategori).
- Inventaris: Melacak tingkat stok dan lokasi produk (ID produk, jumlah, dan lokasi).
- Pesanan: Mencatat pesanan pelanggan (ID pesanan, ID produk, jumlah, dan tanggal pesanan).



Rancangan ERD



Relasi Database Hasil Migration

b. Write SQL scripts to:

- Memasukkan data sampel ke dalam tabel-tabel.

```
INSERT INTO pesanans
(id, created_at, updated_at, deleted_at, jumlah, tanggal_pesan, produk_id)
VALUES(0, NULL, NULL, NULL, 3, '2024-10-10',1);
```

Statistics 1

Name	Value
Updated Rows	1
Query	INSERT INTO pesanans (id, created_at, updated_at, deleted_at, jumlah, tanggal_pesan, produk_id) VALUES(0, NULL, NULL, NULL, 3, '2024-10-10',1)
Start time	Sat Apr 12 06:49:12 WIB 2025
Finish time	Sat Apr 12 06:49:12 WIB 2025

```
INSERT INTO inventaris
(id, created_at, updated_at, deleted_at, jumlah, lokasi, produk_id)
VALUES(0, NULL, NULL, NULL, 10, 'South Jakarta', 1);
```

Statistics 1

Name	Value
Updated Rows	1
Query	INSERT INTO inventaris (id, created_at, updated_at, deleted_at, jumlah, lokasi, produk_id) VALUES(0, NULL, NULL, NULL, 10, 'South Jakarta', 1)
Start time	Sat Apr 12 06:48:11 WIB 2025
Finish time	Sat Apr 12 06:48:11 WIB 2025

```sql
-- Nomor 1B. 1
INSERT INTO produks
(id, created_at, updated_at, deleted_at, name, deskripsi, harga, kategori, produk_image)
VALUES(0, NULL, NULL, NULL, 'Permen', 'Manisan', 10000, 'Snack', NULL);
```

Statistics 1

Name	Value
Updated Rows 1	
Query	INSERT INTO produks
	(id, created_at, updated_at, deleted_at, name, deskripsi, harga, kategori, produk_image)
	VALUES(0, NULL, NULL, NULL, 'Permen', 'Manisan', 10000, 'Snack', NULL)
Start time	Sat Apr 12 06:47:21 WIB 2025
Finish time	Sat Apr 12 06:47:21 WIB 2025

- Melakukan query untuk produk, inventaris, dan detail pesanan.

```sql
select * from inventaris i;
```

aris 1

Enter a SQL expression to filter results (use Ctrl+Space)

id	created_at	updated_at	deleted_at	jumlah	lokasi	produk_id
1	2025-04-12 01:08:53.666	2025-04-12 06:28:23.562	[NULL]	153	West Nehaburgh	1
2	2025-04-12 05:59:03.771	2025-04-12 06:27:55.233	[NULL]	5	North Roosevelt	2
3	2025-04-12 06:06:48.624	2025-04-12 06:06:48.624	[NULL]	489	Enidfurt	3
4	2025-04-12 06:18:10.867	2025-04-12 06:18:10.867	[NULL]	620	New Adelleside	4

produks 1

Enter a SQL expression to filter results (use Ctrl+Space)

	id	created_at	updated_at	deleted_at	name	deskripsi	harg
1	1	2025-04-12 01:08:53.662	2025-04-12 01:21:39.883	2025-04-12 01:23:34.180	Handmade Cotton Chair	test update description	
2	2	2025-04-12 05:59:03.769	2025-04-12 05:59:03.769	[NULL]	Awesome Granite Car	Voluptatem accusantium suscipit soluta nemo ipsa.¶Maxim	
3	3	2025-04-12 06:06:48.623	2025-04-12 06:06:48.623	[NULL]	Ergonomic Rubber Car	Nemo officiis in.¶Fugit nisi minus autem enim doloribus qui	
4	4	2025-04-12 06:18:10.862	2025-04-12 06:18:10.862	[NULL]	Small Concrete Cheese	Laboriosam illo rerum voluptatum consectetur earum modi	

```sql
SELECT * from pesanans p;
```

pesanans 1

Enter a SQL expression to filter results (use Ctrl+Space)

	id	created_at	updated_at	deleted_at	jumlah	tanggal_pesan	produk_id
1	1	2025-04-12 06:01:54.820	2025-04-12 06:01:54.820	[NULL]	5	2025-04-12	2
2	2	2025-04-12 06:27:55.230	2025-04-12 06:27:55.230	[NULL]	4	2025-04-12	2
3	3	2025-04-12 06:28:07.027	2025-04-12 06:28:07.027	[NULL]	5	2025-04-12	1
4	4	2025-04-12 06:28:12.124	2025-04-12 06:28:12.124	[NULL]	1	2025-04-12	1
5	5	2025-04-12 06:28:23.559	2025-04-12 06:28:23.559	[NULL]	4	2025-04-12	1

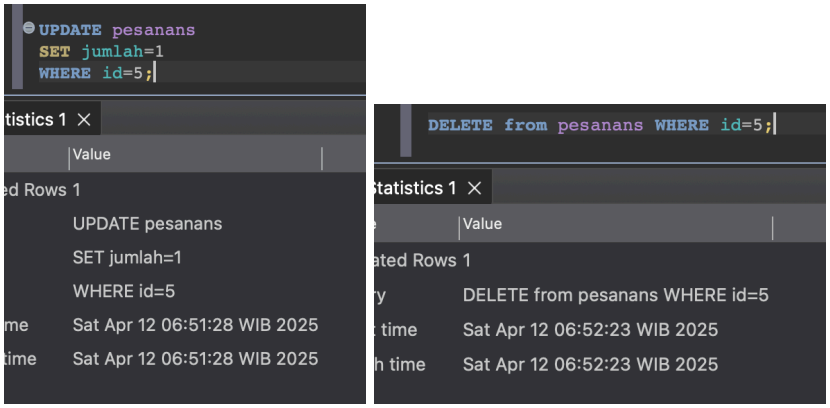
<localhost> Script-31

```sql
SELECT p.name, p.deskripsi, p.harga, i.jumlah, i.lokasi,
    count(p2.jumlah) as total_pesanan,
    sum(p2.jumlah) as total_jumlah_pesanan,
    avg(p2.jumlah) as average_jumlah_per_pesanan
FROM produks p
join inventaris i on i.produk_id = p.id
left join pesanans p2 on p2.produk_id  = p.id
group by p.id;
```

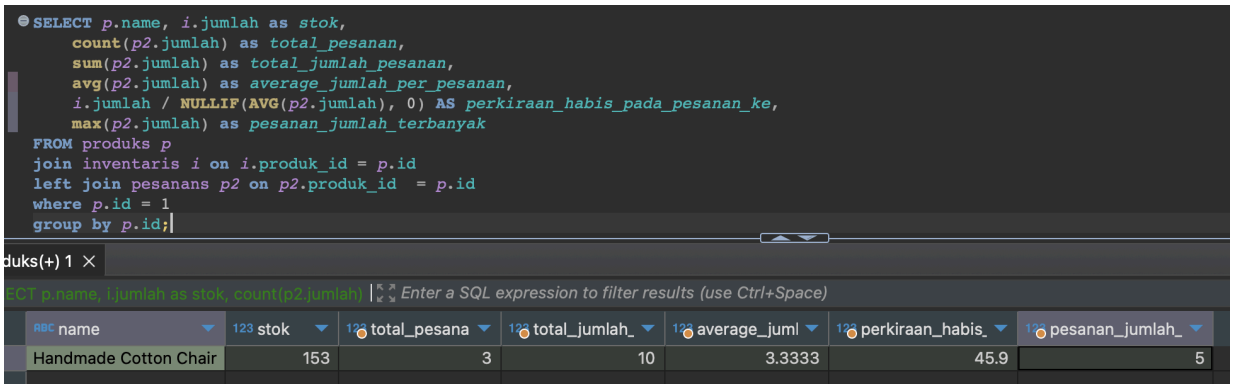
produks(+) 1

Enter a SQL expression to filter results (use Ctrl+Space)

	name	deskripsi	harga	jumlah	lokasi	total_pesana	total_jumlah_pe	average_jum
1	Handmade Cotton Chair	test update description	659	153	West Nehaburgh	3	10	3.3333
2	Awesome Granite Car	Voluptatem accusantium s	2	5	North Roosevelt	2	9	4.5
3	Ergonomic Rubber Car	Nemo officiis in.¶Fugit nis	114	489	Enidfurt	0	[NULL]	[NULL]
4	Small Concrete Cheese	Laboriosam illo rerum volu	453	620	New Adelleside	0	[NULL]	



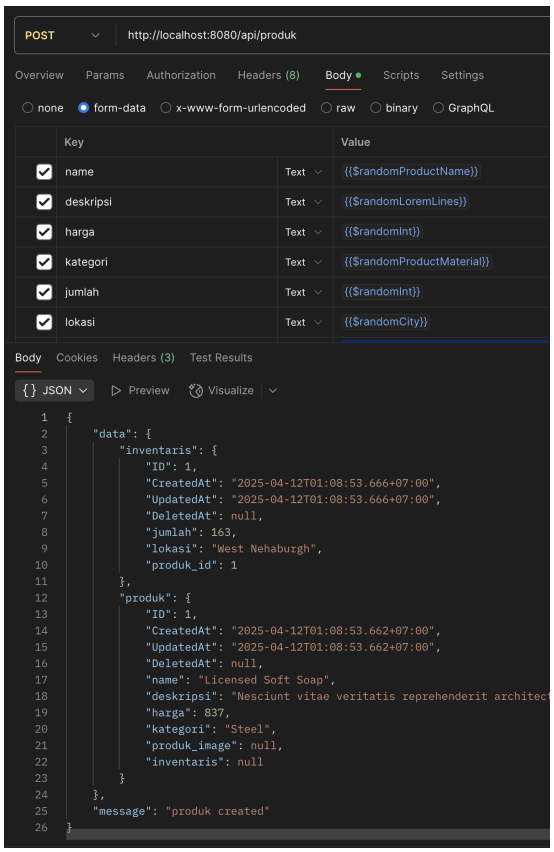
- Melakukan agregasi seperti total pesanan untuk suatu produk atau tingkat stok di lokasi tertentu.



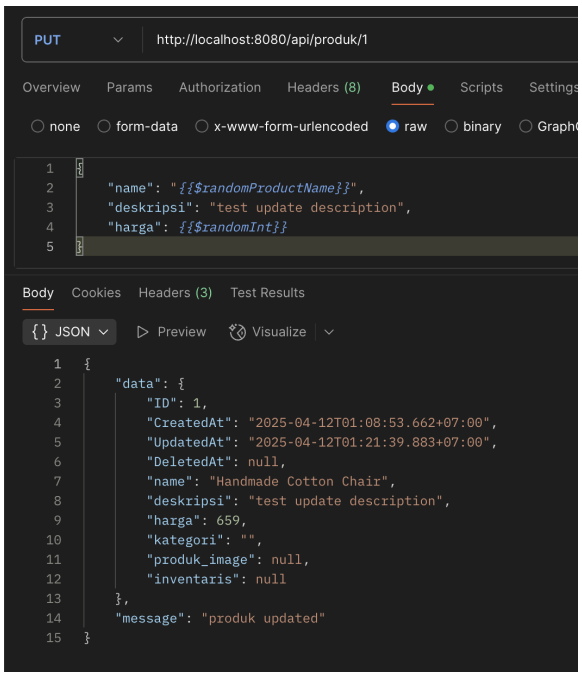
2. RESTful API Development

Buat API untuk hal-hal berikut:

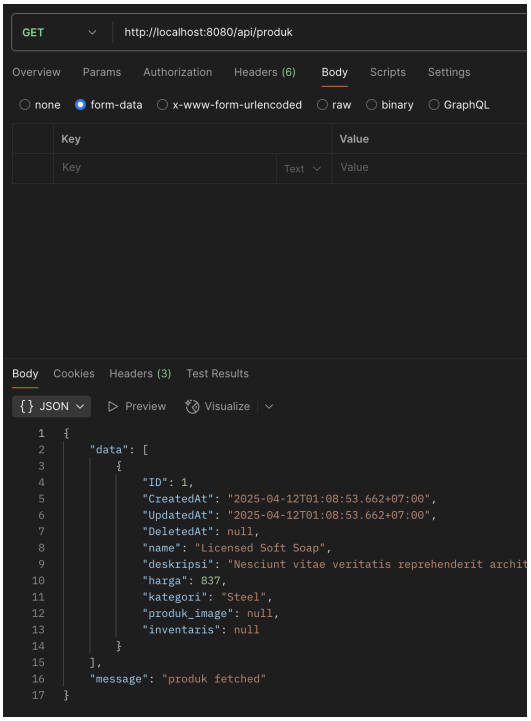
- **Produk**:
  - Menambahkan, melihat, memperbarui, dan menghapus produk.



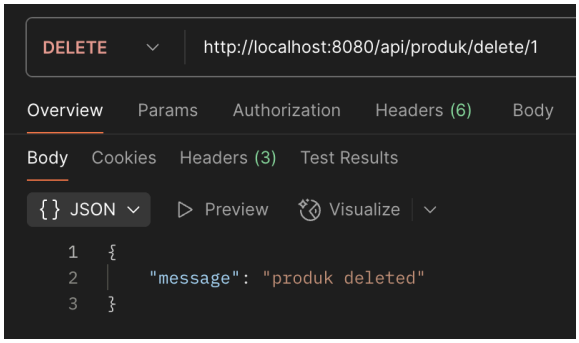
Menambahkan Produk

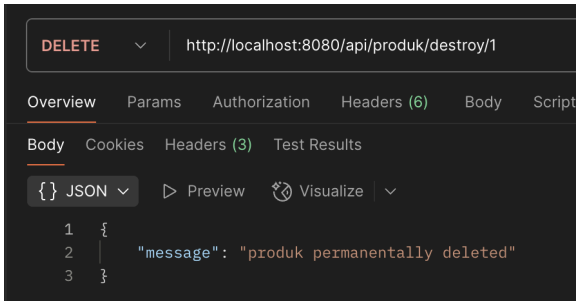


Mengubah Produk



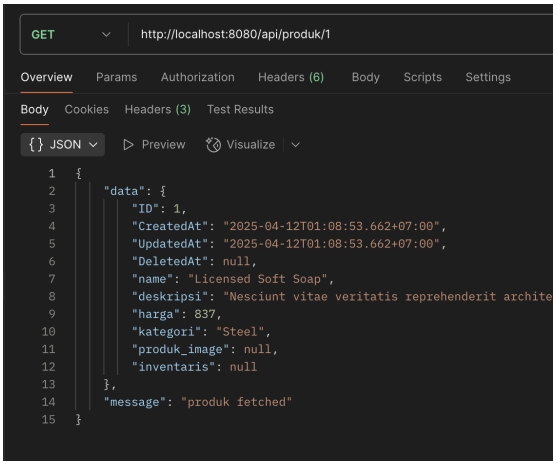
Melihat Produk



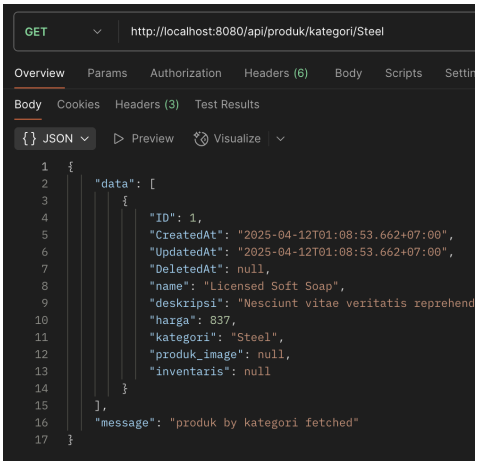


Menghapus Produk

- Melihat detail produk berdasarkan ID atau kategori.

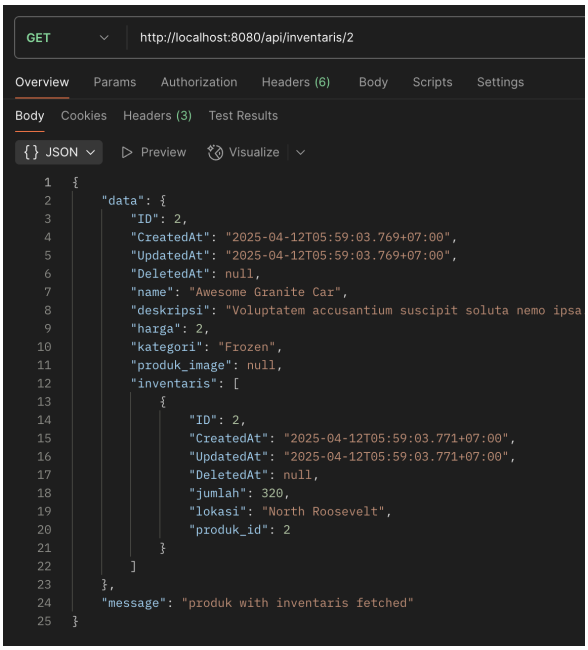


Berdasarkan ID

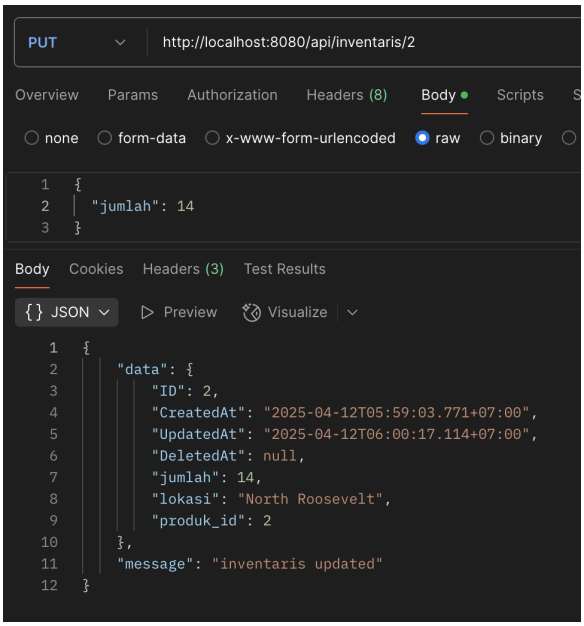


Berdasarkan Kategori

- **Inventaris**:
  - Melihat tingkat stok untuk suatu produk.
  - Memperbarui tingkat stok (menambah atau mengurangi stok).



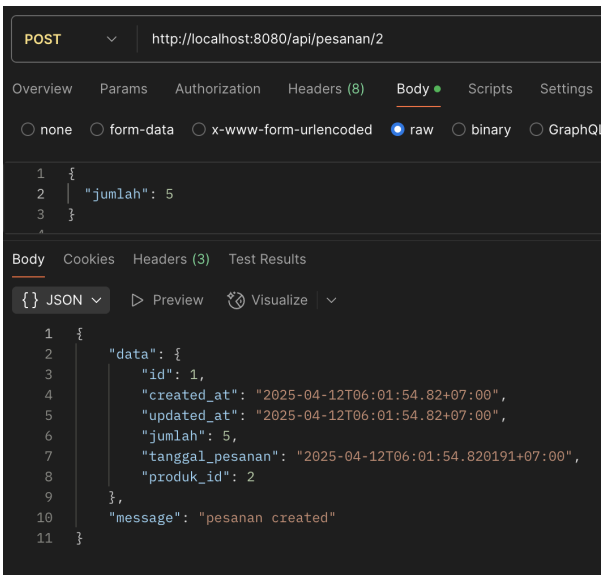
Melihat tingkat stok suatu produk



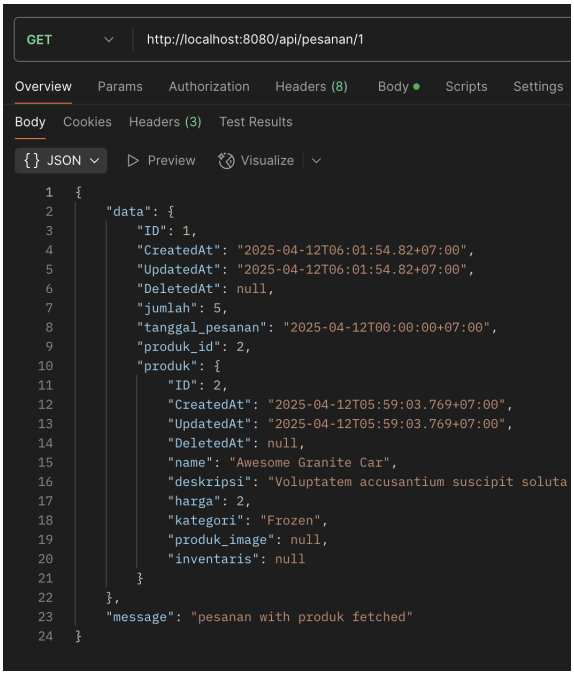
Memperbarui tingkat stok

- **Pesanan**:
  - Membuat pesanan baru.

- Mengambil detail pesanan berdasarkan ID.



Membuat pesanan baru



Mengambil detail pesanan berdasarkan ID

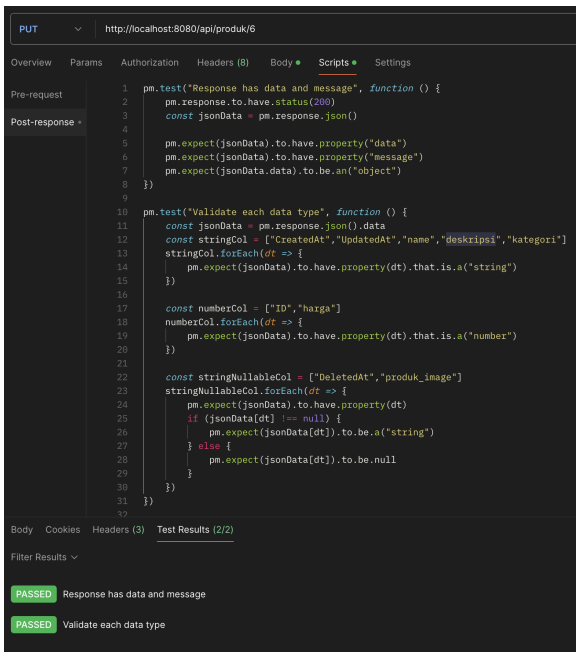
3. API Integration

- Uji API untuk memastikan integrasi dan fungsionalitas berjalan lancar.
- Tulis skrip atau gunakan alat (misalnya, Postman, Curl) untuk menguji semua endpoint.

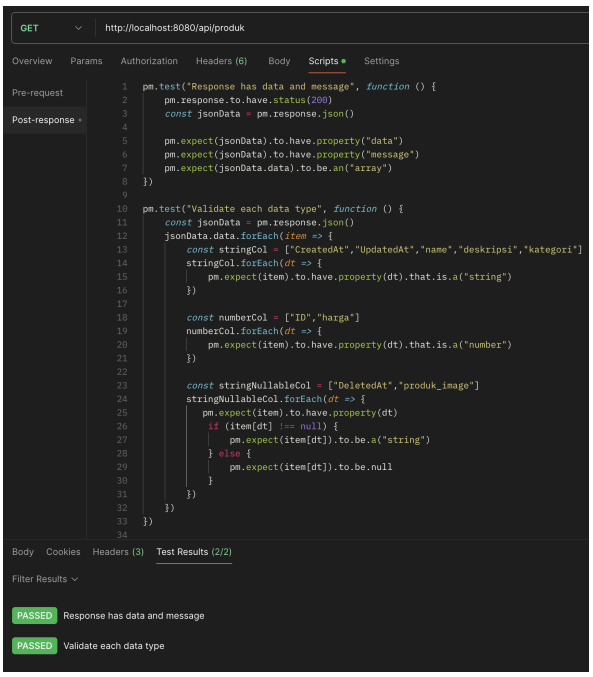
Integration Testing API:

- **Produk**:
  - Menambahkan, melihat, memperbarui, dan menghapus produk.

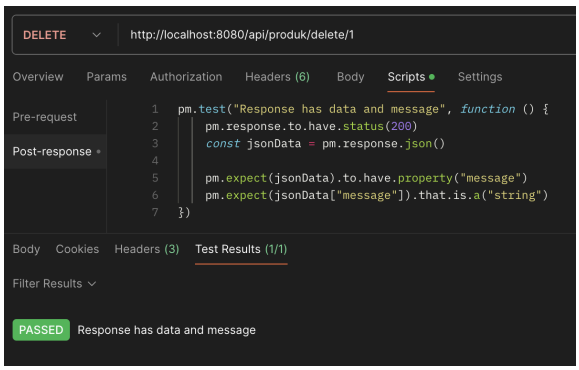
Menambahkan Produk



Mengubah Produk

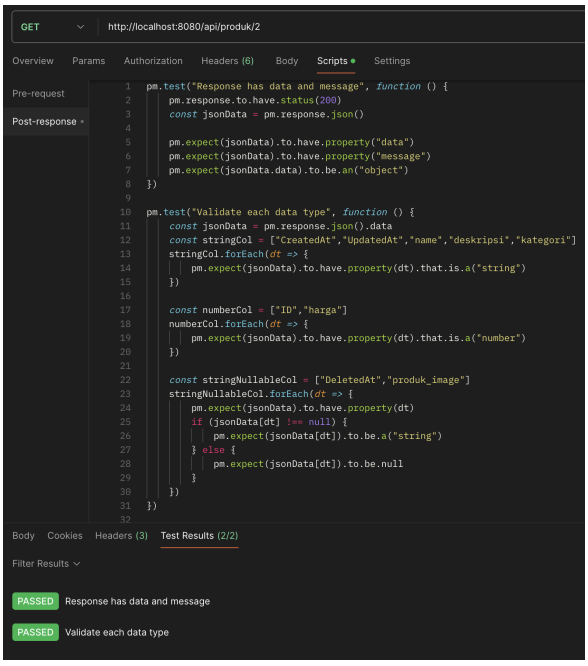


Melihat Produk

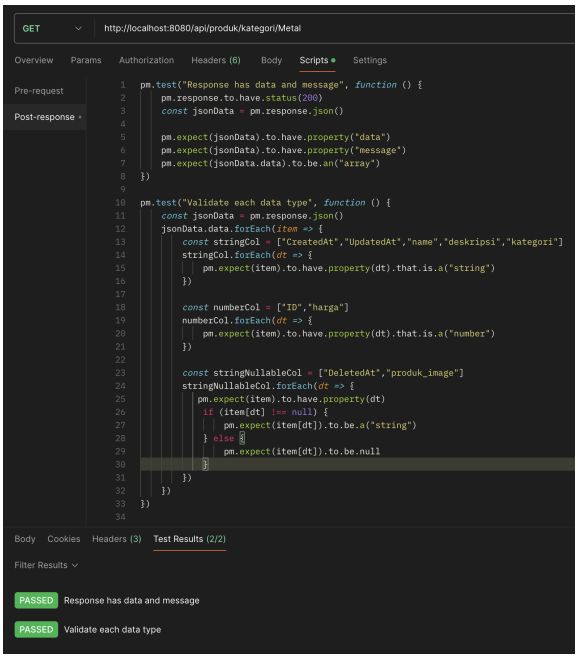


Menghapus Produk

- Melihat detail produk berdasarkan ID atau kategori.



Berdasarkan ID

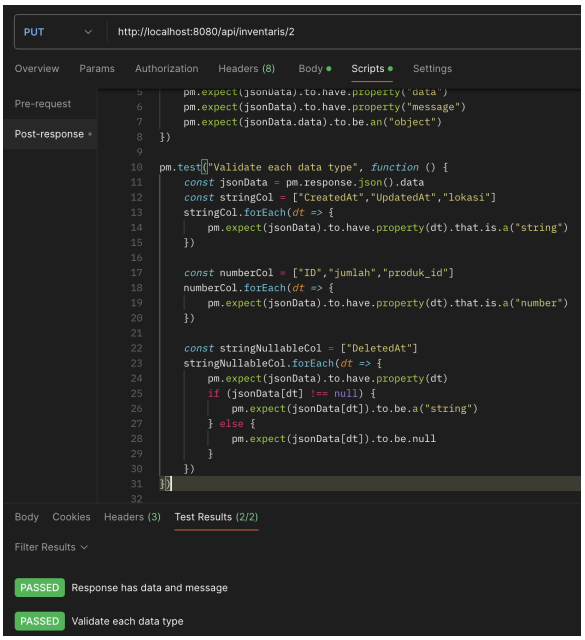


Berdasarkan Kategori

- **Inventaris**:
  - Melihat tingkat stok untuk suatu produk.

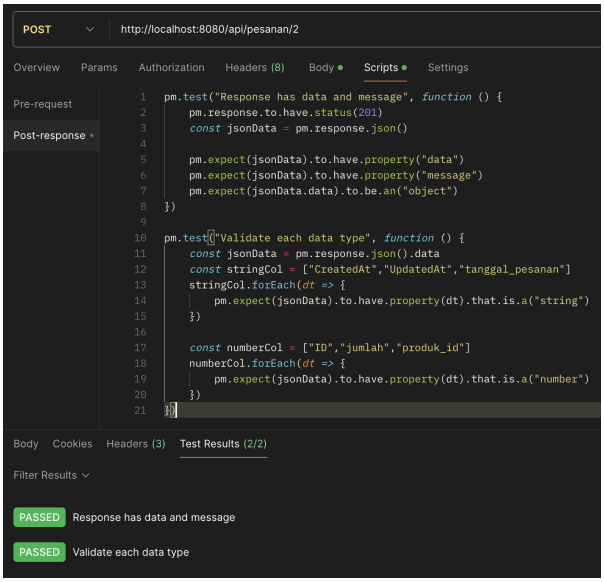
- Memperbarui tingkat stok (menambah atau mengurangi stok).

Melihat tingkat stok suatu produk



Memperbarui tingkat stok

- **Pesanan**:
  - Membuat pesanan baru.
  - Mengambil detail pesanan berdasarkan ID.



Membuat pesanan baru

Mengambil detail pesanan berdasarkan ID

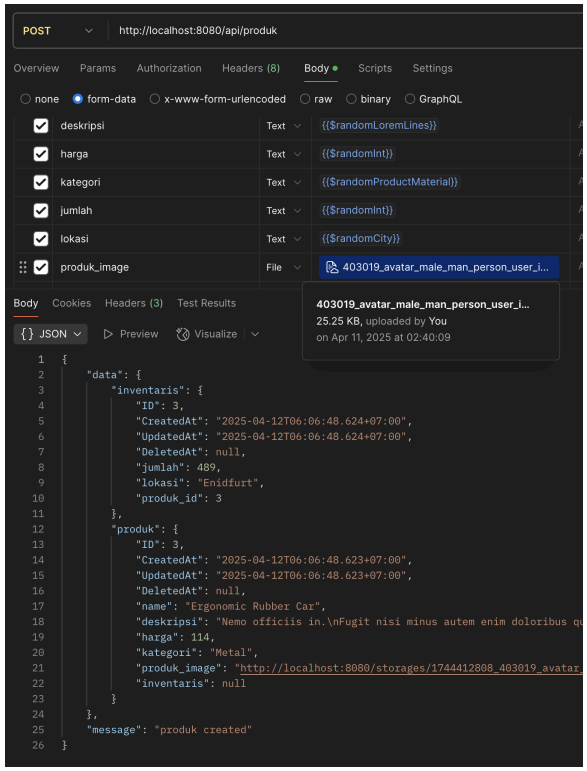
4. Web Server and Routing

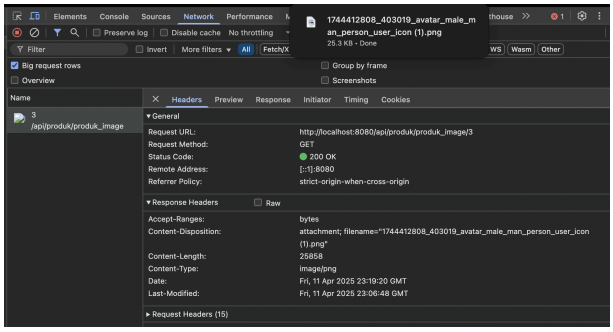
Gunakan framework Gin di Golang untuk:

- Menyusun route untuk semua endpoint API (misalnya, /products, /inventory, /orders).
- Menangani metode HTTP seperti GET, POST, PUT, dan DELETE.

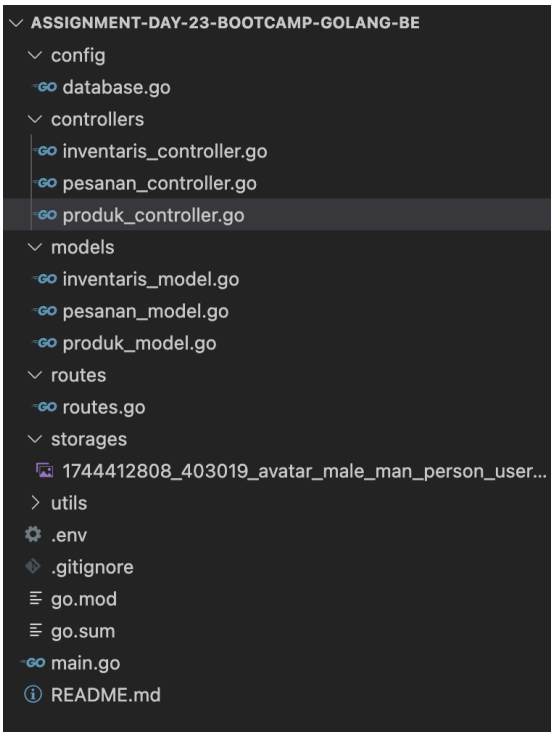
```go
func SetUpRoutes(r *gin.Engine, db *gorm.DB) {
    produkController := controllers.NewProdukController(db)
    inventarisController := controllers.NewInventarisController(db)
    pesananController := controllers.NewPesananController(db)

    api := r.Group("/api")
    {
        produk := api.Group("/produk")
        {
            produk.GET("/", produkController.GetAllProduk)
            produk.GET("/:id", produkController.GetProdukById)
            produk.GET("/produk_image/:id", produkController.GetDownloadProdukImage)
            produk.GET("/kategori/:kategori", produkController.GetProdukByKategori)
            produk.POST("/", produkController.CreateProduk)
            produk.PUT("/:id", produkController.UpdateProdukById)
            produk.DELETE("/delete/:id", produkController.SoftDeleteProdukById)
            produk.DELETE("/destroy/:id", produkController.HardDeleteProdukById)
        }
        inventaris := api.Group("/inventaris")
        {
            inventaris.GET("/:id", inventarisController.GetProdukInventarisByProdukId)
            inventaris.PUT("/:id", inventarisController.UpdateInventarisById)
            inventaris.DELETE("/destroy/:id", inventarisController.HardDeleteInventarisById)
        }
        pesanan := api.Group("/pesanan")
        {
            pesanan.GET("/:id", pesananController.GetDetailPesananById)
            pesanan.POST("/:produk_id", pesananController.CreatePesanan)
            pesanan.DELETE("/destroy/:id", pesananController.HardDeletePesananById)
        }
    }
}
```

5. Mengunggah dan Mengunduh File





Mengunduh image

Mengunggah image pada produk  File disimpan pada folder storages	

6. Dokumentasi API (Postman) & Tautan Lainnya

Postman Collection Export :
https://drive.google.com/file/d/1nyozUV-EN3Cjf6fMf-F5tQmTPS5BX4Dm/view?usp=sharing
Dokumentasi Postman :
https://documenter.getpostman.com/view/18882129/2sB2cYdgA4
DB Dump :
https://drive.google.com/file/d/1oi1RoJBq2nDdW1ozVXXVn5Sr-Hwj1ZbZ/view?usp=sharing
Github Repo :
https://github.com/FlazeFy/Assignment-Day-23-Bootcamp-Golang-BE