Project RESTful API :: Ecommerce

Setup dan Konfigurasi

Database

- Buatlah sebuah database baru
- Restore file ecommerce-db.sql ke dalam database yang baru saja dibuat
- Copy file .env.example ke .env, lalu ubah `DATABASE_NAME`, `DATABASE_HOST`, `DATABASE_USERNAME` dan `DATABASE_PASSWORD` sesuai kebutuhan

Logging

- Ubah `LOG_LEVEL` untuk mendapatkan level log sesuai yang diinginkan

Debug	Semua query ke database (SQL)
Info	Pengaksesan route, handler, service, repository
Error	Semua error dari eksekusi database query, return dari service,

- Output log ada di file app.log di dalam folder /log. Ubah `LOG_PATH` untuk menyimpan di folder yang lain. `LOG_PATH` harus diawali dengan karakter "/".

Session

- User yang berhasil login, akan mendapat sebuah token yang bisa digunakan untuk mengakses protected route. Secara bawaan (by default), token ini tidak memiliki batas waktu kedaluwarsa.
- Ubah `SESSION_LIFETIME` jika ingin menetapkan batas waktu kedaluwarsa. `SESSION_LIFETIME` menggunakan satuan menit. Ganti dengan 0 (angka nol) jika ingin mematikan fitur ini.

HTTP Server

- HTTP Server, secara bawaan, akan melayani request pada port 8080. Ubah `PORT` jika ingin mengganti port bawaan tersebut.
- Jalankan web server dengan cara masuk ke folder /cmd, dan melakukan `go run .`

Instruksi Penggunaan

- Gunakan tool Postman untuk melakukan review
- Postman collection tersedia di dalam source code repository. Gunakan menu File | Import dan memilih file 20241118Project.postman_collection.json

Catatan

- Semua kalkulasi dilakukan on-the-fly. (Best Seller, New Product, Product Rating, Discount Rate / Net Price, Product yang sudah ter-wishlist) Tidak ada yang menggunakan tabel agregat atau CRON atau prekalkulasi.
- Validasi input hanya menggunakan package go-playground/validator/v10 (baked-in dan custom validation). Pesan error diterjemahkan ke Bahasa Indonesia menggunakan package go-playground/universal-translator + go-playground/locales. Tidak ada validasi manual.
- API yang bertanda ⚠️ di dokumentasi ini, mewajibkan user untuk login terlebih dahulu supaya diijinkan akses oleh middleware auth.

API Endpoints

a. Auth

1. POST /api/register (Postman Collection: /auth/register)

Validasi

name	required, max=30
username	required, unique(database table:users), max=40
password	required, max=40

Contoh request

```
{
    "name": "Customer Satu",
    "username": "customer1@mail.com",
    "password": "drowssap"
}
```

Contoh response sukses

```
{
    "status_code": 201,
    "message": "Customer registered successfully.",
    "data": {
        "name": "Customer Satu",
        "username": "customer1@mail.com",
        "created_at": "2024-11-24T23:16:09.850622+07:00"
    }
}
```

Contoh response gagal

```
{
    "status_code": 422,
    "message": "invalid input",
    "data": [
        {
            "field": "name",
            "message": "name wajib diisi",
            "tag": "required",
            "value": "",
            "param": ""
        }
    ]
}
```

2. POST /api/login (Postman Collection: /auth/login)

Validasi

username	required
password	required

Contoh request

```
{
    "username": "customer1@mail.com",
    "password": "drowssap"
}
```

Contoh response sukses

```
{
    "status_code": 200,
    "message": "user authenticated",
    "data": {
        "token": "f29523d7-1df6-42df-b39d-42b34a958944",
        "expired_at": "6107-12-19T01:41:41.210877+07:00"
    }
}
```

Contoh response gagal

```
{
    "status_code": 401,
    "message": "invalid username and/or password."
}
```

b. Home

1. GET /api/banners (Postman Collection: /home/banners)

Contoh response sukses

```
{
    "status_code": 200,
    "message": "Get all banners",
    "data": [
        {
            "photo": "https://placehold.co/600x400?text=Banner+1",
            "title": "Banner Title 1",
            "subtitle": "Banner Subtitle 1",
            "path_page": "https://excitable-circumference.info"
        },
        {
            "photo": "https://placehold.co/600x400?text=Banner+5",
            "title": "Banner Title 5",
            "subtitle": "Banner Subtitle 5",
            "path_page": "https://flowery-abacus.biz/"
        }
    ]
}
```

2. GET /api/best-sellers (Postman Collection: /home/bestsellers)

Contoh response sukses

```
{
    "status_code": 200,
    "page": 1,
    "limit": 6,
    "total_items": 42,
    "total_pages": 7,
    "data": [
        {
            "id": 2,
            "name": "Product 2",
            "thumbnail_url": "",
            "price": "",
            "discount_rate": null,
            "net_price": "",
            "category_id": 8,
            "rating": null,
            "on_my_wishlist": null,
            "created_at": "",
            "is_new": false
        },
        {
            "id": 12,
            "name": "Product 12",
            "thumbnail_url": "",
            "price": "",
            "discount_rate": null,
            "net_price": "",
            "category_id": 9,
            "rating": null,
            "on_my_wishlist": null,
            "created_at": "",
            "is_new": false
        }
    ]
}
```

3. GET /api/categories (Postman Collection: /home/categories)

Contoh response sukses

```
{
    "status_code": 200,
    "message": "Get all categories",
    "data": [
        {
            "id": 1,
            "name": "Category A"
        },
        {
            "id": 20,
            "name": "Category T"
        }
    ]
}
```

4. GET /api/weeklies (Postman Collection: /home/weeklies)

Contoh response sukses

```
{
    "status_code": 200,
    "message": "user authenticated",
    "data": {
        "token": "f29523d7-1df6-42df-b39d-42b34a958944",
        "expired_at": "6107-12-19T01:41:41.210877+07:00"
    }
}
```

5. GET /api/products (Postman Collection: /home/products)

API products mendeteksi guest atau user. Jika user, dan sudah pernah memasukkan product ke dalam wishlist, maka field on_my_wishlist bernilai `true`

Contoh response sukses

```
{
    "status_code": 200,
    "page": 1,
    "limit": 6,
    "total_items": 39,
    "total_pages": 6,
    "data": [
        {
            "id": 2,
            "name": "Product 2",
            "thumbnail_url": "",
            "price": "",
            "discount_rate": null,
            "net_price": "",
            "category_id": 8,
            "rating": null,
            "on_my_wishlist": null,
            "created_at": "",
            "is_new": false
        },
        {
            "id": 12,
            "name": "Product 12",
            "thumbnail_url": "",
            "price": "",
            "discount_rate": null,
            "net_price": "",
            "category_id": 9,
            "rating": null,
            "on_my_wishlist": true,
            "created_at": "",
            "is_new": false
        }
    ]
}
```

6. **GET /api/recommendations (Postman Collection: /home/recommendations)**

   Contoh response sukses

```
{
    "status_code": 200,
    "message": "Get all recommended products",
    "data": [
        {
            "title": "Title Recommendation 1",
            "subtitle": "Subtitle Recommendation 1",
            "photo":
"https://placehold.co/600x400?text=Recommendation+1",
            "product_id": 1
        }
    ]
}
```

7. POST /api/wishlist ⚠️ (Postman Collection: /wishlist/add to wishlist)

Validasi

product_id	required, exists(database table:products)

Contoh request

```
{
    "product_id": 4
}
```

Contoh response sukses

```
{
    "status_code": 201,
    "message": "Added to Wishlist",
    "data": {
        "product_id": 4,
        "created_at": "2024-11-27T23:03:16.559653+07:00"
    }
}
```

8. DELETE /api/wishlist ⚠️ (Postman Collection: /wishlist/remove from wishlist)

Contoh request

```
{
    "product_id": 3
}
```

Contoh response sukses

```
{
    "status_code": 200,
    "message": "Item removed from Wishlist"
}
```

9. GET /api/cart ⚠️ (Postman Collection: /shopping cart/items)

Contoh response sukses

```
{
    "status_code": 200,
    "message": "Get customer cart",
    "data": {
        "items": [
            {
                "product_id": 4,
                "quantity": 1
            }
        ],
        "item_count": 1
    }
}
```

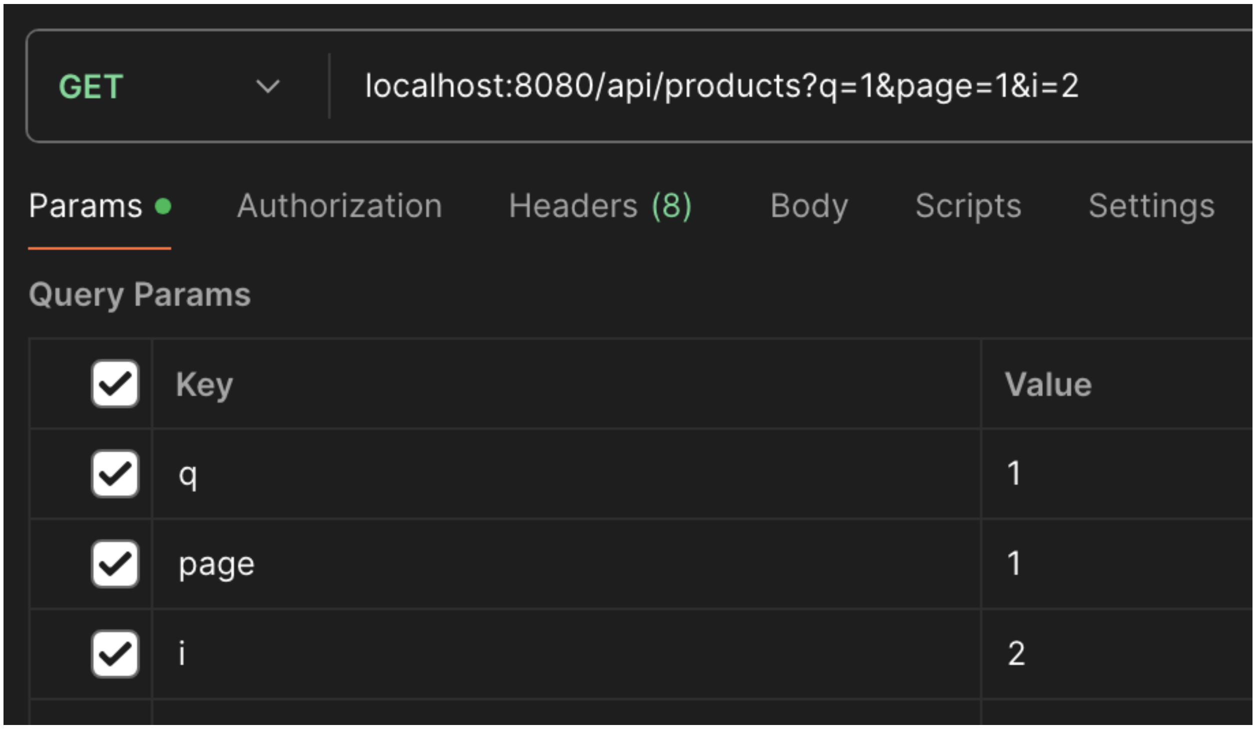
c. Product

1. GET /api/products (Postman Collection: /home/products)

Parameter

q	kata kunci pencarian
page	halaman (pagination)
i	jumlah produk yang ditampilkan di tiap halaman (item per page)

Contoh request



GET localhost:8080/api/products?q=1&page=1&i=2

Query Params

☑	Key	Value
☑	q	1
☑	page	1
☑	i	2

Contoh response sukses

```
{
    "status_code": 200,
        "page": 1,
        "limit": 2,
        "total_items": 90,
        "total_pages": 45,
        "data": [
            {
                "id": 1,
                "name": "Product 1",
                "thumbnail_url":
"https://placehold.co/600x400?text=Product+1",
                "price": "$12.30",
                "discount_rate": 10,
                "net_price": "$11.07",
                "category_id": 5,
                "rating": 0,
                "on_my_wishlist": true,
                "created_at": "2024-11-15T11:16:26.09646+07:00",
                "is_new": true
            },
            {
                "id": 4,
                "name": "Product 4",
                "thumbnail_url":
"https://placehold.co/600x400?text=Product+4",
                "price": "$36.55",
                "discount_rate": 6.85,
                "net_price": "$34.05",
                "category_id": 7,
                "rating": 0,
                "on_my_wishlist": true,
                "created_at": "2024-10-19T17:08:23.864501+07:00",
                "is_new": false
            }
        ]
    }
```

2. POST /api/cart ⚠️ (Postman Collection: /shopping cart/add to cart)

Add to cart menggunakan query `INSERT OR UPDATE`. Artinya, jika produk belum ada di keranjang, lakukan `INSERT`. Sebaliknya, jika produk sudah ada di keranjang, lakukan `UPDATE`.

Jika terdapat field `quantity` pada JSON request, maka keranjang belanja akan di-update dengan `quantity`. Sebaliknya, jika tidak ada field `quantity`, maka produk di keranjang belanja akan ditambah 1 (increment)

Contoh request

```
{
    "product_id":4,
    "quantity":1
}
```

Contoh response sukses

```
{
    "status_code": 200,
    "message": "Product successfully added to cart"
}
```

3. **GET /api/categories/{id}/products (Postman Collection: /categories/{id}/products)**

Contoh response sukses

```
{
    "status_code": 200,
    "page": 1,
    "limit": 6,
    "total_items": 6,
    "total_pages": 1,
    "data": [
        {
            "id": 71,
            "name": "Product 71",
            "thumbnail_url":
"https://placehold.co/600x400?text=Product+71",
            "price": "$47.08",
            "discount_rate": 2.14,
            "net_price": "$46.08",
            "category_id": 1,
            "rating": 0,
            "on_my_wishlist": false,
            "created_at": "2024-10-16T10:57:02.888926+07:00",
            "is_new": false
        },
        {
            "id": 75,
            "name": "Product 75",
                "thumbnail_url":
"https://placehold.co/600x400?text=Product+75",
                "price": "$15.38",
                "discount_rate": 4.25,
                "net_price": "$14.73",
                "category_id": 1,
                "rating": 0,
                "on_my_wishlist": false,
                "created_at": "2024-11-21T17:34:01.000999+07:00",
                "is_new": true
            }
        ]
    }
```

d. Checkout

1. GET /api/product/{id} (Postman Collection: /checkout/product)

Contoh response sukses

```
{
    "status_code": 200,
    "message": "Product Found",
    "data": {
        "id": 1,
        "name": "Product 1",
        "thumbnail_url": "https://placehold.co/600x400?text=Product+1",
        "price": "$12.30",
        "discount_rate": 10,
        "net_price": "$11.07",
        "category_id": 5,
        "rating": 0,
        "on_my_wishlist": true,
        "created_at": "2024-11-15T11:16:26.09646+07:00",
        "is_new": true,
        "variants": [
            {
                "name": "Variant 1A"
            },
            {
                "name": "Variant 1B"
            },
            {
                "name": "Variant 1C"
            }
            ],
            "photos": [
                {
                    "photo_url":
"https://placehold.co/600x400/jpg?text=Product+1+A"
                },
                {
                    "photo_url":
"https://placehold.co/600x400/jpg?text=Product+1+B"
                },
                {
                    "photo_url":
"https://placehold.co/600x400/jpg?text=Product+1+C"
                },
                {
                    "photo_url":
"https://placehold.co/600x400/jpg?text=Product+1+D"
                }
            ]
        }
    }
```

2. GET /api/cart ⚠️(Postman Collection: /shopping cart/items)

Contoh response sukses

```
{
    "status_code": 200,
    "message": "Get customer cart",
    "data": {
        "items": [
            {
                "product_id": 4,
                "quantity": 1
            }
        ],
        "item_count": 1
    }
}
```

3. PUT /api/cart ⚠️(Postman Collection: /shopping cart/update cart)

Contoh request

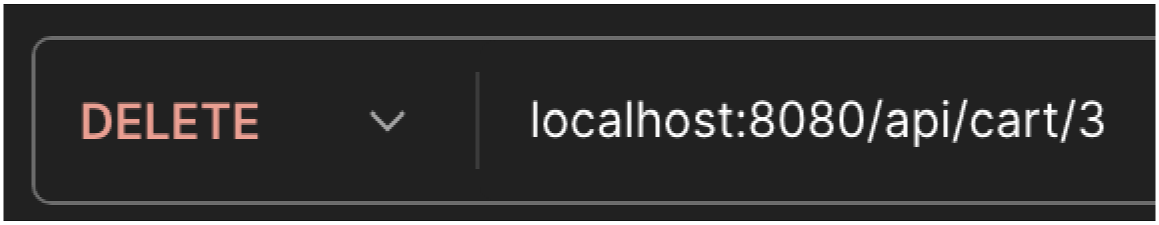
```
{
    "product_id":2,
    "quantity":3
}
```

Contoh response sukses

```
{
    "status_code": 200,
    "message": "Cart successfully updated"
}
```

4. DELETE /api/cart/{product_id} ⚠️(Postman Collection: /shopping cart/remove from cart)

Contoh request



DELETE localhost:8080/api/cart/3

Contoh response sukses

```
{
    "status_code": 200,
    "message": "Product successfully removed from cart"
}
```

5. GET /api/address ⚠️(Postman Collection: /checkout/addresses)

Contoh response sukses

```
{
    "status_code": 200,
    "message": "Customer addresses retrieved successfully",
    "data": [
        {
            "id": 3,
            "full_name": "customer 1",
            "email": "customer1b",
            "detail": "street 2",
            "is_default": true
        }
    ]
}
```

6. POST /api/order ⚠️(Postman Collection: /checkout/checkout)

Validasi

token	not-empty-cart (keranjang belanja tidak boleh kosong)
address_id	required, exists-for-user (address_id harus dimiliki oleh user)
coupon_code	max=20 (maksimal 20 karakter)
shipping	required, is-regular (harus berisi “regular”)
notes	max=128 (maksimal 128 karakter)
payment_method	oneof=bank cod (harus berisi “bank” atau “cod”)

Contoh request

```
{
    "address_id":2,
    "coupon_code":"abcde",
    "shipping":"regular",
    "notes":"lorem ipsum",
    "payment_method":"bank"
}
```

Contoh response sukses

```
{
    "status_code": 201,
    "message": "order created",
    "data": {
        "id": "59",
        "created_at": "2024-11-22T00:32:08.228525+07:00"
    }
}
```

e. Account

1. GET /api/profile⚠️(Postman Collection: /profile/profile)

Contoh response sukses

```
{
    "status_code": 201,
    "message": "Customer Profile retrieved successfully.",
    "data": {
        "name": "a",
        "username": "a",
        "created_at": "2024-11-19T03:02:56.4429+07:00"
    }
}
```

2. POST /api/address ⚠️(Postman Collection: /addresses/create address)

Contoh request

```
{
    "full_name":"f",
    "email":"e@mail.com",
    "detail":"street details"
}
```

Contoh response sukses

```
{
    "status_code": 201,
    "message": "address successfully created",
    "data": {
        "id": 0,
        "full_name": "f",
        "email": "e@mail.com",
        "detail": "street details",
        "is_default": false
    }
}
```

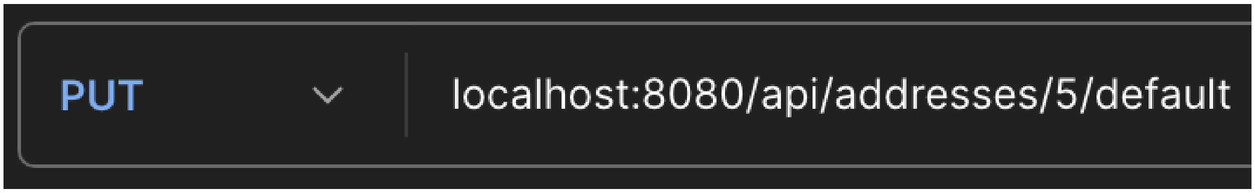
3. GET /api/address ⚠️(Postman Collection: /addresses/list of address)

Contoh response sukses

```
{
    "status_code": 200,
    "message": "Customer addresses retrieved successfully",
    "data": [
        {
            "id": 3,
            "full_name": "customer 1",
            "email": "customer1b",
            "detail": "street 2",
            "is_default": true
        },
        {
            "id": 5,
            "full_name": "f",
            "email": "e@mail.com",
            "detail": "street details",
            "is_default": false
        }
    ]
}
```

4. **PUT /api/address/{address_id}/default ⚠️(Postman Collection: /addresses/set default address)**

   Contoh request



```
PUT    localhost:8080/api/addresses/5/default
```

5. **PUT /api/address/{address_id} ⚠️(Postman Collection: /addresses/edit address)**

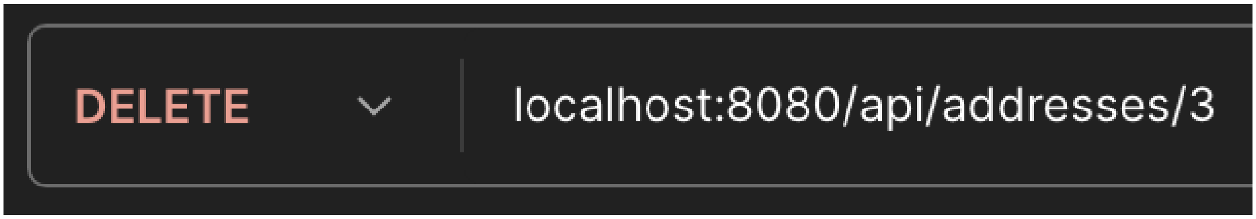
   Contoh request

```
{
    "full_name":"f",
    "email":"f@mail.com",
    "detail":"street details f"
}
```

   Contoh response sukses

```
{
    "status_code": 200,
    "message": "address successfully updated",
    "data": {
        "id": 1,
        "full_name": "f",
        "email": "f@mail.com",
        "detail": "street details f",
        "is_default": false
    }
}
```

6. **DELETE /api/address/{address_id} ⚠️(Postman Collection: /addresses/remove address)**



```
DELETE    localhost:8080/api/addresses/3
```